

1. BGBilling	12
1.1 Описание основной части программы BGBilling	14
1.1.1 Логическая структура биллинга	15
1.1.2 Программная структура биллинга	16
1.1.3 Установка биллинга	18
1.1.3.1 Особенности установки под различные платформы	19
1.1.3.1.1 Linux	20
1.1.3.1.2 Windows	22
1.1.3.2 MySQL/MariaDB	23
1.1.3.3 JDK	29
1.1.3.4 ActiveMQ-сервер	30
1.1.3.5 Установка сервера биллинга (для версий до 8.0(вкл.))	33
1.1.3.6 Установка (обновление) сервера биллинга (для версий с 8.2010)	35
1.1.3.7 Установка планировщика (с версии 8.2107)	37
1.1.3.8 Установка приложений биллинга (BGInetAccess, BGVoiceProcwssor и др.) (с версии 8.2104)	38
1.1.4 Установка и первый запуск клиента биллинга	39
1.1.4.1 Запуск клиента через BGBilling Runner	40
1.1.4.2 Запуск клиента через Java Web Start (устарело)	42
1.1.4.3 Запуск клиента через установку локальной версии BGClient (устарело)	43
1.1.5 Описание интерфейса клиента биллинга	46
1.1.6 Настройка подсистем биллинга	54
1.1.6.1 Конфигурации	55
1.1.6.2 Логирование	56
1.1.6.3 RADIUS-протокол	58
1.1.7 Настройка сервера биллинга	60
1.1.7.1 Конфигурация	61
1.1.7.2 Почтовая подсистема	67
1.1.7.3 Система оповещения	68
1.1.7.4 Закрытый период	69
1.1.8 Модули	70
1.1.9 Установка и настройка плагинов	78
1.1.10 Установка обновлений биллинга	80
1.1.11 Установка лицензии биллинга	84
1.1.12 Настройка SSL между сервером и клиентом	85
1.1.13 Настройка планировщика	87
1.1.14 Справочники	92
1.1.14.1 Общие сведения	93
1.1.14.2 Адрес - страны, города, районы, кварталы, улицы, дома	94
1.1.14.3 Договоры - параметры	98
1.1.14.4 Договоры - группы параметров	100
1.1.14.5 Договоры - группы	101
1.1.14.6 Договоры - шаблоны комментариев	103
1.1.14.7 Договоры - значения списков	104
1.1.14.8 Договоры - обслуживание	106
1.1.14.9 Договоры - скрипты поведения	107
1.1.14.10 Типы платежей	108
1.1.14.11 Типы расходов	110
1.1.14.12 Типы времени	111
1.1.15 Клиенты	113
1.1.16 Договор	114
1.1.16.1 Общие сведения, создание договора	115
1.1.16.2 Обзор карточки договора	117
1.1.16.3 Параметры договора	119
1.1.16.3.1 Общие сведения о параметрах договоров	120
1.1.16.3.2 Копирование параметров	121
1.1.16.3.3 Параметры типа "Текст", "Флаг", "Дата", "E-Mail"	122
1.1.16.3.4 Параметр типа "Адрес"	123
1.1.16.3.5 Параметр типа "Список"	126
1.1.16.3.6 Параметр типа "Мультисписок"	127
1.1.16.3.7 Параметр типа "Телефон"	128
1.1.16.3.8 История изменения параметра	130
1.1.16.4 Группы договоров	131
1.1.16.5 Поиск договоров	132
1.1.16.5.1 Создание собственных алгоритмов поиска договоров	137
1.1.16.6 Баланс	141
1.1.16.7 Лимит договора, режимы договора, управление лимитом	144
1.1.16.8 Режимы баланса договора	145
1.1.16.9 Статус договора	146
1.1.16.10 Тариф и группа тарифов	149
1.1.16.11 Учетные периоды договора	150
1.1.16.12 Примечания	152
1.1.16.13 Дополнительные действия	153

1.1.16.14	Подключение модулей и их услуг к договору	154
1.1.16.15	Карты регистрации договора	157
1.1.16.15.1	FOP-карточки	158
1.1.16.15.2	Полная карта договора	161
1.1.16.16	Шаблоны договоров	163
1.1.16.16.1	Шаблон имени	164
1.1.16.16.2	Лимит, лицо, режим, время жизни, статус	165
1.1.16.16.3	Модули2	166
1.1.16.16.4	Прочие параметры	167
1.1.16.16.5	Создание договора по шаблону	168
1.1.16.17	Субдоговоры	169
1.1.16.17.1	Добавление субдоговоров	170
1.1.16.17.2	Зависимые субдоговоры	172
1.1.16.17.3	Независимые субдоговора	173
1.1.16.18	Переоформление договоров	175
1.1.16.19	Удаление договоров, архив договоров	176
1.1.17	Тарифные планы	177
1.1.17.1	Редактирование тарифного поддрева	180
1.1.17.2	Расширение тарифных планов	183
1.1.17.3	Персональные тарифные планы	184
1.1.17.4	Порядок просмотра тарифных планов	187
1.1.17.5	Стандартные узлы тарифных планов	188
1.1.17.5.1	Услуга	189
1.1.17.5.2	Мультиуслуга	190
1.1.17.5.3	Период	191
1.1.17.5.4	Фильтр по времени	193
1.1.17.5.5	Фильтр по типу времени	195
1.1.17.5.6	Параметры тарификации	196
1.1.17.5.7	Использовать карту зон	197
1.1.17.5.8	Зона	198
1.1.17.5.9	Часть префикса и Диапазон префиксов	199
1.1.17.5.10	Стоимость минуты	201
1.1.17.5.11	Множитель цены	202
1.1.17.5.12	Элемент каталога	203
1.1.17.6	Тарифные опции	204
1.1.17.7	Тарифные переменные	207
1.1.18	Объекты	209
1.1.19	Web-интерфейс пользователя	216
1.1.19.1	Настройка доступа к статистике	217
1.1.19.2	Настройка страницы статистики	225
1.1.19.3	Обзор Web-интерфейса пользователя	230
1.1.19.3.1	Смена тарифных планов	234
1.1.19.3.2	Управление статусом	236
1.1.19.3.3	Управление лимитом	237
1.1.20	Разграничение прав доступа	241
1.1.20.1	Пользователи, группы, права	242
1.1.20.2	Настройка дерева действий	246
1.1.20.3	Доступность пунктов меню в клиенте BGBillingClient	247
1.1.20.4	Пользовательские правила	249
1.1.20.5	Настройка ролей пользователей(с версии 9.2204)	250
1.1.21	Сервис	253
1.1.21.1	Журналы	254
1.1.21.1.1	Ошибки обработки логов	255
1.1.21.1.2	Ошибки периодических процессов	256
1.1.21.1.3	Журнал запросов	258
1.1.21.1.4	Журнал Web-запросов	259
1.1.21.2	Загрузка платежей и расходов из файла	260
1.1.21.2.1	Автоматическая загрузка реестров платежей	266
1.1.21.3	Групповые операции над договорами	267
1.1.21.3.1	Операция "Изменение статуса"	269
1.1.21.3.2	Операция "Добавление группы тарифов"	270
1.1.21.3.3	Операция "Открытие тарифных планов"	271
1.1.21.3.4	Операция "Закрытие тарифных планов"	272
1.1.21.3.5	Операция "Добавление (Удаление) модулей"	273
1.1.21.3.6	Операция "Добавление разрешённых услуг"	274
1.1.21.3.7	Операция "Удаление(Прерывание на период) разрешённых услуг"	275
1.1.21.3.8	Операция "Смена тарифа"	276
1.1.21.3.9	Операция "Добавление скрипта"	277
1.1.21.3.10	Операция "Установка шаблона комментария договору"	278
1.1.21.4	Сообщения пользователям	279
1.1.21.5	Индикатор лицензии	281
1.1.21.6	SQL Редактор	282

1.1.22	Администрирование и оптимизация	283
1.1.22.1	Конфигурация базы данных, память	284
1.1.22.2	Поддержка репликации	285
1.1.22.3	"Мусорные" базы данных	286
1.1.22.4	Настройка типа хранения помесячных и подневных таблиц в MySQL	287
1.1.22.5	Параметры запуска клиента	288
1.1.22.6	Дополнительные дефолтные настройки локализации, tz и т.д.	289
1.1.22.7	Импорт договоров	290
1.2	Расширение функциональности BGBilling	292
1.2.1	Управление динамическим кодом	293
1.2.2	Скрипты поведения	298
1.2.2.1	Создание скрипта поведения	299
1.2.2.2	Привязка динамически загружаемых Java-классов к скриптам поведения	300
1.2.2.3	Написание функций скрипта поведения на языке BGBS	301
1.2.2.4	Привязка скриптов поведения к договору	306
1.2.2.5	Обработка событий внешними обработчиками	308
1.2.3	Скрипты поведения глобальных событий	309
1.2.4	Глобальные скрипты	310
1.2.4.1	Глобальные скрипты с использованием динамических классов Java	311
1.2.4.2	Глобальные скрипты на языке BGBS	312
1.2.4.3	Периодическое выполнение глобальных скриптов	313
1.2.5	Резервные копии	314
1.2.6	Скрипты предобработки RADIUS запросов	315
1.2.7	Интеграция с внешними системами	316
1.2.7.1	Web API	320
1.2.7.2	Динамические Web-Сервисы	326
1.2.8	Общие рекомендации по разработке скриптов	327
1.3	Модули абонплат и разовых услуг	328
1.3.1	Модуль NPray (абонплаты)	329
1.3.1.1	Привязка абонплат к клиентам	331
1.3.1.2	Алгоритм начисления, примеры тарифов	333
1.3.1.2.1	Абонплаты, не зависящие от других модулей	334
1.3.1.2.2	Абонплаты, зависящие от наработки по объёму в других модулях	335
1.3.1.2.3	Абонплаты, зависящие от денежной наработки в других модулях	338
1.3.1.2.4	Абонплаты, пропорциональные количеству телефонов(+линий), логинов и сервисов	340
1.3.1.2.5	Абонплаты, зависящие от тарифных опций	341
1.3.1.2.6	Абонплаты, зависящие от других абонплат	342
1.3.1.3	Методики построения тарифных планов	343
1.3.1.4	Начисление	345
1.3.1.5	Дебетовые абонплаты	346
1.3.1.6	Групповые операции	348
1.3.1.6.1	Групповая операция "Добавление/прерывание абонплат"	349
1.3.2	Модуль RSCM	350
1.3.3	Модуль Subscription	353
1.3.3.1	Привязка подписок к клиентам	354
1.3.3.2	Примеры тарифных планов модуля	356
1.3.3.3	Групповые операции модуля Subscription	357
1.4	Модули Интернета	358
1.4.1	Модуль Inet	359
1.4.1.1	Общий алгоритм модуля, установка и принципы настройки серверов	361
1.4.1.1.1	Установка серверов Inet	362
1.4.1.1.2	Общая часть конфигурации	365
1.4.1.1.3	Слушатель InetRadiusListener	367
1.4.1.1.4	Слушатель InetDhcpListener	384
1.4.1.1.5	Слушатель InetFlowListener	406
1.4.1.1.6	Сбор статистики с помощью InetSnmpWorker	411
1.4.1.1.7	Настройка BGINetAccess сервера	412
1.4.1.1.8	Настройка BGINetAccounting сервера	414
1.4.1.1.9	Общий алгоритм настройки	417
1.4.1.2	Сущности модуля Inet	437
1.4.1.2.1	Трафики	438
1.4.1.2.2	Ресурсы	440
1.4.1.2.3	Типы устройств	447
1.4.1.2.4	Устройства	459
1.4.1.2.5	Опции	466
1.4.1.2.6	Типы сервисов	467
1.4.1.2.7	Сервисы	471
1.4.1.2.8	Сессии	479
1.4.1.3	Тарифные планы Inet	482
1.4.1.4	Монитор соединений Inet	488
1.4.1.5	Личный кабинет (web-статистика)	489
1.4.1.6	Переобработка логов	495

1.4.1.7	Переобсчет	497
1.4.1.8	Отчеты модуля Inet	498
1.4.1.9	Коды ошибок авторизации	499
1.4.1.10	Задачи планировщика модуля Inet	500
1.4.1.11	Настройка WiFi-агента для работы с модулем Inet	501
1.4.1.11.1	Описание WiFi-агента	502
1.4.1.11.2	Установка, настройка и запуск	506
1.4.1.11.3	Связь WiFi-агента с модулем "Карточки"	513
1.4.1.11.4	Защита WiFi-сети от ARP-спуфинга	514
1.4.1.11.5	Настройка ограничения скорости (шейпинг) для трафика WiFi-сети	516
1.4.1.11.6	Настройка REALM'ов	518
1.4.1.11.7	Web-интерфейс WiFi-портала	519
1.4.1.11.8	Активация на портале	520
1.4.1.12	Интеграция модуля Inet	521
1.4.1.12.1	Ericsson (RedBack) SmartEdge RSE	522
1.4.1.12.2	Управление через telnet, ssh, tcp, shell, mikrotik-api	526
1.4.1.12.3	IPoE с Mikrotik в качестве шейпера	533
1.4.1.12.4	СКАТ DPI BRAS	538
1.4.1.12.5	RDP IPoE BRAS (EcoNAT)	544
1.4.1.12.6	Интеграция с модулем Card	548
1.4.1.13	FAQ модуля Inet	549
1.4.2	Модуль IPN	558
1.4.2.1	Привязки услуг (категории трафика)	561
1.4.2.2	Создание источников и интерфейсов	565
1.4.2.3	Управление ресурсами IP-адресов	567
1.4.2.4	Добавление адресов абонентам	570
1.4.2.4.1	Настройка выделения адресов в шаблоне договора	574
1.4.2.5	Настройка сбора и обработки логов	575
1.4.2.5.1	Настройка коллектора в автономном режиме	577
1.4.2.5.2	Настройка коллектора в связке с flow-tools	579
1.4.2.5.3	Запуск коллектора	580
1.4.2.5.4	Наладка приёма данных коллектором в автономном режиме	581
1.4.2.5.5	Настройка обработки данных	582
1.4.2.6	Подсистема аудита	584
1.4.2.7	Тарификация	585
1.4.2.7.1	Создание тарифных планов	586
1.4.2.7.2	Запуск начисления	595
1.4.2.7.3	Начисление наработки за максимальные трафики (IPN)	597
1.4.2.8	Установка типа правила в тарифе.	598
1.4.2.9	Настройка шлюзов	600
1.4.2.9.1	Настройка типов шлюзов	602
1.4.2.9.2	Настройка шлюза	603
1.4.2.9.3	Типы правил	605
1.4.2.9.4	Добавление шлюза в договор.	
1.4.2.9.5	Выделение ресурса VLAN на шлюз.	610
1.4.2.9.6	Настройка портов шлюза.	612
1.4.2.9.7	Настройка шлюзов типа Manad	613
1.4.2.9.8	Настройка шлюзов типа Switch	625
1.4.2.9.9	Настройка шлюза BGRADIUSIPN	630
1.4.2.9.10	Настройка шлюза CISCO	633
1.4.2.9.11	Настройка шлюза DLINK 35xx, 38xx	637
1.4.2.9.12	Настройка сервера/шлюза DHCP	640
1.4.2.9.13	Настройка шлюза Mikrotik RouterOS	642
1.4.2.9.14	Настройка шлюза Cisco2 с коммутаторами	645
1.4.2.9.15	Реализация шлюза на языке BeanShell	652
1.4.2.10	Отчёты модуля, детализация	654
1.4.2.10.1	Детализация трафика за час	655
1.4.2.10.2	Детализация трафика за период	656
1.4.2.11	Web-интерфейс модуля	657
1.4.3	Модуль DialUp	660
1.4.3.1	Базовые понятия и алгоритм работы модуля DialUp	661
1.4.3.1.1	SNMP и NetFlow	665
1.4.3.1.2	Режимы работы RADIUS сервера	666
1.4.3.1.3	Статус соединения	667
1.4.3.1.4	Работа с соединением	668
1.4.3.1.5	Порядок тарификации	669
1.4.3.1.6	RADIUS атрибуты	670
1.4.3.2	Настройка модуля DialUp	671
1.4.3.3	REALMы	674
1.4.3.4	Наборы атрибутов	675
1.4.3.5	Выдача атрибутов соединения и выделение IP адресов	676
1.4.3.6	Настройка NASов	677

1.4.3.6.1	Скрипт предобработки запроса	684
1.4.3.6.2	Пересылка RADIUS Accounting в DialUP	685
1.4.3.7	Настройка RADIUS-сервера для DialUp	686
1.4.3.7.1	Установка BGRadiusDialup на Linux-платформу	687
1.4.3.7.2	Установка BGRadiusDialup на Windows платформу	688
1.4.3.7.3	Настройка radius.properties	689
1.4.3.7.4	Администрирование BGRadiusDialup	690
1.4.3.8	Настройка встроенного коллектора	699
1.4.3.8.1	Переобработка NetFlow трафиков	701
1.4.3.9	Настройка абонентов DialUp	702
1.4.3.9.1	Вкладка "IP-адрес"	703
1.4.3.9.2	Вкладка "Атрибуты RADIUS"	704
1.4.3.9.3	Вкладка "Ограничения"	706
1.4.3.9.4	Вкладка "Логи"	709
1.4.3.9.5	Вкладка "Пароль"	710
1.4.3.9.6	Перенос логинов	711
1.4.3.10	Настройка тарифных планов DialUp	713
1.4.3.10.1	Простейший тариф	714
1.4.3.10.2	Разделение стоимости по времени суток	715
1.4.3.10.3	Учётные периоды	716
1.4.3.10.4	Зависимость стоимости от объема	717
1.4.3.10.5	Комбинированные зависимости	720
1.4.3.10.6	Детализация по тарифу DialUp	721
1.4.3.10.7	Использование узла "Мультиуслуга"	722
1.4.3.10.8	Указание в тарифе свойств соединения	723
1.4.3.10.9	Уровни	728
1.4.3.10.10	Тарифные опции в DialUp	729
1.4.3.10.11	Тарифы с переоценкой всего потребленного трафика	730
1.4.3.10.12	Ограничение по NASam	731
1.4.3.10.13	Узел "Конфигурация тарифа"	732
1.4.3.11	Переобсчёт соединений	735
1.4.3.12	Монитор соединений	736
1.4.3.13	Интеграция с модулем "Карточки"	741
1.4.3.14	Отчёты	742
1.4.3.14.1	Отчёт по сессиям, детализация	743
1.4.3.14.2	Отчёт по наработке логинов	747
1.4.3.15	Web-интерфейс DialUp	748
1.4.3.16	Начисление наработки за максимальные трафики	749
1.4.3.17	Динамическое управление DNS-сервером	751
1.4.3.17.1	Пример настройки DNS сервера	752
1.4.3.17.2	Пример настройки модуля	754
1.4.3.17.3	Пример настройки логина	755
1.4.3.18	Настройка автозакрытия соединений	756
1.4.3.19	Поддержка CallBack	757
1.4.3.20	Настройка RADIUS-сервера с различными шлюзами CISCO	758
1.4.3.21	Настройка WiFi-агента для работы с модулем Dialup	759
1.4.3.21.1	Описание WiFi-агента (DialUp)	760
1.4.3.21.2	Установка, настройка и запуск (DialUp)	764
1.4.3.21.3	Связь WiFi-агента с модулем "Карточки" (DialUp)	771
1.4.3.21.4	Защита WiFi-сети от ARP-спуффинга (DialUp)	772
1.4.3.21.5	Настройка ограничения скорости (шейпинг) для трафика WiFi-сети (DialUp)	774
1.4.3.21.6	Настройка REALM'ов (DialUp)	776
1.4.3.21.7	Web-интерфейс WiFi-портала (DialUp)	777
1.5	Модули Телефонии	778
1.5.1	Модуль Phone	779
1.5.1.1	Подготовка логов	781
1.5.1.2	Настройка загрузки и обработки логов	782
1.5.1.3	Географические коды, карты зон и цен	787
1.5.1.3.1	Карта зон	791
1.5.1.3.2	Карта цен	795
1.5.1.4	Управление ресурсами номеров	797
1.5.1.4.1	Добавление ресурсов	798
1.5.1.4.2	Слежение за ресурсами	800
1.5.1.5	Учёт абонентского трафика	802
1.5.1.5.1	Тарифы на местную связь	807
1.5.1.5.2	Тарифы на МГМН-связь	812
1.5.1.5.3	Специфичные тарифные узлы модуля	820
1.5.1.5.4	Отчёты в клиенте	825
1.5.1.5.5	Отчёты в Web-интерфейсе	832
1.5.1.6	Учёт операторского трафика	833
1.5.1.6.1	Редактирование правил	834
1.5.1.6.2	Отчёты операторов	838

1.5.1.6.3	Транзитные операторы	839
1.5.1.6.4	Операторские отчеты	840
1.5.1.7	Тарификация при работе по агентской схеме	845
1.5.1.7.1	Составление тарифов при агентской схеме	848
1.5.1.7.2	Отчетность	849
1.5.1.8	Отключение абонентов	850
1.5.1.9	Web-интерфейс Phone	852
1.5.1.10	Черно-белые списки	857
1.5.2	Модуль VoiceIP	860
1.5.2.1	Базовые понятия и алгоритм работы модуля VoiceIP	861
1.5.2.2	Настройка модуля VoiceIP	864
1.5.2.3	Настройка режимов поиска и типов логинов	866
1.5.2.4	Настройка NASов модуля VoiceIP	869
1.5.2.4.1	Скрипт преобработки RADIUS-запроса	872
1.5.2.5	Монитор	873
1.5.2.6	Настройка RADIUS-сервера для VoiceIP	875
1.5.2.6.1	Установка BGRadiusVoip на платформу Linux	876
1.5.2.6.2	Установка BGRadiusVoip на платформу Windows	877
1.5.2.6.3	Настройка файла radius.properties	878
1.5.2.6.4	Управление BGRadiusVoip	879
1.5.2.7	Клиентская настройка	881
1.5.2.8	Интеграция VoiceIP с модулем "Карточки"	884
1.5.2.9	Настройка тарифных планов VoiceIP	885
1.5.2.9.1	Тарификация по префиксам в модуле VoiceIP	887
1.5.2.9.2	Тарификация по зонам модуля VoiceIP	889
1.5.2.9.3	Тарификация по карте цен модуля VoiceIP	890
1.5.2.9.4	Смешанный режим	891
1.5.2.9.5	Модификация стоимости звонка	892
1.5.2.9.6	Импорт/экспорт тарифов	893
1.5.2.10	Работа с операторами	894
1.5.2.10.1	Старая схема	895
1.5.2.10.2	Новая схема	897
1.5.2.10.3	Оценка операторов	899
1.5.2.11	Отчёты VoiceIP	900
1.5.2.12	Web-интерфейс VoiceIP	902
1.5.2.13	Переобсчёт сессий	903
1.5.2.14	Создание договоров абонентом через Web	904
1.5.3	Модуль Voice	905
1.5.3.1	Базовые понятия и алгоритм работы модуля Voice	906
1.5.3.1.1	Обработка логов CDR	907
1.5.3.1.2	Работа по протоколу Radius	908
1.5.3.2	Настройка модуля	911
1.5.3.2.1	Настройка обсчета по логам CDR.	912
1.5.3.2.2	Настройка обсчета по протоколу Radius.	913
1.5.3.3	Общий алгоритм первичной настройки модуля Voice.	914
1.5.3.3.1	Общий алгоритм первичной настройки модуля Voice для обсчета логов CDR.	915
1.5.3.3.2	Общий алгоритм первичной настройки модуля Voice для обсчета по Radius.	917
1.5.3.4	Сущности модуля Voice	919
1.5.3.4.1	Типы аккаунтов	920
1.5.3.4.2	Аккаунты	923
1.5.3.4.3	Ресурсы номеров	929
1.5.3.4.4	Типы устройств модуля Voice	930
1.5.3.4.5	Устройства модуля Voice	932
1.5.3.4.6	Географические справочники	934
1.5.3.4.7	Сессии модуля Voice	944
1.5.3.5	Настройка и установка серверов	945
1.5.3.5.1	Установка серверов Voice	946
1.5.3.5.2	Общая часть конфигурации Voice-серверов.	951
1.5.3.5.3	Настройка BGVoiceProcessor	952
1.5.3.5.4	Настройка BGVoiceRadius	953
1.5.3.6	Обработка логов CDR в модуле Voice	955
1.5.3.6.1	Обработчик логов CDR	962
1.5.3.6.2	Runtime обработчик логов CDR	963
1.5.3.7	Настройка NAS-ов для Radius-запросов.	965
1.5.3.7.1	Обработчик процессора протокола модуля Voice	970
1.5.3.8	Тарификация модуля Voice.	972
1.5.3.8.1	Тарифы на местную связь модуля Voice.	973
1.5.3.8.2	Тарифы на МГМН и зонную связь.	974
1.5.3.8.3	Специфичные тарифные узлы модуля Voice	980
1.5.3.9	Клиентские отчеты модуля Voice	981
1.5.3.10	Монитор соединений модуля Voice	983
1.5.3.11	Переобсчет сессий.	985

1.5.3.12	Учет операторского трафика модуля Voice	986
1.5.3.12.1	Учет операторского трафика для CDR	987
1.5.3.12.2	Учет операторского трафика для Radius	989
1.5.3.13	Тарификация абонента по агентской схеме	992
1.5.3.14	Управление доступом абонентов	995
1.5.3.15	Личный кабинет модуля Voice	996
1.5.3.16	Коды ошибок авторизации при Radius запросах	997
1.5.3.17	Переобработка Radius-логов	998
1.5.4	Mango Office	999
1.5.5	Модуль RunTel	1001
1.6	Модули ТВ	1005
1.6.1	Модуль CerberCrypt	1006
1.6.1.1	Настройка модуля CerberCrypt	1007
1.6.1.1.1	Настройка каналов/пакетов	1008
1.6.1.1.2	Тестовый	1013
1.6.1.1.3	CDCAS 3.0	1014
1.6.1.1.4	CerberCrypt	1015
1.6.1.1.5	CTI CAS 4.0	1016
1.6.1.1.6	Conax	1017
1.6.1.1.7	Conax 4.0	1018
1.6.1.1.8	Gospell	1019
1.6.1.1.9	Griffin	1020
1.6.1.1.10	NordE/CTI	1021
1.6.1.1.11	Pisys Irdeto	1022
1.6.1.1.12	enigo CAS comteza	1023
1.6.1.1.13	XCrypt	1024
1.6.1.1.14	DRECrypt	1025
1.6.1.2	Настройка клиентов	1026
1.6.1.3	Дополнительные возможности для Irdeto Pisys, CTI/NordE итд	1028
1.6.1.4	Типы оборудования	1031
1.6.1.5	«Мультирум»	1032
1.6.1.6	Настройка задач планировщика	1033
1.6.1.6.1	Сопоставление договоров картам	1034
1.6.1.6.2	Установка актуальных статусов пакетов в картах договоров	1035
1.6.1.6.3	Начисление CerberCrypt	1036
1.6.1.6.4	Блокировка должников	1037
1.6.1.6.5	Обновление подписок	1038
1.6.1.6.6	Постепенное продление подписок	1039
1.6.1.6.7	Контроль синхронизации	1040
1.6.1.7	Настройка тарифных планов	1041
1.6.1.8	Возможности Web-интерфейса модуля	1042
1.6.1.9	Создание договоров пользователем через Web	1044
1.6.1.10	Виртуальный кинозал	1045
1.6.1.11	Лог синхронизаций	1047
1.6.2	Модуль TV	1048
1.6.2.1	Продукты, Сервисы, Каналы	1050
1.6.2.1.1	Продукты TV	1051
1.6.2.1.2	Сервисы TV	1055
1.6.2.1.3	Каналы TV	1057
1.6.2.2	Типы аккаунтов TV	1058
1.6.2.3	Типы устройств TV	1061
1.6.2.4	Устройства TV	1062
1.6.2.5	Аккаунты TV	1064
1.6.2.6	Тарифы TV	1067
1.6.2.7	Переобсчет TV	1069
1.6.2.8	Интеграция	1071
1.6.2.8.1	Установка серверных приложений TV	1072
1.6.2.8.2	24часаТВ	1076
1.6.2.8.3	Compass (compass.tv, ТИТВ)	1083
1.6.2.8.4	CTI TVEngine	1088
1.6.2.8.5	IPTV Портал (iptvportal.ru)	1093
1.6.2.8.6	FrontStage Middleware (Mediastage.tv, TelecomTV, BCC)	1105
1.6.2.8.7	LifeStream (Смотрёшка, Ifstrm.tv)	1107
1.6.2.8.8	Ministra TV platform (ex Stalker Middleware)	1111
1.6.2.8.9	MOOVI	1119
1.6.2.8.10	TVIP Media (TVIP TMS)	1123
1.6.2.8.11	Smarty (microimpulse.ru)	1127
1.6.2.8.12	Wink	1130
1.6.2.8.13	Модуль Inet (продукты TV - опции Inet)	1136
1.6.2.8.14	NetUP IPTV	1137
1.6.3	Модуль Vidimax	1141
1.6.4	Модуль Megogo	1143

1.7 Модули платежных систем	1145
1.7.1 SSLv3	1146
1.7.2 Модуль Alfabank (АО "Альфа банк"	1147
1.7.3 Модуль Assist	1149
1.7.3.1 Назначение модуля Assist	1150
1.7.3.2 Настройка модуля Assist	1151
1.7.3.3 Оплата через систему Assist	1153
1.7.3.4 Мониторинг и администрирование платежей	1154
1.7.3.5 Настройка автоматической установки статусов платежей	1156
1.7.3.6 Настройка сети, фаервола и т.д.	1157
1.7.3.7 Важные замечания	1158
1.7.4 Модуль Buyemoney (beta-версия)	1159
1.7.4.1 Назначение модуля Buyemoney (beta-версия)	1160
1.7.4.2 Структура и настройка модуля	1161
1.7.4.3 Настройка gpg-подписи для Yandex.Деньги	1163
1.7.4.4 Настройка WebMoney	1166
1.7.4.5 Использование модуля Buyemoney	1167
1.7.5 Модуль BVCom	1170
1.7.5.1 Оплата через систему BVCom	1171
1.7.5.2 Мониторинг платежей	1173
1.7.5.3 Возврат платежей	1174
1.7.6 Модуль Chronopay	1175
1.7.7 Модуль Gorod	1176
1.7.8 Модуль MOBI.Деньги	1179
1.7.9 Модуль Moneta	1180
1.7.10 Модуль MPS	1182
1.7.10.1 Настройка модуля MPS	1184
1.7.10.1.1 ОСМП/Qiwi, Empray, Pegas, Rapida, Comepay	1188
1.7.10.1.2 CyberPlat	1191
1.7.10.1.3 XPlat	1192
1.7.10.1.4 Eport	1193
1.7.10.1.5 SFOUR PayBox Alternative	1194
1.7.10.1.6 ОПТИМА plus	1195
1.7.10.1.7 Elecsnet	1196
1.7.10.1.8 Юникасса	1197
1.7.10.1.9 Quickpay	1198
1.7.10.1.10 Sberbank	1199
1.7.10.1.11 Сбербанк (sbrf)	1200
1.7.10.1.12 Bisys	1201
1.7.10.1.13 Paynet	1202
1.7.10.1.14 Click	1204
1.7.10.1.15 MBank	1205
1.7.10.1.16 Kaspі	1206
1.7.10.1.17 ВТБ	1207
1.7.10.2 Сертификаты	1208
1.7.10.3 Менеджер платежей	1211
1.7.10.4 Сверка платежей MPS	1212
1.7.10.5 Web-интерфейс модуля MPS	1213
1.7.11 Модуль MTSBank (ПАО «МТС-Банк»)	1214
1.7.12 Модуль Netpay	1215
1.7.13 Модуль OnPay	1216
1.7.14 Модуль PayKeeper	1217
1.7.15 Модуль Paylinks	1219
1.7.15.1 Назначение модуля Paylinks	1220
1.7.15.2 Настройка модуля Paylinks	1221
1.7.15.3 Использование модуля Paylinks	1222
1.7.16 Модуль Paymaster	1225
1.7.17 Модуль PayOnline	1228
1.7.18 Модуль Payture	1232
1.7.18.1 Назначение модуля Payture	1233
1.7.18.2 Настройка модуля Payture	1234
1.7.18.3 Проведение платежей Payture	1235
1.7.18.4 Мониторинг платежей Payture	1236
1.7.18.5 Возврат платежей Payture	1237
1.7.19 Модуль PSB (Промсвязьбанк)	1238
1.7.20 Модуль PSCB (АО «Петербургский социальный коммерческий банк»)	1239
1.7.21 Модуль Qiwi	1240
1.7.22 Модуль RBK.Money	1244
1.7.23 Модуль RFI e-Commerce (шлюз ЗАО «РФИ БАНК»)	1247
1.7.24 Модуль Robokassa	1249
1.7.25 Модуль RuRuPay	1251
1.7.26 Модуль Sberbank (АО "Сбербанк")	1254

1.7.27 Модуль SimplePay	1257
1.7.28 Модуль Tinkoff	1258
1.7.29 Модуль Uniteller	1259
1.7.30 Модуль WebMoney	1263
1.7.31 Модуль ЮKassa	1267
1.8 Модуль Card	1272
1.8.1 Дилеры	1273
1.8.2 Работа с карточками	1274
1.8.3 Генератор логинов и паролей для модуля "Карточки"	1283
1.8.3.1 Установка генератора	1284
1.8.3.2 Использование генератора	1285
1.8.4 Web-интерфейс	1287
1.8.5 Web-активация	1288
1.8.6 Суперкарты	1291
1.8.7 IVR-Система	1292
1.8.8 Удалённые платежи	1294
1.8.8.1 Настройка модуля Card	1295
1.8.8.2 Настройка дилера	1298
1.8.8.3 Web-клиент	1301
1.8.8.4 Сверка платежей	1303
1.9 Модуль RentSoft	1304
1.9.1 Инструкции по активации услуг компании RentSoft	1305
1.9.2 Клиентская часть модуля RentSoft	1306
1.9.3 Работа с юридическими лицами	1307
1.9.4 Web интерфейс модуля RentSoft	1310
1.10 Модуль TrayInfo	1312
1.10.1 Активация TrayInfo клиентом	1313
1.10.2 Настройка утилиты TrayInfo	1314
1.10.3 Настройка утилиты TrayInfo3	1315
1.10.4 Настройка утилиты TrayInfo для Mac Os	1316
1.10.5 Отображение в утилите произвольной информации вместо баланса	1317
1.11 Модуль Bill	1318
1.11.1 Настройка модуля Bill	1319
1.11.1.1 Настройка позиций	1323
1.11.1.1.1 Экстакторы	1325
1.11.1.1.2 Детализация по тарифу	1326
1.11.1.2 Номерной пул	1327
1.11.1.3 Типы документов	1328
1.11.1.4 Настройка реквизитов организации(ий) и банковских счетов	1330
1.11.2 Настройка параметров договора	1332
1.11.3 Выставление счетов и счетов-фактур администратором	1334
1.11.4 Работа со счетами	1336
1.11.4.1 Первичная подготовка для курьерской службы	1338
1.11.5 Работа со счетами-фактурами, Актами и УПД	1339
1.11.5.1 Настройка УПД	1340
1.11.6 Работа с актами-сверки	1343
1.11.7 Панель просмотра документов	1344
1.11.8 Генерация печатных форм	1346
1.11.9 Отчёты договора модуля	1347
1.11.10 Web-интерфейс абонента	1349
1.11.11 Тонкости интеграции с внешними (1С) системами	1352
1.11.12 Групповые операции модуля Bill	1353
1.12 Модуль DBA	1354
1.13 Модуль DrWeb	1356
1.13.1 Управление подписками	1358
1.13.2 Web-интерфейс модуля DrWeb	1359
1.13.3 Настройка тарифных планов модуля DrWeb	1360
1.14 Модуль E-Mail	1361
1.14.1 Установка и настройка модуля E-Mail	1362
1.14.1.1 Домены	1363
1.14.1.2 Настройка конфигурации	1364
1.14.1.3 Хранилище почтовых аккаунтов	1366
1.14.1.3.1 LDAP база	1367
1.14.1.3.2 SQL база	1368
1.14.1.4 Настройка планировщика для модуля E-Mail	1369
1.14.2 Использование модуля E-Mail	1370
1.15 Модуль Reports	1373
1.15.1 Отчёты основного модуля	1374
1.15.2 Отчёты модуля DialUP	1382
1.15.3 Отчёты модуля IPN	1387
1.15.4 Отчёты модуля IP Телефония	1389
1.15.5 Отчёты плагина CRM	1391

1.15.6	Отчёты модуля Телефония (Phone)	1392
1.15.7	Отчёты модуля Карточки	1394
1.15.8	Отчёты модуля RSCM	1395
1.15.9	Отчёты модуля CerberCrypt	1396
1.15.9.1	Отчет "Количество абонентов"	1397
1.15.9.2	Отчет "Наработка пакетов"	1398
1.15.10	Создание собственных отчетов	1399
1.15.10.1	Настройка фильтра	1400
1.15.10.2	Отчёты JasperReports.	1414
1.15.10.3	Табличные отчёты	1418
1.15.11	Мобильные отчеты	1420
1.15.11.1	Настройка сервера	1421
1.15.11.2	Собственные отчеты.	1422
1.15.12	Печать квитанций из LibreOffice	1428
1.16	Модуль Mobile	1435
1.17	Модуль Smarthouse	1437
1.17.1	Система видеонаблюдения Forpost (Форпост)	1438
1.18	Модуль СОРМ	1439
1.18.1	Рекомендации по СОРМ 3	1440
1.18.2	Начало работы. Общая конфигурация.	1443
1.18.3	Настройка выгрузки справочных данных	1449
1.18.4	Выгрузка. Настройка расписания выгрузки	1454
1.18.5	Статистика выгрузок	1455
1.18.6	Переопределение выгружаемых файлов	1456
1.18.7	Просмотр сформированных файлов	1459
1.18.8	Формирование радиус-пакетов для СОРМ	1461
1.19	Модуль BotManager	1463
1.19.1	Создание ботов	1464
1.19.2	Создание диалогов	1465
1.19.3	Создание элементов диалога	1469
1.19.4	Отправка собственных сообщений в чаты	1471
1.20	Модуль RCMts	1473
1.21	Плагины	1475
1.21.1	Плагин Assistant	1476
1.21.1.1	Заполнение элементов формы	1480
1.21.1.2	Примеры Действий	1482
1.21.1.3	Создание таблицы в мастере	1485
1.21.1.4	Мобильное приложение для плагина	1492
1.21.1.5	Экспорт/Импорт мастеров плагина	1494
1.21.2	Плагин Bonus	1497
1.21.2.1	Бонусные приходы	1498
1.21.2.2	Бонусные расходы	1499
1.21.2.3	Бонусный баланс	1502
1.21.2.4	Бонусные программы	1503
1.21.2.4.1	Операционная программа	1504
1.21.2.4.2	Динамические программы	1505
1.21.2.5	Web-интерфейс плагина Bonus	1508
1.21.2.6	Шаблоны договоров плагина Bonus	1509
1.21.2.7	Групповые операции плагина Bonus	1510
1.21.3	Плагин CashCheck	1512
1.21.3.1	Назначение и структура плагина, архитектура системы	1513
1.21.3.2	Настройка плагина	1515
1.21.3.2.1	Настройка плагина в биллинге	1516
1.21.3.2.2	Настройка внешнего вида чеков (динамический код)	1517
1.21.3.2.3	Разделение по отделам в ККМ, задание типа платежа, налога, признак электронного чека и т.д.	1520
1.21.3.2.4	Автопечать платежей	1521
1.21.3.3	Использование плагина	1522
1.21.3.3.1	Очередь печати	1523
1.21.3.3.2	Лог распечатанных платежей	1524
1.21.3.3.3	Выбор принтера, отчёты, сервис	1525
1.21.3.3.4	Печать чека	1526
1.21.3.3.5	Настройка галочки в диалоге прихода платежа	1528
1.21.3.4	Фискализаторы	1529
1.21.3.4.1	OFD.ru Ferma	1530
1.21.3.4.2	АТОЛ-Онлайн v4	1531
1.21.3.4.3	Комтет Касса	1532
1.21.3.4.4	Локальные ККТ (сервер печати)	1534
1.21.3.4.5	Пустой фискализатор	1535
1.21.3.4.6	Фискализатор на динамическом классе	1536
1.21.3.5	CashCheck-сервер (сервер печати)	1537
1.21.3.5.1	Настройка сервера печати и оборудования, поддерживаемые устройства	1538
1.21.3.5.2	Запуск сервера печати	1546

1.21.3.5.3	Запуск двух копий сервера	1547
1.21.3.5.4	Тестирование	1548
1.21.3.5.5	Анализ ошибок и логирование	1549
1.21.4	Плагин CRM	1550
1.21.4.1	Журнал звонков	1551
1.21.4.2	Журнал проблем	1553
1.21.4.3	Журнал задач	1556
1.21.4.4	Журнал работ	1559
1.21.5	Плагин Dispatch	1560
1.21.5.1	Общий обзор	1561
1.21.5.2	Типы контактов	1562
1.21.5.3	Поиск по контактам	1563
1.21.5.4	Методы отправки	1564
1.21.5.5	Создание и редактирование рассылки	1566
1.21.5.6	Подписка на рассылки	1568
1.21.5.7	Создание и редактирование сообщений	1571
1.21.5.7.1	Рассылки модулей	1573
1.21.5.8	Условия отправки	1574
1.21.5.8.1	Отправка по событию	1575
1.21.5.8.2	Отправка по балансу	1576
1.21.5.8.3	Ограничение частоты отправки	1577
1.21.5.8.4	Ограничение по группе договора	1578
1.21.5.8.5	Ограничение по адресу подписчика	1579
1.21.5.8.6	Указание сервиса модуля Inet	1580
1.21.5.8.7	Указание логина модуля VoiceIP	1581
1.21.5.8.8	Указание логина модуля DialUp	1582
1.21.5.8.9	Пользовательское условие отправки	1583
1.21.5.9	Web-интерфейс плагина Dispatch	1584
1.21.5.10	Конвертирование рассылок с версии 5.1	1586
1.21.6	Плагин Documents	1587
1.21.7	Плагин HelpDesk	1600
1.21.7.1	Настройка групп и категорий	1602
1.21.7.2	Статусы тем	1604
1.21.7.3	Работа с сообщениями	1605
1.21.7.4	Работа с пакетами обращений	1607
1.21.7.5	Web-интерфейс клиента	1608
1.21.7.6	Оповещение (для пользователей биллинга)	1611
1.21.7.7	Автозакрытие тем	1614
1.21.8	Плагин Organizer	1615
1.21.8.1	Общий обзор плагина Organizer	1616
1.21.8.2	Добавление задания	1618
1.21.8.3	Просмотр текущих заданий	1619
1.21.8.4	Поиск заданий	1620
1.21.8.5	Выполнение заданий	1621
1.21.8.6	Журнал	1622
1.21.9	Плагин SBPilot	1623
1.21.10	Плагин КЛАДР	1628
1.21.11	Плагин ФИАС	1631
1.22	Инструкция по обновлению	1632
1.22.1	Инструкция по обновлению с версии 7.1 на версию 7.2	1633
1.22.2	Инструкция по обновлению с версий 3.5-7.1 до версии 7.2	1634
1.22.3	Инструкция по обновлению с версии 7.2 на версию 8.0	1641
1.22.4	Инструкция по обновлению на версию 8.YYMM	1642
1.23	Личный кабинет (не поддерживается с версии 9.2204+)	1645
1.23.1	Установка личного кабинета	1647
1.23.2	Настройка личного кабинета	1651
1.23.2.1	Основная конфигурация (mygbilling-conf.groovy)	1654
1.23.2.2	Конфигурация меню (mygbilling-menu.groovy)	1659
1.23.2.3	Конфигурация приема платежей (mygbilling-payment.groovy)	1662
1.23.3	Кастомизация личного кабинета	1666
1.23.3.1	Создание собственной страницы	1669
1.24	Содержание	1673

BGBilling

Добро пожаловать

на главную страницу документации **BG Биллинг**. На данной странице расположены полезные ссылки, недавно измененные страницы и последние комментарии. В панели слева расположено дерево страниц документации, в нижней части страницы - список вложенных (дочерних страниц). На страницах документации присутствуют ссылки "Предыдущая" и "Следующая", также срабатывающие при нажатии на клавиатуре клавиш влево и вправо.

[Перейти к содержанию](#)

Ссылки

- [Установка биллинга](#)
- [Документация в формате PDF одним файлом](#)
- [API-документация \(Javadoc\) для разработки скриптов поведения](#)
- [Инструкция по обновлению с версии 7.1 на версию 7.2](#)
- [Инструкция по обновлению с версий 3.5-7.1 до версии 7.2](#)
- [RSS обновлений в документации](#)

Недавно обновлено



[Кирилл Сергеев](#)

[Операционная программа обновлено апр 03, 2023 изменения](#)
[Установка планировщика \(с версии 8.2107\) обновлено мар 21, 2023 и изменения](#)
[Привязка подписок к клиентам обновлено мар 15, 2023 изменения](#)
[Дебетовые абонплаты обновлено мар 15, 2023 изменения](#)



[Валерий Зернов](#)

[Загрузка платежей и расходов из файла обновлено мар 07, 2023 изменения](#)



[Кирилл Сергеев](#)

[Модуль Sberbank \(АО "Сбербанк"\) обновлено фев 16, 2023 изменения](#)



[Валерий Зернов](#)

[Заполнение элементов формы обновлено фев 06, 2023 изменения](#)
[image2023-2-6_13-56-7.png](#) прикреплено фев 06, 2023



[Кирилл Сергеев](#)

[Тарифные опции обновлено янв 29, 2023 изменения](#)
[image2023-1-29_1-14-22.png](#) прикреплено янв 29, 2023
[image2023-1-29_1-13-56.png](#) прикреплено янв 29, 2023
[image2023-1-29_1-11-24.png](#) прикреплено янв 29, 2023
[image2023-1-29_1-10-10.png](#) прикреплено янв 29, 2023
[Тарифные переменные обновлено янв 29, 2023 изменения](#)
[image2023-1-29_1-6-54.png](#) прикреплено янв 29, 2023
[image2023-1-29_1-5-52.png](#) прикреплено янв 29, 2023
[image2023-1-29_1-2-7.png](#) прикреплено янв 29, 2023
[image2023-1-29_1-1-49.png](#) прикреплено янв 29, 2023
[image2023-1-29_0-59-57.png](#) прикреплено янв 29, 2023

Последние комментарии

[image2023-1-29_0-59-12.png](#) прикреплено янв 29, 2023

Описание основной части программы BGBilling

Как построено данное руководство

Данное руководство в пошаговом режиме описывает развёртывание системы BGBilling.

Мы настоятельно рекомендуем вам прочитать раздел [Описание основной части программы](#) в полном объёме и последовательно. Это поможет вам лучше представить общую идеологию системы и не теряться в главах-описаниях модулей и плагинов. Последующие за первым разделы вы можете читать в зависимости от тех модулей/плагинов, которые вы будете использовать.

В руководстве принято несколько простых соглашений:

1. Все названия файлов, пунктов меню, частей графического интерфейса, пути файловой системы, URL и т.п. выделены **жирным** шрифтом;
2. Названия пунктов меню выглядят так: **Меню 1=>Пункт 1**;
3. Всё заключённое в скобки <текст>, либо {текст} следует трактовать как (подставьте вместо этих скобок то, что описано в тексте);
4. Иногда названия скриптов указывается таким образом: **radius.sh(bat) ps**, что следует трактовать как: запустите **radius.sh ps** для Linux-платформы, либо **radius.bat ps** для Windows-платформы.

Вам также потребуются навыки запуска консольных приложений с набором аргументов. Для пользователей Linux это не составит проблемы, для Windows мы рекомендуем сразу установить и использовать консольные оболочки, например, [FAR Manager](#), либо пользоваться пунктом меню **Пуск=>Выполнить**, где сначала запускать командный интерпретатор **cmd** (чёрное окно с командной строкой), а далее в нем вводить команды.

Данное руководство описывает вопросы настройки непосредственно биллинга, вопросы настройки вспомогательного ПО, примеры тарифов и более подробные описания решений на базе биллинга вы можете найти в [Wiki](#). Вы также можете описать там ваши интересные разработки.

Логическая структура биллинга

В системе BGBilling можно выделить следующие основные подсистемы:

- ядро системы;
- плагины ядра;
- модули.

Ядро системы выполняет следующие функции:

- подключение модулей и плагинов;
- ведение перечня услуг;
- ведение справочников;
- ведение базы договоров;
- ведение базы объектов;
- ведение баланса договоров, истории приходов/расходов;
- СРД - система разграничения доступа пользователей к функциям ядра, плагинов и модулей;
- некоторые дополнительные функции.

Плагины - это программные компоненты, расширяющие функционал ядра.

Плагины устанавливаются в систему и могут быть активированы/деактивированы администратором в меню **Плагины=>Настройка плагинов**. После инсталляции в систему плагины нельзя удалить, можно только деактивировать. Потребность в отключении плагина может возникнуть, например, при истечении тестового периода лицензии на плагин.

При отсутствии лицензии на активный плагин система будет выдавать предупреждения.

Модули - это программные компоненты, расширяющие функционал ядра и предоставляющие, обычно, функционал связанный с тарификацией тех или иных услуг в биллинге.

Экземпляры модулей и перечни услуг в них создаются администратором в меню **Модули=>Редактор модулей и услуг**. Экземпляры модулей могут быть впоследствии удалены вместе со всеми данными.

Отличия плагинов и модулей в следующем:

1. плагины никогда не предоставляют функционала по тарификации услуг, соответственно, услуг в плагинах нет;
2. плагины не подключаются к договору в явном виде посредством услуг как модули, они существуют глобально в системе;
3. один и тот же модуль, в отличие от плагина, может быть создан в биллинге в нескольких экземплярах (например, вы можете создать DialUp и VPN модули как 2 экземпляра модуля DialUp).

Услуги - способ разделения наработки договора по типам. Например, в модуле DialUp могут быть определены услуги: Входящий трафик, Исходящий трафик, Время. Наработка в договоре будет начислена с разбиением по данным услугам.

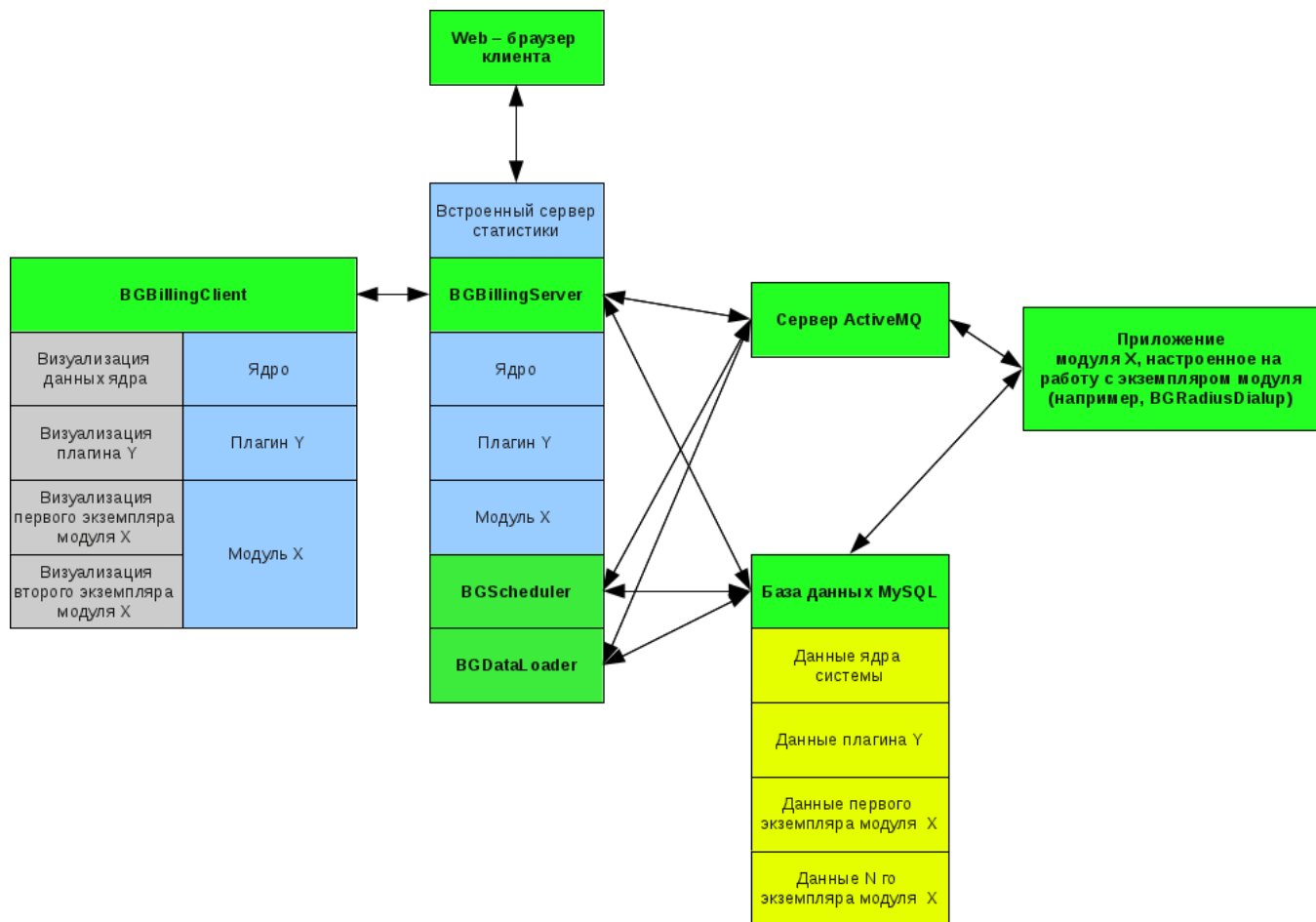
Договор - это ключевое понятие системы. В терминах BGBilling договор - это отдельная сущность, содержащая набор параметров, подключённых к нему модулей и тарифного плана, по которым услуги модулей тарифицируются. Каждый договор обладает отдельным балансом.

Баланс представляет из себя остаток на счету договора, историю его платежей и расходов (списаний). Подключённые к договору модули начисляют в баланс наработку по различным услугам, уменьшая исходящий остаток. Баланс в системе ведётся ежемесячно для всех договоров, т. е. по истечении каждого месяца исходящий остаток баланса переходит на следующий месяц.

Биллинг работает в единой валюте, мультивалютность не поддерживается.

Программная структура биллинга

Данная биллинговая система выполнена в клиент-серверном варианте. Общая структура изображена на рисунке.



На представленной схеме цветами выделены следующие виды компонентов:

- **Зелёным** - отдельный процесс в операционной системе, запущенная программа;
- **Синим** - библиотеки модулей, плагинов, либо ядра (серверные или клиентские части);
- **Жёлтым** - часть базы данных;
- **Серым** - визуальное отображение в клиентском приложении.

Можно выделить несколько основных частей:

- **Серверная часть (BGBillingServer)** - обрабатывает запросы клиента и Web-запросы;
- **Клиентская часть (BGBillingClient)** - визуализирует работу с сервером, АРМ-оператора и администратора биллинга;
- **Web интерфейс пользователя (Web браузер клиента)** - позволяет пользователям просматривать и модифицировать свои параметры, а также получать оперативные отчёты по модулям (просмотр сессий, звонков и т.д.);
- **Сервер ActiveMQ** - сервер для обмена событиями между серверными приложениями биллинга;
- **База данных MySQL** - единое хранилище и связующее звено компонентов биллинговой системы.

Можно заметить, что приложения **BGBillingServer**, **BGScheduler**, **BGDataLoader** используют общие библиотеки (BGBillingServer/lib), но физически являются разными процессами.

Связь клиента с сервером биллинга осуществляется через HTTP-протокол, также к серверу может обращаться браузер клиента провайдера для получения доступа к странице статистики. К серверу биллинга могут одновременно обращаться большое число клиентских приложений. Более того, под видом клиента для получения данных или их модификации к серверу могут обращаться сторонние приложения (например, бухгалтерское ПО). При этом сервер биллинга также производит авторизацию и контроль прав доступа этого клиентского приложения.

Все приложения, за исключением BGBillingClient, называются **серверными приложениями** биллинга, либо просто **приложениями** биллинга. Связь между всеми серверными приложениями осуществляется через базу данных и сервер ActiveMQ.

Также на схеме изображено, что экземпляр модуля (отдельный пункт в меню **Модули**) является ничем иным, как обособленным блоком данных в БД.

Преимущества клиент-серверной технологии заключаются в:

- возможности удалённого управления серверной частью с помощью клиента;
- одновременном доступе неограниченного количества рассредоточенных операторов к данным биллинговой системы;
- независимости автономной работы сервера от наличия запущенного клиентского приложения;
- наличии единой точки доступа к биллингу: отсутствие базы данных на машине оператора позволяет жёстко контролировать права доступа, гарантировать целостность данных биллинга.

Установка биллинга

Особенности установки под различные платформы

Linux

Windows

Linux

Установка всего серверного ПО производится под пользователем root.

Если установка производится в виртуальной машине, то может быть ситуация, что BGBillingServer долго стартует. В этом случае скорее всего дело в генераторе случайных чисел, и Вам нужно будет поменять в файле **java.security** в JDK/JRE:

```
[root@localhost ~]#  
securerandom.source=file:/dev/random
```

на

```
[root@localhost ~]#  
securerandom.source=file:/dev/urandom
```

Стандартные действия при установке

При установке каждого серверного приложения необходимо удалить Windows скрипты и установить права исполнения для **.sh**-файлов:

```
rm -f *.bat && rm -f *.exe && rm -f *.ini && chmod 744 *.sh
```

В различных дистрибутивах Linux существуют разные схемы автоматического запуска служб при старте сервера. Рассмотрим **systemd** и **sysvinit**.

systemd

В поставляемых приложениях биллинга в каталоге **scripts** присутствуют файлы **.service** для systemd. Отредактируйте их, поправив пути к приложениям, скопируйте в каталог **/lib/systemd/system** (или **/etc/systemd/system**), выполните команду **systemctl daemon-reload**, а затем выполните для каждого файла (приложения):

```
[root@localhost ~]#  
systemctl enable <app>
```

Например, для приложения BGBillingServer файл **bgbilling.service**:

```
[root@localhost ~]#  
  
[Unit]  
Description=BGBilling Server  
After=network.target mysql.service activemq.service  
#Required=network.target mysqld.service activemq.service  
  
[Service]  
Type=forking  
ExecStartPre=/opt/bgbilling/BGBillingServer/script/wait-for.sh 127.0.0.1:3306 -t 60  
ExecStart=/opt/bgbilling/BGBillingServer/server.sh start  
ExecStop=/opt/bgbilling/BGBillingServer/server.sh stop  
  
[Install]  
WantedBy=multi-user.target
```

Обратите внимание на строку "After=network.target mysql.service activemq.service". Если в системе есть такие сервисы и они стартуют, то systemctl ожидает их старта и только после этого запускает сервис **bgbilling**. В зависимости от дистрибутива, сервис mysql может называться по другому, например, mysqld.service или mariadb.service.

Строка "ExecStartPre=/opt/bgbilling/BGBillingServer/script/wait-for.sh 127.0.0.1:3306 -t 60" выполняется перед стартом сервиса **bgbilling** - данный скрипт ждет 60 секунд, пока не откроется порт 127.0.0.1:3306. Таким образом выполняется ожидание старта MySQL сервера, т.к. в некоторых дистрибутивах сервис **mysql** считается запущенным еще до того как был открыт его порт. Или MySQL может быть на другой машине - в этом случае нужно поменять адрес на правильный или закомментировать строку (символ #).

После того, как мы скопировали **bgbilling.service** в каталог **/lib/systemd/system**, мы можем поставить его в автозагрузку:

```
[root@localhost ~]#  
systemctl enable bgbilling
```

и запустить:

```
[root@localhost ~]#  
systemctl start bgbilling
```

sysvinit

Эта система самая старая и простая и поддерживается большинством дистрибутивов. Со всеми серверными приложениями биллинга в каталоге **scripts** поставляются скрипты запуска с командами start и stop. Все поставляемые скрипты ориентированы на командный интерпретатор Bash, либо совместимый (проверена работа с Dash), ссылка на который должна располагаться в файле **/bin/sh**. В случае, если у вас используется другой интерпретатор, либо отсутствует ссылка - поправьте скрипты

Рассмотрим способ добавления службы **bgbilling**.

1) Скопируйте скрипт **bgcommonrc** в **/etc/init.d**, установите права на выполнение.

```
chmod 755 /etc/init.d/bgcommonrc
```

2) Выполните команду **runlevel**, чтобы узнать уровень запуска.

```
[root@bill-2 init.d]# runlevel  
N 3
```

3) Скопируйте скрипт службы в **/etc/init.d**, установите права на выполнение.

```
chmod 755 /etc/init.d/bgbilling
```

4) Перейдите в папку **/etc/rcN.d** (N - требуемый уровень запуска), где выполните команду.

```
ln -s /etc/init.d/bgbilling S99bgbilling
```

Для запуска/остановки службы используйте **/etc/init.d/bgbilling start (stop)**. Префикс ссылки **S99** задаёт порядок старта сервиса.

Windows

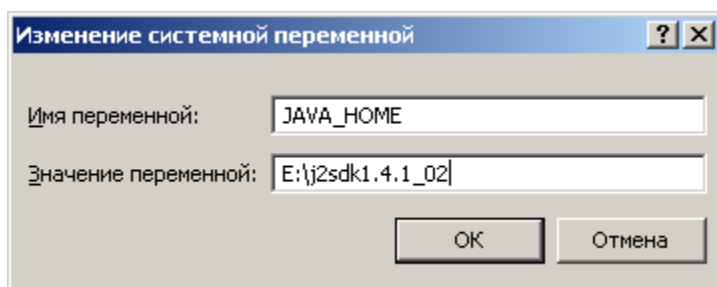
Следует учитывать, что ОС семейства Windows не являются оптимальными для запуска серверных приложений. Их применение снижает производительность дисковой подсистемы и оптимальность использования ресурсов аппаратуры. Кроме того, операционные системы данной серии менее гибко управляемы. Для запуска высоконагруженных биллинговых систем используйте ОС *NIX семейств. Ещё одним негативным фактором использования Windows является усложнение предоставления тех. поддержки по причине отсутствия полноценного shell доступа.

ОС семейства Windows **НЕ РЕКОМЕНДУЮТСЯ** разработчиками BGBilling для установки серверной части программы, однако хорошо подходят для запуска клиентского приложения.

Некоторые конфигурационные или шаблонные файлы компонентов системы (например, настройки BGCashCheckServer) используют кодировку UTF-8. Следует учесть, что, по традиции, в операционной системе Windows свои подходы к любой технологии и поэтому сохранённое, например, в "блокноте" не является валидным UTF-8, обратите на это внимание. Пользуйтесь текстовыми редакторами, сохраняющими правильно.

Установка всего серверного ПО производится под пользователем, обладающим администраторскими привилегиями на машине.

Для проверки и установки системных переменных окружения нажмите правой клавишей мышки по ярлыку **Мой компьютер** затем выберите **Свойства=>Дополнительно=>Переменные среды**. В нижнем окошке (**Системные переменные**) нажмите **Создать**, либо поправьте интересующую переменную.



Здесь и далее обратите внимание на необходимость установки переменной именно как системной, а не как пользовательской. Достаточно распространённая ошибка при настройке в дальнейшем службы - иначе работать не будет или будет работать неправильно. Также не забудьте перезагрузить систему после правки любой системной переменной окружения.

Все серверные приложения устанавливаются в данной ОС как службы и доступны через меню **Пуск=>Настройка=>Панель управления=>Администрирование=>Службы**. Следует учитывать, что ОС Windows не позволяет настроить порядок запуска служб, предоставляя взамен механизм зависимостей. Поэтому все службы по умолчанию помечены зависимыми от MySQL и ActiveMQ. В случае, если данные службы устанавливаются на отдельных машинах, необходимо удалить зависимость в .ini файле службы перед её инсталляцией (например **server.ini**, **scheduler.ini**).

Стандартные действия при установке

При установке каждого серверного приложения необходимо удалить все *.sh скрипты.

```
del /F *.sh
```

MySQL/MariaDB

- Установка
 - Linux
 - MariaDB
 - MariaDB из репозитория MariaDB
 - MySQL
 - Windows
- Настройка
 - Кодировки
 - utf8mb4
 - utf8
 - Дополнительная конфигурация службы Linux
 - Действия при смене datadir

MySQL/MariaDB-сервер (далее - MySQL) используется как постоянное хранилище для большинства данных биллинговой системы. Доступ к нему со стороны приложений биллинга осуществляется посредством сетевого соединения, поэтому MySQL может быть установлен на отдельной машине с любой поддерживаемой ОС. Для небольших баз, либо тестовых целей возможна установка MySQL на одну машину с сервером биллинга и другими серверными приложениями - весь процесс установки описан под этот случай и все конфигурации по умолчанию также ориентированы на этот случай.

Для работы биллинга необходим MySQL/MariaDB-сервер версии 5.7 и новее. Служба MySQL-сервера должна быть запущена до момента старта всех серверных приложений биллинга.

Подключение к MySQL для каждого приложения настраивается в .properties- или .xml- файле, например **data/data.properties** для сервера биллинга, планировщика и загрузчика логов, и **application.xml** для приложения BGInetAccess.

```
db.driver=com.mysql.jdbc.Driver
db.url=jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-
8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false
db.user=bill
db.pswd=bgbilling
```


Если база данных MySQL и приложения биллинга установлены на одной машине, то ничего менять не надо. В противном случае вместо 127.0.0.1 указывается IP-адрес машины с БД. Параметры **db.user** и **db.pswd** определяют имя пользователя и пароль, под которыми приложение будет подключаться к базе данных. Возможна настройка отдельного пользователя MySQL для каждого серверного приложения биллинга, это позволит сразу видеть источник запроса на MySQL-сервере.

Обратите внимание на параметры URL **characterEncoding** и **connectionCollation** - значения должны соответствовать реальной картине с кодировками (см. ниже).

Пользователь **bill** с паролем **bgbilling** создаётся при начальном создании БД при установке сервера биллинга (скрипт **dump.sql**).

Установка

Linux

-  Для установки MySQL/MariaDB требуются привилегии root или sudo.
- Обратите внимание, что для заливки дампа базы помимо сервера MySQL вам понадобится клиентское приложение mysql.

Возможно несколько вариантов установки сервера БД, рассмотрим их.

MariaDB

Для установки MariaDB на Linux, Вы можете воспользоваться предусмотренным системой способом установки. Например, для Linux с пакетным менеджером yum:

```
yum update
yum install mariadb-server mariadb-client
```

Для Linux с пакетным менеджером apt:

```
apt-get update
apt-get install mariadb-server mariadb-client
```

MariaDB из репозитория MariaDB

Для установки MariaDB версии отличной от поставляемой с текущей ОС воспользуйтесь репозитарием MariaDB: <https://downloads.mariadb.org/mariadb/repositories/>

Пример для Debian 9 и MariaDB 10.2

```
apt-get install software-properties-common dirmngr
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 0xF1656F24C74CD1D8
add-apt-repository 'deb [arch=amd64,i386,ppc64el] http://ams2.mirrors.digitalocean.com/mariadb/repo/10.2/debian stretch main'

apt-get update
apt-get install mariadb-server mariadb-client
```

Пример для CentOS 7 и MariaDB 10.2

```
{ \
  echo '[mariadb]'; \
  echo 'name = MariaDB'; \
  echo 'baseurl = http://yum.mariadb.org/10.2/centos7-amd64'; \
  echo 'gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB'; \
  echo 'gpgcheck=1'; \
} > /etc/yum.repos.d/MariaDB.repo

yum install MariaDB-server MariaDB-client
```

MySQL

Для некоторых временных зон может потребоваться выполнить команду

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
```

после чего в **my.cnf (server.cnf)** прописать

```
default-time-zone="Europe/Moscow"
```

Если не выполнять команду можно указывать в формате **default-time-zone="+03:00"**

Если не сделать этого биллинг для временной зоны MSK ругается

java.sql.SQLException: The server time zone value 'MSK' is unrecognized or represents more than one time zone. You must configure either the server or JDBC driver (via the serverTimezone configuration property) to use a more specific time zone value if you want to utilize time zone support.

Для установки MySQL воспользуйтесь репозитарием MySQL для вашей операционной системы: <https://dev.mysql.com/downloads/>

Пример для Debian или Ubuntu

Выполните:

```
curl -fsSL https://dev.mysql.com/get/mysql-apt-config_0.8.9-1_all.deb -o mysql-apt-config.deb
dpkg -i mysql-apt-config.deb
```

выберите версию MySQL, нажмите OK, затем:

```
apt-get update
apt-get install mysql-server mysql-client
```

При установке Вам предложат указать пароль для пользователя root MySQL.

Подробнее: <https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/>

Пример для CentOS 7, Red Hat Enterprise Linux 7

```
wget http://repo.mysql.com/mysql57-community-release-el7.rpm
rpm -ivh mysql57-community-release-el7.rpm
yum update
yum install mysql-server mysql-client
```

Подробнее: <https://dev.mysql.com/doc/mysql-yum-repo-quick-guide/en/>

Windows

Для установки MySQL-сервера на Windows-машине загрузите последнюю версию с сайта <http://dev.mysql.com/downloads/mysql/>. Рекомендуем установить MySQL Server в корень диска, например в папку **C:\MySQL**.

Служба сервера обычно устанавливается и стартует автоматически. Обратите внимание, что для заливки дампа базы помимо сервера MySQL вам понадобится клиентское приложение mysql.

Настройка

В зависимости от дистрибутива, конфигурация MySQL/MariaDB может находиться в файле **/etc/my.cnf** и/или в директориях **/etc/my.cnf.d/** или в **/etc/mysql/**.

Рекомендуемые параметры конфигурации с описанием параметров:

```
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.7/en/server-configuration-defaults.html

[client]
# - mysqldump
default-character-set = utf8mb4

[mysqld]
#
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci

# ,
#bind-address = 127.0.0.1

# ,
#datadir = /var/lib/mysql
# , tmpfs
#tmpdir = /tmp
#log-error = /var/lib/mysql/mysqld.log
# PID
#pid-file = /var/run/mysqld/mysqld.pid
# ,
#log-slow-queries = /var/log/slowquery.log
#long_query_time = 1

#
max_connections = 1000

# -
default-storage-engine = InnoDB

# SQL-
sql_mode =

# DNS
skip-name-resolve

# ,
max_allowed_packet = 64M

# InnoDB -
innodb_file_per_table=1
```

```

#
#innodb_log_group_home_dir =

# InnoDB, 70-80% RAM
#
innodb_buffer_pool_size = 512M

# ( ),
# - 25% innodb_buffer_pool_size, 64M-512M
innodb_log_file_size = 128M
# -
innodb_log_files_in_group = 2
innodb_log_buffer_size = 16M

# sync , - ,
innodb_flush_log_at_trx_commit = 2
innodb_flush_method = O_DIRECT

# InnoDB, : 16 - 256
innodb_thread_concurrency = 128

# ( )
innodb_lock_wait_timeout = 120

# -
table_open_cache = 1024

# , RAM
max_heap_table_size = 64M
tmp_table_size = 64M

#
read_buffer_size = 2M
read_rnd_buffer_size = 8M
sort_buffer_size = 8M
join_buffer_size = 8M
key_buffer_size = 32M

#
query_cache_size = 64M
# ,
query_cache_limit = 4M

bulk_insert_buffer_size = 64M
myisam_sort_buffer_size = 64M

# mysql >= 8, LOAD DATA LOCAL INFILE (: phone)
# data.properties &allowLoadLocalInfile=true
local_infile=ON

[mysqldump]
# mysqldump
max_allowed_packet = 64M

[mysqld_safe]
# ( )
open-files-limit = 4096

```

Из предложенных параметров настройке обычно подлежат следующие параметры:

- **innodb_buffer_pool_size** - размер буфера InnoDB, в которой MySQL будет хранить данные для чтения таблиц и их индексов. Чем больше значение, тем меньше операций с диском. Однако сервер при этом не должен уходить в swap;
- **innodb_log_file_size** - размер файлов логов транзакций InnoDB, он не должен быть слишком маленьким по сравнению с **innodb_buffer_pool_size**, но следует также учитывать, что при старте MySQL будет перечитывать данные логи. Диапазона 64M-512M для биллинга обычно достаточно ([подробнее](#)).

- ⚠ В параметре **sql_mode** запрещается установка режимов **STRICT_TRANS_TABLES** и **STRICT_ALL_TABLES**, это приведёт к неработоспособности некоторых компонентов биллинговой системы (legacy).
- В параметре **wait_timeout** запрещается установка значения, меньше чем 28800 - биллинг работает с пулом соединений к БД, которые могут быть неактивны, он сам следит за их закрытием, поэтому нет необходимости, чтобы MySQL закрывал соединения по таймауту. Если уже у Вас к БД подключаются другие приложения, которым необходимо маленькое значение параметра **wait_timeout**, то добавьте в URL подключения к БД (**db.url**) всех приложений параметр **&interactiveClient=true** (в этом случае для соединений биллинга будет работать параметр **interactive_timeout** вместо **wait_timeout**).
- В случае использования Nginx как проху для MySQL для балансировки нагрузки следует установить значение конфигурации Nginx **proxy_timeout** не меньше чем максимум от (**wait_timeout, interactive_timeout**).

См. также [wiki](#).

Кодировки

Рекомендуемая для использования кодировка для новых инсталляций - **utf8mb4**.

Проверить текущие кодировки БД можно запросами (из sql-клиента):

```
SHOW VARIABLES LIKE 'character_set%';
SHOW VARIABLES LIKE 'colla%';
```

utf8mb4

"Правильный" UTF-8, длиной от 1 до 4 байт. Рекомендуется, вместе с collation utf8mb4_unicode_ci. Конфигурация MySQL/MariaDB:

```
[client]
default-character-set = utf8mb4

[mysqld]
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

В параметрах подключения приложений биллинга к БД (**db.url**) указываем useUnicode=true и characterEncoding=UTF-8. Обратите внимание, что значение "UTF-8" для utf8mb4 здесь такое же, как для utf8 (в зависимости от БД будет учтено нужное). В данный момент (на старых версиях бд и /или jdbc возможно это не так) требуется указывать connectionCollation с актуальным collation, по умолчанию для utf8mb4 используется utf8mb4_general_ci.

```
useUnicode=true&characterEncoding=UTF-8&connectionCollation=utf8mb4_unicode_ci&
```

utf8

"Ограниченный" "старый" UTF-8, длиной от 1 до 3 байт.

```
[client]
default-character-set=utf8

[mysqld]
character-set-server=utf8
collation-server=utf8_unicode_ci
init_connect="SET collation_connection = utf8_unicode_ci;"
```

В параметрах подключения приложений биллинга к БД (**db.url**):

```
useUnicode=true&characterEncoding=UTF-8&connectionCollation=utf8_unicode_ci&
```

Если collation установлен именно "utf8_unicode_ci" — также обратите внимание на параметр "connectionCollation=utf8_unicode_ci" в db.url в data.properties (см. выше). Здесь и в других местах его приходится дополнительно прописывать, т.к. для MySQL collation по-умолчанию для utf8 является utf8_general_ci.

Дополнительная конфигурация службы Linux

Для стабильной работы MySQL/MariaDB в Linux рекомендуется увеличить лимиты открытых пользователем файлов (файловых дескрипторов). Для этого в файле `/etc/security/limits.conf` необходимо указать:

```
mysql      soft  nofile      10000
mysql      hard  nofile      10000
```

В некоторых дистрибутивах (например, RHEL/CentOS 7) **systemd** не использует PAM и лимиты из файла `limits.conf` игнорируются. Поэтому в дополнение к редактированию `limits.conf` рекомендуется создать файл дополнительной конфигурации для сервиса **systemd**. Например, если файл сервиса (каталог `/lib/systemd/system`) называется `mysqld.service`, то создаем каталог `/etc/systemd/system/mysqld.service.d` и в нем файл с именем `override.conf` и содержимым:

```
[Service]
LimitNOFILE=10000
```

При использовании `ntp` мы также рекомендуем настроить сервис MySQL/MariaDB так, чтобы он стартовал после выполнения синхронизации времени. Поэтому общее содержимое `override.conf` может быть таким:

```
[Unit]
After=time-sync.target

[Service]
LimitNOFILE=10000
```

Действия при смене `datadir`

Убедитесь, что у пользователя `mysql` есть права на новую директорию, если это не так, то дайте ему их (например, новая директория - это `/new/data/dir`):

```
chown -R mysql:mysql /new/data/dir
```

Если в системе включен SELinux, то ему также нужно будет указать, что MySQL может работать с новой директорией:

```
semanage fcontext -a -t mysqld_db_t "/new/data/dir(/.*)?"restorecon -Rv /new/data/dir
```

JDK

Для запуска и работы биллинга требуется Java SE Development Kit (или сокращенно - **JDK**). Необходимо загрузить именно Java SE JDK (JRE + средства разработки), а не JRE (только Java-машина), т.к. биллинг использует динамическую компиляцию кода, кроме того, средства разработки могут быть полезны при расследовании нештатных ситуаций в системе.

Приложения биллинга, в общем случае, могут быть запущены на любой платформе, для которой выпускается JDK. Это Windows, Linux, Solaris, MacOS. В официальной поставке включены скрипты запуска только для Linux (sh-скрипты, скрипты сервисов) и Windows (Batch).

JDK должна быть установлена на всех серверах с компонентами биллинга, в т.ч. на клиентских машинах. Также JDK необходим для запуска [ActiveMQ](#).

Для биллинга версии

- 8.0 требуется JDK8 (<https://adoptopenjdk.net>)
- 8.2101+ требуется JDK15 (<https://adoptopenjdk.net>)
- 9.2204+ требуется JDK17 (<https://adoptium.net>)

Linux

Загрузите **jdk**, скопируйте его в директорию **/opt/java** (создайте, если её нет), перейдите в неё, распакуйте архив, появится подкаталог, например **jdkXXX**. Для более удобного обновления Java в дальнейшем рекомендуем перейти в папку **/opt/java** и создать символическую ссылку **/opt/java/jdk** к папке **/opt/java/jdkXXX**.

Путь **/opt/java/jdk** является **JAVA_HOME** - системной переменной, являющейся путём к Java-машине. Вы можете указать **JAVA_HOME** глобально для всех приложений, создав файл **/etc/profile.d/java_home.sh** с содержимым:

```
#!/bin/sh

JAVA_HOME=/opt/java/jdk
```

Или же вы можете указать **JAVA_HOME** в скриптах запуска (например, в файле **setenv.sh** в **BGBillingServer**).

При использовании Gentoo Linux обнаружена проблема с некорректным определением java текущей временной зоны. Данная ошибка связана с тем, что Oracle Java определяет временную зону по содержимому файла **/etc/sysconfig/clock**, который отсутствует в данном дистрибутиве.

Для решения проблемы создайте этот файл самостоятельно, заполнив следующим содержимым:

```
# The ZONE parameter is only evaluated by system-config-date.
# The timezone of the system is defined by the contents of /etc/localtime.
ZONE="Asia/Yekaterinburg"
UTC=false
ARC=false
```

Название временной зоны вы можете получить из названия каталога и файла в **/usr/share/zoneinfo**. Правильность установки зоны можно проверить по отметкам времени, выводимому в логе **BGBillingServer/log/server.log**, либо в любом другом логе.

Windows

Загрузите установочный **jdkXXX.exe** файл и запустите его установку. Рекомендуем устанавливать ближе к корню диска, например **C:\Java\JDK**. Иначе при установке в **Program Files** путь будет содержать пробелы, что неудобно при использовании в batch-файлах и командной строке.

Установка производится мастером, смените каталог установки на не содержащий пробелы, выбрав опцию **Change destination folder** в начале установки.

Проверьте, что **системная переменная JAVA_HOME** указывает на каталог установки JDK, а также на наличие в переменной **PATH** пути до исполнимого файла **java.exe**. Команда **java -version** в консоли должна возвращать правильную версию Java-машины.

ActiveMQ-сервер

MQ (Message Queue)-сервер необходим для передачи сообщений между различными приложениями - компонентами системы. Он также важен для работы, как и сервер базы данных. В качестве MQ-сервера используется [Apache ActiveMQ](#).

- [Подробная установка](#)
 - [Linux](#)
 - [systemd](#)
 - [sysvinit](#)
- [Настройка](#)

Подробная установка

Загрузите ActiveMQ с официального сайта (<https://activemq.apache.org/>)

Linux

- ❗ Убедитесь, что имя сервера с ActiveMQ указано в файле `/etc/hosts`. Имя сервера можно получить командой `uname -n`.
- При распаковке в системе Linux обратите внимание, чтобы tar/zip поддерживал длинные имена файлов (проблема проявлялась в старых дистрибутивах Linux и FreeBSD), иначе распаковка пройдет некорректно и для нормальной работы activeMQ не будет хватать нужных файлов.

Распакуйте архив в каталог `/opt/activemq`, создайте символическую ссылку, например:

```
ln -s /opt/activemq/apache-activemq-5.16.1 /opt/activemq/current
```

Пропишите в файле `/opt/activemq/apache-activemq-5.16.1/bin/env` переменную `JAVA_HOME`

```
# Location of the java installation
# Specify the location of your java installation using JAVA_HOME, or specify the
# path to the "java" binary using JAVACMD
# (set JAVACMD to "auto" for automatic detection)
JAVA_HOME="/opt/java/jdk17"
JAVACMD="auto"
```

systemd

Для системы с `systemd` вызовите создайте файл `activemq.service` в каталоге `/lib/systemd/system` (или `/etc/systemd/system`) с содержимым:

```
[Unit]
Description=Apache ActiveMQ
After=network-online.target

[Service]
Type=forking
WorkingDirectory=/opt/activemq/current/bin
ExecStart=/opt/activemq/current/bin/activemq start
ExecStop=/opt/activemq/current/bin/activemq stop
Restart=on-abort
RestartSec=60
User=root
Group=root

[Install]
WantedBy=multi-user.target
```

Если вы хотите, чтобы ActiveMQ запускался под пользователем `activemq`, то создайте пользователя, отредактируйте `activemq.service` и дайте пользователю права на каталог `/opt/activemq/current`.

Затем выполните:

```
systemctl daemon-reload
systemctl enable activemq
systemctl start activemq
```

Логи выполнения хранятся в **data/activemq.log** и **data/wrapper.log**, по ним можно проследить безаварийный старт сервиса.

sysvinit

Укажите в скрипте запуска **/opt/activemq/current/bin/linux/wrapper.conf** переменную **wrapper.java.command**. Например:

Создайте ссылку на службу (init.d).

```
ln -s /opt/activemq/current/bin/linux-x86-64/activemq /etc/init.d/activemq
```

Настройте автоматический запуск **службы** и запустите её. При работе на одной машине с приложениями биллинга служба должна стартовать раньше всех приложений биллинга (регулируется префиксом ссылки).

Логи выполнения хранятся в **data/activemq.log** и **data/wrapper.log**, по ним можно проследить безаварийный старт сервиса.

Настройка

Главный конфигурационный файл ActiveMQ, использующийся по умолчанию - **conf/activemq.xml**. Логин и пароль (те, что указываются в биллинге в **mq.user** и **mq.pswd**) расположены в файле **conf/credentials.properties**.

Обратите внимание на строчку:

```
<authenticationUser username="${activemq.username}" password="${activemq.password}" groups="user" />
```

Значения **\${activemq.username}** и **\${activemq.password}** являются "ссылками" на значения из файла **credentials.properties** и используются для аутентификации пользователя MQ (в нашем случае пользователь MQ - это биллинг).

В последней строчке (**<import resource="jetty.xml"/>**) включается веб-консоль activeMQ, при этом в **jetty.xml** по умолчанию указан хост "0.0.0.0". Рекомендуется поменять значение на 127.0.0.1, даже если строка (**<import resource="jetty.xml"/>**) закомментирована (т.е. даже если веб-консоль отключена - на случай, если понадобится её включить):

```
<bean id="jettyPort" class="org.apache.activemq.web.WebConsolePort" init-method="start">
    <!-- the default port number for the web console -->
    <property name="host" value="127.0.0.1"/>
    <property name="port" value="8161"/>
</bean>
```

Также рекомендуется использовать фаервол, разрешая доступ только для необходимых портов с необходимых сетей.

В ветке **plugins** указан параметр, при котором все сообщения, у которых истек **timeToLive** будут удаляться (по умолчанию они переносятся в очередь **ActiveMQ.DLQ**):

```
<!-- drop messages that have been sent to the DLQ -->
<discardingDLQBrokerPlugin dropAll="true"/>
```

Ниже описывается использование системных ресурсов для **NON_PERSISTENT**, **PERSISTENT**-сообщений и временных очередей. При превышении данных ресурсов отправка сообщений будет замедлена:

```
<systemUsage>
  <systemUsage>
    <memoryUsage>
      <memoryUsage limit="256 mb"/>
    </memoryUsage>
    <storeUsage>
      <storeUsage limit="10 gb"/>
    </storeUsage>
    <tempUsage>
      <tempUsage limit="1 gb"/>
    </tempUsage>
  </systemUsage>
</systemUsage>
```

В этом отрывке указывается тип коннектора для работы с сервером, интерфейс и порт:

```
<transportConnectors>
  <transportConnector name="openwire" uri="tcp://127.0.0.1:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
</transportConnectors>
```

К этому порту будут подключаться к серверу MQ приложения биллинговой системы. Если все компоненты биллинга установлены на одном сервере, то можно оставить значение **uri=tcp://127.0.0.1:61616**. Иначе нужно указать IP-адрес интерфейса, на который будут идти подключения или установить **uri=tcp://0.0.0.0:61616**, чтобы порт был открыт на всех интерфейсах.

Параметры подключения к серверу ActiveMQ указываются в каждом серверном приложении в .properties файле, например в **data/data.properties** для сервера биллинга.

```
mq.url=failover:(nio://127.0.0.1:61616)
mq.user=bill
mq.pswd=bgbilling
```

Значения **mq.user** и **mq.pswd** должны совпадать со значениями из **conf/credentials.properties** ActiveMQ.

Для локальной машины **mq.url=failover:(nio://127.0.0.1:61616)**, для нескольких серверов (должна быть настроена поддержка сети серверов в каждом из MQ-серверов):

```
mq.url=failover:(tcp://mq1.core.provider.org:61616,tcp://mq2.core.provider.org:61616)
```

В последнем случае подключение будет к случайному серверу из списка. Если подключение невозможно, то идет попытка подключения к следующему указанному серверу MQ, и так до тех пор, пока не установится подключение. Если второй сервер играет роль "запасного", например, он установлен на слабой машине и должен принять работу только, если прервется работа первого сервера, то можно указать, чтобы подключение не устанавливалось к случайному, а попытки шли в указанном порядке:

```
mq.url=failover:(tcp://mq1.core.provider.org:61616,tcp://mq2.core.provider.org:61616)?randomize=false
```

Установка сервера биллинга (для версий до 8.0(вкл.))

Для работы сервера биллинга необходима установка и запуск MySQL/MariaDB и ActiveMQ-сервера.

Извлеките из архива **BGBillingServer_X.Y.ZZ.zip** файл **dump.sql** и **BGBillingServer** (X.Y - номер версии, ZZ - билда) в каталог установки. Стандартный каталог установки для Linux **/opt/bgbilling**, для Windows - C:\.

Перенесите файл **dump.sql** на машину с MySQL-сервером, если это отдельная машина. Перейдите в каталог в **dump.sql**, запустите

```
mysql --default-character-set=utf8 < dump.sql
```

для создания базы данных. Дамп нужно заливать от пользователя MySQL **root**, так как дамп содержит GRANT PRIVILEGES и подобные команды.

При необходимости скорректируйте **параметры подключения к БД** и **ActiveMQ** в **data/data.properties**. Там же можно скорректировать прослушиваемый порт, адрес, порт управления.

Для запуска клиента биллинга через **BGClientRunner** дополнительно нужно установить клиентские библиотеки на сервер, для этого загрузите в каталог сервера биллинга следующие архивы с <https://bgbilling.ru/download/X.Y/> - **update_X.Y.zip** и **update_lib_X.Y.ZZ.zip** и на остановленном сервере выполните скрипты **./bg_installer.sh update_X.Y.zip** и **./bg_installer.sh update_lib_X.Y.ZZ.zip**

При успешном запуске (см.далее) в папке log биллинга должны появиться **server.log** и **server.out**. В первом должно быть примерно следующее:

```
INFO 13.07.2005 19:42:42 Starting BGBillingServer..
INFO 13.07.2005 19:42:42 HTTP port: 8080
INFO 13.07.2005 19:42:42 Browsing installed modules..
...
INFO 13.07.2005 19:42:42 Starting listen admin port 2005
```

Linux

Выполните **стандартные действия**, предшествующие установке приложения на Linux.

Установите переменную **JAVA_HOME** в файле **setenv.sh**, если она не установлена глобально в системе. Если же переменная установлена глобально, иногда имеет смысл переопределить ее в данном скрипте, на случай, если обновление ОС или установка другого JDK переопределит текущий глобальный **JAVA_HOME**.

```
JAVA_HOME=/opt/java/jdk
```

Создайте службы сервера, планировщика и загрузчика логов. Для этого используйте скрипты из **BGBillingServer/script**. Для **sysvinit** скрипт **bgcommonrc** также необходимо перенести в /etc/init.d, он содержит общие переменные для скриптов сервера, планировщика и загрузчика логов.

Запустите сервер, планировщик задач и загрузчик логов.

systemd

```
systemctl start bgbilling
systemctl start bgscheduler
```

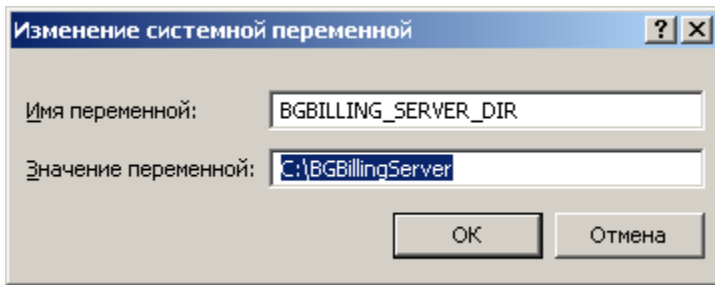
sysvinit

```
/etc/init.d/bgbilling start
/etc/init.d/bgscheduler start
```

Windows

Выполните **стандартные действия**, предшествующие установке приложения на Windows.

Установите **переменную окружения** **BGBILLING_SERVER_DIR=C:\BGBillingServer**.



После этого необходимо перезагрузить компьютер.

Проинсталируйте службу сервера, планировщика. Для этого перейдите в папку **C:\BGBillingServer** и запустите **server_install.bat**, **scheduler_install.bat**.

Зайдите в управление службами и запустите службы **BGBillingServer**, **BGScheduler**.

Если с лаунчерами служб какая-то проблема - не стартуют, долго висят в запуске итд - нужно разбираться с `jsl.exe` и параметром `-debug`.
Подробнее на [wiki](#).

Установка (обновление) сервера биллинга (для версий с 8.2010)

Для работы сервера биллинга необходима установка и запуск JDK, MySQL/MariaDB и ActiveMQ-сервера.

1) Создайте папку **/opt/bgbilling**

```
mkdir /opt/bgbilling
```

2) Загрузите в нее архив с сервером биллинга. (текущую версию и название файла уточняйте на сайте <https://bgbilling.ru/download>)

Ниже пример команды для загрузки архива

```
cd /opt/bgbilling
wget https://bgbilling.ru/download/8.2101/last/BGBillingServer_8.2101-2101251439.zip
```

3) Распакуйте архив

```
unzip BGBillingServer_8.2101-2101251439.zip
```

4) Загрузите дампы БД (если БД находится на другом сервере замените **localhost** на IP сервера БД)

```
mysql -h localhost -uroot -p --default-character-set=utf8mb4 < dump.sql
```

5) Скопируйте файл **/opt/bgbilling/BGBillingServer/data/data.properties.orig** в **/opt/bgbilling/BGBillingServer/data/data.properties**.

Если у вас БД и/или ActiveMQ сервер находятся на других серверах пропишите их адреса в файле **/opt/bgbilling/BGBillingServer/data/data.properties**

6) Если у вас на сервере используется **systemd** скопируйте файлы для запуска сервера из папки биллинга в каталог **/etc/systemd/system**

```
cp /opt/bgbilling/BGBillingServer/script/bgbilling.service /etc/systemd/system
systemctl enable bgbilling.service

cp /opt/bgbilling/BGBillingServer/script/bgscheduler.service /etc/systemd/system
systemctl enable bgscheduler.service
```

7) если у вас есть купленная лицензия скопируйте ее в каталог **/opt/bgbilling/BGBillingServer/data/lic.properties**

8) запустите скрипт **/opt/bgbilling/BGBillingServer/bin/update.sh**

```
cd /opt/bgbilling/BGBillingServer/bin
update.sh
```

если при запуске **update.sh** получаете следующее сообщение

 **# ./update.sh**

The JAVA_HOME environment variable is not defined

This environment variable is needed to run this program

пропишите в файле **/opt/bgbilling/BGBillingServer/bin/setenv.sh** путь к папке где у вас установлена JDK

JAVA_HOME=/opt/java/jdk15

в процессе работы скрипта может быть выдано сообщение



Установить все модули и плагины указанные в файле лицензий (yes/no)

если вы ранее скопировали купленную лицензию ответьте - **yes**

если у вас тестовая лицензия - **no**

9) После загрузке ядра будет выдано следующее меню

```
Select:
 1) Install new component          3) Show installed components
 2) Update installed components
 0) Exit
```

Если вы в предыдущем шаге ответили **no** то выберите пункт 1, будет выведен список доступных компонентов биллинга, выберите нужный вам и установите его, затем повторите для других

10) для выхода выберите 0

Обновление сервера биллинга (в пределах одной версии)

1) На сервере биллинга запустите скрипт **/opt/bgbilling/BGBillingServer/bin/update.sh**

2) Из меню выбрать пункт 2

3) Будет выведен список доступных версий для обновления или сообщение о том, что нет доступных версий (если у вас установлена последняя)

4) Введите номер версии на который хотите обновить (1 для обновления на последнюю)

5) Для выхода нажмите **0**

Обновление сервера биллинга (переход на новую версию)

1) на сервере в файле **data/files.dat** указать версию на которую хотите перейти

2) На сервере биллинга запустите скрипт **/opt/bgbilling/BGBillingServer/bin/update.sh**

2) Из меню выбрать пункт 4

3) Для выхода нажмите **0**

4) обновите все приложения биллинга используя скрипт **update.sh** в каталоге приложения

Установка планировщика (с версии 8.2107)

С версии 8.2107 планировщик вынесен из сервера биллинга и устанавливается как отдельное приложение

- 1) Загрузите архив с планировщиком (https://bgbilling.ru/download/X.YYYY/last/BGScheduler_X.YYYY-ZZZZZZZZ.zip)
- 2) Распакуйте архив в каталог **/opt/bgbilling**
- 3) В конфиге (**/opt/bgbilling/BGScheduler/conf/application.xml**) укажите параметр **app.id** (каждое приложение биллинга должно иметь уникальное значение, например для планировщика можно указать 1)
- 4) Запустите скрипт **/opt/bgbilling/BGScheduler/bin/update.sh** (сервер биллинга должен быть к этому моменту запущен)
- 5) Попробуйте запустить планировщик с помощью скрипта **/opt/bgbilling/BGScheduler/bin/start.sh**, результат запуска см. в каталоге с логами **logs**
- 6) Если ошибок нет, настройте автозапуск планировщика, скопируйте файл сервиса из каталога **script** в **/usr/lib/systemd/system/** и активируйте его командой

```
systemctl enable bgscheduler.service
systemctl start bgscheduler.service
```

Установка приложений биллинга (BGInetAccess, BGVoiceProcwssor и др.) (с версии 8.2104)

В зависимости от установленных модулей биллинга, может быть необходимо установить и настроить дополнительные приложения. Они могут быть установлены на том же сервере, что и сервер биллинга, но так же могут быть вынесены на другие сервера.

Все приложения устанавливаются по одинаковой схеме:

- 1) Загружаете архив с установщиком приложения например (https://bgbilling.ru/download/beta/last/BGInetAccess_beta-XXXXXXXXXX.zip)
- 2) Распаковываете архив в папку (рекомендуем в `/opt/bgbilling/BGInetAccess`)
- 3) Переходите в папку `/opt/bgbilling/BGInetAccess/bin` и запускаете команду `chmod 0755 *.sh`
- 4) Если приложение устанавливается на сервер, отличный от того где установлен сервер биллинга, не забываем установить JDK (рекомендуем ставить в каталог `/opt/java/jdkXX`)
- 5) Если приложение устанавливается на сервер, отличный от того где установлен сервер биллинга, в файле `/opt/bgbilling/BGInetAccess/conf/application.xml` редактируем URL сервера биллинга

```
<param name="server.update.url" value="http://127.0.0.1:8080/bgbilling"/>
```

(обратите внимание на протокол **http** или **https**, на IP адрес, порт и контекст (bgbilling))

- 6) Запускаем скрипт `/opt/bgbilling/BGInetAccess/bin/update.sh` (обратите внимание при этом сервер биллинга должен быть запущен)
- 7) Настройте приложение указав необходимые параметры в файле `/opt/bgbilling/BGInetAccess/conf/application.xml`
- 8) Попробуйте вручную запустить приложение, используя скрипт `/opt/bgbilling/BGInetAccess/bin/start.sh`
- 9) Если приложение успешно запустилось настройте автозапуск приложения (заготовки скриптов для systemd доступны в каталоге `/opt/bgbilling/BGInetAccess/script`)

Установка и первый запуск клиента биллинга

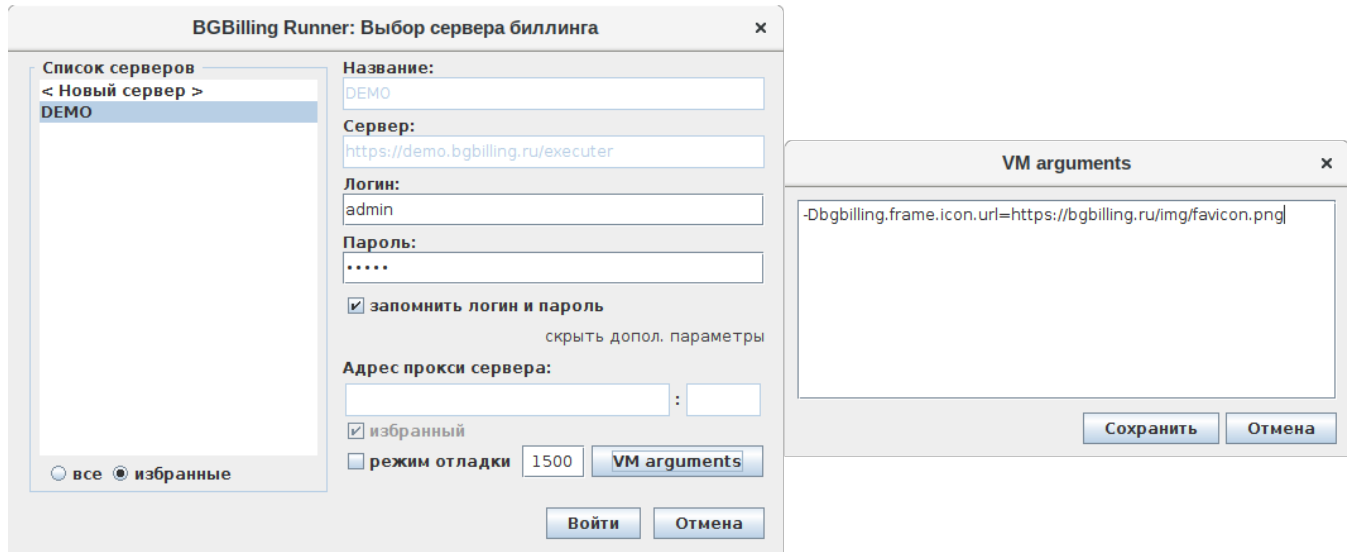
При работе некоторых антивирусов некоторые вкладки могут очень долго открываться. Это происходит, когда антивирус пытается проверить интернет-трафик, нарушая при этом работу SOAP-протокола Web-сервисов, через который клиент биллинга обращается к серверу биллинга. Если такое странное поведение наблюдается - рекомендуем добавить IP-адрес сервера биллинга в исключения в настройках Вашего антивируса.

Например, для ESET NOD32 необходимо открыть расширенные настройки ESET Smart Security, выбрать раздел "Интернет и электронная почта" - "Фильтрация протоколов" - "Исключенные IP адреса" и добавить в список адрес сервера.

В данный момент доступно несколько способов запуска клиента биллинга.

Рекомендуемым на данный момент является запуск с использованием программы **BGBilling Runner**

Запуск клиента через BGBilling Runner



С версии 7.0+ (в сборках от 01 декабря 2019 и позднее) доступен новый способ запуска клиента биллинга **BGBilling Runner**. Данный вариант является рекомендуемым способом запуска клиента биллинга.

Для запуска клиента биллинга данным способом, необходимо:

1. На клиентском компьютере должна быть установлена виртуальная машина Java (**JRE**)
2. Загрузить архив с **BGBilling Runner** на клиентский компьютер (например отсюда <https://bgbilling.ru/download/runner/>)
3. Распаковать архив в домашнем каталоге пользователя или на сетевом диске если клиент будут запускать несколько пользователей.
4. Убедиться, что установлена и корректно настроена системная переменная **JAVA_HOME**
5. При необходимости настроить адреса сервера(ов) биллинга по умолчанию (что то типа **http[s]://host[:port]/[bgbilling]/executer**), которые будут использоваться при запуске **BGBilling Runner**. Доступно несколько способов задания списка доступных серверов:
 - a. в файле **config.json** из архива **BGBilling Runner**
 - b. на локальном сервере, при запуске **BGBilling Runner** пытается загрузить список серверов с сайта <http://billing.local/servers.json> (пример файла **config_http.json** см. архиве **BGBilling Runner**). Адрес сайта и имя файла могут быть заданы в файле **config.json**, параметр **"configURL" : "none"**. Значение **"none"** отключает загрузку с сайта.
 - c. ручной ввод данных сервера в интерфейсе **BGBilling Runner**.
6. Запустить **BGBilling Runner** с помощью скриптов запуска **runner.sh**(Linux) или **runner.bat**(Windows)
7. Выбрать необходимый сервер, ввести ни необходимости логин и пароль и **Войти** на сервер.

Внимание



При установке базы биллинга в ней создаётся единственный пользователь **admin** с паролем **admin**. После первого входа желательно поменять пароль в целях безопасности.

Если при запуске клиента нужно передать дополнительные параметры это можно сделать в редакторе **VM arguments**. Для каждого сервера можно настроить свои параметры, при этом если необходимо, что бы параметры сохранялись между запусками клиента, их необходимо их добавлять в режиме редактирования, иначе они будут применяться только для текущего запуска.

Алгоритм работы BGBilling Runner

Программа использует похожий на **JavaWebStart** алгоритм запуска программы с дополнительными возможностями.

После запуска **BGBilling Runner**, выбора сервера, и ввода логина и пароля, **BGBilling Runner** обращается к серверу биллинга за списком библиотек необходимых для запуска клиента биллинга. Сервер возвращает список библиотек и их контрольные суммы. **BGBilling Runner** проверяет наличие запрашиваемых библиотек в локальном кэше клиента (в домашнем каталоге пользователя, каталог **./bgbilling/libs**). Отсутствующие библиотеки или отличающиеся по контрольной сумме, загружаются с сервера и сохраняются в кэше. При наличии в кэше всех необходимых библиотек, **BGBilling Runner** запускает клиент биллинга в виде отдельного java процесса. Кроме списка библиотек, с сервера загружает контрольную сумму всех библиотек, которая передается клиенту биллинга при запуске и используется в процессе работы клиента для контроля совпадения версии клиента и сервера, в случае обновления сервера, его контрольная сумма меняется и в клиенте выводится сообщение о необходимости перезапуска (обновления) клиента.

BGBilling Runner позволяет одновременно запускать несколько клиентов биллинга разных версий (в том числе, например 7.1 и 8.0) без необходимости переустановки клиента.

В процессе работы **BGBilling Runner** сохраняет вручную добавленные сервера, а также логины и пароли, в домашнем каталоге пользователя, в файле **./bgbilling/servers.json**

В **BGBilling Runner** встроен механизм очистки кэша от редко используемых библиотек. По умолчанию библиотеки не использованные для запуска клиента более 15 дней удаляются из кэша. Данный срок может быть изменен с помощью ключа запуска **-Dbg.client.runner.cache.period=15**. Дата последнего использования библиотеки хранится, в домашнем каталоге пользователя, в файле **./bgbilling/cache.json**

При первом запуске **BGBilling Runner**, если в домашнем каталоге пользователя есть файл **~/bgbilling/config**, то программа попытается загрузить из него данные о серверах, и сохраненных логинах и паролях. Данные будут сохранены в файле **~/bgbilling/servers.json**. При последующих запусках если существует файл **~/bgbilling/servers.json**, то загрузка данных из **~/bgbilling/config** производиться не будет.

Обновление BGBilling Runner

В сборка доступных после 1 марта 2019 года в версиях 7.1+ реализован механизм автоматического обновления библиотеки программы **BGBilling Runner**. Для активации данного режима необходимо убедиться в следующем:

- 1) у вас на рабочей станции установлен **BGBilling Runner** версии **1903011835** или старше.
- 2) на сервере установлена сборка сервера старше 1 марта 2019 года
- 3) в каталоге установленного сервера есть скрипт **update_runner.sh** (если скрипта нет, его можно загрузить с ftp://ftp.bgbilling.ru/pub/bgbilling/runner/update_runner.sh)

Для обновления **BGBilling Runner** необходимо запустить скрипт **update_runner.sh** (можно периодически запускать скрипт из планировщика) который проверит наличие новой версии на сервере <ftp://ftp.bgbilling.ru/pub/bgbilling/runner/> и при наличии, загрузит ее на сервер биллинга в каталог **runner**. Далее при запуске **BGBilling Runner** на рабочей станции, происходит сравнение версии на рабочей станции и на сервере, если версия на сервере больше версии на рабочей станции, версия с сервера копируется на рабочую станцию в папку **BGBillingRunner/lib.update**. При следующем запуске **BGBillingRunner** файлы из папки **BGBillingRunner/lib.update** переносятся в папку **BGBillingRunner/lib**.

Параметры запуска BGBilling Runner

При запуске скриптов **runner.sh(Linux)** или **runner_w2k.bat(Windows)** можно задать параметр **autostart=default<server_key>**, где **default** - ключевое слово обозначающее сервер по умолчанию заданный в конфигурационных файлах (параметр **"defaultServerKey"**); **<server_key>** - значение параметра **"key"** из конфигураций сервера. При наличии этого параметра запуск указанного клиента биллинга будет происходить автоматически.

Часто задаваемые вопросы

1. При подключение выдается ошибка

Ошибка



"Выбранный сервер не поддерживает данный режим запуска клиента. Необходимо настроить сервер."

Данная проблема возможна по следующим причинам

- Сервер биллинга не поддерживает данный способ запуска, необходимо обновить сервер до более поздних версий
- Сервер биллинга находится например за nginx, на котором не настроен доступ к сервису [http://host\[:port\]/bgbilling/clientUpdate/](http://host[:port]/bgbilling/clientUpdate/)
- Если подключаетесь к только что установленному серверу биллинга, необходимо выполнить процедуру обновления сервера (загрузить в каталог сервера архивы, например https://bgbilling.ru/v7.2/download/kernel/update_7.2.zip и [update_lib_7.2_XX.zip](https://bgbilling.ru/v7.2/download/lib/update_lib_7.2_XX.zip) с <ftp://ftp.bgbilling.ru/v7.2/>, после чего на остановленном сервере запустить скрипт **./bg_installer.sh update_7.2.zip** и **./bg_installer.sh update_lib_7.2_XX.zip**)

2. Использование самоподписанного сертификата на сервере биллинга

Для отключение проверки валидности сертификата в **config.json** или **config.json** загружаемого с сервера, нужно указать параметр **"anySSL" : "true"**. Это работает только для новых пользователей, тех кто в первый раз запускает **BGBilling Runner**. Для тех кто уже запускал **BGBilling Runner** ранее необходимо данные настройки сделать в конфиге пользователя, в домашнем каталоге пользователя, в папке **./bgbilling** файл **servers.json**.

3. Запуск клиента биллинга в debug режиме

Если перед нажатием кнопки **Войти** в **BGBilling Runner** щелкнуть по ссылке по **"показать дополнительные параметры"** и установить галочку **режим отладки**, то в этом случае после запуска клиента биллинга, в домашнем каталоге пользователя будет создан файл **debug_XX.XXXXXXXXXX.log** с логами обмена запросами клиента и сервера.

4. Смена иконки приложения BGBilling

В случае одновременного запуска нескольких копий клиента биллинга можно для разных копий задать разные иконки с помощью ключа **-Dbgbilling.frame.icon.url=<url>** в конфигурации запуска в разделе **VM arguments**.

Запуск клиента через Java Web Start (устарело)

ВАЖНО



В связи с удалением поддержки данной технологии начиная с JDK 9, применение данного варианта запуска клиента биллинга становится не возможным.

Данная возможность реализована, если сервер биллинга установлен на машине с ОС Linux.

Для запуска клиента через JavaWebStart необходимо выполнить несколько условий

- Настроить сертификат для подписи библиотек
- Установить сервер и выполнить обновления сервера (загрузить в каталог сервера архив, например https://bgbilling.ru/v7.2/download/kernel/update_7.2.zip, после чего на остановленном сервере запустить скрипт "`./bg_installer.sh update_7.2.zip`")
- Настроить (отредактировать) файл **BGBillingServer/webroot/client/bgclient.jnlp**
- Добавить адрес сервера в биллинга в список сайтов-исключений

По умолчанию библиотеки клиента для Java Web Start подписываются самоподписанным сертификатом. Поэтому необходимо в контрольной панели Java (jcontrol) на машине, где будет запускаться клиент, добавить URL к серверу биллинга (например, <http://x.x.x.x:8080> или <http://billing.internal.provider.ru:8080>) в список сайтов-исключений.

Или же Вы можете использовать подписанный сертификат, для этого его нужно импортировать в **BGBillingServer/bgclient_keystore**:

```
# PKCS12-
openssl pkcs12 -export -name bgbilling -in public.crt -inkey private.key -out bgclient.p12

# PKCS12- JKS-
keytool -importkeystore -destkeystore .bgclient_keystore -srckeystore bgclient.p12 -srcstoretype pkcs12 -alias
bgbilling

#
keytool -list -v -keystore .bgclient_keystore
```

Необходимо отредактировать файл **BGBillingServer/webroot/client/bgclient.jnlp** указав правильное значение codebase и db.server.0.url в зависимости от URL доступа к биллингу, затем запустить `bgclient_javawebstart.sh`.

Теперь при открытии URL <http://x.x.x.x:8080/bgbilling/client/bgclient.jnlp> запустится клиент биллинга.

Запуск клиента через установку локальной версии BGClient (устарело)

Клиентское приложение единое для Windows и UNIX-систем, различия в установке незначительны, поэтому процесс описан в одной главе.

- 1) Загрузите клиент биллинга **BGBillingClient_X.X_Y.zip** (X.X - номер версии, Y - билда) и распакуйте его в произвольное место. На машине, где установлен клиент должна стоять JDK (допускается JRE);
- 2) Выполните [стандартные действия для Linux](#), либо [стандартные действия для Windows](#);
- 3) Для Linux пропишите переменную JAVA_HOME в начале .sh скриптов:

```
cd ${0%${0##*/}}.  
  
JAVA_HOME=/opt/java/jre
```

- 4) В каталоге **BGBillingClient** найдите файл **client.properties**.

```
db.server.0.title=MyBilling  
db.server.0.url=http://127.0.0.1:8080/bgbilling/executer  
db.server.0.proxy.host=  
db.server.0.proxy.port=
```

В случае, если клиент установлен на одну с сервером машину, ничего менять не надо. Если необходимо подключиться к серверу с другого рабочего места укажите вместо 127.0.0.1 адрес машины, где установлен сервер BGBilling и, при необходимости, хост и порт прокси, если подключение идёт не напрямую.

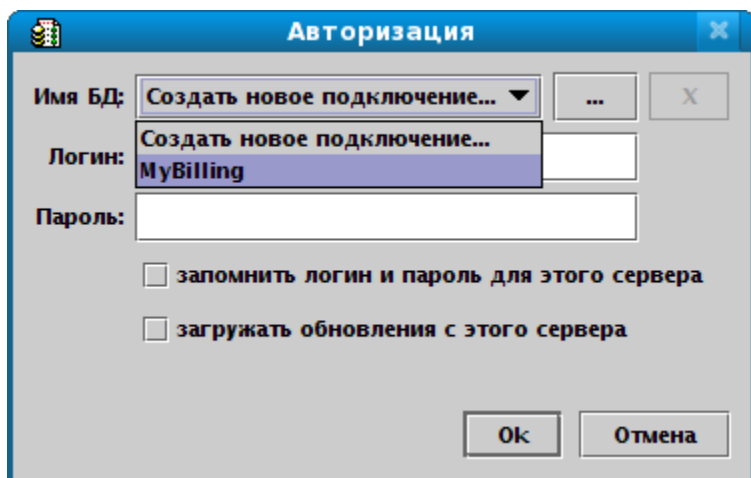
Аналогичным образом вы можете добавить ещё один сервер BGBilling, доступный для подключения. Нужно лишь добавить аналогичный набор записей ниже, исправив server.0 на следующий номер. Например:

```
db.server.1.title=NextBilling  
db.server.1.url=http://www.bill.com:8080/bgbilling/executer  
db.server.1.proxy.host=my.proxy.com  
db.server.1.proxy.port=3128
```

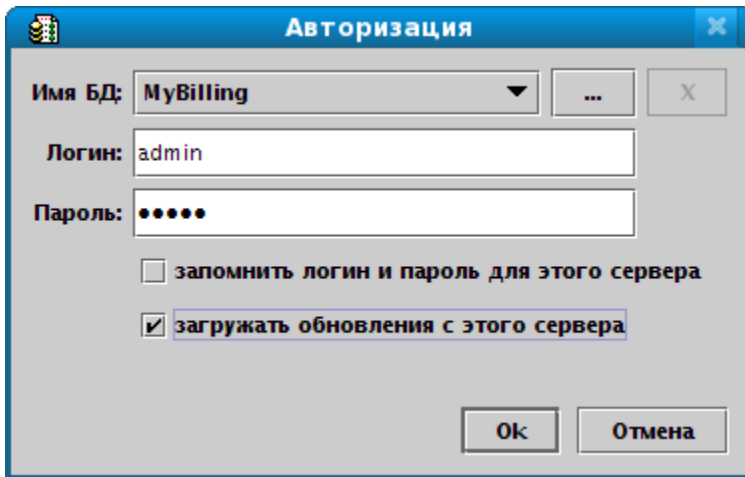
При запуске клиента вам будет предложено выбрать сервер для подключения. Кроме редактирования файла возможно добавление серверов в окне логина, как это сделать описано далее.

- 5) Запустите клиент с помощью пакетного файла **bgbilling_w9x.bat** для Win98/ME, **bgbilling_w2k.bat** для Windows2000/XP/2003, **bgbilling.sh** для Linux. Если клиент не стартует, либо после старта обнаруживаются проблемы, запустите DEBUG-версию **bgbilling_debug.bat**, либо **bgbilling_debug.sh**, при этом в каталоге **BGBillingClient** должен появиться файл **log**, который вы можете передать разработчикам при разборе проблемы.

- 6) Если клиент стартовал, вы увидите окошко авторизации (см. рисунок).



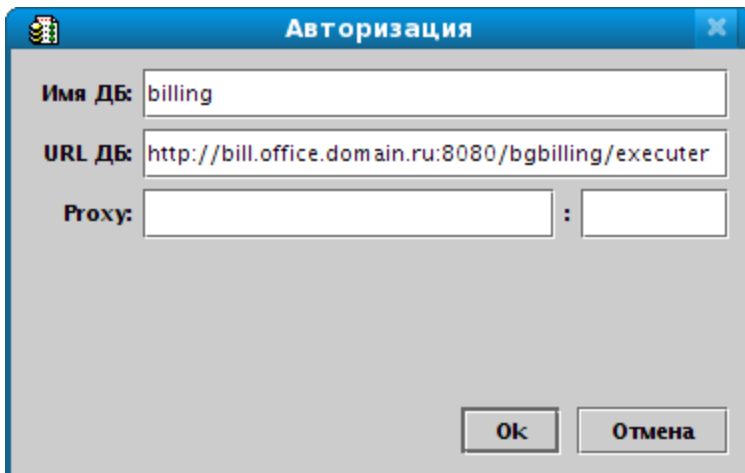
В списке подключений необходимо выбрать требуемый сервер.



При установке базы биллинга в ней создаётся единственный пользователь **admin** с паролем **admin**. После первого входа желательно поменять пароль в целях безопасности.

Обратите внимание на опцию **загружать обновления с этого сервера**, её необходимо установить, чтобы библиотеки установленных на сервере модулей могли быть получены клиентом. Опцию можно снимать при подключении к сторонним серверам для предотвращения получения нежелательных обновлений.

При выборе в списке позиции **Создать новое подключение** и нажатии кнопки ... открывается редактор подключений где вы можете создать подключение к новому серверу биллинга.



Созданные таким методом подключения, также как и сохранённые логины и пароли запоминаются в файле **HOME_DIR/bgbilling/config**, где **HOME_DIR** - домашний каталог пользователя.

Вы можете изменить файл, в котором сохраняются пароли и дополнительные соединения установив опцию **-Dlocal.setting.file.name=<имя отличное от config>** в скрипте запуска клиента, например так:

```
start javaw -Dupdate.folder=lib.update -Djavax.net.ssl.trustStore=.keystore -Dsun.net.client.defaultConnectTimeout=1000 -Xmx256m -Duser.language=ru -Duser.region=RU -Dlocal.setting.file.name=config_v.4.5 -cp %CLASSPATH% bitel.billing.ShellFrame
```

Это может быть полезно, если на одной машине запускается несколько клиентов для разных версий биллинга.

Если ошибок авторизации не выдаётся, а просто происходит очистка окошек ввода логина и пароля, то, скорее всего, проблема в сервере.

Проверьте лог **C:\BGBilling\log\server.out** на наличие ошибок (Exception). Вероятнее всего, сервер не может соединиться с базой данных. Попробуйте взять логин и пароль из файла **data\data.properties** и с их помощью соединиться с БД. Для этого используйте консольный клиент **mysql**, расположенный в директории **C:\mysql\bin**. Наберите в командной строке

```
mysql -ubill -pbgbilling bgbilling
```

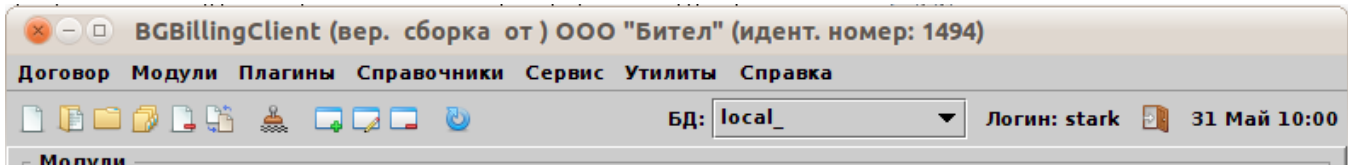
Если соединение не прошло, проверьте, запущен ли сервер MySQL.

7) Если вы устанавливали модули или плагины, то при загрузке клиента должно быть выведено сообщение об установке обновлении системы, после чего потребуются перезапуск клиента.

Если после логина требуется сменить текущий сервер биллинга (в случае, если у вас их несколько, см. выше как можно их добавлять) вы можете воспользоваться выпадающим списком расположенным справа от надписи **БД:** на панели инструментов. При первом входе на новый сервер биллинга будет запрошен логин и пароль, при последующих переключениях авторизация не требуется. Открытые вкладки клиента при переключении на другой сервер и возврате на исходный сохраняются.

Описание интерфейса клиента биллинга

Так выглядит основное рабочее окно программы **BGBillingClient** (приведена только верхняя область окна):



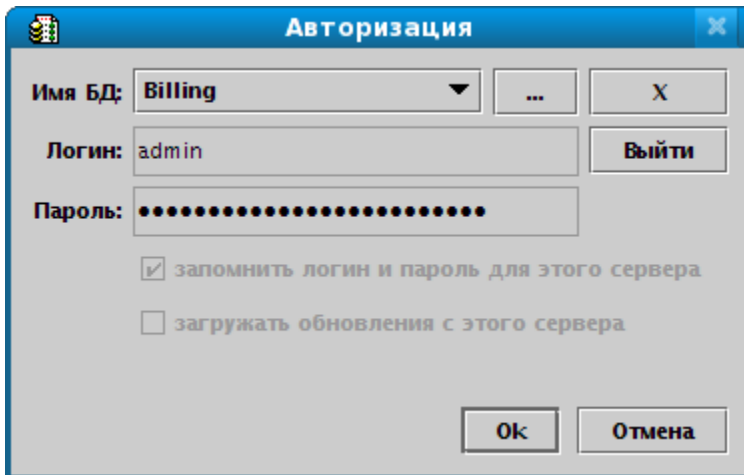
Интерфейс клиента биллинга построен на вкладках, открывающихся в основном окне. Это позволяет оператору держать одновременно открытыми несколько договоров/редакторов/справочников и т.п. Во вкладках открываются договоры, редакторы справочников, редакторы свойств модулей и пр.

Для закрытия вкладки (вкладок) можно нажать крестик на вкладке в нижней области окна, либо вызвать пункт меню **Договор=>Закрыть вкладку (Закрыть вкладки)**. Пункты меню продублированы на панели инструментов кнопками **Закрыть вкладку (Закрыть вкладки)**.

Панель инструментов расположена ниже меню. Первые семь кнопок дублируют часто используемые пункты меню **Договор**. Расположенные далее кнопки **Новый элемент, Редактировать, Удалить элемент, Обновить** действуют на текущую вкладку биллинга и являются универсальными. В контексте текущей вкладки они позволяют:

- создать новую сущность на выбранной вкладке;
- редактировать существующую сущность;
- удалить выбранную сущность;
- обновить информацию на вкладке данными с сервера биллинга.

Далее идёт выпадающий список подключений к серверу (БД). Если сменить текущий сервер, то снова вызовется диалог авторизации с вводом логина/пароля к серверу. Переключение между серверами, на которые уже произошла авторизация, происходит без вызова диалога. Если такой сервер выбрали в диалоге авторизации (диалог вызвали для не авторизованного сервера, а потом сменили на уже авторизованный), то правка логина/пароля недоступна, в диалоге при этом появляется кнопка **Выйти** - она разрывает соединение с сервером и он становится не авторизованным.

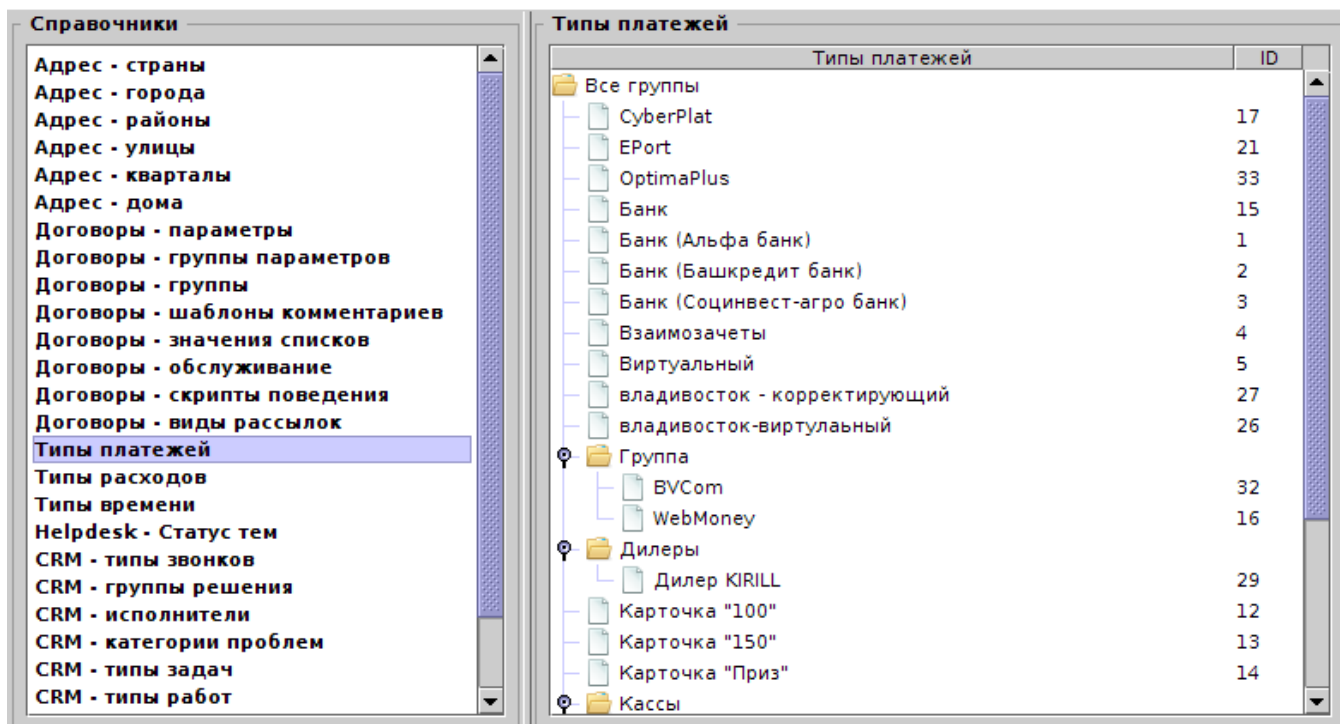


При смене сервера запоминаются все текущие вкладки и активная вкладка, они восстанавливаются, если вернуться к старому серверу. Далее на панели идёт вывод текущего логина и кнопка разрыва соединения с текущим сервером биллинга.

Далее на панели инструментов идет отображение текущего логина и кнопка завершения сеанса. И в завершении панели отображается актуальное время на сервере биллинга.

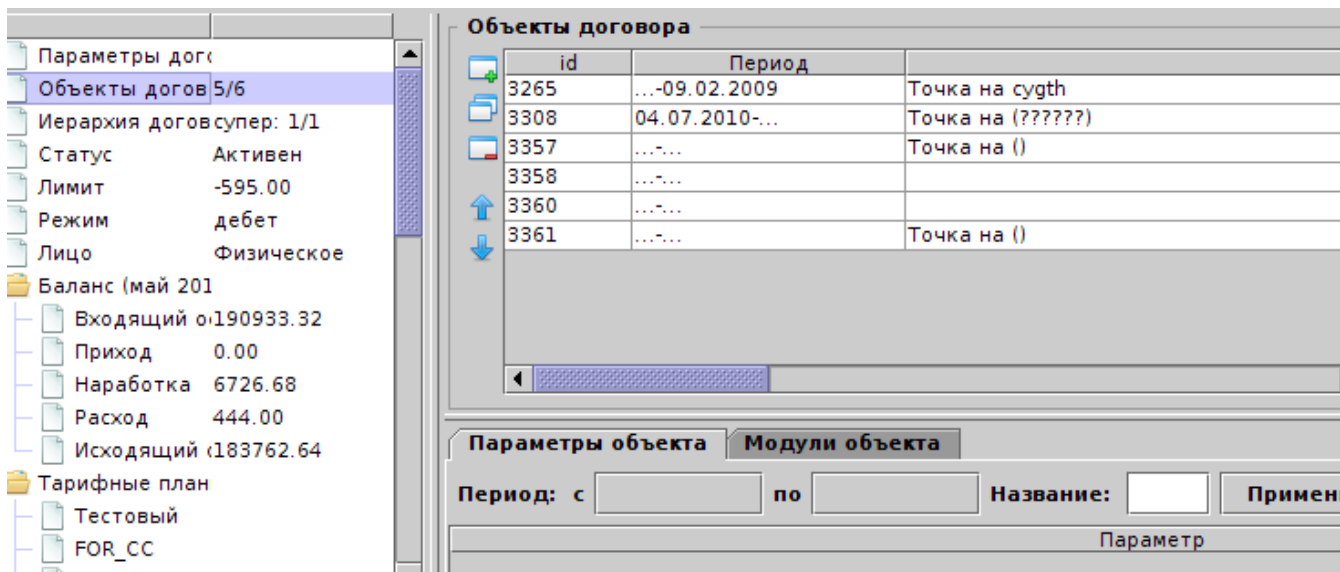
Меню и панель инструментов могут быть настроены редактированием файла **BGBillingServer/data/menu.xml** и **toolbar.xml**. Установленные плагины и модули могут дополнять содержимое меню и панели инструментов новыми пунктами.

Выберите пункт меню **Справочники=>Другие**. В открывшемся справочнике типов платежей выберите корневой узел, нажмите **Новый элемент** в панели инструментов. В появившемся редакторе введите название типа платежа, установите галочку **Элемент группы** и нажмите **ОК**.



Выберите получившийся тип платежа и удалите его кнопкой **Удалить** на панели инструментов. Далее закройте вкладку **Редактор справочников** одним из вышеописанных способов.

Возможны случаи, когда в редакторах вкладок используются кнопки не со стандартной панели инструментов. Следует использовать ближайшую к редактору панель инструментов. Пример такого случая приведён на снимке ниже.



Горячие клавиши

Во всех текстовых полях работают горячие клавиши **копировать/вырезать/вставить** - **Ctrl+C/Ctrl+X/Ctrl+V**, что позволяет использовать в работе с клиентом буфер обмена. Для копирования в буфер обмена информации, содержащейся в таблице, выберите требуемое количество строк таблицы мышью, нажимая кнопки **Shift, Ctrl**, либо **Ctrl+A** для выбора всех строк и нажмите **Ctrl+C**.

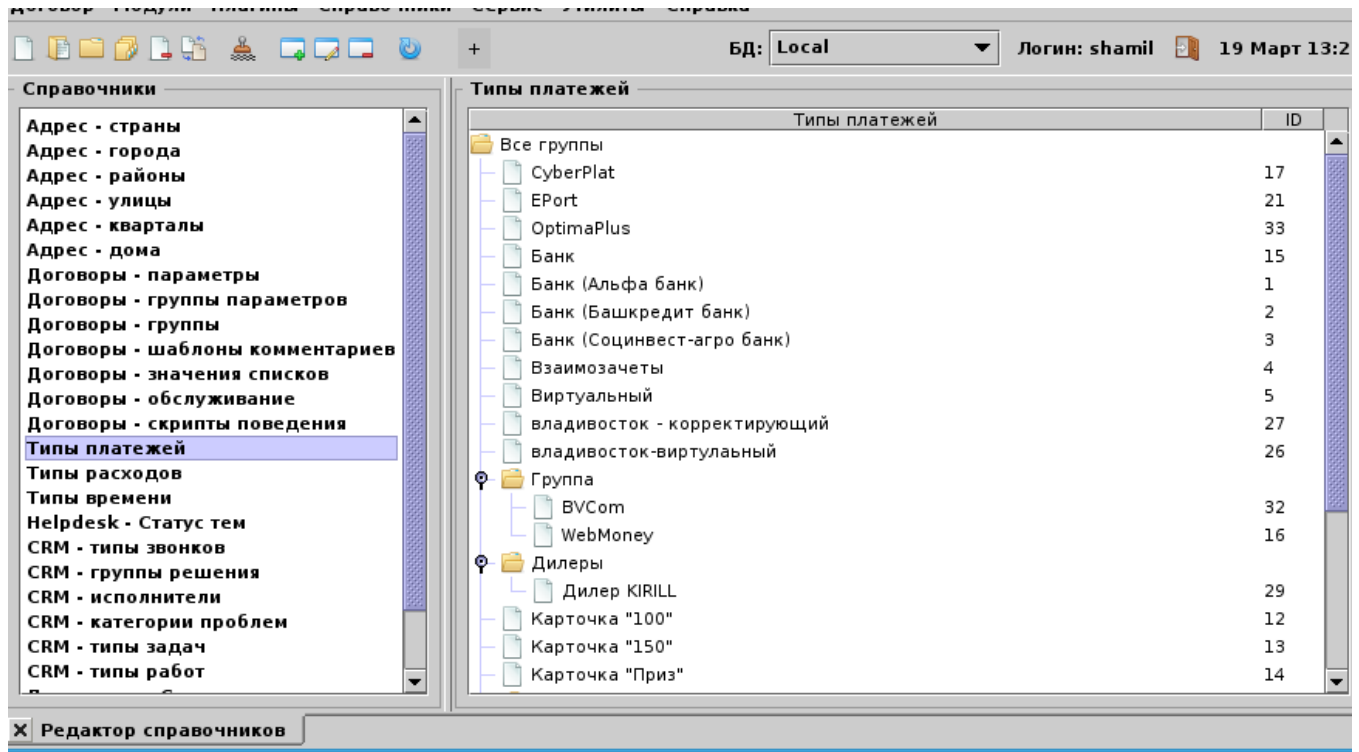
Для быстрого доступа к главному меню можно использовать связки **Ctrl+1, Ctrl+2**. Далее нажать клавишу, соответствующую подчеркнутому символу в названии пункта (необходимость включения одинаковой раскладки зависит от операционной системы). Например, **Ctrl+1**, затем клавиша **J(O)** - "Открыть договор". Или **Ctrl+2**, затем **1** - открыть вкладку модуля, первого в списке выпадающего меню ("**1**. Интернет"). Или же выбрать пункт стрелками вверх и вниз и нажать **Enter**.

Также каждому пункту может быть присвоена своя связка с помощью файла menu.xml, который находится в BGBillingServer/data/menu.xml. Например, есть пункты меню "Открыть договор", "Предыдущая вкладка", "Следующая вкладка", "Заккрыть вкладку", им присвоены (через menu.xml) комбинации, соответственно, **Ctrl+O**, **Alt+Z**, **Alt+X**, **Alt+Q**.

Коды

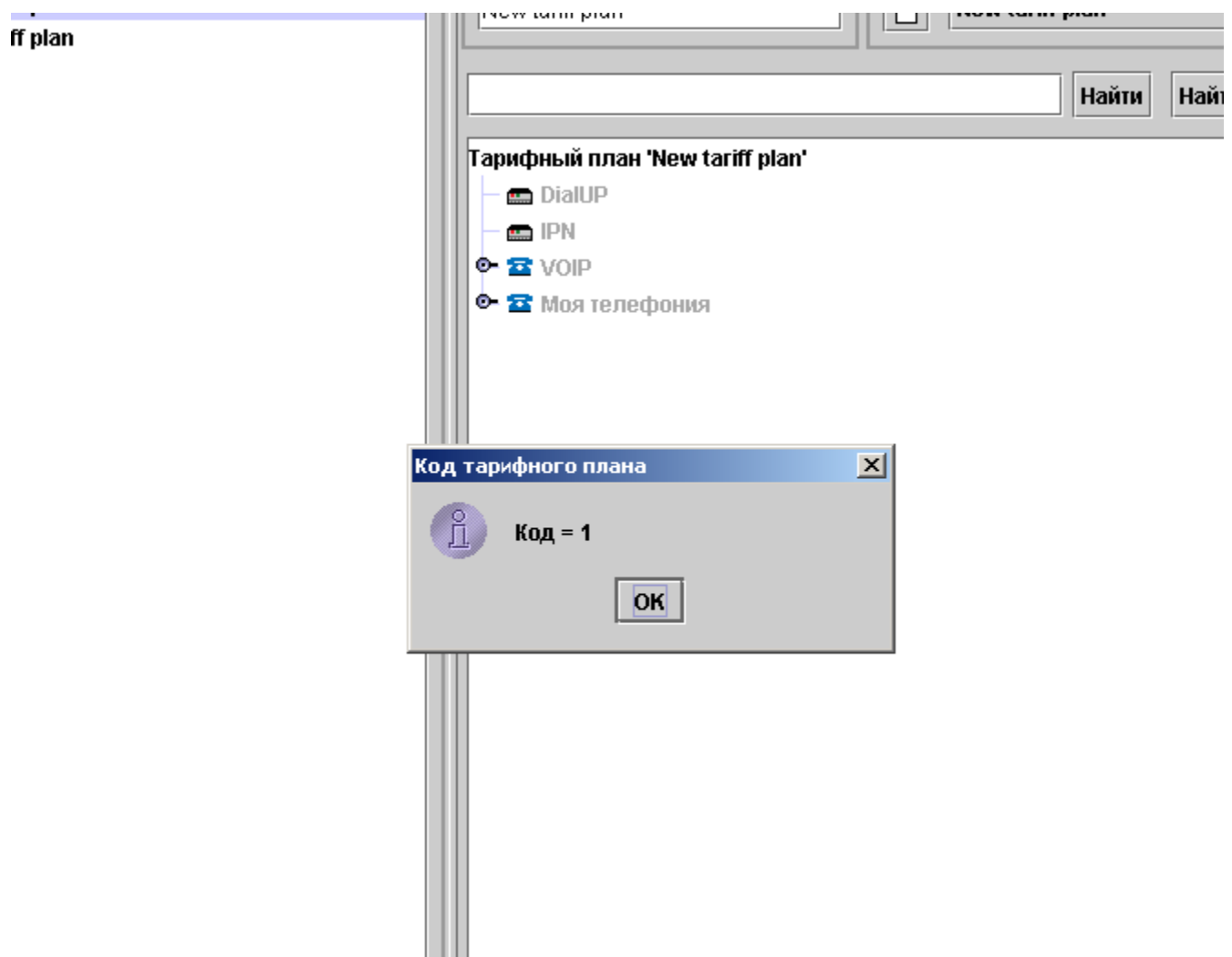
При настройке конфигураций, разработке расширений довольно часто необходимо получить внутренний идентификатор справочного значения, либо сущности биллинга. Идентификатор может отображаться в таблице значений в столбце ID, в редакторе открытой сущности, либо возвращаться по сочетанию клавиш **Ctrl + i**.

На снимках экрана ниже пример идентификаторов (кодов) типов платежей и тарифного плана (получен нажатием Ctrl + i в редакторе тарифных планов).



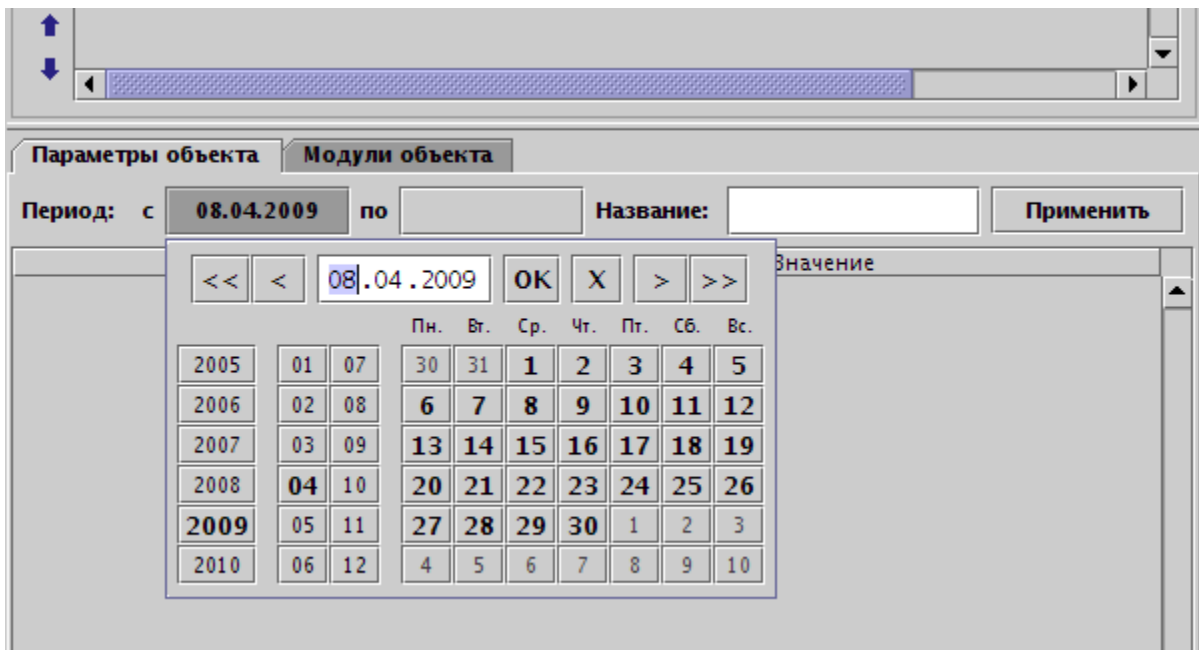
The screenshot shows a software interface with a sidebar on the left and a main window on the right. The sidebar contains a list of categories, with "Типы платежей" (Payment Types) selected. The main window displays a table of payment types with their corresponding IDs.

Типы платежей	ID
Все группы	
CyberPlat	17
EPort	21
OptimaPlus	33
Банк	15
Банк (Альфа банк)	1
Банк (Башкредит банк)	2
Банк (Социнвест-агро банк)	3
Взаимозачеты	4
Виртуальный	5
владивосток - корректирующий	27
владивосток-виртуальный	26
Группа	
BVCom	32
WebMoney	16
Дилеры	
Дилер KIRILL	29
Карточка "100"	12
Карточка "150"	13
Карточка "Приз"	14



Редактирование даты

При редактировании дат открывается календарь с установленным предыдущим значением даты, либо с текущей датой, если предшествующее значение отсутствовало.

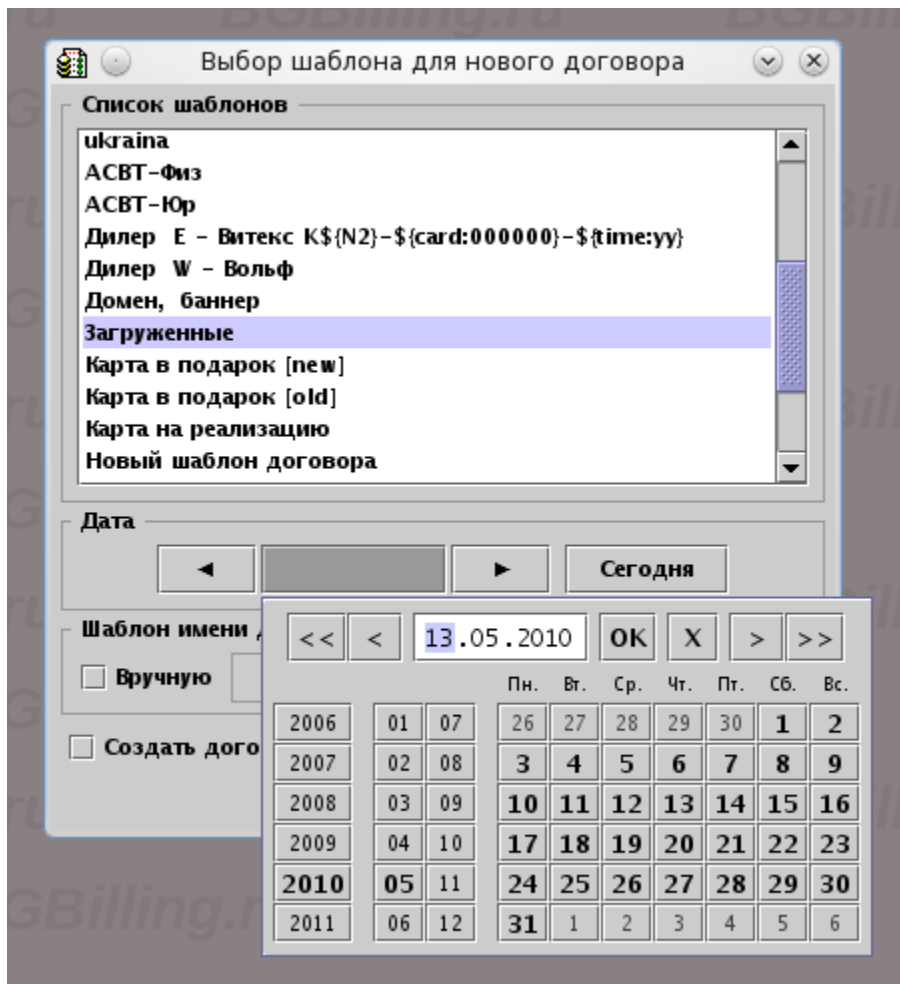


Кнопка **OK** устанавливает выбранную дату. **X** - оставляет существующее значение даты в поле ввода (отмена). Стрелки влево и вправо позволяют проматывать года и месяцы, стрелки продублированы кнопками с установленными значениями годов. Текущий год и месяц выделены жирным шрифтом.

Ctrl+стрелка вверх/Ctrl+стрелка вниз изменяют год, **Ctrl+стрелка влево/Ctrl+стрелка вправо** - месяц.

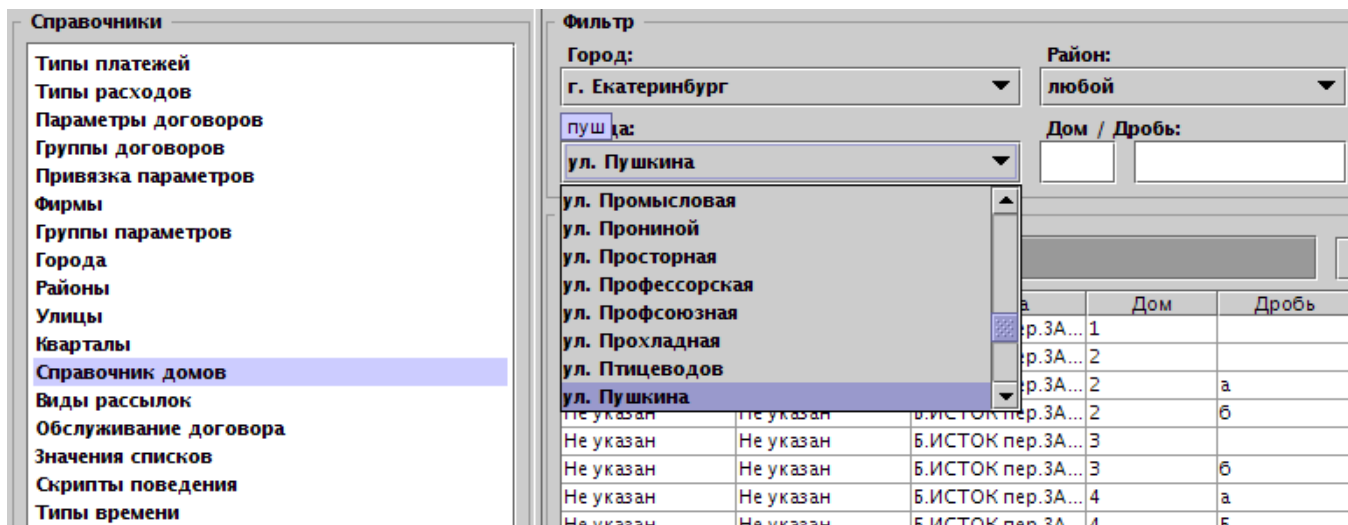
Также можно редактировать дату при выделенном поле даты (т.е. имеющим фокус), но не нажатом, без открытого календаря. В этом случае **Ctrl+стрелка вверх/Ctrl+стрелка вниз** изменяют месяц, **Ctrl+стрелка влево/Ctrl+стрелка вправо** - дату, **Ctrl+Insert** - устанавливает текущее число, **Ctrl+Delete** - очищают значение.

Для тренировки вы можете нажать меню **Договор=>Новый договор** либо кнопку **Новый договор** на панели инструментов. Нажатие поля ввода даты вызывает календарь. При вновь созданной базе биллинга список шаблонов будет пуст.



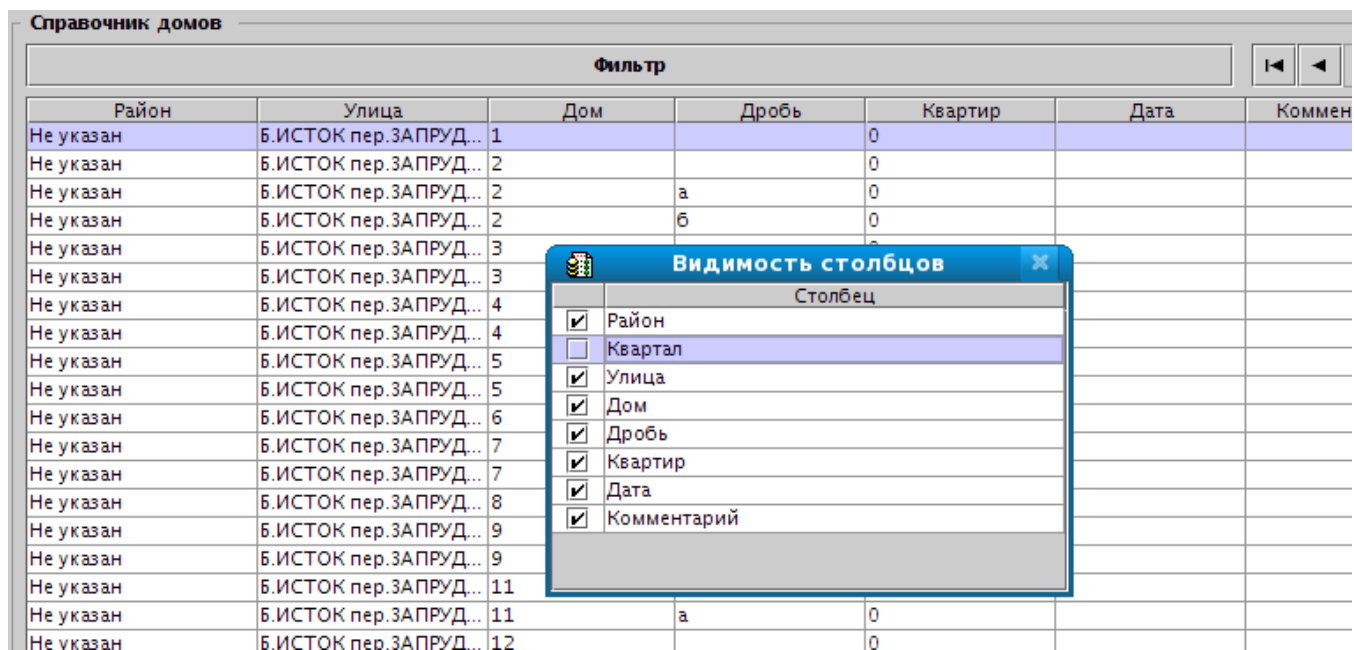
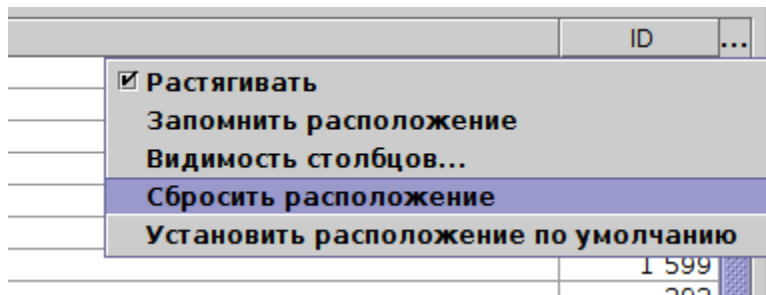
Выпадающие списки

При наборе с клавиатуры на выделенном выпадающем списке (ComboBox) производится автоматический поиск первой совпадающей по подстроке записи. При этом выше отображается набранный отрывок, а при навигации кнопками **вверх/вниз** выделение переходит по совпадающим полям. **Esc** - обнуляет отрывок поиска.



Таблицы

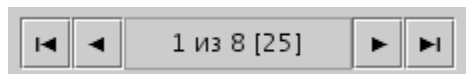
Практически все таблицы в биллинге настраиваемые. Можно изменять размер столбцов, положение (перемещением за заголовков), их видимость (щелкнуть кнопкой мыши на квадратике в правом верхнем углу таблицы - **Видимость столбцов...**), затем сохранить настройки (пункт меню **Запомнить расположение**). Пункт меню **Расширить** расширяет столбцы по ширине таблицы, **Сбросить расположение** - сбрасывает размер, положение и видимость на значения по умолчанию.



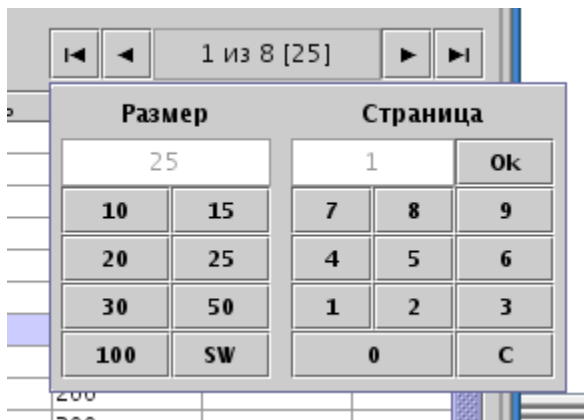
После перенастройки таблицы данные сохраняются на сервере для данного пользователя и автоматически применяются при дальнейшей работе пользователя с сервером.

Постраничный вывод

Вывод таблицы и списков данных с большим количеством пунктов организован в биллинге постранично. Для перемотки страниц и настройки количества записей на странице используется такой элемент управления:



Кнопки позволяют переходить на первую, предыдущую, последующую и последнюю страницы. В квадратных скобках отображается текущий размер страницы. При нажатии по средней части элемента отображается следующий диалог:



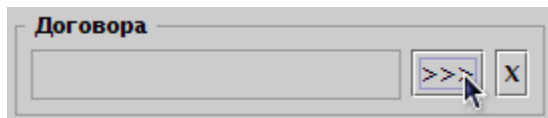
Правая область используется для быстрого перехода на нужную страницу. Для этого на изображённой справа клавиатуре мышью набирается номер страницы и кнопкой **ОК** осуществляется переход. Кнопка **С** позволяет отменить неправильный набор.

Левая область задаёт размер страницы. Нажатием кнопки можно установить predetermined значение. Для установки пользовательского значения нажмите кнопку **SW**, далее наберите на правой клавиатуре нужное значение и нажмите **ОК**.

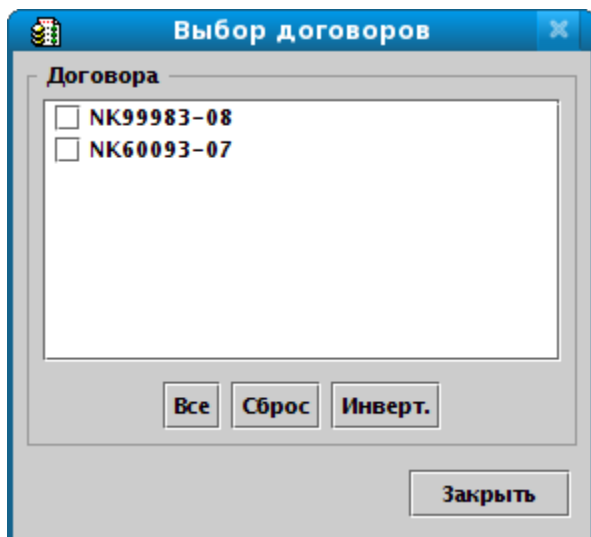
Сохранённые настройки записываются в файл `$USER_HOME/.bgbilling/config`.

Выбор договора

В элементе интерфейса выбор договора:



Необходимо обозначить один или несколько договоров, при этом отображаются только договоры, вкладки которых открыты в данный момент в клиенте. Диалог выбора договоров вызывается кнопкой >>>.



Отключение фонового рисунка

В клиентском приложении есть возможность отключить фоновый рисунок (например, при использовании терминал сервера), добавьте в `BGBillingClient/client.properties`

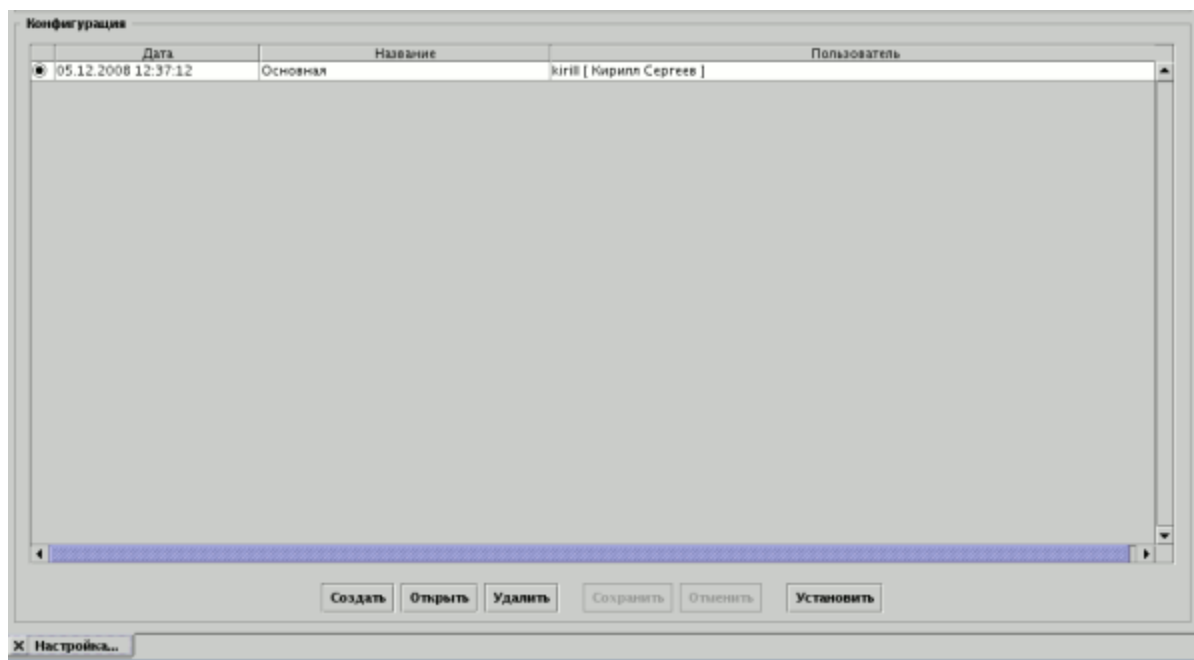
```
bg.enable=0
```

Настройка подсистем биллинга

Конфигурации

Очень большое количество редко меняющихся настроек поведения системы вынесено в конфигурации. Конфигурация - это текстовый блок, состоящих из записей вида: **<ключ>=<значение>**. На одной строке может быть только одна такая запись, символ # в начале строки означает комментарий. Порядок записей в тексте значения не имеет. При необходимости указания порядка в ключе вводятся дополнительные числовые индексы.

Конфигурации вводятся либо в текстовых **properties**-файлах (опции подключения к БД, базовые настройки), либо в редакторах конфигурации в клиенте биллинга, сохраняясь в базе данных. Ядро биллинга, каждый экземпляр модуля биллинга, плагины обладают различными конфигурациями, конфигурация ядра доступна в меню **Сервис=>Настройка=>Конфигурация**.



На приведённом снимке экрана изображён типовой редактор конфигурации. В таблице указан перечень конфигураций, из которых в текущий момент активен только один, установка активной конфигурации производится кнопкой **Установить**. Установка конфигурации позволяет осуществлять быстрый переход на заранее подготовленную конфигурацию. Создание новой конфигурации производится кнопкой **Создать**. Открытие - двойным кликом мыши, либо кнопкой **Открыть**.

В значениях параметров конфигурации возможна подстановка ранее указанных значений с помощью подстановок **{@имя параметра}**. Рассмотрим пример подстановки.

```
#
howYou=how you
#
some.kind.of.config.record=Thats {@howYou} should use macro!
```

Т.е. при такой конфигурации при взятии значения **some.kind.of.config.record** получаем в результате строку **"Thats how you should use macro!"**. Подставляемое значение должно быть обязательно определено ранее подстановки.

В большинстве случаев при смене конфигурации необходим перезапуск сервера, использующего данную конфигурацию. Например, при установке опций для работы RADIUS-сервера в конфигурации привязанного модуля DialUP необходим перезапуск RADIUS-сервера.

Для быстрого комментирования отдельных строк и блоков: **ctrl+shift+C**.

Логи́рование

Вы можете пропустить, либо бегло изучить этот раздел и вернуться к нему, если вам понадобится изменение стандартных параметров логирования.

По умолчанию логи серверов сохраняются в папке **log** приложения. В качестве подсистемы логирования используется библиотека log4j. Конфигурирование логирования заключается в правке файла **data/log4j.xml** (log4j-radius.xml - для RADIUS-сервера, log4j-collector.xml - для коллектора). Это xml-файл определенной структуры.

Код

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j='http://jakarta.apache.org/log4j/'>

  <appender name="APPLICATION" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="${log.dir.path}${log.prefix}.log" />
    <param name="MaxFileSize" value="100MB" />
    <param name="MaxBackupIndex" value="2" />
    <param name="Append" value="false" />

    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{MM-dd/HH:mm:ss} %5p [%t] %c{1} - %m%n" />
    </layout>

    <filter class="ru.bitel.common.logging.Log4JMDCFilter">
      <param name="key" value="nestedContext" />
      <param name="value" value="${log.prefix}" />
    </filter>
  </appender>

  <appender name="MQ" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="${log.dir.path}${log.prefix}.mq.log" />
    <param name="MaxFileSize" value="100MB" />
    <param name="MaxBackupIndex" value="2" />
    <param name="Append" value="false" />

    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{MM-dd/HH:mm:ss} %5p [%t] %c{1} - %m%n" />
    </layout>

    <filter class="ru.bitel.common.logging.Log4JMDCFilter">
      <param name="key" value="nestedContext" />
      <param name="value" value="mq" />
    </filter>
  </appender>

  <appender name="ASYNC" class="ru.bitel.common.logging.Log4jAsyncAppender">
    <appender-ref ref="APPLICATION"/>
  </appender>

  <root>
    <priority value="INFO" />
    <appender-ref ref="ASYNC" />
  </root>

</log4j:configuration>
```

Логи́рование в системе основано на категориях и контекстах. Разные контексты (server/script или collector/processor/loader) разнесены по разным аппендерам. Аппендер - это куда и как добавляется запись лога - т.е., например в файл server.log такого вида:

```
server 04-06/12:04:49 INFO [main] DefaultServerSetup - Init DB connection pools
server 04-06/12:04:49 INFO [main] DefaultServerSetup - Init trash pools..
server 04-06/12:04:49 INFO [main] DefaultServerSetup - Init trash pool trash_1
server 04-06/12:04:49 INFO [main] Server - Starting BGBillingServer..
server 04-06/12:04:49 INFO [main] Server - HTTP port: 6565
server 04-06/12:04:49 INFO [main] Server - Starting HTTP connector..
server 04-06/12:04:49 INFO [main] Server - HTTPS port: -1
```

root - это главная категория, по умолчанию priority=@value='INFO', т.е. все логирование в режиме info. **Чтобы переключить логирование в режим debug, необходимо указать здесь 'DEBUG'.**

```
<root>
  <priority value="DEBUG" />
  <appender-ref ref="ASYNC" />
</root>
```

Также в каждом аппендере можно указать фильтр по приоритету. Например, в root указать DEBUG, а в аппендере APPLICATION добавить указанную ниже ветку, чтобы логирование для всех контекстов, кроме APPLICATION было в режиме DEBUG.

```
<param name="Threshold" value="ERROR" />
```

```
<appender name="MQ" class="org.apache.log4j.RollingFileAppender">
  <param name="Threshold" value="ERROR" />
  <param name="File" value="${log.dir.path}${log.prefix}.mq.log" />
  <param name="MaxFileSize" value="100MB" />
  <param name="MaxBackupIndex" value="2" />
  <param name="Append" value="false" />

  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d{MM-dd/HH:mm:ss} %5p [%t] %c{1} - %m%n" />
  </layout>

  <filter class="ru.bitel.common.logging.Log4JMDCFilter">
    <param name="key" value="nestedContext" />
    <param name="value" value="mq" />
  </filter>
</appender>
```

RADIUS-протокол

Вы можете пропустить этот раздел при первичной установке системы и вернуться к нему, если вам понадобится работа с протоколом RADIUS.

RADIUS - протокол авторизации и аккаунтинга (передачи информации о соединении). В качестве транспорта используется UDP. Протокол бинарный, определяет формат передачи пакетов нескольких типов, в каждом из которых могут быть определены пары "**атрибут-значение**". NAS (Network Access Server) - сервер, через который происходит выход клиента с RADIUS-авторизацией.

Типы пакетов могут быть следующими:

1. AUTHENTICATION_REQUEST

Запрос авторизации, отправляется NASом RADIUS-серверу и содержит помимо идентификационной информации соединения, указанной выше, информацию о логине и пароле пользователя. Логин передаётся в открытом виде, пароль шифруется.

2. AUTHENTICATION_REJECT

Отказ в авторизации, может содержать атрибут с кодом ошибки.

3. AUTHENTICATION_ACCEPT

Пользователь авторизован. В данном пакете могут содержаться атрибуты, устанавливающие характеристики соединения пользователя (IP-адрес, скорость, максимальную длину сессии, частоту Update-пактов и т.п.).

4. ACCOUNTING_REQUEST

Запросы аккаунтинга могут быть трёх типов: **Start, Stop, Update**. Различаются они атрибутом **Acct-Status-Type** который равен 1, 2 или 3 соответственно. Данный тип запросов передаёт на RADIUS-сервер информацию о ходе соединения (соединение началось, завершилось или текущее состояние соединения).

5. ACCOUNTING_RESPONSE

Ответ RADIUS-сервера о том, что он получил запрос аккаунтинга. Ответ не содержит никаких атрибутов. Исключение составляет ответ MPD-серверу, который может содержать атрибут, информирующий NAS о необходимости разрыва соединения.

Посредством RADIUS-атрибутов передаётся вся информация пакета. Все используемые атрибуты должны быть описаны в файле **dictionary.xml**. Фактически файл определяет соотнесение кодов атрибутов их строковым обозначениям и типам. Он необходим при разборе RADIUS-пакета и при его создании, для выяснения кода атрибута по его имени.

Обратите внимание, что строковые обозначения атрибутов в нашем словаре могут отличаться от обозначений в словарях производителей оборудования. Для взаимодействия это совершенно не важно, т.к. в пакете атрибут обозначает числовой код.

Такой файл есть в любом приложении, которому может понадобится работа с RADIUS-протоколом. Во всех конфигурациях атрибуты должны указываться с именем таким же как и в данном конфигурационном файле. Атрибуты идентифицируются по имени, недопустимы атрибуты с одинаковыми названиями в пределах **dictionary.xml**. Перегрузка словаря приложением происходит при изменении конфигурации связанного с приложением экземпляра модуля.

Рассмотрим формат описания атрибута в словаре.

```
<attribute name="cisco-Fax-Connect-Speed" type="string" add="yes" code="8" />
```

Где:

- **code** - числовой код атрибута;
- **name** - строковое обозначение, отображается в логах, указывается в конфигурации;
- **type** - тип атрибута, возможные типы далее;
- **add** - при установке в **yes** в пакете имя строкового атрибута дублируется в поле со значением, это особенность CISCO устройств.

Возможные типы атрибутов:

- **octets** - последовательность байт;
- **string** - текстовая строка;
- **integer** - беззнаковое целое число, 4 байта;
- **ipaddr** - IP-адрес, 4 байта;
- **abinary** - текстовая строка с бинарными кодами.

Значения атрибутов в текстовых конфигурациях указываются в виде: **<NAME>=<VALUE>**, например: **Framed-IP-Address=192.168.168.2**. Для текстовой строки с бинарными кодами бинарный код указывается как **\0x<code>**, где **<code>** - код в 16-ичной системе счисления. Например: **cisco-SSG-Command-Code=\0xC SERVICE A**.

Тегированные атрибуты указываются в виде: **<NAME>:<TAG>=<VALUE>**, например: **ERX-Activate-Service:2=testtest**. Тегированный атрибут в словаре должен быть помечен атрибутом **tag**, который определяет логику для разных типов атрибутов.

Таблица. Логика тегирования

Тип тегирования /Тип атрибута	integer	string
1	Тег - первый байт значения, значение от 0x01 до 0x1F, если тегирования нет - его значение 0. Оставшиеся три байта - само число.	Тег - первый байт значения, значение от 0x01 до 0x1F, если тегирования нет - его значение 0. Оставшиеся байты - значение строки.
2	Не поддерживается.	Если значение первого байта от 0x01 до 0x1F, то это тег, иначе - первый байт - это начало значения, тегирования нет.

При необходимости указания нескольких атрибутов они разделяются точкой с запятой, например: **Framed-IP-Address=192.168.168.2;Service-Type=1;ERX-Activate-Service:2=testtest**. Для указания точки с запятой в теле атрибута нужно писать её парно. Например: **Framed-IP-Address=192.168.168.2;cisco-SSG-Service-Info=QU;;1024000;;512000;;D;;1024000;;512000**.

Строковые представления для некоторых перечислимых числовых RADIUSатрибутов в данный момент не поддерживаются. Например, значению 1 атрибута **Service-Type** соответствует Login.

Настройка сервера биллинга

Конфигурация

После установки сервера и клиента сервер должен быть настроен, для этого откройте в клиенте пункт меню **Сервис=>Настройка=>Конфигурация**. На тулбаре кликните по иконке "Новый элемент" для создания новой конфигурации сервера.

Если в дальнейшем где-то упоминается **Конфигурация сервера**, то следует знать, что речь идёт именно об этой конфигурации. В биллинге много различных редакторов конфигурации, ориентироваться в них на стадии освоения системы может быть затруднительно.

Вид и поведение: Metal [system] [crossplatform] Шрифт: Обычный Применить

Закрытый период: Основной 14.07.2006 Установить

Конфигурация

Поиск:

Тип	Наименование	Значение
Строка	web.auth.class	
Логический	web.api.enable	false
Логический	api.enable	false
Строка	mail.encoding	utf-8
Строка	mail.smtp.host	
Строка	mail.smtp.localhost	
Строка	mail.smtp.port	25
Строка	mail.smtp.user	
Строка	mail.smtp.pswd	
Строка	mail.from.email	
Логический	mail.debug	false
Логический	xslt.cache	true
Строка	web.admin.password	
Строка	web.mail.server.url	
Число	contract.parameter.rule.write.group	1
Число	contract.parameter.rule.write.user	3
Число	contract.parameter.rule.read.group	1
Число	contract.parameter.rule.read.user	3
Число	password.length.min	5
Число	password.length.auto	8
Число	password.length.max	10
Строка	password.chars	0123456789
Строка	client.gui.contract.limit.values	=-1000;=-500;=-300;=-150;=-100;=-50;=-10;=0;=5/1;=30/1;...
Число	logon.counter.max	2147483647
Число	logon.timeout.period	0
Число	logon.timeout.lock	21600
Число	logon.lock.cache.size	100
Число	logon.lock.cache.expired	600
Строка	logon.timeout.action	+
Строка	temp.dir.path	
Логический	debug	false

Справка по данному параметру отсутствует

Сохранить Отменить Установить Шаблон

Настройка...

ПАРАМЕТР	ТИП	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ	ОПИСАНИЕ
Управление API			
api.enable	логический	false	Включение Cuscom API
web.api.enable	логический	false	Включение Web API
Системные настройки сервера			
temp.dir.path	строка	-	Путь к временному каталогу, используется обработчиком логов для загрузки логов по FTP и сервером биллинга для хранения промежуточных файлов. Если не указан, то используется каталог BGBillingServer/tmp
Система алармов - экстренных оповещений			
alarm.mail	строка	-	На какой адрес высылать оповещения, указать обязательно!
alarm.disabled	строка	-	Алармы, которые надо игнорировать. Ключи через запятую.
alarm.send.to.log	строка	-	Алармы, которые направляются в log. Необходимо указать список ключей через запятую или символ "**", чтобы ВСЕ алармы писались в лог.
BGSecure			
bgsecure.usergroup.convert	логический	false	При переходе на 9.2204 и выше позволяет включить конвертирование пользовательских групп в Роли для новой схемы разграничения прав. true - включить. По умолчанию - false

bgsecure.usergroup.convert.user.domain.to.role	логический	true	Создавать отдельный набор прав, если у пользователя в настройках задан домен
bgsecure.usergroup.convert.user.action.to.role	логический	true	Создавать отдельный набор прав, если у пользователя есть персональные права
bgsecure.check	число	1	Проверка прав пользователей. 0 - не проверять
bgsecure.log	число	-1	Логирование действий в журнале событий. -1 - не логировать, bgsecure.log=1 - логировать
Протокол			
server.response.debug	логический	false	server.log XML, DEBUG
header.name.remote.addr	строка	X-Real-IP	Заголовок HTTP-пакета, в котором передаётся IP-адрес клиента, если параметр не указан или не передан, то используется request.getRemoteAddr() нужен при проксировании запросов с помощью nginx
max.post.size	число	10 000 000	Максимальный размер запроса к серверу (в байтах), запросы большего размера обрезаются, что может привести к некорректной работе сервера.
XSLT			
xslt.cache	логический	true	Кэширование XSLT-шаблонов памяти: true - включить Необходимо отключать опцию на момент модификации любых XSLT-шаблонов
contractcard.l	строка	-	Заголовок и адрес к шаблону карточки (доступны в свойствах договора), например card_inet.xml:Капра регистрации
Личный кабинет (web)			
web.mail.server.url	строка	-	Адрес ЛК (в письме с логином и паролем)

Конфигурация

```

#-----
#   BGBillingClient
#-----
#
client.gui.contract.tree.order=parameters objects hierarchy status limit mode face balance tariff modules
groups web tariffGroup script addAction memo
#
client.gui.contract.limit.values=-2000;=-500;=-300;=-150;=-50;=-30;=-10;=0;-5/1;-50/1;-100/1;-15/3;-50/3;-100/3
#
#
#   .
#   .
#   1234. :      ,
#   , , . 13,
#   , :
#   .
client.gui.contract.searchDefaultFilters=13
#
client.gui.pattern.limit.values=-2000;-500;-300;-150;-100;-50;-10;0;5;30;100;15;50;100
#
#-----
#   -
#-----
#
contract.param.pattern.keys=bank,firm
#
contract.param.pattern.bank.title= (
contract.param.pattern.firm.title= ( , , /)
#
contract.param.pattern.bank.pids=12,34
contract.param.pattern.firm.pids=1,23,8,4,25
#
#-----
#   " "
#-----
#   - 1, 0 -
address.unique.check=1
#   ( ${comment} - )

```

```

addr.format=({index})(, {city})(, {area})(, {quarter})(, {street})(, . {house})({frac})(, . {flat})(
${room})(, {pod} .)(, {floor} .)
#
#address.create=1
#
#-----
#
#-----
#           (int), (string), (date)
##
address.country=countPeople,test
address.country.countPeople.title=
address.country.countPeople.type=int
address.country.test.title=
address.country.test.type=string
#   ,   country   city
#   ,   country   area
#   ,   country   quarter
#   ,   country   street
#
address.street.boxIndexRange.title=
address.street.boxIndexRange.type=string
#
address.house=dateConnecting, test, floorRange, entranceRange
address.house.dateConnecting.title=
address.house.dateConnecting.type=date
address.house.test.title=
address.house.test.type=string
#
address.house.entranceRange.title=
address.house.entranceRange.type=string
#
address.house.floorRange.title=
address.house.floorRange.type = string
#-----
#
#-----
#           , 1 -
check.double.tariff=0
#
allow.future.payment=0
allow.future.charge=0
#
allow.closed.payment=0
allow.closed.charge=0
#   "   , 1 - , 2 - .
contract.status.monitor.saldo.show.mode=1
#           , 1 -
reject.limit.update=0
#
#tariffOption.dontCheckOnActivateAlreadyActivated=1
#
#-----
#
#-----
#   ,
contract.status.list=0:;1: ;2:;3:;4:;5:
#   ,
contract.status.no.manual.set=1,5
#   (   ,   )
contract.status.deprecated=
#           , 1 -
independ.subcontract.status.change=0
# ... ...
depend.subcontract.status.change=1
#
#-----
#   ,
#-----
#   ,
credit.contract.active.status=0

```

```

# C ,
# ,
credit.contract.open.by.payment.status=2,3
# C , ,
credit.contract.override.future.to.active.status=2
# , credit.contract.open.by.payment.status
#do.not.open.contract.on.payment=1
# ,
#do.not.open.groups.on.payment=
#
#-----
#
#-----
# , 1 -
closed.date.enabled=1
# ,
#closed.date.groups.id=1,2,3 ( 9.XXXX closed.date.roles.id - id )
# ,
#closed.date.users.id=1,58,666
# ( )
#closed.date.disabled.ActionDeleteContractCharge=1
# ( )
#closed.date.disabled.ActionDeleteContractPayment=1
# ( )
#closed.date.disabled.ActionDeleteContractService=1
# ( )
#closed.date.disabled.ActionDeleteContractTariffGroup=1
# ( )
#closed.date.disabled.ActionDeleteContractTariffPlan=1
# ( )
#closed.date.disabled.ActionUpdateContractCharge=1
# ( )
#closed.date.disabled.ActionUpdateContractDate1=1
# ( )
#closed.date.disabled.ActionUpdateContractDate2=1
# ( )
#closed.date.disabled.ActionUpdateContractPayment=1
# ( )
#closed.date.disabled.ActionUpdateContractService=1
# ( )
#closed.date.disabled.ActionUpdateContractTariffGroup=1
# ( )
#closed.date.disabled.ActionUpdateContractTariffPlan=1
# ( )
#closed.date.disabled.ActionObjectUpdate=1
# ( )
#closed.date.disabled.ActionContractStatusChange=1
#
#-----
#
#-----
#
scheduler.periodic.thread.count=5
# ( )
scheduler.nonperiodic.thread.count=5
#
#-----
#
#-----
#
loader.add.process=1
#
#-----
#
#-----
# (1-, 0-).
# print, error ; BGBillingServer
log.function.process=1
#
#-----
# -

```

```

#-----
# (UTF-8 , )
dynamic.src.encoding=UTF-8
# BGBillingServer
dynamic.src.dir=dyn
#
#-----
# Web-
#-----
# :
# 1 - ( )
# 2 -
#web.sub.contract.auth.mode=1
# ( )
#web.error.all=1
# xml xsl
#
web.xslt=http://127.0.0.1:8080/bgbilling/xsl/
# xml https xsl
#
web.xslt.https=https://127.0.0.1:8443/bgbilling/xsl/
# XML - 1
web.add.contract=0
# Web-
web.exit.redirect=about:blank
# Web- (Web-)
webquery.log=0
# Web
multipart.max.post.size=1048576
# ( )
server.error.404=/error/error404.html
server.error.403=/error/error403.html
#
#-----
# Web-,
#-----
# MD5- Web-, " => Digest"
#web.admin.password=21232F297A57A5A743894A0E4A801FC3
# Web- ; 1:
# 0 -
# 0 - , 1 - , 2 -
web.auth.modes=0:1
# ru.bitel.bgbilling.kernel.auth.ModuleAuth
# web.auth.class=
# , 0 -
web.max.day.request.count=0
#
password.length.min=5
password.length.max=10
#
password.length.auto=6
#
password.chars=1234567890
#
web.auth.deny.statuses=
#
web.auth.deny.periodClose=true|false
#
#-----
# Web-, ,
#-----
#
#
logon.counter.max=20
#
logon.timeout.period=0
#
logon.timeout.lock=21600
#
logon.lock.cache.size=100
#

```

```


logon.lock.cache.expired=600
#      (+ ^)
logon.timeout.action=+
#
#-----
# Web-, ,
#-----
#      , E-Mail,
contract.password.forgot.email.param.id=< >
#
contract.password.forgot.expire.hour=24
#
contract.password.forgot.link=http://host:8080/bgbilling/webexecuter?webexecuter?
action>Password&mid=0&module=contract.basic
#
contract.password.forgot.email.subject=
#
contract.password.forgot.email.body=      {contract} -      ( {hour} )      .
#
contract.password.forgot.char.array=1234567890QWERTYUIOPLKJHGFDSA ZXC VBNMqwertyuioplkjhgfsazxcvbnm
#
#-----
# Web-,
#-----
#
limit.max.current.msg=      .      /
limit.max.nopayed.msg=      .
#
#-----
converter.groups.to.labels=true -      .      1 .      - true.

```

Значение большинства параметров пояснены в комментариях, либо присутствует ссылка на главу с более подробным описанием подсистемы.

Если вы не знаете, что ввести в качестве параметра в конфигурацию, возьмите этот параметр из примера, приведённого выше. После сохранения конфигурации выберите строку с ней в таблице и нажмите **Установить**.

В поле **Вид и поведение** вы можете изменить текущую тему оформления клиента.

Установка темы оформления отличной от Metal может привести к проблемам с отрисовкой. Смена темы оформления производится  пользователем на его страх и риск, нарушение в отрисовке клиента не считается ошибкой ПО.

Почтовая подсистема

Необходимо обязательно настроить опции почты в конфигурации сервера:

ПАРАМЕТР	ТИП	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ	ОПИСАНИЕ
mail.smtp.host	строка	localhost	Адрес SMTP сервера
mail.smtp.user	строка	-	Логин для авторизации на SMTP сервере
mail.smtp.pswd	строка	-	Пароль для авторизации на SMTP сервере
mail.from.email	строка	-	E-mail адрес отправителя
mail.from.name	строка	BGBilling Server	Имя отправителя
mail.to.email	строка	-	E-mail адрес Администратора биллинга
mail.to.name	строка	Support team	Имя Администратора биллинга
mail.encoding	строка	UTF-8	Кодировка писем
mail.smtp.localhost	строка	-	Имя хоста, отправляемое в команде EHLOW SMTP-серверу, если необходимо отличное от имени хоста, где запущено приложение биллинга. Параметр общий для для всех приложений биллинга, отправляющих почту, он может быть также указан в скрипте запуска приложения ключём -Dmail.smtp.localhost=<host> либо в .properties файле приложения
mail.debug	логический	false	Включение отладки SMTP-обмена в .out log, true - включить, false - выключить

Данные настройки используются для отправки почтовых сообщений всеми приложениями биллинга. Настройка почтовой подсистемы очень важна, т.к. иначе не будет работать система экстренного оповещения "алармы". Для проверки корректности настройки подсистемы произведите модификации конфигурации сервера биллинга и попытайтесь отправить отчёт по балансу в договоре. Для этого откройте любой созданный договор, выберите в дереве узел **Баланс** и нажмите кнопку с изображением конверта над таблицей.

В **mail.smtp.host** укажите ваш почтовый сервер, адрес и имя, подставляемые в поля ОТ письма (**mail.from.email**, **mail.from.name**). Также укажите адрес и имя администратора (**mail.to.email**, **mail.to.name**). В кодировке можете указать **windows-1251**, либо **UTF-8**, либо **KOI8-R** на ваш выбор.

Если SMTP-сервер требует авторизации, логин и пароль указываются в параметрах **mail.smtp.user**, **mail.smtp.pswd**.

Для поддержки отправки почты через SSL-соединение (например, smtp.gmail.com) добавьте следующие настройки (SMTP порт 465).

```
mail.properties.mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.properties.mail.smtp.socketFactory.fallback=false
mail.properties.mail.smtp.quitwait=false
mail.properties.mail.smtp.port=465
mail.properties.mail.smtp.socketFactory.port=465
```

Система оповещения

В опции **alarm.mail** укажите почтовый ящик для отправки экстренных оповещений о нештатной ситуации в биллинге (потеря связи с БД, недостаток памяти и т.п.). При наступлении нештатной ситуации приложение биллинга вышлет письмо со следующей темой: [**<имя приложения>**] **<название ошибки>**, где:

- **<имя приложения>** - название процесса биллинга, уникально для каждого процесса биллинга, может быть переопределено передачей ключа `-Dapp.name=<новое имя>` в скрипте старта процесса (.sh, .bat);
- **<название ошибки>** - краткое описание ошибки.

Например: "[BGBillingServer] Достигнут лимит одновременных подключений к Master базе". Если, например, у вас установлено несколько серверов биллинга и необходимо получение алармов на один ящик, то вы можете скорректировать скрипт `server.sh` следующим образом (имя приложения изменено на `BGBillingServer2`):

```
if [ "$1" = "start" ]; then
    #starting
    nohup ${JAVA_HOME}/bin/java -Dapp.name=BGBillingServer2 ...
else
```

Аналогично может быть скорректировано имя любого приложения. В теле письма содержится идентификатор проблемы, время регистрации и краткое описание, например:

```
ID : db.master.connection.limit.over
   : 10.03.2009 16:12:01

   .
   Master .

Connections pool to Master status Idle: 0; Active: 4; maxActive: 4; maxIdle: 4
```

Для каждого типа оповещения определен минимальный интервал между отправками писем. Интервал существует для избежания большого потока писем одного содержания. Он может быть изменён установкой опции конфигурации сервера биллинга **alarm.min.interval**. **<key>=<seconds>**, где:

- **<key>** - идентификатор события, например `db.master.connection.limit.over`;
- **<seconds>** - минимальное время в секундах между отправками писем по этому типу события.

Например:

```
alarm.min.interval.db.slave.connect.error=240
```

Хоть это и не рекомендуется, но можно отказаться от получения каких-либо оповещений. Для этого в конфигурации перечислить ключи (ID) оповещений через запятую. Например:

```
# , . .
alarm.disabled=bad.java
```

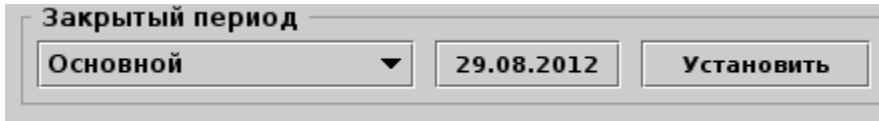
Также существует возможность отправки оповещений не только на почту, но и в log-файл. Это достигается путем добавления в конфигурацию сервера опции

```
#, log. "*"
alarm.send.to.log=<keys>
```

Log-файл располагается в стандартном каталоге логов сервера и называется ***.alarm.log**. Также стоит отметить следующее: если ключ аларма указан и в опции **alarm.disabled**, и в **alarm.send.to.log**, то на почту оповещение не придет, но в лог запишется; если ключ указан только в **alarm.send.to.log**, то оповещение придет и на почту, и в log-файл; если же ключ указан в **alarm.disabled**, но его нет в **alarm.send.to.log**, то оповещение будет полностью проигнорировано.

Закрытый период

В меню настройки сервера (пункт меню **Сервис=>Настройка=>Конфигурация**) также существует возможность указания закрытого периода для сущностей ядра и модулей системы. Проверка закрытого периода работает только при установленной опции конфигурации сервера биллинга **closed.date.enabled=1**. Также имеется возможность выборочного отключения проверки закрытого периода. Флаги отключения см. [конфигурация ядра](#), и в описании конфигурации модулей.



Назначение закрытого периода в том, чтобы не допускать изменения или удаления сущностей в периоде, уже закрытом для изменений в балансе. Для закрытого периода устанавливается его правая граница. Логика работы проверки на закрытый период для изменения сущностей, имеющих период работы (две даты, например, тарифный план), следующая:

- если и левая, и правая границы периода сущности лежат левее закрытой даты, то невозможно любое изменение этих сущностей;
- если левая граница периода сущности лежит в закрытом периоде, а правая нет (либо не существует, либо лежит правее закрытой даты), то возможно изменение параметров данной сущности, за исключением изменения левой границы ее периода (т.к. она лежит в закрытом периоде) и установки правой границы внутри закрытого периода;
- если обе границы периода сущности лежат правее закрытой даты, то возможно любое изменение этой сущности за исключением тех, которые приводят к пересечению периода сущности с закрытым периодом.

Для сущностей с одной датой (например, приход) логика работы проверки на закрытый период для изменения следующая:

- если дата сущности лежит внутри закрытого периода, то невозможно любое изменение этой сущности;
- если дата сущности лежит вне закрытого периода, то возможно любое изменение этой сущности, за исключением установки ее даты внутри закрытого периода.

Логика проверки на закрытый период для удаления сущностей едина для обоих типов (для сущностей с двумя датами в проверке участвует только левая граница ее периода, т.к. для проверки на пересечение этого достаточно):

- если дата сущности лежит внутри закрытого периода, то удаление ее невозможно;
- если дата сущности лежит вне закрытого периода, то ее удаление возможно.

По умолчанию в системе присутствует только один закрытый период "Основной". При необходимости, возможно определение в конфигурации сервера биллинга дополнительных периодов следующим образом:

```
closed.date.types=<id1>:<title1>;<id2>:<title2>..
```

Где:

- **<idN>** - уникальный код периода, целое число больше 1;
- **<titleN>** - название периода.

Например:

```
closed.date.types=2: ;3:
```

Для привязки какой-либо проверки периода к иному от основного необходимо указать в конфигурации ядра, либо модуля:

```
closed.date.type.<key>=<typeId>
```

Где:

- **<key>** - ключ проверки, тот же, что используется для её отключения;
- **<typeId>** - код периода.

Например, привязка проверки периодов при правке платежей и расходов к периоду с кодом 2:

```
closed.date.type.ActionUpdateContractPayment=2  
closed.date.type.ActionDeleteContractPayment=2
```

Модули

Модули биллинга необходимы для расширения функциональности системы, в основном для тарификации различных услуг. В этой главе предполагается, что ваш сервер биллинга установлен в **C:\BGBillingServer** на ОС Windows. Если вы установили его в другое место, используйте ваши пути.

Установка модуля производится инсталлятором, который расположен в каталоге **BGBillingServer**.

Для установки модуля нужно запустить команду

```
# Linux
./bg_installer.sh install { }
# Windows
bg_installer.bat install { }
```

Эта команда скачает новый модуль с сайта установит его в систему.

В качестве альтернативы (если например с этой машины нет выхода в интернет), вы можете скачать zip-файл модуля сами, положить его рядом с инсталлятором и запустить

```
# Linux
./bg_installer.sh { }
# Windows
bg_installer.bat { }
```

Если инсталляция прошла успешно, будет выведено сообщение (например):

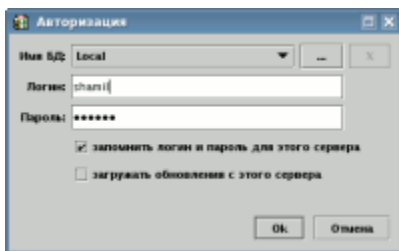
```
Module dialup version 3.5 was successfull installed!
```

Для установки обновления того же модуля биллинга (версии пакетов совпадают) необходимо выполнить:

```
#Linux
./bg_installer.sh { }!
#Windows
bg_installer.bat { }!
```

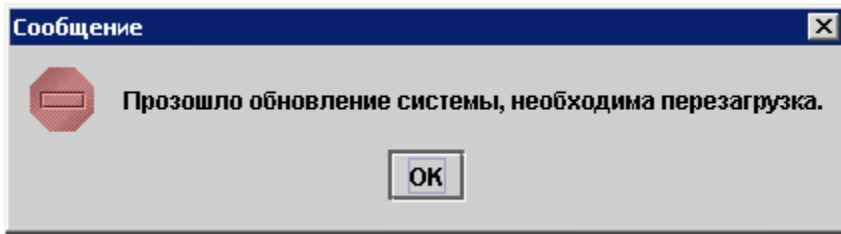
Символ "!" после имени пакета указывает инсталлятору, что установка необходима даже, если такой модуль такой версии уже стоит.

После того, как установили модуль, необходимо перезапустить сервер биллинга. После этого должен быть обновлён клиент. Для этого запустите его и подсоединитесь к обновлённому серверу биллинга.

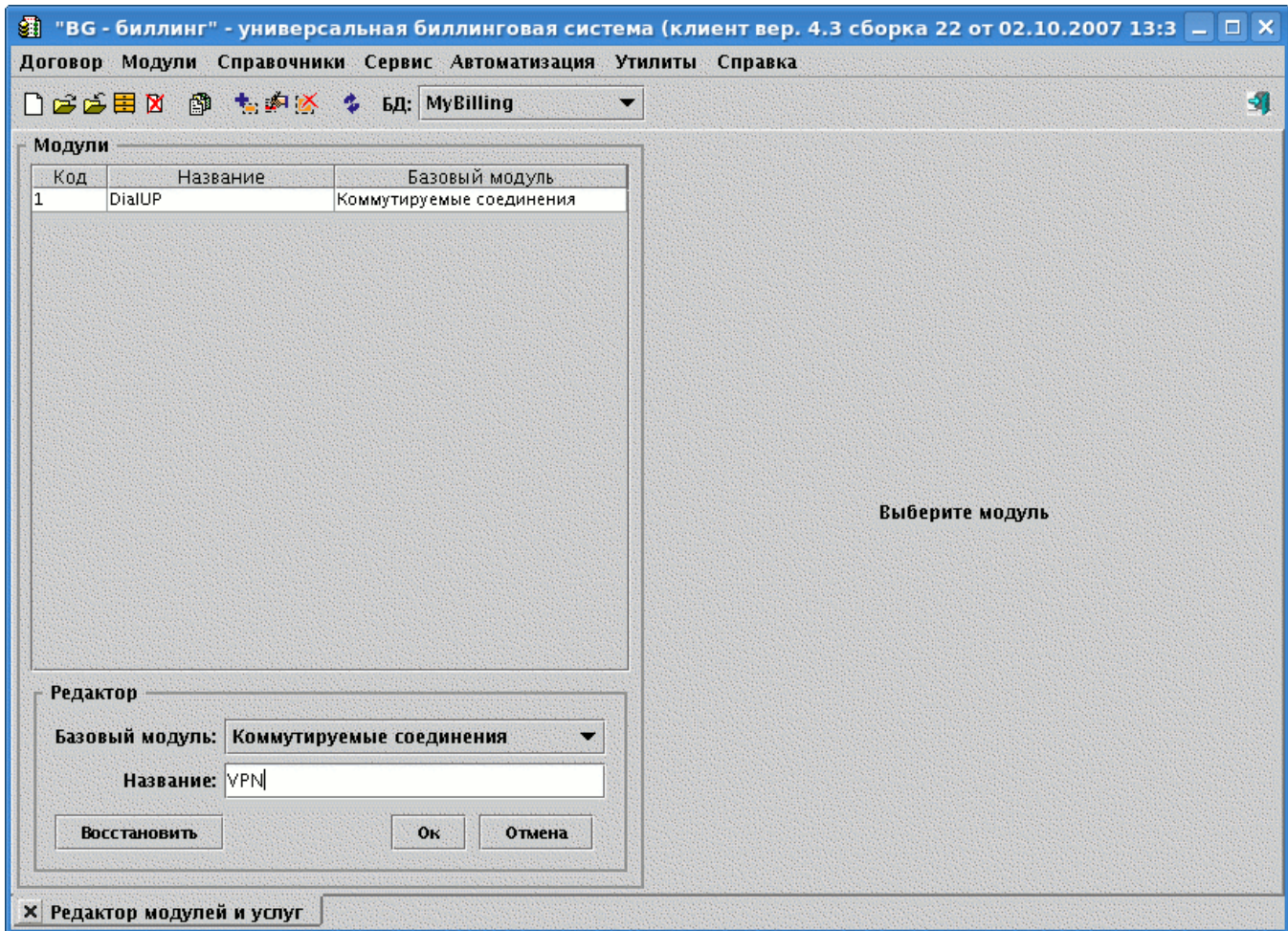


При подключении установите в окошке логина галочку **загружать обновления с этого сервера**. Либо произведите вход на сервер, а после вызовите пункт меню **Сервис=>Настройка=>Принудительное обновление клиента**.

Если все прошло успешно, будет выведено окошко с сообщением.



Данное сообщение означает, что код клиентской части, необходимый для работы с модулем, скопирован с сервера. Иначе говоря, модуль и его библиотеки установлены, но нет обособленной единицы (экземпляра) модуля со своими данными. Для того чтобы её создать, следует запустить **Модули** => **Редактор модулей и услуг**.



Слева расположена таблица созданных экземпляров модулей: **код экземпляра (ID модуля)**, его произвольное название и базовый модуль. Базовый модуль - ничто иное, как программный код (библиотеки), осуществляющий поддержку функционала модуля. А сам экземпляр - это совокупность данных в базе с уникальным кодом.

От одного базового модуля можно породить неограниченное количество экземпляров модулей. Например, можно сделать два экземпляра от базового модуля **DialUp (Коммутируемые соединения)**: DialUp и VPN.

Каждому из них будут соответствовать свои таблицы в базе, свои списки логинов, NASов и т.п. Код, обрабатывающий все эти данные на сервере, будет общим. Каждый из экземпляров будет обслуживать собственный RADIUS-сервер.

После создания экземпляра модуля ему присваивается уникальный код, далее в руководстве для краткости называется **Код модуля** и используется во многих конфигурациях. Экземпляр модуля в дальнейшем руководстве также часто называется просто модуль, необходимо понимать разницу и выяснять нужное значение по контексту.

Для создания нового экземпляра модуля нажмите кнопку **Новый элемент** на общей панели инструментов, для редактирования - кнопка **Редактировать**, либо двойной клик по строке таблицы. Изменить можно только название экземпляра.

Удаление экземпляра модуля: кнопка **Удалить**. Вам может понадобиться эта кнопка после истечения тестовой лицензии на какой-либо модуль, решение о приобретении которого принято не было.

Удаляя модуль, вы удалите и все связанные с ним данные.

Справа расположены услуги, существующие в модуле.

Услуга - это способ разделения наработки, а также возможность активации тех или иных сервисов в некоторых модуля путём добавления **Разрешённых услуг договора**. Каждой услуге также соответствует уникальный код и он также часто используется в конфигурациях как **Код услуги**.

Для редактирования списка услуг экземпляра модуля выберите строку с экземпляром модуля и воспользуйтесь панелью инструментов над таблицей с услугами. С помощью кнопки **Новый элемент/Редактировать элемент** вы можете изменить параметры услуги. В частности задать название, флаг используемости, а также единицу измерения (штука, секунда, байт).

Код	Название	Базовый модуль
130	Assist.Ru	Assist.RU
148	Buy e-Money	Buy e-Money
145	BVCom	BVCom
43	CerberCrypt2	CerberCrypt
240	CerberCryptTest	CerberCrypt
8	DBA	DBA
21	Dial-Up	Коммутируемые...
139	DrWeb	Dw.Web AV-DESK
209	DrWebN	Dw.Web AV-DESK...
1	E-Mail	E-Mail
186	Enaza	Enaza
144	HD2	HD
179	INET	Inet
208	Internet	Inet
33	IPN	Выделенные к...
213	Mobi.Деньги	Mobi.Деньги

Код	Название	Используется	Ед. измерения
81	MPS	<input checked="" type="checkbox"/>	
308	MPS(копия)	<input checked="" type="checkbox"/>	

Редактор услуги

Название:

Единица измерения:

Используется

Ok Отмена

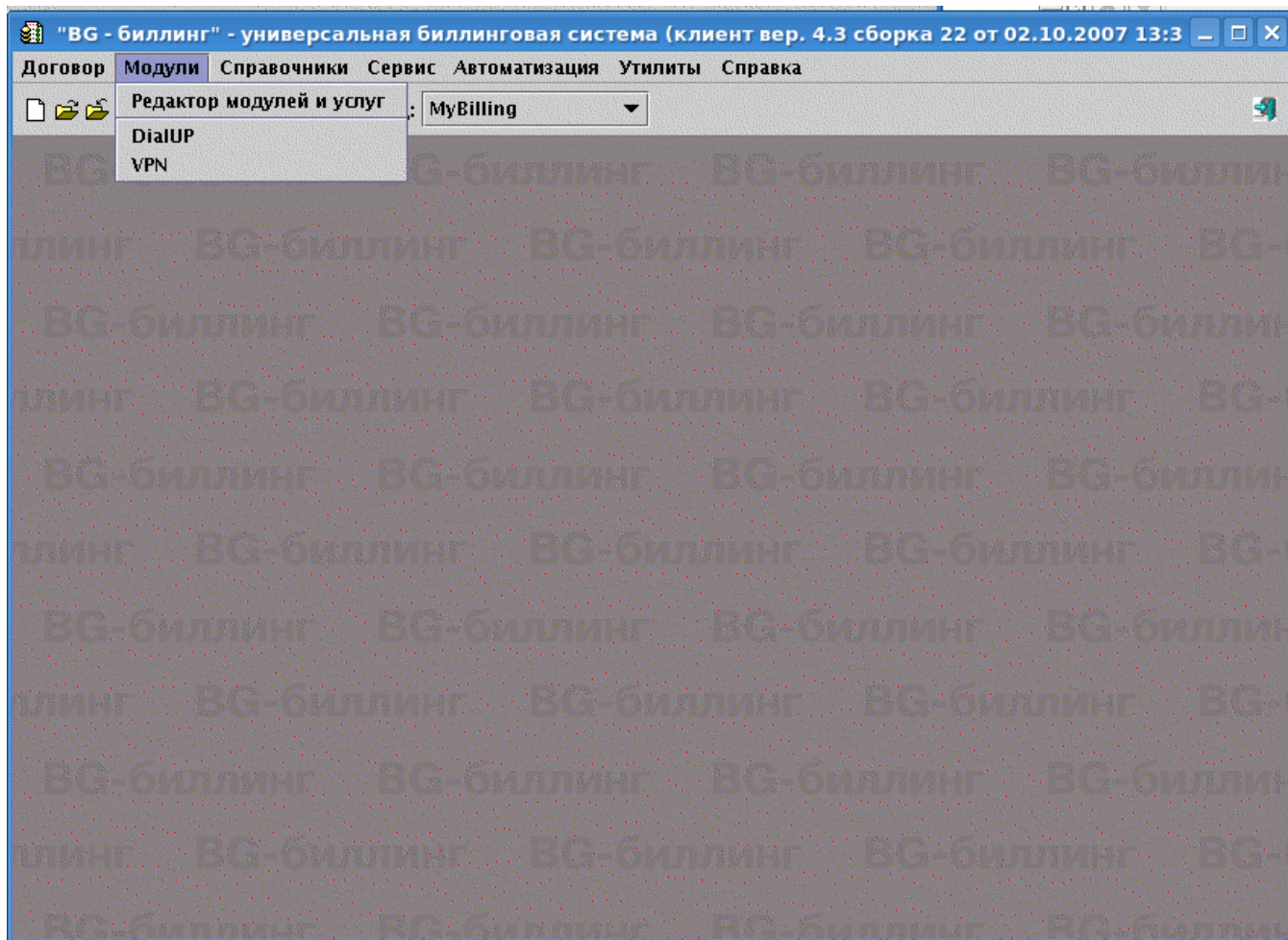
При необходимости создать множество однотипных услуг, можно воспользоваться кнопкой **Клонировать услугу**. После этого останется лишь скорректировать название.

Код	Название	Базовый модуль
130	Assist.Ru	Assist.RU
148	Buy e-Money	Buy e-Money
145	BVCom	BVCom
43	CerberCrypt2	CerberCrypt
240	CerberCryptTest	CerberCrypt
8	DBA	DBA
21	Dial-Up	Коммутируемые соедин...
139	DrWeb	Dw.Web AV-DESK
209	DrWebN	Dw.Web AV-DESK6.0
1	E-Mail	E-Mail
186	Enaza	Enaza
144	HD2	HD
179	INET	Inet
208	Internet	Inet
33	IPN	Выделенные каналы
213	Mobi.Деньги	Mobi.Деньги

Код	Название	Используется	Ед. измерения
81	MPS	<input checked="" type="checkbox"/>	
308	MPS(копия)	<input checked="" type="checkbox"/>	

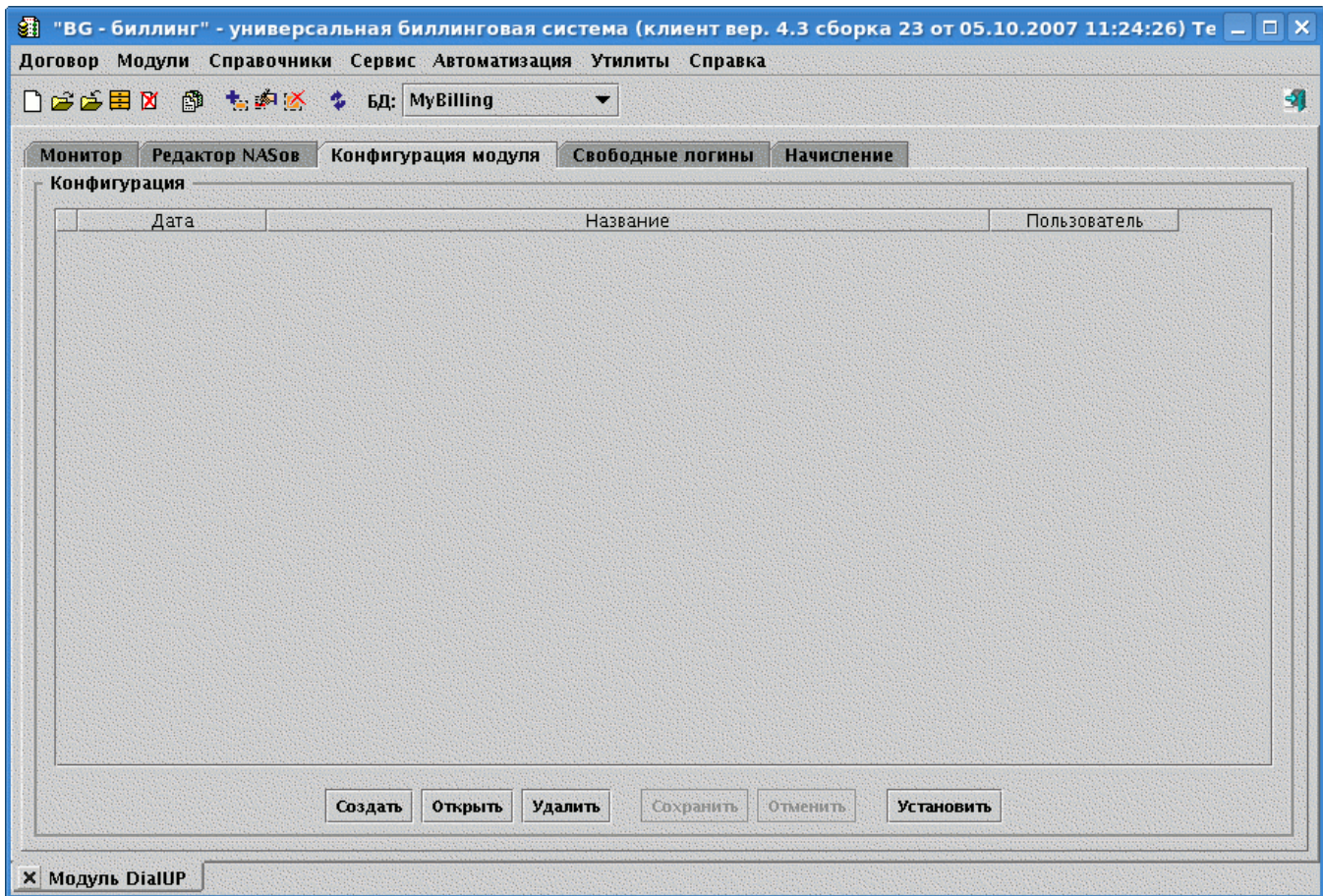
Редактор модулей и услуг

После создания экземпляра модуля биллинга созданные экземпляры должны появиться в меню **Модули**.



При открытии соответствующего модулю пункта подменю открывается вкладка с настройками модуля. Для каждого модуля набор вкладок разный. Однако в большинстве модулей присутствует вкладка **Конфигурация модуля**, её редактирование аналогично **Конфигурации сервера биллинга**.

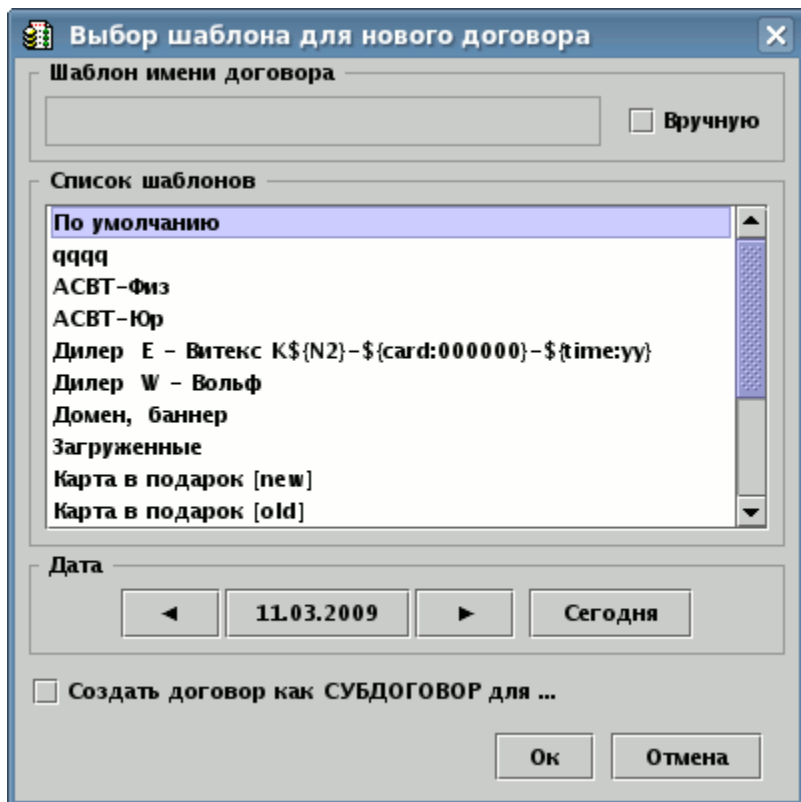
Здесь задаются настройки, специфичные для данного модуля и его агентов (например, RADIUS-сервера модуля DialUp). Конфигурация сохраняется в БД и поэтому доступна всем серверным компонентам биллинговой системы.



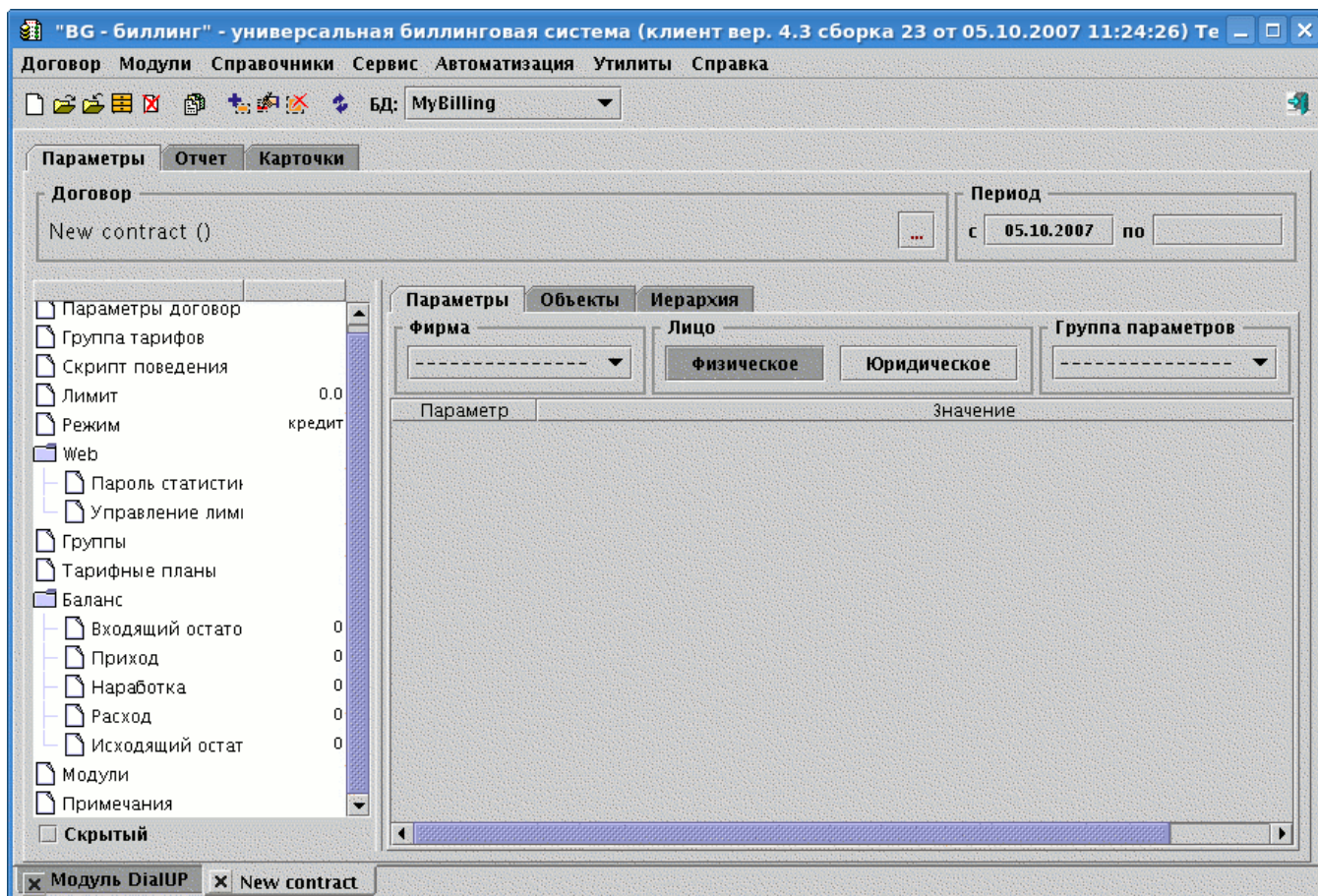
Существуют три точки "выхода" модуля в графическом интерфейсе клиента:

1. Пункт меню **Модули**, описан выше;
2. Договор абонента, узел дерева **Модули**;
3. Вкладка **Отчёты**.

Для примера создадим договор (New Contract) с шаблоном по умолчанию (более подробно о договорах, их свойствах и шаблонах см. далее). Для этого нажмём кнопку **Новый договор..** на панели инструментов (самая левая кнопка), далее выберем шаблон **По умолчанию** и текущую дату.

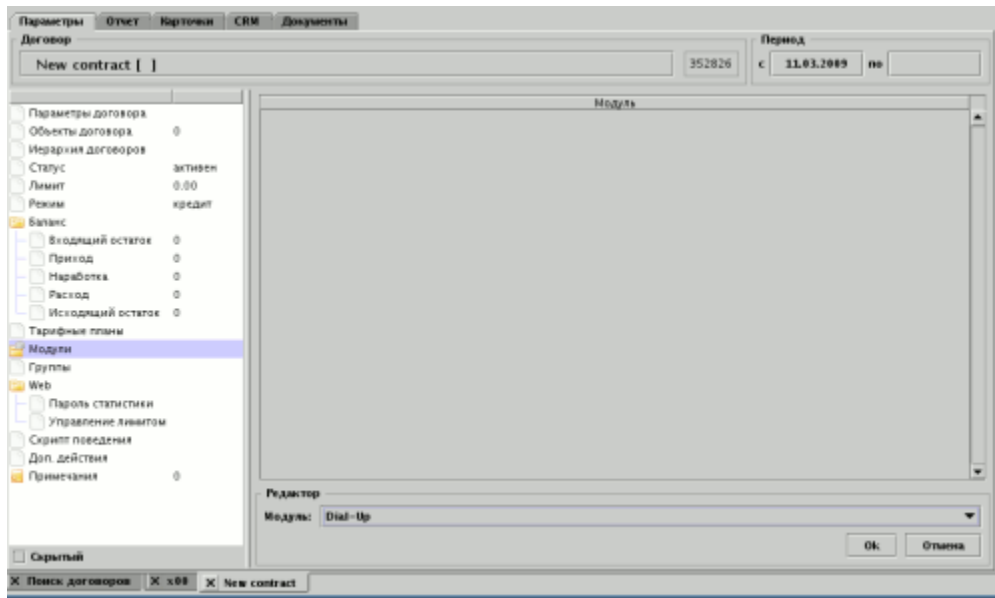


Должен открыться созданный договор **New contract**.

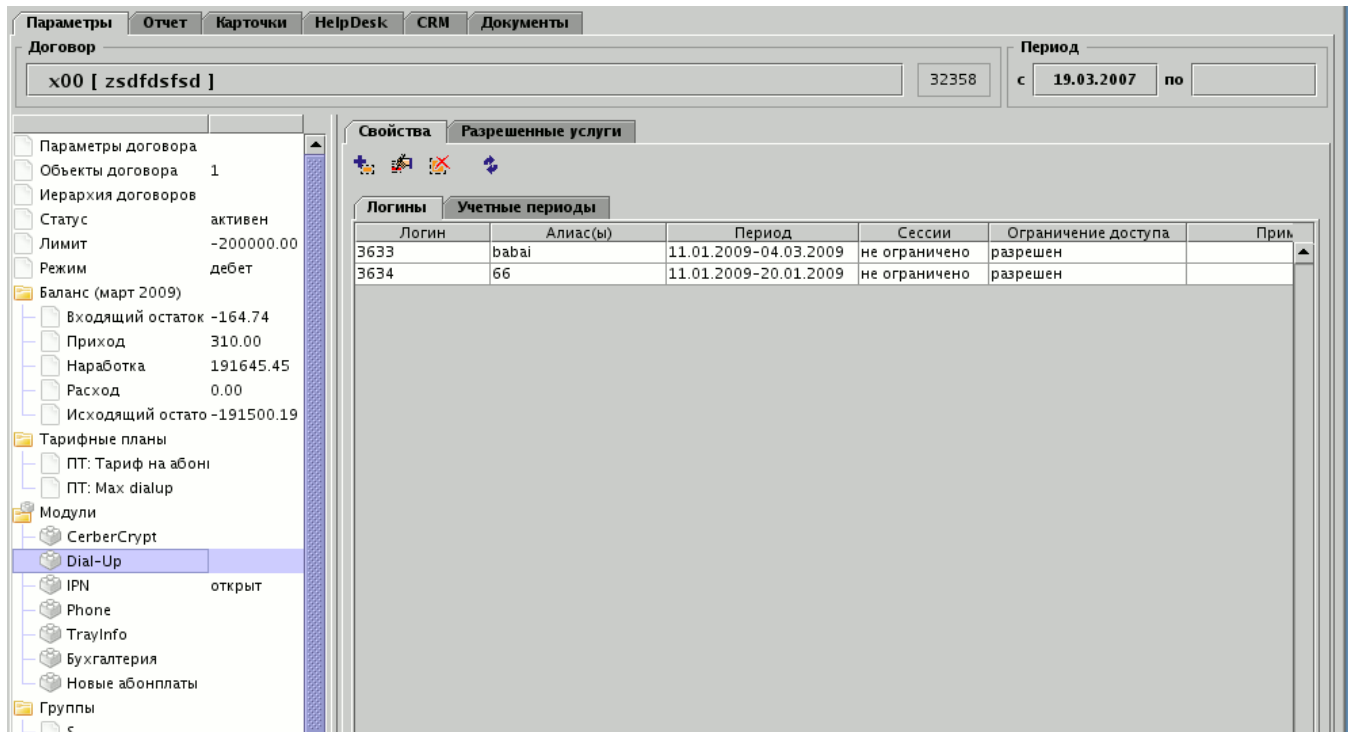


В дальнейшем, если вы его случайно закроете, вы можете найти его, нажав вторую кнопку на панели инструментов **Открыть договор..** и нажав в ней кнопку >>>. Более подробно о поиске договоров написано далее.

Для подключения к договору модуля необходимо перейти на пункт дерева **Модули**, далее нажать **Новый элемент**, выбрать в открывшемся редакторе требуемый экземпляр модуля, нажать **ОК**.

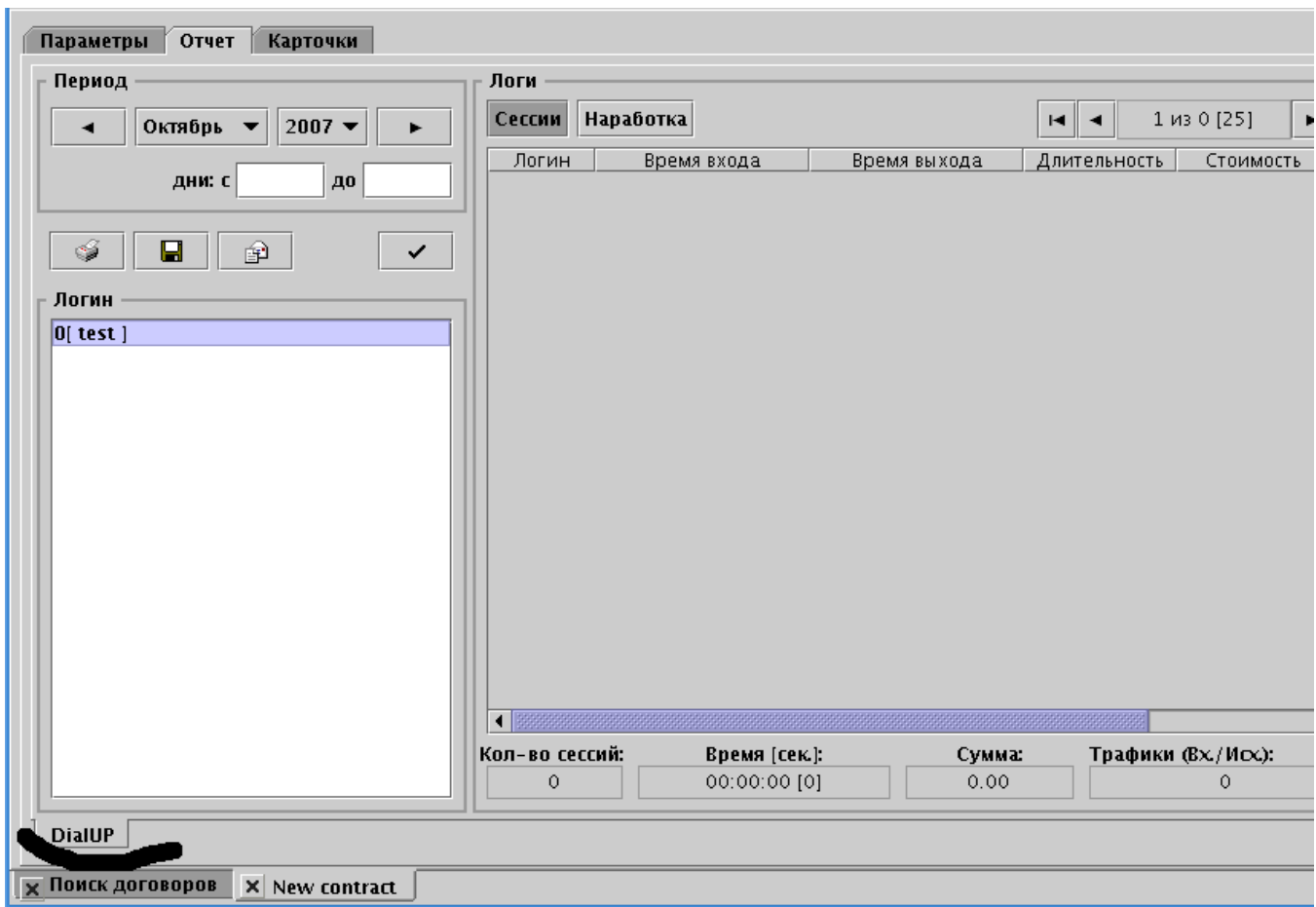


В результате в дереве появится ветка со свойствами модуля для конкретного договора. При её открытии откроется специфичный для модуля редактор. На снимке экрана это редактор логинов DialUp-модуля.



Вкладка **Разрешённые услуги** присутствует не для всех модулей биллинга, её назначение различно и описывается в документации конкретного модуля.

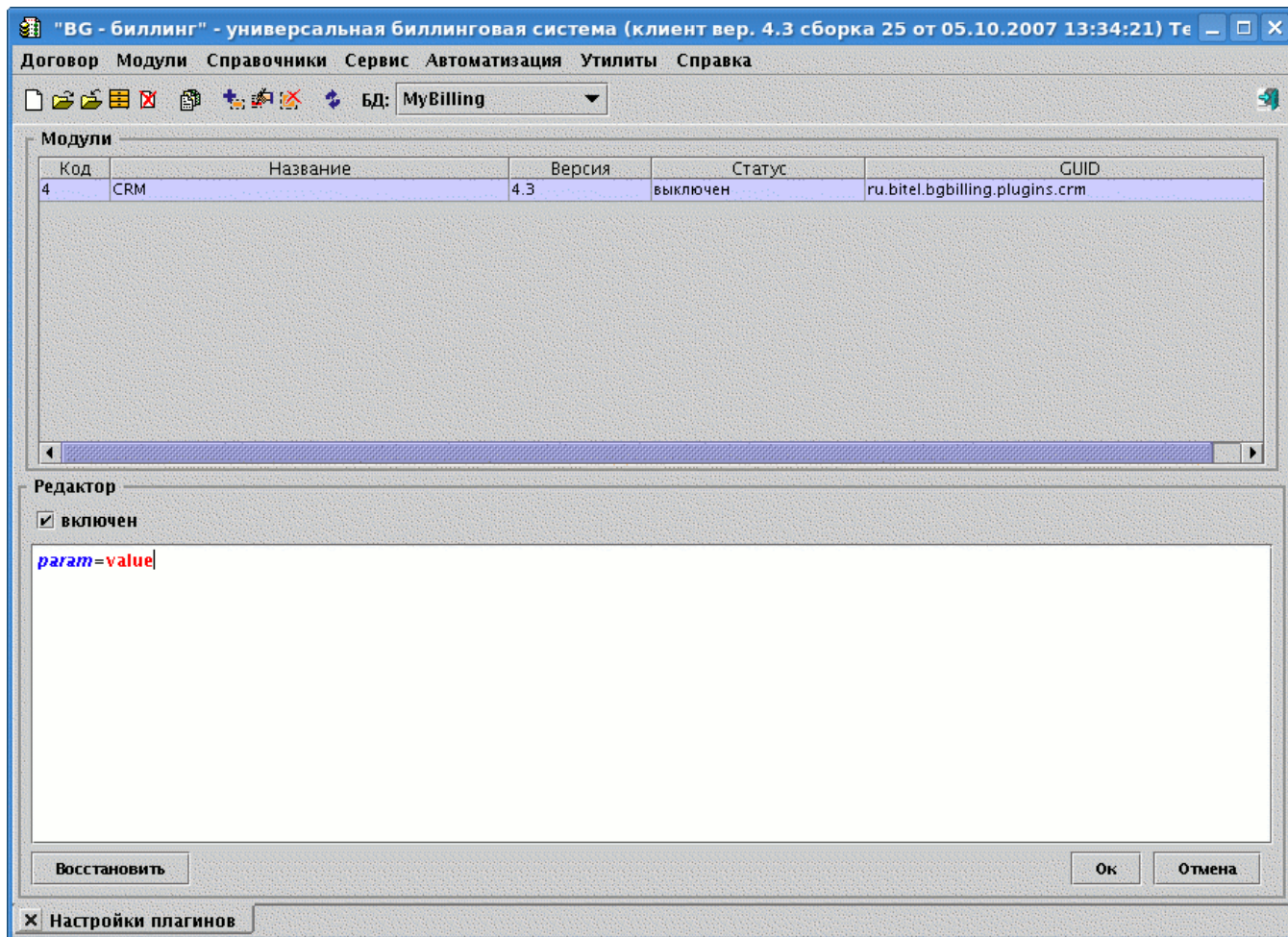
Также на вкладке **Отчёт** договора отобразится панель оперативных отчётов данного модуля.



Аналогично в договор можно добавить другие модули.

Установка и настройка плагинов

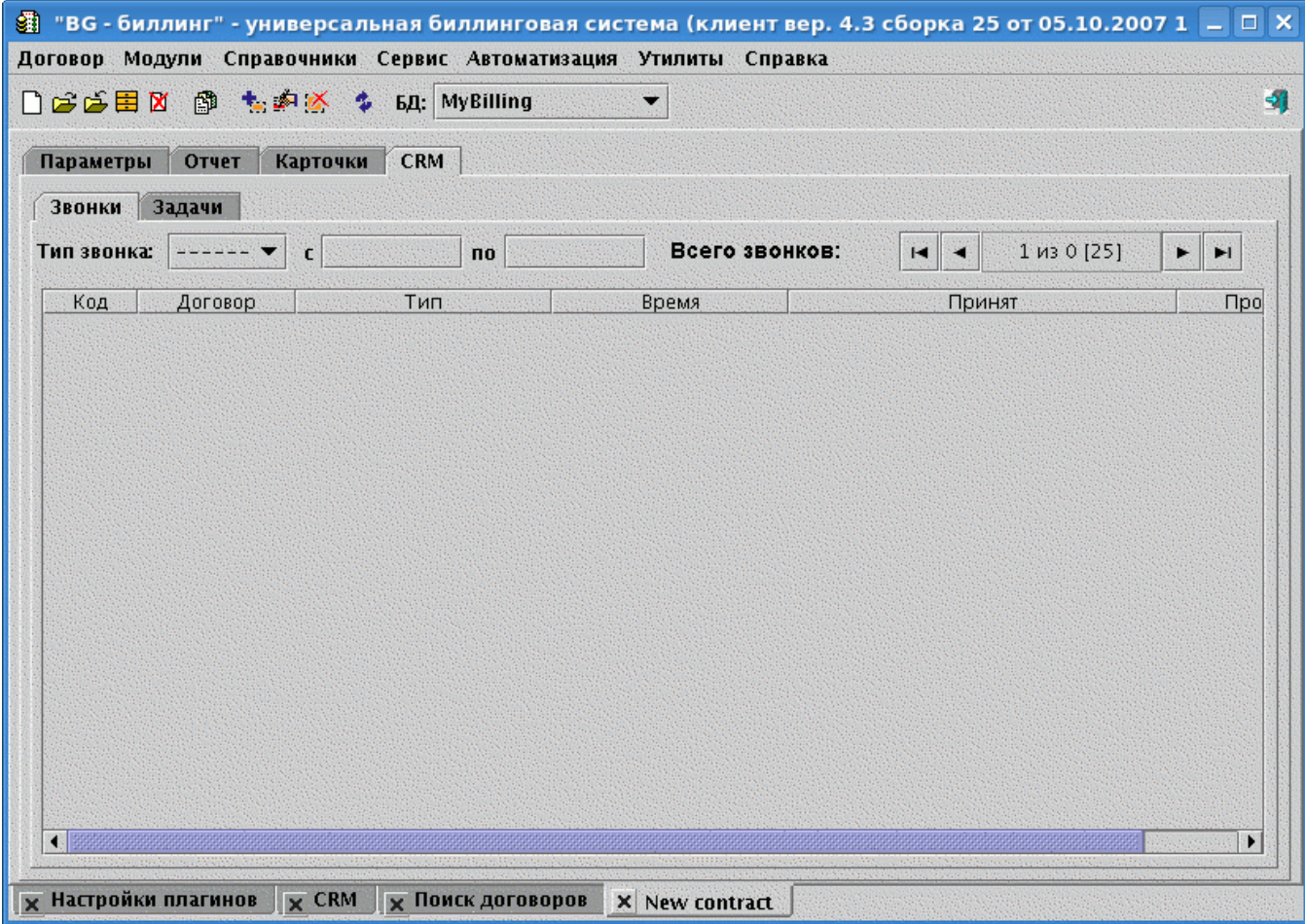
Установка плагинов не отличается от установки модулей, также используется утилита **bg_installer** (см. предыдущую секцию документации). После инсталляции плагина перезапустите сервер биллинга и зайдите в клиенте в меню **Плагины=>Настройка плагинов**.



Для активации плагина необходимо дважды кликнуть по строке таблицы, поставить в редакторе галочку **включён** и нажать **ОК**. Здесь же вносится конфигурация плагина, конкретные ключи и значения необходимо смотреть в документации конкретного плагина. После активации плагина сервер начинает проверять лицензию на него, в случае отсутствия, либо просроченной лицензии плагин необходимо отключить.

После активации плагина необходимо переподключиться клиентом к серверу биллинга. В зависимости от конкретного плагина он может "проявляться" в различных точках клиента. В этом отличие плагинов от модулей, которые появляются только в определённых 3х точках "входа" в интерфейс.

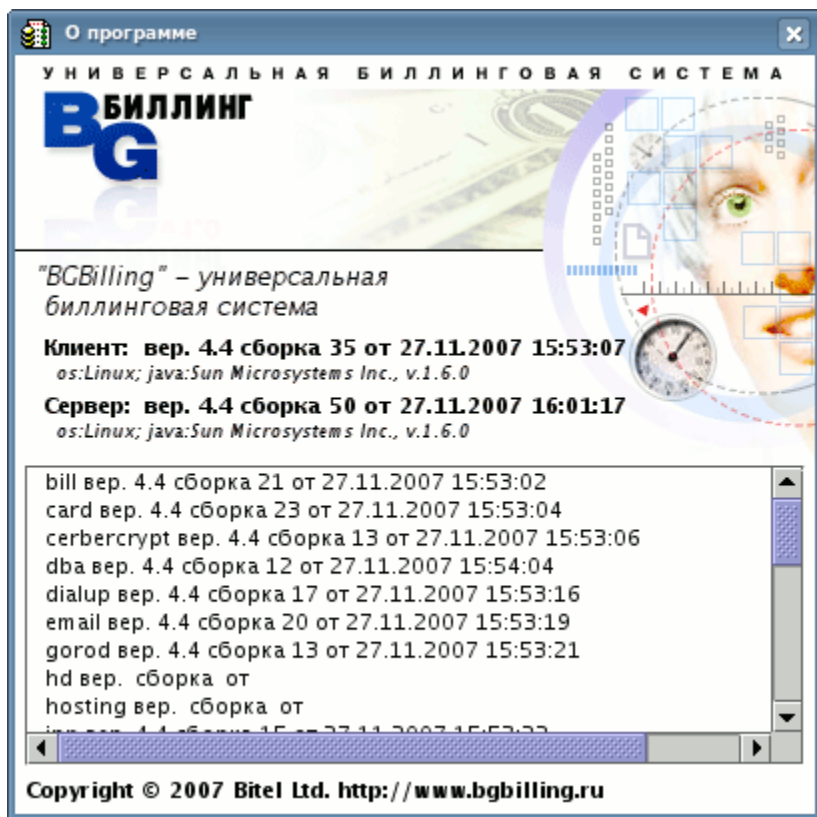
Например, плагин CRM добавляет пункт меню **Плагины=>CRM** и вкладку в договоре **CRM**.



Установка обновлений биллинга

По мере обнаружения ошибок в текущей версии программы выпускаются обновления в виде новых билдов (сборок). Каждый новый билд модуля сопровождается комментарием к сделанным исправлениям. Комментарии доступны на странице загрузки продукта.

Периодически следует сверять текущие установленные билды модулей и плагинов с доступными на сайте. Для получения информации по версиям и билдам компонентов биллинга воспользуйтесь меню **Справка=>О программе**.



В верхней области отображаются версии и билды клиента и сервера. Обновление клиента и сервера доступны на сайте единым пакетом. Ниже перечислены установленные в системе модули и плагины, их версии и билды.

С версии 8.2104+ используется новый механизм установки обновлений

На сервере биллинга запустить скрипт **update.sh** (при этом на время обновления произойдет остановка сервера биллинга и планировщика, учитывайте это при выборе момента запуска обновления)

```
< /home/bill/BGBillingServer/bin
.и      Имя
/..
/win
*application.sh
*bg_installer.sh
*on_crush.sh
*setenv.sh
*snapshot.sh
*start.sh
*status.sh
*stop.sh
*update.sh
*update_runner.sh
```

После запуска выберите из меню пункт 2

```
Java version is 15.0 (15.0.1 2020-10-20)
Проверка доступности файлов для версии: beta; набор: 2201200226
Version: beta
Set: 2201200226
```

```
Select:
 1) Установить новые компоненты          3) Показать установленные компоненты
 2) Обновить установленные компоненты    4) Переустановить текущие компоненты
 0) Exit
=> 
```

выберите сборку на которую хотите обновиться, 1 - самая последняя

```
Select:
 1) Установить новые компоненты          3) По
 2) Обновить установленные компоненты    4) Пе
 0) Exit
=> 2
```

```
Select:
 1) 2201241348  2) 2201240323  3) 2201210258
 0) Exit
=> 
```

После загрузки обновлений выберите из меню 0 для выхода и запуска сервера биллинга и планировщика

```
Select:
 1) Установить новые компоненты          3) Показать установленные компоненты
 2) Обновить установленные компоненты    4) Переустановить текущие компоненты
 0) Exit
=> 
```

Для версий младше 8.0

Для установки модулей и плагинов используется утилита **bg_installer.sh (.bat)**.

```
./bg_installer.sh <___>
```

Для Linux.

```
bg_installer.bat <___>
```

Для Windows.

Файл с модулем должен быть загружен и располагаться в каталоге BGBillingServer.

При установке обновлений следует останавливать процесс сервера биллинга, планировщика и загрузчика логов и запускать их после установки пакетов.

Для установки обновления пакет модуля, либо плагина необходимо установить повторно. Поддерживается автоматическое обновление с FTP-сервера ftp://bgbilling.ru. Для обновления системы остановите сервер биллинга, планировщик и загрузчик логов а затем выполните команду:

```
./bg_installer.sh update
```

Для Linux платформы.

```
bg_installer.bat update
```

Для Win платформы.

Инсталлятор проверит наличие обновлений и предложит их установить выбрав нажав клавишу **y**.

```
[bill@bill-reg BGBillingServer]$ ./bg_installer.sh update
Update starting..
Update from ftp://bgbilling.ru/pub/bgbilling
Server version is 4.4
Changing dir to /pub/bgbilling/4.4
Checking updates for ru.bitel.bgbilling.plugins.crm..
Found update for crm build 50 packet crm_4.4_52.zip updating to build 52
Checking updates for ru.bitel.bgbilling.plugins.documents..
Checking updates for card..
Checking updates for dialup..
Checking updates for email..
Checking updates for ipn..
Checking updates for npay..
Checking updates for reports..
Checking updates for voiceip..
Checking updates for bill..
Checking updates for gorod..
Checking updates for server..
Found update for BGBillingServer build 102 packet update_4.4.zip updating to build 104
Checking updates for client..
Install 2 updates (y/n):
```

Возможно установить только все обновления сразу. Также необходимо останавливать сервер, планировщик и загрузчик логов до обновления и стартовать после.

Не забывайте делать резервную копию каталога BGBillingServer при установке обновлений. Обновление перетирает все XSLT-шаблоны и многие другие файлы, заменяя их стандартными из пакета.

Для Linux также есть скрипт **update.sh**, который самостоятельно останавливает службы, запускает процесс обновления и стартует службы после. Кроме того, он сохраняет лог процесса обновления в файл.

Для предотвращения перетирания файла при обновлении вы можете **перед** его модификацией создать копию с именем **<file_name>.orig** (например, style.css.orig). При установке пакета инсталлятор будет проверять перед записью каждого файла наличие файла с таким же именем в текущей установке. Если файл существует, но отличается от того, что в пакете, то предпринимается попытка найти файл **<file_name>.orig**. Если orig-файл существует и **не** отличается от файла из нового пакета, то ничего не будет перезаписано, система сообщит: **File doesn't changed <filePath>**. Если же и оригинальный файл не совпадает со вновь предлагаемым из новой версии пакета, файл будет записан.

Перечень перезаписанных файлов сообщается после завершения процедуры установки, либо обновления после фразы **REPLACED FILES:**. Вы должны вновь внести в данные файлы требуемые корректировки и снова создать **.orig**-копию файла (из нового файла из дистрибутива).

Обновление клиента происходит автоматически при последующем подключении к серверу биллинга и установленной опции **Загружать обновления с данного сервера**.

При выходе новых версий биллинга предоставляется инструкция по переходу на новую версию ПО, далее в пределах версии обновление производится по описанному выше алгоритму.

В системе обновления биллинга работает механизм предотвращения повторного выполнения SQL-запросов обновления базы данных, что значительно ускоряет обновление системы. Выполненные SQL-запросы кэшируются и хранятся в базе данных биллинга. В случае, если по какой-либо причине необходимо повторное выполнение SQL-запросов при обновлении модуля (или ядра), то необходимо запустить

```
./bg_installer.sh killhash <entity_id>
```

для Linux или

```
bg_installer.bat killhash <entity_id>
```

для Windows,

где **<entity_id>** - код сущности, для которой необходимо уничтожить кэш SQL-запросов (для ядра = 0; для плагина = код плагина из **Плагины => Настройки плагинов**; для экземпляра модуля = **m<mid>**), где **<mid>** - код экземпляра модуля из **Модули => Редактор модулей и услуг**).

Например:

```
./bg_installer.sh killhash m10
```

Обновление других серверных приложений

Все сторонние приложения получают обновления от сервера биллинга (серверные приложения, CashCheck-сервер итд). Единый набор серверных библиотек биллинга на всех приложениях обеспечивает унифицированную среду для работы скриптов и расширений. Для обновления приложения используется скрипт **update.sh (.bat)**. В момент обновления BGBillingServer должен быть запущен. Также нужно убедиться что в application.xml (или соответствующем) прописан путь до корня сервлетов сервера биллинга ("server.update.url"), см. [Установка приложений биллинга \(BGInetAccess, BGVoiceProcwssor и др.\) \(с версии 8.2104\)](#).

Следите, чтобы все ваши серверные приложения были обновлены!

Данная схема распространяется только на серверные приложения, связанные с ядром через JMS. Изолированные приложения обновляются отдельно. У них нет скрипта update.*. К таким приложениям относятся, например, DHCP-сервер модуля IPN.

Снапшоты

В данный момент есть версия скрипта только для ОС Linux! Для Windows выполняйте требования по резервному копированию перед обновлением!

Для сохранения текущего состояния библиотек биллинга, каталога webroot, данных по установленным модулям и плагинам в БД с BGBilling поставляется скрипт **snapshot.sh**. Для создания снапшота вызовите перед обновлением:

```
./snapshot.sh create
```

Снапшоты сохраняются архивами в каталог **BGBillingServer/snapshots**. Для восстановления снапшота команда:

```
./shapshot.sh restore <FILE>
```

, где **<FILE>** - имя файла со снапшотом.

Восстановив сервер из снапшота необходимо обновить с него все другие серверные приложения. Функционал скрипта реализован исключительно сторонними приложениями и может быть легко скорректирован.

Установка лицензии биллинга

Для корректной работы BGBilling необходимо наличие лицензии. Лицензия представляет собой файл **lic.properties** с набором текстовых строк, который выдаётся при покупке BGBilling. Лицензия включает в себя описание разрешенных к использованию модулей и плагинов, количества договоров и период действия.

Полученную лицензию необходимо поместить в каталог **BGBillingServer/data** заменив существующий **lic.properties**, после чего перезапустить сервер биллинга. Загруженный с сайта биллинг содержит тестовую лицензию. Текущую лицензию можно просмотреть в [индикаторе лицензии](#).

С версии 7.2+ новую лицензию можно установить в индикаторе лицензий, без перезагрузки биллинга.

До 4.5 версии включительно информация о лицензии размещалась в файле **data.properties**. Для сохранения обратной совместимости, данный файл также просматривается, если не был обнаружен **lic.properties**.

Для того, чтобы получать оповещения о том, что лицензии скоро заканчиваются, в конфигурации сервера можно настроить такие параметры:

```
# 1- , 0 - ( )
need.lic.alarm=1
# , , X
lic.alarm.contract.limit=X
# , , Y
lic.alarm.days.limit=Y
# . , .
lic.alarm.modules=ipn,dilaup
```

Настройка SSL между сервером и клиентом

По умолчанию обмен между клиентом и сервером биллинга, а также между Web-браузером клиента и сервером биллинга, производится по протоколу HTTP. В дополнение к нему вы можете настроить HTTPS-режим соединений, позволяющий безопасно подключаться к серверу биллинга из-за пределов вашей сети клиентом, либо получать безопасный доступ к статистике вашим абонентам.

Вы можете пропустить эту главу при первичном знакомстве с биллингом

Для работы по HTTPS требуется серверный сертификат - пара private key и public key, public key обернут сертификатом. Сертификат и private key должны находиться в хранилище сертификатов - .keystore.

Работа сертификатов основывается на паре асинхронных ключей. Необходимо создать их, для этого воспользуемся утилитой keytool, которая идет вместе с JRE/JDK:

```
keytool -keystore .keystore -alias bgbilling -genkey -keyalg RSA -dname "cn=bill.provider.ru, email=email@provider.ru,ou=Provider Billing, o=Provider, c=RU" -validity 1001
```

Параметр -validity указывает сколько дней будет действителен сертификат.

Утилита попросит ввести пароль для хранилища и пароль для закрытого ключа (private key).

Оба пароля должны совпадать и быть прописаны в конфиге сервера (по умолчанию биллинг пытается использовать значение 'bgbilling'):

```
keystore.password=bgbilling
```

cn - common name - имя сервера. Пользователь заходит на <https://bill.provider.ru/>, сервер возвращает сертификат, браузер проверяет имя на совпадение - если не совпадает, то выводится предупреждение.

ou - organization unit, o - organization, c - country

После работы утилиты появится файл .keystore, в нем в алиасе bgbilling - закрытый ключ и сертификат. Закрытый ключ нельзя передавать никому.

Просмотреть хранилище можно командой:

```
keytool -keystore .keystore -list -rfc
```

Экспортировать сертификат, чтобы удаленная сторона могла добавить его в доверенные:

```
keytool -keystore .keystore -alias bgbilling -exportcert -file bgbilling.cer
```

В данный момент хранилище уже готово для работы, но текущий сертификат подписан самим собой, т.е. не подписан доверенным сертификатом - таким образом, когда клиент зайдет на <https://bill.provider.ru/> - ему выведется предупреждение, например, "К сертификату нет доверия, так как он является самоподписанным." и предложат уйти с сайта, продолжить работать или добавить сертификат в доверенные и продолжить работать.

Если вы хотите, чтобы браузер сразу доверял сертификату, необходимо его подписать одним из Certification Authority (CA), например VeriSign, Inc. Для этого создаем запрос на подпись сертификата - Certificate Signing Request (CSR).

```
keytool -keystore .keystore -alias bgbilling -certreq -file bgbilling.csr
```

Созданный запрос нужно отправить CA, в ответ вы получите сертификат, подписанный им (или цепь сертификатов).

Сначала необходимо импортировать сертификат CA в доверенные (или верхний из цепи сертификатов, например ROOT), т.к. при импорте ответа проверяется, что ответ подписан доверенным сертификатом.

```
keytool -keystore .keystore -alias verisign -importcert -file verisign.cer
```

Далее импортируем ответ, он заменит наш самоподписанный сертификат:

```
keytool -keystore .keystore -alias bgbilling -importcert -trustcacerts -file response.cer
```

Более подробную информацию о работе keytool можно получить здесь: <http://java.sun.com/javase/6/docs/technotes/tools/windows/keytool.html>

Скопируйте файл .keystore в каталог **BGBillingServer**. В файле **data.properties** сервера укажите:

```
connector.https=:8443
```

и перезапустите его. Если нужно поднять биллинг на конкретном ip (интерфейсе) , то вместо "*" можно указать его, например "81.30.30.30:8443".

В результате сервер начинает слушать на 2х портах. Не стоит убирать **http.port**, т.к. сервер должен обращаться к самому себе за некоторыми ресурсами (XSL-файлы) по этому протоколу.

При этом клиенты для просмотра Web-статистики также могут использовать 2 вида протоколов HTTP и HTTPS. Для обращения через безопасное соединение необходимо набирать **https://<IP адрес сервера биллинга>:8443/bgbilling/webexecuter**.

Редактируем файл **client.properties** в клиенте

```
db.server.0.url=http://127.0.0.1:8080/bgbilling/executer
```

меняем на

```
db.server.0.url=https://bill.provider.ru:8443/bgbilling/executer
```

Руководство по интеграция существующего сертификата и приватного ключа SSL в хранилище keystore вы можете найти в [Wiki](#).

Обновление keystore до pkcs12-версии:

```
keytool -importkeystore -srckeystore .keystore -destkeystore keystore.p12 -deststoretype pkcs12
```

Настройка планировщика

Специально для выполнения периодических процессов (снятие абонплат, очистка таблиц, восстановление лимитов на договоре и т. д.) в системе **BGBilling** предусмотрен планировщик задач, выполняющий задачи в заданные интервалы времени.

Для просмотра/добавления/редактирования/удаления задач зайдите в **Сервис=>Администрирование=>Планировщик заданий**. Перед вами откроется окно управления планировщиком с фильтруемым списком задач. Фильтрация осуществляется только по тем модулям/плагинам, которые имеют задачи.

BGBillingClient

Договор Модули Плагины Справочники Сервис Утилиты Справка

БД: local | Логин: artur | 25 Ноябрь 11:49

Планировщик | Состояние планировщика

Фильтр
Модуль: Все

...	Название задачи	Коммен...	Период	Модуль	Статус	Приоритет	Месяцы	Дни месяца	Дни недели	Часы	...
29	HelpDesk => рассылка уведомлений...		...-...	HelpDesk	выключена	нормальн...	*	*	*	*	...
30	Загрузка платежей из файлов		08.09.2008-...	Ядро	выключена	нормальн...	*	*	*	*	...
31	Установка статусов договоров		09.01.2009-...	Ядро	выключена	нормальн...	*	*	*	*	...
32	Пользовательские рассылки		12.03.2009-...	Ядро	выключена	максимал...	*	*	*	*	...
33	Генератор событий таймера	test for ...	18.05.2009-...	Ядро	выключена	нормальн...	*	*	*	*	...
34	Валидатор		16.06.2009-...	Ядро	выключена	нормальн...	*	*	*	*	...
35	HelpDesk => рассылка персональны...		31.08.2010-30...	HelpDesk	выключена	нормальн...	*	*	*	*	...
36	Обработка задач Dr.Web (режим 1)		...-...	DrWeb	выключена	нормальн...	*	*	*	*	...
37			...-...	Assist.Ru	выключена	нормальн...	*	*	*	*	...
38	Выполнение глобальных скриптов п...		03.11.2009-...	Ядро	выключена	нормальн...	*	*	*	*	...
39	Синхронизация E-Mail аккаунтов		19.10.2009-...	E-Mail	выключена	нормальн...	*	*	*	*	...
40	Выполнение глобальных скриптов п...		03.11.2009-...	Ядро	выключена	нормальн...	*	*	*	*	...
41	HelpDesk => авто закрытие тем че...	topic.av...	24.09.2009-...	HelpDesk	выключена	нормальн...	*	*	*	*	...
43	Закрытие статуса NPay договоров ...		25.01.2010-...	Новые абон...	выключена	нормальн...	*	*	*	*	...
51	Начисление NPay абонплат		12.02.2010-...	Новые абон...	выключена	нормальн...	*	*	*	*	...
53	Начисление Dr.Web		...-...	DrWeb	выключена	минималь...	*	*	*	*	...
54	Начисление RSCM		29.07.2010-...	RSCM	выключена	нормальн...	*	*	*	*	...
58	Модуль HD => Обновление зон моду...		14.09.2010-14...	HD2	выключена	максимал...	*	*	*	*	...
61	Модуль HD => Удаление зон с серве...		14.09.2010-14...	HD2	выключена	максимал...	*	*	*	*	...
62	Модуль HD => Обновление DNS slav...		21.09.2010-21...	HD2	выключена	максимал...	*	*	*	*	...
63	Проверка наличия новых пользоват...		...-...	Ядро	выключена	нормальн...	*	*	*	*	...
64	Задача отправки рассылки		19.10.2010-...	Dispatch	включена	минималь...	*	*	*	*	...
68	BVCom => Проверка результатов з...		02.08.2011-02...	BVCom	выключена	нормальн...	*	*	*	*	...
69	Модуль HD => Обновление Whois да...		03.02.2011-...	HD2	выключена	минималь...	*	*	*	*	...
70	Модуль Softkey. Обновление доступ...		07.07.2011-07...	Softkey	выключена	минималь...	*	*	*	*	...
72	Установка статусов договоров	Устано...	30.12.2011-30...	Ядро	выключена	минималь...	*	*	*	*	...
74		Переве...	30.01.2012-...	E-Mail	выключена	минималь...	*	*	*	*	...

Статус планировщика
В очереди 10 "тяжелых" задач

✕ Планировщик заданий

Для создания/редактирования/удаления задач воспользуйтесь стандартной панелью инструментов клиента биллинга. При создании /редактировании задачи снизу появится редактор, представленный на рисунке ниже.

BGBillingClient

Договор Модули Плагины Справочники Сервис Утилиты Справка

БД: local Логин: artur 25 Ноябрь 11:43

Планировщик Состояние планировщика

Фильтр
Модуль: Все

№	Название задачи	Коммен...	Период	Модуль	Статус	Приоритет	Месяцы	Дни месяца	Дни недели	Часы
41	HelpDesk => авто закрытие тем че...	topic.av...	24.09.2009-...	HelpDesk	выключена	нормальн...	*	*	*	*
43	Закрытие статуса NPay договоров ...		25.01.2010-...	Новые абон...	выключена	нормальн...	*	*	*	*
51	Начисление NPay абонплат		12.02.2010-...	Новые абон...	выключена	нормальн...	*	*	*	*
53	Начисление Dr.Web		...-...	DrWeb	выключена	минималь...	*	*	*	*
54	Начисление RSCM		29.07.2010-...	RSCM	выключена	нормальн...	*	*	*	*
58	Модуль HD => Обновление зон моду...		14.09.2010-14...	HD2	выключена	максимал...	*	*	*	*
61	Модуль HD => Удаление зон с сервере...		14.09.2010-14...	HD2	выключена	максимал...	*	*	*	*
62	Модуль HD => Обновление DNS slav...		21.09.2010-21...	HD2	выключена	максимал...	*	*	*	*
63	Проверка наличия новых пользоват...		...-...	Ядро	выключена	нормальн...	*	*	*	*
64	Задача отправки рассылок		19.10.2010-...	Dispatch	включена	минималь...	*	*	*	*

Период с 25.11.2013 по

Задача
Модуль: BVCom
Название: BVCom => Проверка результатов запроса на возврат денежных средств
Приоритет: нормальный
Состояние: выключена

Время запуска Параметры запуска Комментарий
Минуты: *
Часы: *
Дни месяца: 1
Дни недели: *
Месяцы: *

Восстановить Ок Отмена

Статус планировщика
В очереди 10 "тяжелых" задач

Планировщик заданий

Максимальная частота запуска заданий планировщиком - один раз в минуту. Данная частота устанавливается, если во всех полях **Время запуска** будет поставлена *. Т.е. следует читать так: "любая минута,любой час, любой день месяца, любой день недели, любой месяц".

Если столь высокая частота не требуется, можно уточнить параметры. Например, запуск задачи один раз в месяц 1-го числа в 00:10 можно задать установкой **Дни месяца** в 1, **Часы** в 0, **Минуты** в 10. При необходимости нескольких значений временных фильтров можно указывать через запятую. Для кратных количеств можно указывать строки вида */n, где n - кратность. Например, */8 для часов установит часы в строку вида 0,8,16. Также можно указывать диапазоны чисел через '-'. Например: 8-16.

Параметр **Приоритет** показывает приоритет потока, который будет выполнять задачу. Без особой необходимости его не следует менять. Не забудьте поставить состояние **Включено**. Все задачи сгруппированы по модулям, к которым относятся. Для начала нужно выбрать модуль из выпадающего списка **Модуль**, затем задачу этого модуля из соответствующего выпадающего списка.

Каждая задача может иметь специфические **Параметры запуска**, которые должны быть введены в поле **Параметры запуска** в виде <ключ>=<значение>, каждая пара на новой строке. Параметры к каждой отдельной задаче и рекомендации по частоте её запуска даны далее в руководстве к ядру системы и её модулям.

В качестве тренировки вы можете добавить задачу восстановления временно изменённых лимитов, установив параметры задачи, как показано на рисунке.

25	Пользовательские рассылки		...-...	Ядро	выключена	нормальн...	*	*	*	13
26	Восстановление лимитов		09.02.2004	Ядро	включена	нормальн...	*	*	*	*

Период с 09.02.2004 по

Задача
Модуль: Ядро
Название: Восстановление лимитов
Приоритет: нормальный
Состояние: включена

Время запуска Параметры запуска Комментарий
Минуты: 0
Часы: 0
Дни месяца: *
Дни недели: *
Месяцы: *

Восстановить Ок Отмена

Статус планировщика
В очереди 10 "тяжелых" задач

Планировщик заданий

Для немедленного запуска задачи планировщика используется контекстное меню, доступное по щелчку правой кнопкой мыши на задаче.

24	Пользовательские рассылки		18.05.2009-
25	Пользовательские рассылки		-
20	Начисление CerberCrypt		-
21	Синхронизация CerberCrypt	Выполнить сейчас	мет состояние кар...
22	Блокировка CerberCrypt		-
23	Установка договоров картам С...		-
33	Генератор событий таймера	test for tk ural	18.05.2009-

Планировщик автоматически перезагружает список периодических заданий при их редактировании в клиенте биллинга.

Кроме плановых заданий планировщик также исполняет все "тяжёлые" задачи биллинга. Например, начисление абонплат, перерасчёты. Это позволяет более оптимально использовать ОЗУ, выделяя большой объем памяти только планировщику. Также можно разгрузить сервер биллинга, вынося планировщик на отдельную машину. Количество "тяжёлых" задач, стоящих в очереди на обработку, также отображается в окне управления планировщиком в нижней части.

41	HelpDesk => авто закрытие тем че...	topic.av...	24.09.2009-...	HelpDesk	выключена	нормальн...	*	*	*	*	*
43	Закрытие статуса NPay договоров ...		25.01.2010-...	Новые абон...	выключена	нормальн...	*	*	*	*	*
51	Начисление NPay абонплат		12.02.2010-...	Новые абон...	выключена	нормальн...	*	*	*	*	...

Статус планировщика

В очереди 10 "тяжелых" задач

Планировщик заданий

Поэтому планировщик должен работать, даже если список его периодических задач пуст. Задания на выполнение начислений и т.п. передаются сервером биллинга планировщику через базу данных. При этом сам планировщик может быть запущен на другой машине.

На вкладке **Состояние планировщика** можно посмотреть, чем в данный момент занят планировщик заданий. Отображаются как выполняемые периодические задачи, так и вся очередь тяжелых задач с отображением их текущего статуса (выполняется или ожидает выполнения).

BGBillingClient			
Договор		Модули	
Плагин		Справочники	
Сервис		Утилиты	
Справка			
БД: local		Логин: artur	
25 Ноябрь 12:38			
Планировщик			
Состояние планировщика			
Выполняемые периодические задачи			
№	Описание	Количество	...
64	Плагин Рассылки: Задача отправки сообщений рассылок.	1	▲
Очередь тяжелых задач			
№	Описание	Статус	Время запуска...
2	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
3	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
4	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
5	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
6	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
7	Модуль Cerbercrypt. Задача начисления наработки. Код модуля: 43. Коды договоров: 352715. За месяц: 2013.10	В очереди	
8	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
9	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
10	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	
11	Модуль Cerbercrypt. Задача синхронизации состояния базы с состоянием на контроллерах. Код модуля: 43. По карт...	В очереди	

Планировщик заданий

На вкладке **Статистика** отображается список выполненных планировщиком задач, а также длительность их выполнения с возможностью фильтрации по задаче и дате выполнения.

BGBillingClient (вер. 7.1.61 / 18.09.2017 11:27:33) Бител (идент. номер: 2538)

Договор Модули Плагины Справочники Сервис Утилиты Справка

БД: local | Логин: admin | 21 сентября 15:24

Планировщик | Состояние планировщика | Статистика

Фильтр

Период с 01.08.2017 по 30.08.2017 | Задача Не выбрано (id = -1)

Всего записей: 20 | 1 из 1 [100]

Код задачи	Название задачи	Время запуска	Время окончания	Время выполнения, Ч:М:С.мс
82	Задача отправки рассылки	10.08.2017 15:30:06.732	10.08.2017 15:30:06.958	00:00:00.226
0		10.08.2017 15:30:06.732	10.08.2017 15:30:07.125	00:00:00.393
40	Выполнение глобальных скрипто...	10.08.2017 15:31:01.735	10.08.2017 15:31:32.704	00:00:30.969
40	Выполнение глобальных скрипто...	10.08.2017 15:32:01.738	10.08.2017 15:32:01.750	00:00:00.012
40	Выполнение глобальных скрипто...	10.08.2017 15:33:01.741	10.08.2017 15:34:37.110	00:01:35.369
84	Генерация заданий на обработк...	10.08.2017 18:18:22.849	10.08.2017 18:18:31.911	00:00:09.062
85	Установка Баланса	10.08.2017 18:21:26.344	10.08.2017 18:21:38.363	00:00:12.019
84	Генерация заданий на обработк...	10.08.2017 19:13:32.397	10.08.2017 19:13:33.032	00:00:00.635
84	Генерация заданий на обработк...	10.08.2017 19:14:14.799	10.08.2017 19:14:16.017	00:00:01.218
84	Генерация заданий на обработк...	10.08.2017 19:15:20.731	10.08.2017 19:15:21.371	00:00:00.640
84	Генерация заданий на обработк...	10.08.2017 19:17:03.287	10.08.2017 19:17:11.176	00:00:07.889
84	Генерация заданий на обработк...	10.08.2017 19:20:12.030	10.08.2017 19:20:44.577	00:00:32.547
85	Установка Баланса	10.08.2017 19:21:05.694	10.08.2017 19:21:07.648	00:00:01.954
85	Установка Баланса	10.08.2017 19:21:20.699	10.08.2017 19:21:31.015	00:00:10.316
0		22.08.2017 13:03:20.021	22.08.2017 13:03:20.654	00:00:00.633
0		22.08.2017 13:03:20.022	22.08.2017 13:03:20.631	00:00:00.609
0		22.08.2017 13:03:20.025	22.08.2017 13:03:20.658	00:00:00.633
0		22.08.2017 13:03:20.028	22.08.2017 13:03:20.659	00:00:00.631
0		22.08.2017 13:03:20.021	22.08.2017 13:03:20.669	00:00:00.648
0		22.08.2017 13:03:20.671	22.08.2017 13:03:20.731	00:00:00.060

Редактор модулей и услуг | Планировщик заданий

Перенос планировщика на другую машину может быть произведен копированием каталога **BGBillingServer** на другую машину и перенастройкой **data.properties** с указанием URL к БД MySQL на другой машине. Если планировщик перенесен на другую машину, при каждом обновлении биллинга необходимо копирование папки **lib** на машину планировщика, а также перезапуск самого планировщика.

Параметры работы планировщика настраиваются в [конфигурации сервера](#).

Стандартные задачи планировщика

Вместе с ядром биллинга устанавливаются перечисленные ниже задачи. Некоторые из них могут быть настроены сразу, другие связаны с различными подсистемами биллинга и их следует настраивать по мере их настройки и ввода в эксплуатацию.

Восстановление лимитов - возвращение к исходному значению временно изменённых лимитов. Периодичность запуска - 1 раз в день, в 0 часов, 0 минут. Данная задача восстанавливает лимиты после их временного понижения, более подробно о временном понижении лимитов читайте [здесь](#). Задача может быть настроена сразу, параметров не содержит. Также задача восстанавливает лимиты договоров после просроченных "обещанных платежей" - временных понижений лимитов абонентами провайдера.

Удаление старых договоров - очистка базы биллинга от ненужных договоров с помещением их в архив. Периодичность запуска - любая, оптимально - несколько раз в сутки. Данная задача выполняет чистку базы от старых договоров по правилам, указанным в **Менеджере договоров**. О настройке **Менеджера договоров** и параметрах запуска задачи описано [далее](#).

Генератор заданий на загрузку логов - запускается раз в час и помещает в базу данных задания для загрузки логов с источников за предыдущий час. Данная задача необходима, если вы установили модуль Phone, хотя возможен и более удобен режим автоматической загрузки логов с вызовом скрипта **dataloader** с параметрами по мере подготовки логов. Данная задача не обязательна.

Рассылка о понижении лимита - напоминает пользователям с пониженным на время лимитом о необходимости оплаты. При этом анализируется не станет ли текущий баланс клиента ниже лимита после его возвращения на исходное значение. В параметрах задачи должны быть указаны:

```
#
email.subject=
#     E-Mail, E-Mail,
email.pid=<>
```

Данная рассылка должна проводится раз в сутки, оптимальное время запуска - середина дня, для предоставления возможности пользователям пополнения баланса. Текст, содержащийся в письме, варьируется в XSL-шаблоне **debt_mail.xml**.

Установка статусов договоров - задача должна запускаться раз в сутки в 0 часов 0 минут и изменять актуальные [статусы договоров](#) в соответствии с заданиями.

Валидатор - задача должна запускаться раз в период минимальной загрузки системы (например, 3 часа ночи 0 минут). Выполняет проверку базы биллинга на корректность данных, диагностируя ошибки случайных сбоев. Оповещает о наличии ошибок в журналах, которые не были исправлены. В параметрах задачи должно быть указано:

```
email=<EMail >
```

Задача должна отправлять отчет при любом результате проверки. В данный момент производятся проверки журналов ошибок, таблицы баланса на правильность переноса остатков и совпадения суммарных наработок/платежей/расходов с суммами в соответствующих таблицах. В перспективе задача будет вызывать подсистемы валидации модулей и плагинов.

Генератор событий таймера - выполняет генерацию определенного события с заданной периодичностью. В качестве параметров могут быть указаны:

```
flag=< >  
cid=< id , >  
gid=< id , >
```

Флаг позволяет различать текущий таймер от других, позволяя обрабатывать только нужные. При этом задача будет выполняться только для указанных договоров или для договоров из указанных групп договоров.

Справочники

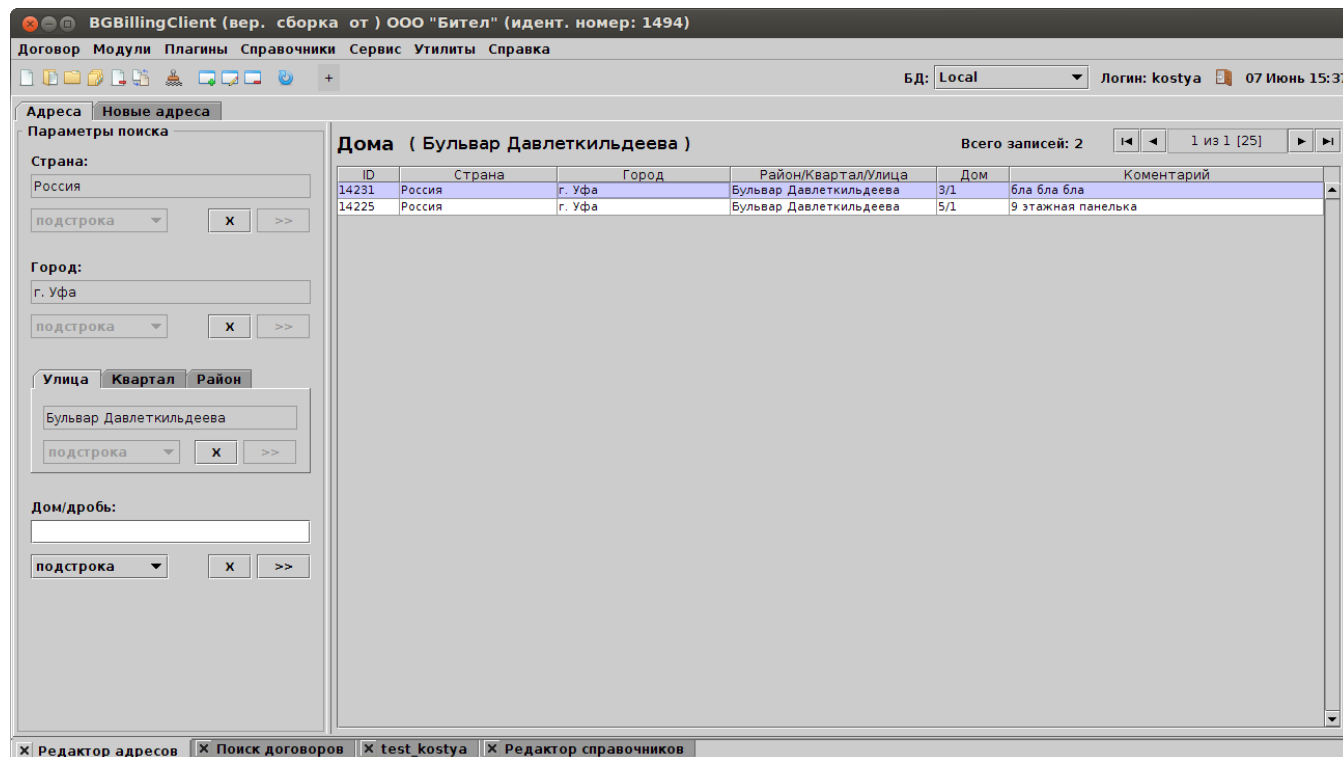
Общие сведения

Система справочников необходима для настройки биллинга. В справочниках задаются значения, которые могут принимать различные параметры договоров. Для редактирования справочников выберите пункт меню **Справочники=>Другие**. В разделе перечислены назначения справочников ядра, дополнительно в списке могут появиться справочники установленных плагинов.

Адрес - страны, города, районы, кварталы, улицы, дома

Используйте новый справочник адресов. Начиная с версии 7.1 старый справочник адресов будет убран.

Адресный справочник необходим для правки параметров договоров и объектов типа "Адрес". Основная сущность справочника - дом. Страны, города, кварталы, районы и улицы - это обычные перечисления, которые используются в справочнике домов. Города привязаны к странам. Кварталы, районы и улицы привязаны к городам. Дома привязаны к улицам. Адресный справочник доступен в меню **Справочники=>Адреса** или при нажатии комбинации клавиш **Alt+A**.



Заполнение нового (пустого) справочника

- 1) Нажимаем кнопку >> в поле поиска **Страна**
- 2) Справа открывается таблицы с списком стран (в новом справочнике будет пустая)
- 3) Нажимаем иконку с + на тулбаре, открывается редактор **Страны**
- 4) Вводим название страны, например **Россия**, нажимаем **Ok**
- 5) В таблице стран появляется запись со строкой **Россия**, двойным кликом по этой записи переходим в список **Городов** для страны **Россия**
- 6) Нажимаем иконку с + на тулбаре, открывается редактор **Города**
- 7) Вводим название города, например **Москва**, нажимаем **Ok**
- 8) В таблице городов появляется строка с названием города **Москва**, двойным кликом по этой записи переходим в список Улиц для города **Москва**
- 9) Повторяем действия для **Улиц** и **Домов**
- 10) Для ввода **Кварталов** и **Районов** необходимо в левой панели поиска нажать кнопку **X** в разделе **Город** затем нажать кнопку >> в разделе **Город**, выбрать из списка двойным кликом нужный город, далее в левой панели выбрать закладку **Район** или **Квартал** и нажать кнопку >>, далее нажимаем иконку с + на тулбаре, открывается соответствующий редактор

Поиск сущностей в адресном справочнике

Для поиска нужного адреса, необходимо сначала выбрать страну, город, улицу/квартал/район и дом. Для поиска каждой из сущностей в параметрах поиска доступно несколько режимов:

1. подстрока - искомая сущность должна содержать введенную подстроку;
2. начинается - искомая сущность должна начинаться с введенной подстроки;
3. заканчивается - искомая сущность должна заканчиваться на введенную подстроку;
4. равна - искомая сущность должна совпадать с введенной подстрокой.

После того, как необходимая сущность найдена необходимо двойным кликом выбрать ее. После выбора найденной сущности в таблице выведутся все привязанные к ней сущности, параметр поиска заблокируется, причем в строку поиска будет введено полное название искомой сущности. Как видно из скриншота, приведенного выше, после выбора улицы, данный параметр заблокировался для редактирования, а в таблице справа вывелись все дома, относящиеся к данной улице, которые заведены в справочнике.

Редактирование сущностей справочника

Для редактирования сущностей, необходимо выбрать ее в таблице и нажать кнопку редактирования на глобальной панели инструментов. Все редакторы схожи, кроме редактора домов. Редактор позволяет отредактировать название сущности и добавить к ней параметры, которые были заведены в [конфигурации сервера](#).

Редактор Параметры

Страна: Россия Город: г. Уфа

Наименование: Бульвар Давлеткильдеева

Восстановить Ок Отмена

Редактор Параметры

Индекс Добавить Удалить

Индекс	Параметр	Значение
450098	Тестовый параметр	test1234455

450098 Сохранить Отмена

Восстановить Ок Отмена

Редактор домов обладает большей функциональностью. В нем можно привязать дом к району, кварталу. Также можно указать индекс дома и кол-во квартир в нем.

Редактор **Параметры**

Страна: Россия Город: г. Уфа Улица: Бульвар Давлеткильдеева

Наименование: 5/1

Район: Октябрьский район Квартал: ----- Индекс: 450098 Квартир: 108

Комментарий
9 этажная панелька

Восстановить Ок Отмена

Для улицы можно завести специальный параметр **boxIndexRange** по которому при заведение дома может автоматически вычисляться почтовый индекс. Поиск индекса идет до первого совпадения.

Заполнение параметра **boxIndexRange**.

Пример 1. Все дома на улице с одинаковым индексом

450000:*

Пример 2. Несколько диапазонов с разными индексами

450001:1-10,31-34;450002:11-30;45000:*

Пример 3. Несколько диапазонов с разными индексами, исключение для конкретных номеров домов и дома с индексами.

450023:7,3,5/a,1;450001:1-10,31-;450002:11-30

Для дома можно завести специальные параметры диапазонов, по которым будет автоматически рассчитываться подъезд и этаж указанной квартиры. Это параметры **floorRange** - для определения этажа по квартире, **entranceRange** - для определения подъезда по квартире. Они также заводятся в [конфигурации сервера](#).

Заполнение параметра **entranceRange**.

Пример 1. Заполнение параметра для простого 9 этажного панельного дома, в котором 3 подъезда, диапазон квартир 1-108, кол-во квартир 4 на этаже и соответственно в одном подъезде 36 квартир.

1-108/36/1,

где первый параметр - диапазон квартир в доме/нескольких подъездах, второй параметр - количество квартир в подъезде, третий параметр - с какого подъезда начинается данный диапазон квартир.

Пример 2. Заполнение параметра для сложного дома.

1-72/36/1;73-132/60/3;133-168/36/4;169-187/19/5

Из примера видно, что в данном доме первые 2 подъезда стандартные по 36 квартир в подъезде, в 3 подъезде 60 квартир, в 4 - опять 36 квартир, а в 5 - 19 квартир. Соответственно ";" разделяются диапазоны квартир в нескольких подъездах.

Заполнение параметра **floorRange**.

Пример 1. Заполнение параметра для простого 9 этажного панельного дома, в котором 3 подъезда, диапазон квартир 1-108, кол-во квартир 4 на этаже и соответственно в одном подъезде 36 квартир.

1-108/4/1/9,

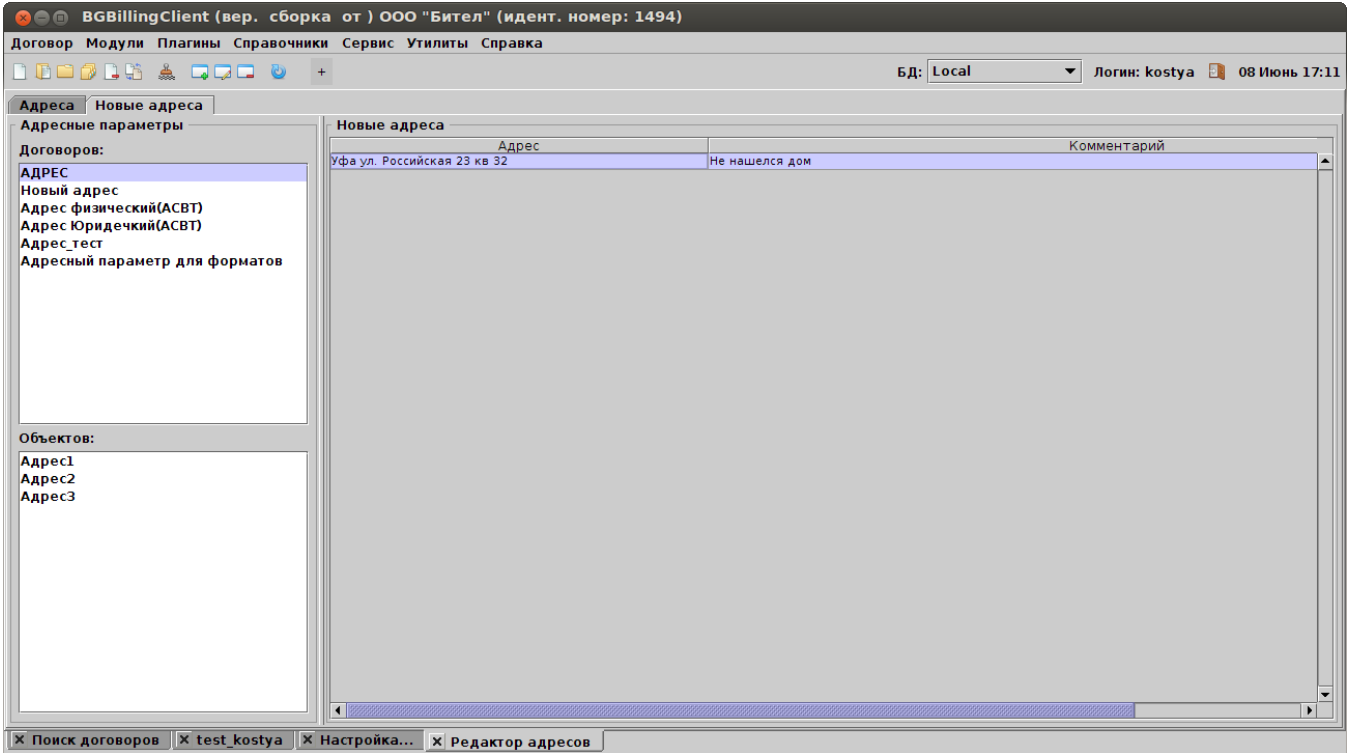
где первый параметр - диапазон квартир в доме/нескольких этажах, второй параметр - количество квартир на этаже, третий параметр - с какого этажа начинается данный диапазон квартир, четвертый параметр - на каком этаже заканчивается данный диапазон квартир.

Пример 2. Заполнение параметра для сложного дома.

1-72/4/1/9;73-132/4/1/15;133-168/4/1/9;169-183/3/1/5;184-187/2/6/7.

Из примера видно, что в данном доме первый диапазон стандартный по 4 квартиры на этаже с 1 по 9й этаж (соответствует первым 2 подъездам в доме), второй диапазон 15 этажей и по 4 квартиры на этаже (соответствует 3 подъезду в доме), третий диапазон 9 этажей по 4 квартиры на этаже (соответствует 4 подъезду в доме), четвертый диапазон с 1 по 5 этажи по 3 квартиры на этаже, пятый диапазон с 6 по 7й этаж по 2 квартиры на этаже. Четвертый и пятый диапазоны соответствуют 5 подъезду в доме. Соответственно, ";" разделяется диапазон квартир на нескольких этажах.

На вкладке Новые адреса выводятся адреса, заведенные через интерфейс "Пользовательское значение адреса" в редакторе адресного параметра договора.



В левой части интерфейса отображаются адресные параметры договора и параметры объектов. В правой части новые адреса в таблице. При двойном клике открывается редактор адресного параметра договора, в нем можно повторно попробовать найти данный адрес. Если он не находится, то необходимо данный адрес завести в справочник адресов. Для этого необходимо переключиться на вкладку **Адреса**. После заведения нового адреса в справочник, необходимо переключиться обратно на вкладку **Новые адреса** и открыть редактор адресного параметра договора, после чего произвести поиск нового адреса и заполнить адресный параметр договора.

Договоры - параметры

В дополнение к стандартному набору параметров договор может иметь набор пользовательских атрибутов. Для этого их надо перечислить и указать их тип.

Существуют следующие типы параметров договоров:

- **Текстовое поле** - простой текстовый параметр, например Фамилия;
- **Список** - значение параметра может быть выбрано из определённого перечня, перечень значений определяется в справочнике **Значения списков** (см. далее);
- **Адрес** - структурированный адресный параметр, ссылается на справочники **Городов, Улиц, Районов и Домов**;
- **Флаг** - параметр имеющий только два значения - да/нет;
- **E-Mail** - в значении данного параметра договора могут быть введены E-Mail адреса;
- **Договор** - параметр договора, ссылающийся на другой договор (необходим для реализации агентских договоров);
- **Дата** - параметр типа дата;
- **Телефон** - параметр типа телефон. Подробнее см. [здесь](#).
- **Разделитель** - визуальный разделитель в списке параметров.
- **Мультисписок** - может быть выбрано одно или несколько значений из определенного перечня, перечень значений определяется в справочнике **Значения мультисписков** (см. далее);

Справочники		Параметры		
Адрес - страны		↑	↓	
Адрес - города		Код	Тип поля	Наименование поля
Адрес - районы		1	Текст. поле	Краткое название ...
Адрес - улицы		2	Текст. поле	Полное название о...
Адрес - кварталы		3	Текст. поле	Должность руково...
Адрес - дома		4	Текст. поле	Ф.И.О. руководите...
Договоры - параметры		5	Текст. поле	Факс
Договоры - группы параметров		6	Текст. поле	Телефон(ы)
Договоры - группы		7	Текст. поле	Контактное лицо
Договоры - шаблоны комментариев		9	Текст. поле	Адрес (юридический)
Договоры - значения списков		8	Текст. поле	E-mail
Договоры - обслуживание		10	Текст. поле	Адрес (почтовый)
Договоры - скрипты поведения		11	Текст. поле	Адрес (физический)
Договоры - виды рассылок		12	Текст. поле	Адрес для рассылк...
Типы платежей		14	Текст. поле	Наличие договора
Типы расходов		13	Текст. поле	Комментарии
Типы времени		19	Текст. поле	АДРЕС
Helpdesk - Статус тем		20	Электронный адрес	Адрес(e-mail)
CRM - типы звонков		22	Обслуживание договора	Обслуживание
CRM - группы решения		25	Адрес	Новый адрес
CRM - исполнители				

Также существует возможность включить ведение истории параметров договоров каждого договора. Для того, чтобы включить ведение истории нужного параметра, двойным щелчком откройте редактор параметра и установите галочку в поле **"История"**. Нажмите **"ОК"**.

дический)	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
нование: Должность руководителя	<input checked="" type="checkbox"/> история
Найти	
Найти далее	

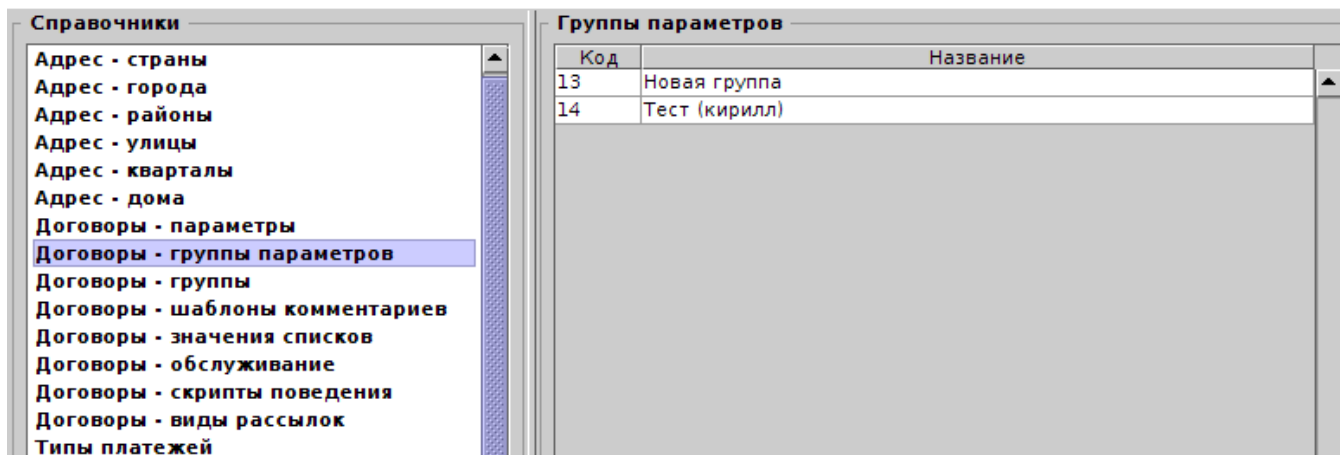
О просмотре истории параметра подробнее см. [здесь](#).

Договоры - группы параметров

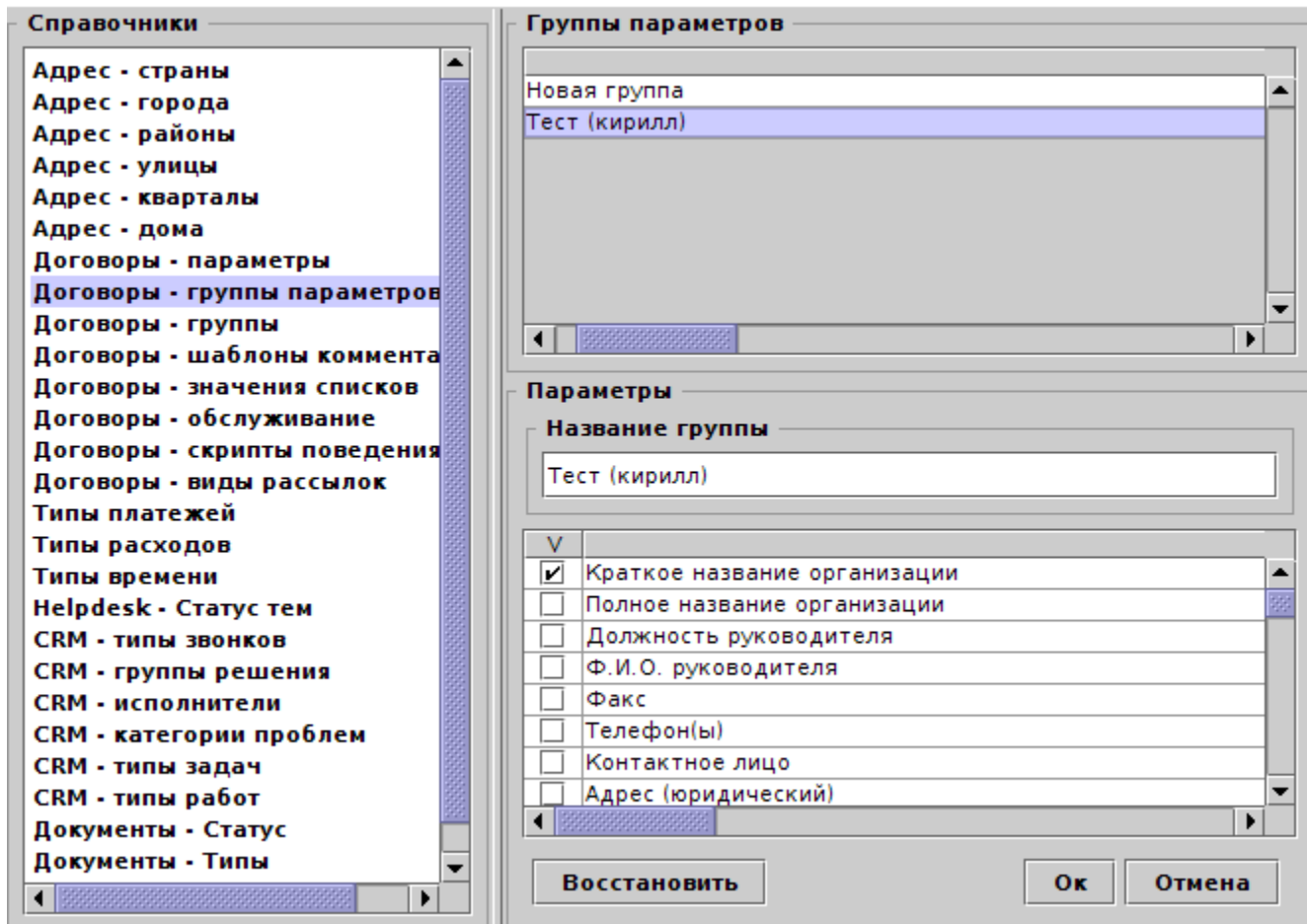
Иногда для разных групп договоров существуют параметры, которые не совпадают друг с другом. Так, для договора на частное лицо параметр "Юр. адрес" не имеет смысла. Чтобы при редактировании договора отображались только нужные параметры, следует создать группу параметров и в привязке параметров указать необходимый набор атрибутов.

Группа параметров привязывается к договору, определяя какими параметрами он обладает.

Ниже изображен справочник групп параметров - это простой перечень, где редактирование осуществляется кнопками панели инструментов.



При добавлении новой группы или редактировании существующей открывается панель привязки параметров, которая определяет какие параметры есть у договоров группы.



Договоры - группы

Группы необходимы для логического объединения договоров одного типа, поиска. Например: "Телефония", "Интернет" и т.п.. У одного договора может быть установлено сразу несколько групп, они сохраняются битовой маской.

Справочники

- Адрес - страны
- Адрес - города
- Адрес - районы
- Адрес - улицы
- Адрес - кварталы
- Адрес - дома
- Договоры - параметры
- Договоры - группы параметров
- Договоры - группы**
- Договоры - шаблоны комментариев
- Договоры - значения списков
- Договоры - обслуживание
- Договоры - скрипты поведения
- Типы платежей
- Типы расходов
- Типы времени
- Helpdesk - Статус тем
- CRM - типы звонков
- CRM - группы решения
- CRM - исполнители
- CRM - категории проблем
- CRM - типы задач
- CRM - типы работ
- Документы - Статус
- Документы - Типы
- Документы - Журналы

Группы договоров

Код	V	Наименование группы	R
0	<input checked="" type="checkbox"/>	Б	<input type="checkbox"/>
1	<input checked="" type="checkbox"/>	С	<input checked="" type="checkbox"/>
2	<input checked="" type="checkbox"/>	Е	<input checked="" type="checkbox"/>
3	<input checked="" type="checkbox"/>	IMPORT(чистится! - не ставить просто так!!)	<input checked="" type="checkbox"/>
4	<input checked="" type="checkbox"/>	Частные лица	<input checked="" type="checkbox"/>
5	<input checked="" type="checkbox"/>	L	<input checked="" type="checkbox"/>
6	<input checked="" type="checkbox"/>	M	<input checked="" type="checkbox"/>
7	<input checked="" type="checkbox"/>	S	<input checked="" type="checkbox"/>
8	<input type="checkbox"/>	U	<input checked="" type="checkbox"/>
9	<input checked="" type="checkbox"/>	V	<input checked="" type="checkbox"/>
10	<input type="checkbox"/>	Дольваоылдва	<input checked="" type="checkbox"/>
11	<input checked="" type="checkbox"/>	X	<input checked="" type="checkbox"/>
12	<input type="checkbox"/>	Z	<input checked="" type="checkbox"/>
13	<input checked="" type="checkbox"/>	Организации	<input checked="" type="checkbox"/>
14	<input checked="" type="checkbox"/>	Каточки	<input checked="" type="checkbox"/>
15	<input checked="" type="checkbox"/>	Сотрудники	<input checked="" type="checkbox"/>
16	<input checked="" type="checkbox"/>	VIP-клиенты	<input checked="" type="checkbox"/>
17	<input type="checkbox"/>	Конверты	<input checked="" type="checkbox"/>
18	<input checked="" type="checkbox"/>	Хостинг	<input checked="" type="checkbox"/>
19	<input checked="" type="checkbox"/>	Фиктивные	<input checked="" type="checkbox"/>
20	<input checked="" type="checkbox"/>	Закрытые	<input checked="" type="checkbox"/>
21	<input checked="" type="checkbox"/>	карта - подарок	<input checked="" type="checkbox"/>
22	<input type="checkbox"/>	Группа 222	<input checked="" type="checkbox"/>

Редактор

Наименование группы
Частные лица

Группа используется: да нет Редажируемый элемент

Восстановить Ок Отмена

Для того, чтобы убрать группу у всех договоров, необходимо в справочнике групп выбрать группу, нажать правой кнопкой мыши и в появившемся меню активировать пункт **Удалить у всех договоров**. Признак **Группа используется** определяет видимость группы в редакторе в договоре и во всех фильтрах поиска. Признак **Редажируемый элемент** означает то, что помеченные группы вручную можно устанавливать /убирать в договоре.

Иногда бывает необходимо вычислять битовую маску групп, чтобы указать значение в настройках некоторых подсистем биллинга, либо чтобы подставлять значения в sql-запрос с фильтрацией по группе. Для этого существует возможность множественного выбора групп договоров в таблице. При выборе одного и более элемента внизу отобразится панель с вычисленной битовой маской выделенных групп и кнопкой "Скопировать в буфер обмена".

Справочники

- Договоры - параметры
- Договоры - группы параметров
- Договоры - группы**
- Договоры - шаблоны ком
- Договоры - именованные
- Договоры - значения спи
- Договоры - скрипты пове
- Договоры - обслуживани
- Типы платежей
- Типы расходов
- Типы возвратов
- Типы резервов
- Типы времени
- CRM - типы звонков
- CRM - группы решения
- CRM - исполнители
- CRM - категории проблем
- CRM - типы задач
- Helpdesk - Статус тем
- Бонусы - приходы

Группы договоров

рассчитывать количество договоров

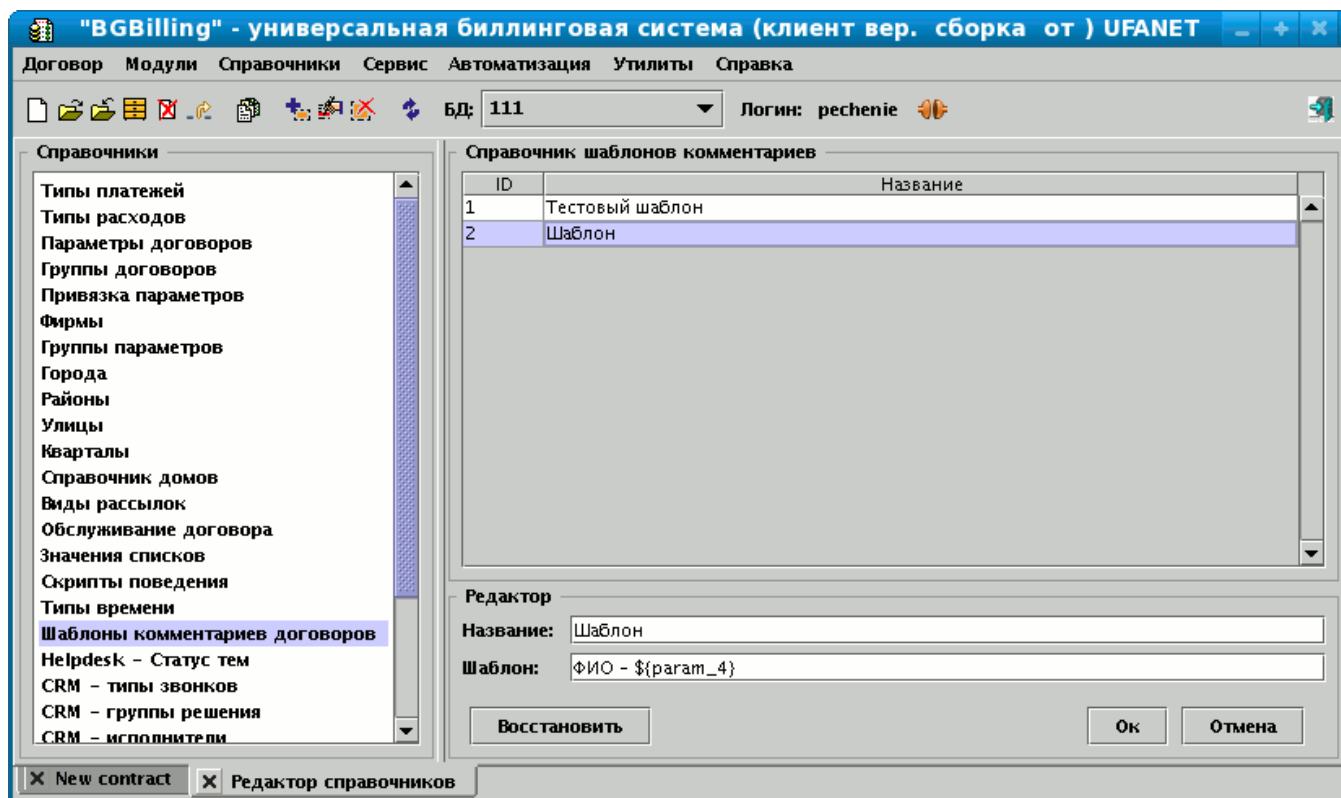
ID	Активна	Правка	Наименование	Комментарий	Договоров	...
46	<input type="checkbox"/>	<input type="checkbox"/>	Группа 46		0	
47	<input type="checkbox"/>	<input type="checkbox"/>	Группа 47		0	
48	<input type="checkbox"/>	<input type="checkbox"/>	Группа 48		0	
49	<input type="checkbox"/>	<input type="checkbox"/>	Группа 49		0	
50	<input type="checkbox"/>	<input type="checkbox"/>	Группа 50		0	
51	<input type="checkbox"/>	<input type="checkbox"/>	Группа 51		0	
52	<input type="checkbox"/>	<input type="checkbox"/>	Группа 52		0	
53	<input type="checkbox"/>	<input type="checkbox"/>	Группа 53		0	
54	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Группа 54		0	
36	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Загруженные		0	
20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Закрытые		0	
62	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	импорт		0	
21	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	карта - подарок		0	
14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Каточки		0	
13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Организации		0	
17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Расторгнутые договоры	тестирование разблокировки р...	0	
15	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Сотрудники		0	
19	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Фиктивные		0	
18	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Хостинг		0	
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Частные лица		0	

Битовая маска выбранных групп договоров равна 3940352

Скопировать в буфер обмена

Договоры - шаблоны комментариев

Зачастую приходится вводить однотипные комментарии в имени договора, которые совпадают с какими-либо его параметрами (Ф.И.О., электронный адрес и прочее). Для автоматизации этого процесса можно использовать шаблоны комментариев. Для редактирования справочника шаблонов комментариев выберите пункт меню **Справочники->Другие** и затем **Шаблоны комментариев** в панели слева.



Для создания нового шаблона комментария нажмите на кнопку **Создать** на панели сверху. Для редактирования шаблона выделите необходимую строку и нажмите на кнопку **Редактировать** на панели сверху, либо дважды щелкните по ней мышью. Для удаления выбранного шаблона нажмите на кнопку **Удалить**.

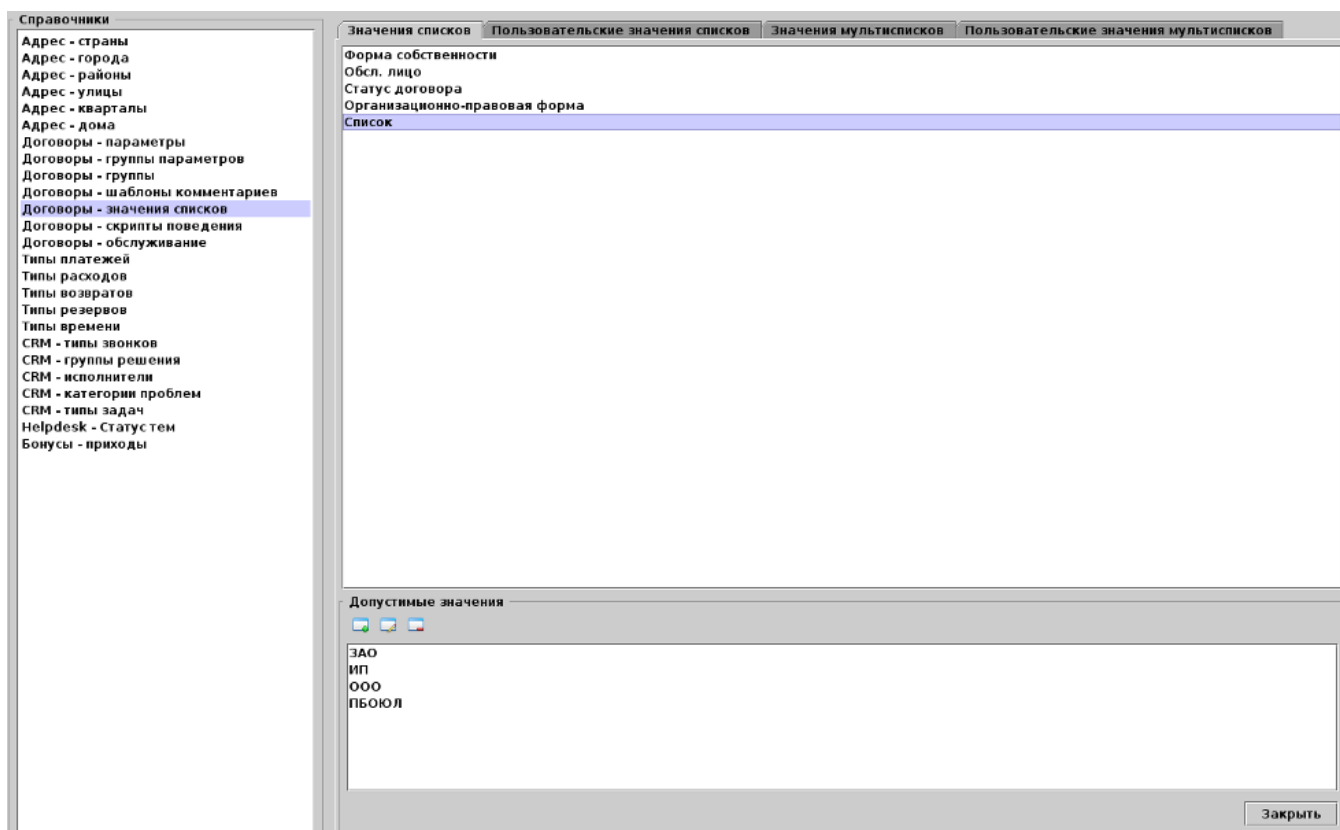
В шаблоне указывается **Название** шаблона, а также сам **Шаблон**. Шаблон - это произвольная строка, в которой возможна подстановка значений из параметров договора, путём включения макросов **\${param_<pid>}**, где **<pid>** - код **параметра договора**. Например: **\${param_4}** - подстановка значения параметра договора с кодом 4. При изменении параметров договора комментарий автоматически изменяется с учётом новых значений параметров.

С версии 8.2107+ поддерживается подстановка из данных **Клиента**, **\${client:/customer/fio/last}**, где **client** префикс, **/customer/fio/last** - путь к нужным данным в json клиента.

В договоре шаблон комментария указывается при **редактировании имени и комментария договора**.

Договоры - значения списков

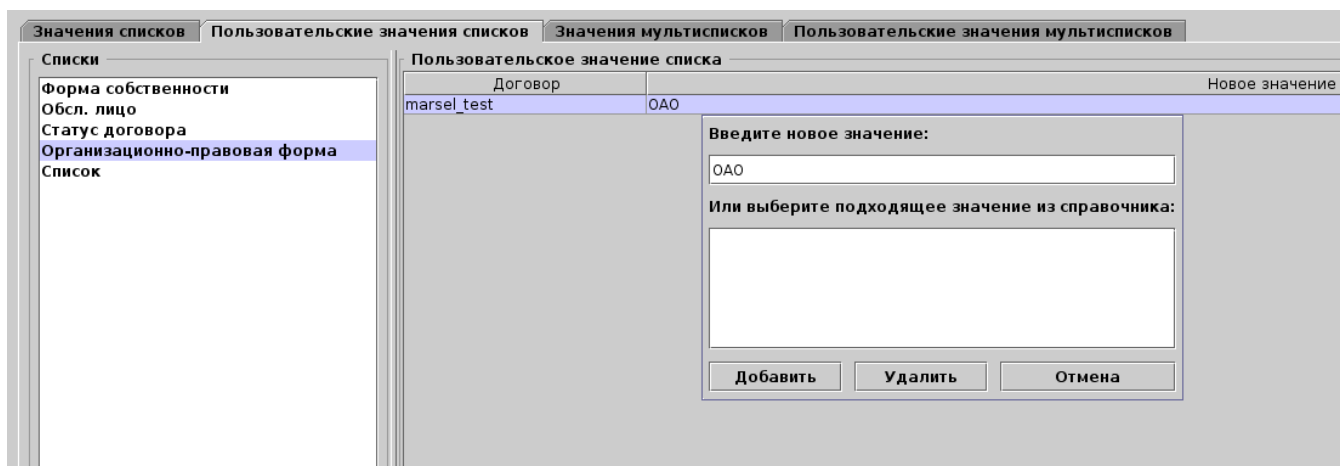
В данном справочнике указываются допустимые значения для списковых параметров.



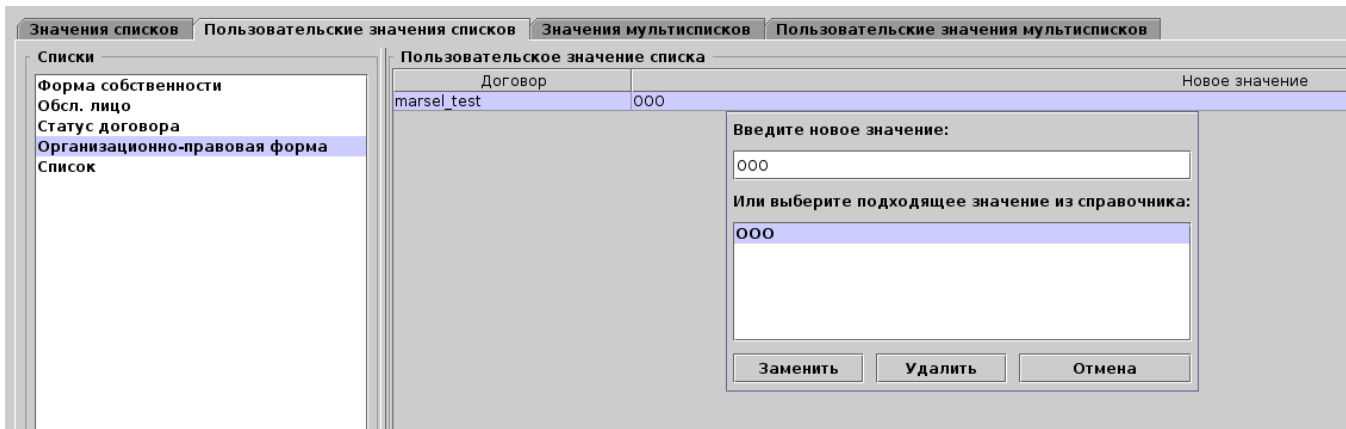
В верхнем списке выбирается требуемый списковый параметр, по двойному клику мыши либо открывается перечень значений. Редактирование перечня осуществляется нижней панелью инструментов.

На вкладке **Пользовательские значения списков** можно добавить новое значение справочника и/или заменить пользовательское значения списка на существующее значение.

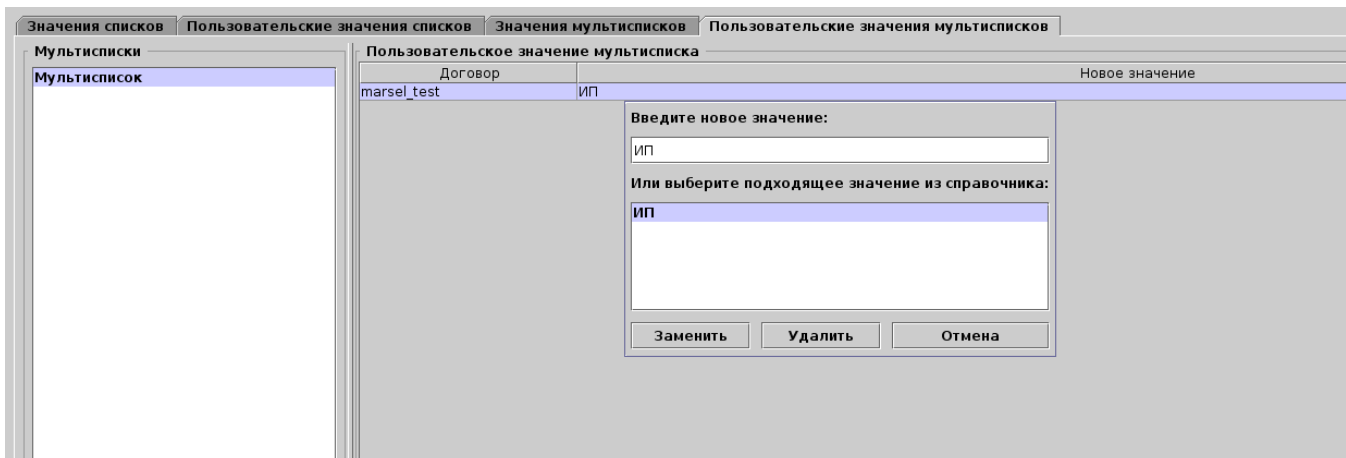
Добавить - добавится новое значение в справочник и пользовательское значение в договоре заменится на добавленное. **Удалить** - значение удалится из договора.



Заменить - пользовательское значение заменится на значение из справочника.



Аналогично с **Значения мультисписков** и **Пользовательские значения мультисписков**



Договоры - обслуживание

Здесь указывается перечень обслуживающих лиц. В дальнейшем эти лица могут быть привязаны в параметрах конкретного договора как обслуживающие какие-то адреса клиентов. Использование данного параметра нежелательно в связи с появлением в биллинге системы объектов, позволяющей выделять отдельные точки клиента в объекты с указанием в параметрах адреса объекта, обслуживающих лиц и пр.

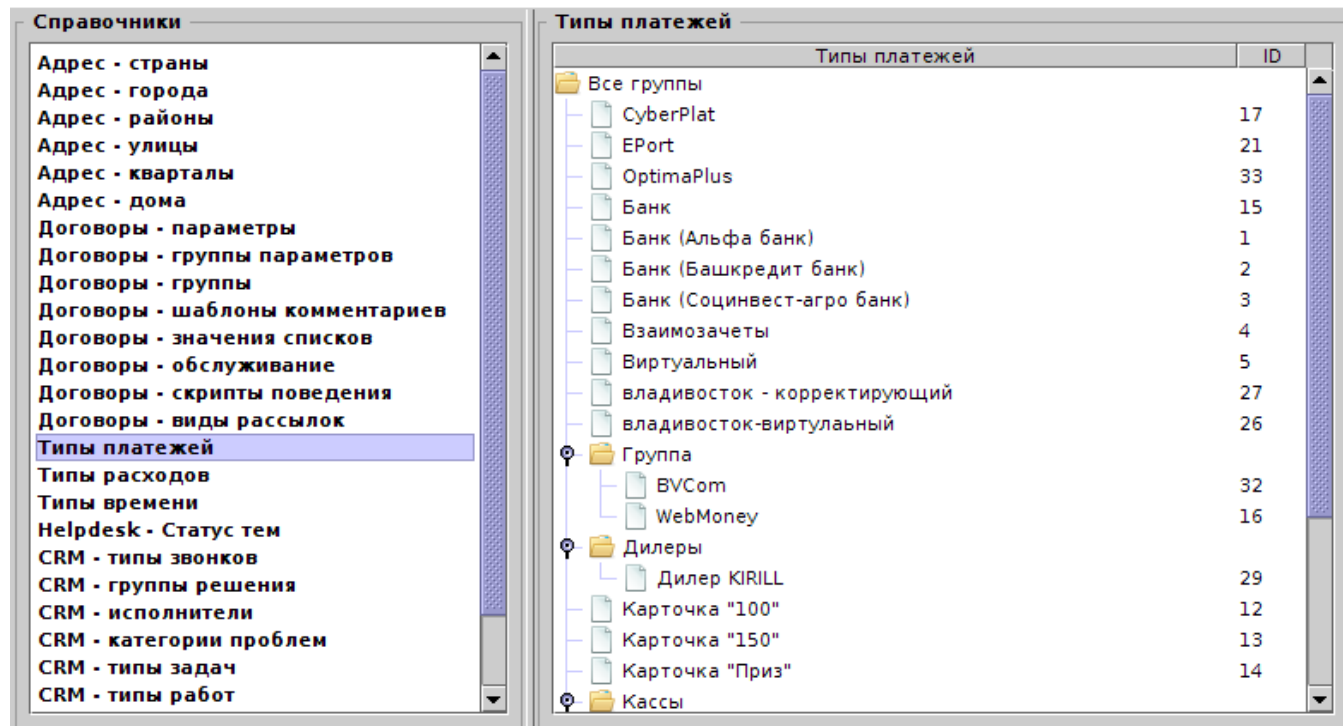
Договоры - скрипты поведения

Перечень скриптов поведения, которые могут быть назначены договору. Скрипт поведения определяет реакцию на определённые события относящиеся к договору, более подробно о скриптах VGBS описано [далее](#).

Типы платежей

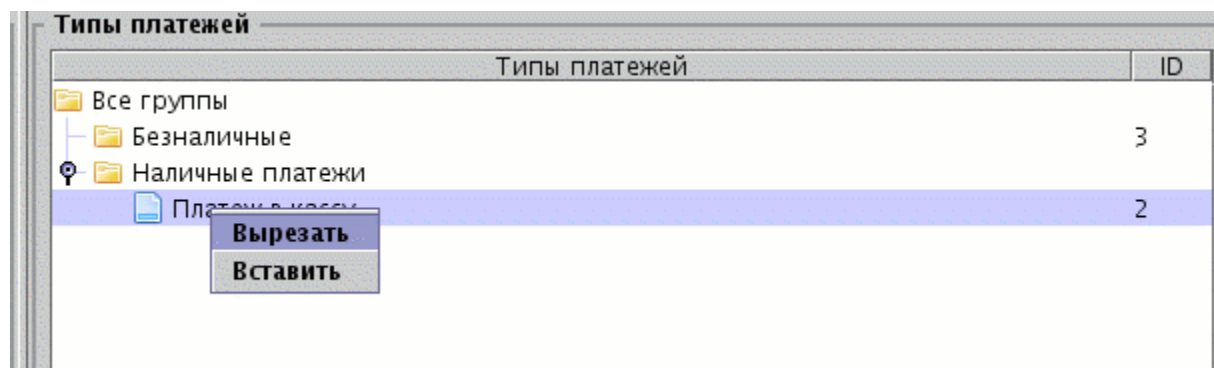
Типы платежей необходимы для разделения потока платежей клиентов по видам. Далее по ним может быть построена аналитика. Примеры типов платежей: "Наличный", "Через банк", "Дилер X", "Платёж за подключение" и т.п..

Для добавления нового типа, либо подгруппы нужно выбрать группу для добавления (группа **Все группы** существует изначально) и нажать **Новый элемент** на панели инструментов.



В справочник можно добавить как группу платежей (см. снимок выше) так и непосредственно платёж, выбирая галочку **Группа**, либо **Элемент группы**. Группы типов платежей предназначены исключительно для визуально более лёгкого восприятия перечня типов платежей в справочнике. Нигде больше они в данный момент не используются.

После добавления платёж, либо группу можно перемещать между группами-предками. Для этого выберите узел, нажмите правой кнопкой мыши, в появившемся меню выберите **Вырезать**, далее выберите новую группу для переносимого узла вызовите меню правой кнопкой мыши и выберите **Вставить**.



Параметр **Нередактируемый элемент** показывает, что данный тип платежа нельзя заносить и редактировать через редактор платежей в договоре. Платежи такого типа используются, например, для платежных систем, автоматической загрузки платежей из реестра и т. п. Нередактируемые платежи заносятся системой без непосредственного участия человека, это мера защиты от правки такого платежа оператором, например, при активации интернет-карточки. Для правки такого платежа пользователь, обладающий правами изменять справочники, должен снять признак нередактируемости с типа платежа, скорректировать платёж в договоре и вернуть признак.

Для того чтобы добавлять платежи договорам в справочнике должен быть хотя бы один **Редактируемый** тип платежа. Создайте его сразу.

Если галочка **Нередактируемый элемент** стоит у группы, то типы платежей внутри группы также становятся нередактируемыми.

Следует обратить внимание на столбец ID - это код типа платежа. Во всех конфигурациях, где необходимо указание типа платежа, указывается именно этот код.

Типы расходов

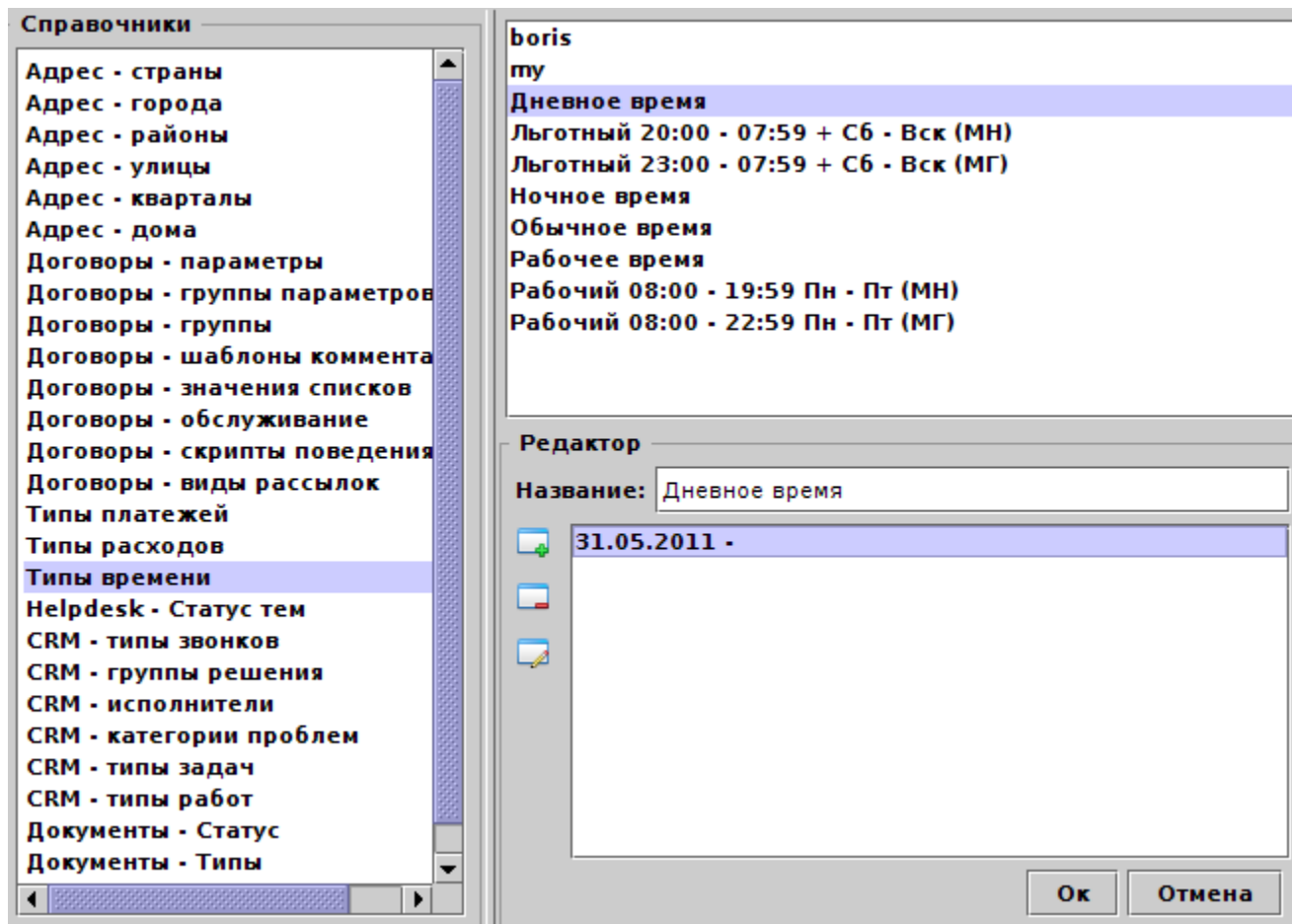
Расход - это единоразовое снятие с баланса договора некоторой суммы. Так же как и платёж расходы разделяются по типам, отображающим их назначение. Например: "Переподключение", "Вызов мастера" и т.п. Расходы также бывают редактируемые и нередатируемые. Для занесения расхода в договор биллинга в справочнике должен быть заведён хотя бы один редактируемый тип расхода. Создайте его сразу.

Справочники		Типы расходов	
		Типы расходов	ID
Адрес - страны		Все группы	
Адрес - города		1	
Адрес - районы		fgb	
Адрес - улицы		Ticket	39
Адрес - кварталы		Ticket2	41
Адрес - дома		11	38
Договоры - параметры		mb	42
Договоры - группы параметров		Абонплата	44
Договоры - группы		Возврат денег в модуле BVCom	57
Договоры - шаблоны комментариев		Перенос средств	51
Договоры - значения списков		Подписка на Антивирус Касперского	58
Договоры - значения списков		Покупка валюты	55
Договоры - скрипты поведения		Разовые	
Договоры - виды рассылок		AV-Desk	54
Типы платежей		Ключи OEM	53
Типы расходов		расходы за подключение	
Типы времени		45345	
Helpdesk - Статус тем		1111	50
CRM - типы звонков		Ticket	56
CRM - группы решения		материалы	47
CRM - исполнители			
CRM - категории проблем			
CRM - типы задач			
CRM - типы работ			

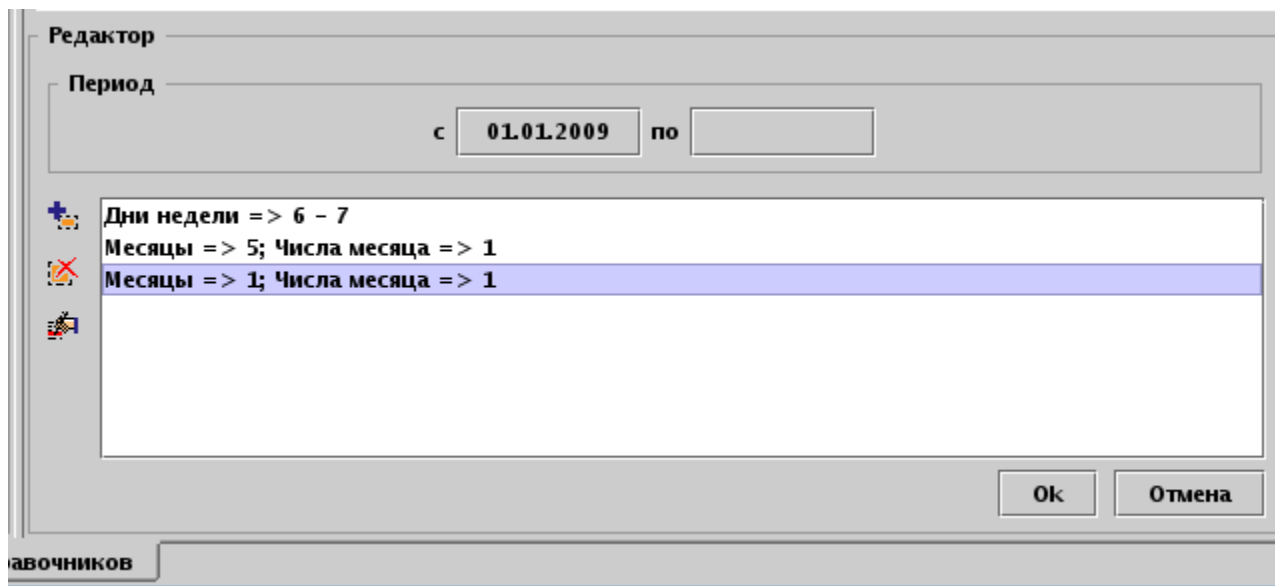
Редактирование справочника аналогично справочнику типов платежей.

Типы времени

Справочник предназначен для создания глобальных категорий времени и использования их в дальнейшем в тарифных планах в узлах типа **Организация по типу времени**.



При открытии типа времени для редактирования открывается редактор с одним или несколькими периодами, бесконечный период выглядит как просто тире. Периоды позволяют внутри одного типа времени задавать разные маски для разных интервалов дней. Например, можно вести учет выходных и праздничных дней на несколько лет. При открытии редактирования периода отображается редактор следующего вида:



В верхней области необходимо определить, собственно, период действия. В приведенном примере начато определение выходных и праздников на 2009 год.

Далее задаются несколько правил. Каждое правило устанавливает маски на часы, дни недели, дни месяца, месяцы. Месяцы могут принимать значения от 1 до 12, часы от 0 до 23, дни недели от 1 до 7, месяцы от 1 до 12. Маски в пределах правила соединяются условием **"И"**. Т.е., например, часы от 0 до 8 **И** дни недели 6-7. Правила внутри типа времени соединяются условием **ИЛИ**. Совпадение хотя бы с одним правилом времени даёт основание отнести его к данному типу.

Несколько особенностей поведения типов времени:

- 1. пустая дата начала или окончания периода означает бесконечность;
- 2. при пустом списке периодов в типе времени любое время относится к данному типу;
- 3. пустой набор правил периода также "включает" в себя любое время, попавшее в данный период.

Клиенты

С версии 8.2107 добавлена возможность создавать Клиентов и связывать их с договорами, тем самым логически объединяя несколько договоров одного клиента

Настройка клиентов

При вводе параметров клиентов происходит валидации вводимых значений, при необходимости изменить условия валидации это можно сделать через следующие параметры в конфигурации сервера (после изменения требуется перезагрузка клиента)

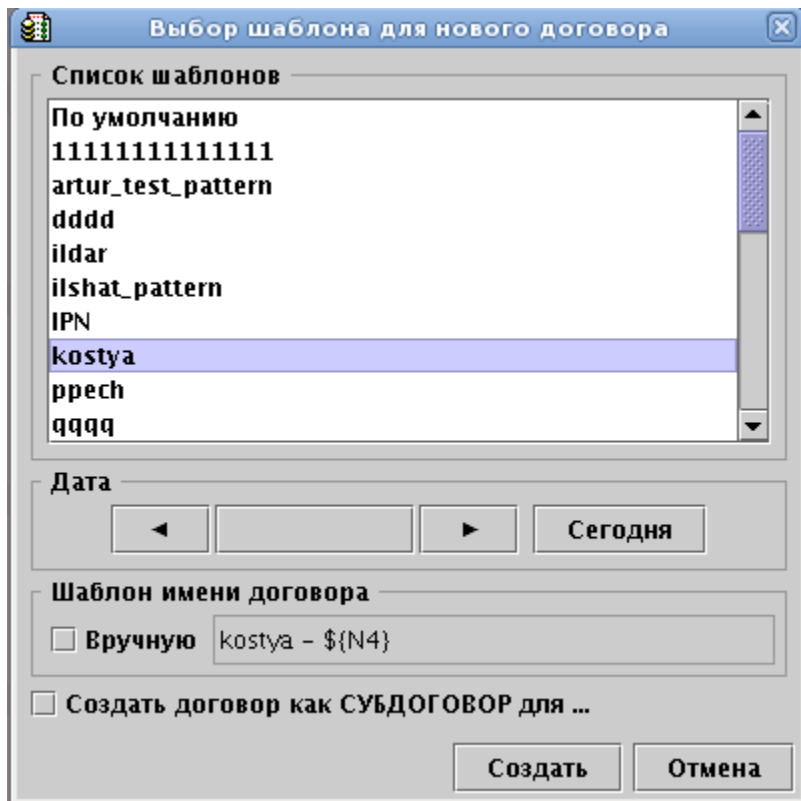
```
client.gui.customer.u.inn.regex=^(\\.{0}|[0-9]{10}|[0-9]{12})$
client.gui.customer.u.kpp.regex=^(\\.{0}|[0-9]{9})$
client.gui.customer.u.ogrn.regex=^[0-9]{13}$
client.gui.customer.u.bank.account.regex=^[0-9]{20}$
client.gui.customer.u.bank.bik.regex=^[0-9]{9}$
client.gui.customer.f.fio.last.regex=^[ \\-]*$
client.gui.customer.f.fio.first.regex=^[ \\-]*$
client.gui.customer.f.fio.middle.regex=^[ \\-]*$
client.gui.customer.f.inn.regex=^[0-9]{12}$
client.gui.customer.f.document.passport_rf.series.regex=^[0-9]{4}$
client.gui.customer.f.document.passport_rf.number.regex=^[0-9]{6}$
client.gui.customer.f.document.passport_rf.code.regex=^[0-9]{3}-[0-9]{3}$
```

Договор

Общие сведения, создание договора

Договор - основная рабочая единица системы **BGBilling**. В терминах **BGBilling** договор - это отдельный баланс (кошелёк) и набор параметров. К договору подключаются установленные в системе модули, посредством добавления услуг этих модулей, что позволяет абоненту использовать свой баланс на различные виды услуг, предоставляемых модулями.

Создание договора производится выбором пункта меню **Договор=>Новый договор**, либо кнопки **Новый договор** на стандартной панели инструментов. При создании договора открывается диалог следующего вида:



Если система только что установлена, в списке шаблонов будет предложен только один шаблон **По умолчанию**. Перед созданием договора необходимо выбрать дату и шаблон, по которому будет создан договор. Шаблон имени договора можно задать вручную. Также данный диалог позволяет создать **субдоговор**. Для этого нужно выбрать соответствующий пункт, после чего появятся дополнительные параметры.

Выбор шаблона для нового договора

Список шаблонов

- По умолчанию
- 111111111111111
- artur_test_pattern
- dddd
- ildar
- ilshat_pattern

Дата

30.06.2011 Сегодня

Шаблон имени договора

Вручную kostya - \${N4}

Субдоговор для

зависимый баланс test_kostya X

Копировать параметры супер договора

Параметры договора

- Краткое название организации
- Полное название организации
- АДРЕС

Все Сброс Инверт.

Создать Отмена

При создании субдоговора следует указать какой будет баланс, **зависимый** или **независимый**. Также необходимо выбрать супердоговор, у которого можно скопировать заполненные параметры договора. Настройка копирования производится в конфигурации сервера. В нее следует добавить одну или несколько подобных записей, в зависимости от количества шаблонов.

```
contract.params.copy.<pattern_id>=1,2,19
```

где **pattern_id** - код шаблона, по которому создается договор, а значение этой переменной является список id параметров договора.

Если данной переменной нет в конфигурации сервера, то в списке параметров договора будут отображаться все заполненные параметры супердоговора, а если есть, то только указанные параметры.

После нажатия **Создать** открывается вкладка со вновь созданным договором **New contract**. При использовании шаблонов система способна также самостоятельно вести последовательную нумерацию вновь создаваемых договоров с использованием порядкового номера договора, года создания.

Обзор карточки договора

Карточка вновь созданного договора выглядит следующим образом.

The screenshot shows the BGVillingClient application window. The title bar reads "BGVillingClient (вер. сборка от) ООО "Бител" (идент. номер: 1494)". The main window has a menu bar with "Договор", "Модули", "Плагины", "Справочники", "Сервис", "Утилиты", and "Справка". Below the menu bar is a toolbar with icons. The main area is divided into a left sidebar and a right panel. The left sidebar contains a tree view with categories like "Параметры договора", "Баланс", "Модули", "Web", and "Примечания". The right panel shows a form for "New contract []" with a unique identifier "302117" and a period "с 22.12.2008 по". Below this is a "Параметры" section with a table of fields and values. The table has columns "Параметр" and "Значение". The "Телефон(ы)" field is highlighted with a red number 4. The "Исходящий остаток" field in the "Баланс" section is highlighted with a red number 5. The "New contract []" field is highlighted with a red number 1, the unique identifier "302117" with a red number 2, and the period "с 22.12.2008 по" with a red number 3.

Числом 1 отмечен номер договора. Номер может содержать любые буквы и цифры и, для вновь созданного договора по шаблону **По умолчанию**, устанавливается в **New contract**. После номера в скобках указывается комментарий договора, для нового договора он пуст. Числом 2 отмечен уникальный идентификатор договора.

Рекомендуется присваивать договорам единообразную буквенно-цифровую нумерацию вида **B5555-07** (пример), при этом префикс можно использовать для обозначения типа договора (телефония, интернет, IP-телефония и т.п.), а порядковый номер - для последовательной нумерации.

В комментарии можно указывать название организации, либо Ф.И.О. частного лица. Для изменения номера и комментария кликните на них. В появившемся диалоге введите требуемые значения.

The dialog box titled "Редактор номера договора" has a blue header. It contains three main input areas: "Номер договора" with a text box containing "New contract", "Шаблон комментария" with a dropdown menu showing "Без шаблона", and "Комментарий (Наименование клиента)" with a text box containing "ФИО -". At the bottom, there are three buttons: "Восстановить", "Ок", and "Отмена".

Комментарий для договора может задаваться [шаблоном комментария](#). Если выбран пункт **Без шаблона**, то комментарий можно задавать вручную, в противном случае (если выбран какой-либо шаблон), то комментарий формируется автоматически.

Кнопка **Восстановить** возвращает поля редактора номера в исходное состояние.

Период договора (3) определяет временной диапазон существования контракта. Дата начала устанавливается в момент создания договора и может быть впоследствии изменена. При закрытии даты завершения система автоматически закрывает периоды действия всех услуг в договоре (период действия логинов, абонплат, ИП адресов и т.п.).

В левой области карточки договора расположено дерево навигации (5). Оно позволяет переключать вкладки редактирования характеристик договора.

Первым при открытии договора отображается вкладка с параметрами договора (сами параметры обозначены числом 4, более подробно чуть далее).

Более подробно об остальных вкладках по тексту ниже. В частности об объектах вы можете прочитать [здесь](#), при первом ознакомлении этот пункт можно пропустить, а об иерархии договоров описано в отдельной [главе](#).

Параметры договора

Общие сведения о параметрах договоров

Набор параметров договора состоит из фиксированных и настраиваемых параметров. К фиксированным относятся:

- **Лицо - Физическое**, либо **Юридическое**, параметр обозначает тип контрагента по договору;
- **Группа параметров** - определяет набор настраиваемых параметров договора, которые будут использованы в данном договоре, перечень групп настраивается в справочнике **Справочники=>Другие=>Договоры - группы параметров**.

О создании настраиваемых параметров договоров, их привязке к группам описано в описании справочника "Договор - параметры".

Набор настраиваемых параметров отображаются в виде таблицы. Если группа параметров не установлена для договора, то договору доступны все заведённые в справочнике **Параметры договоров** параметры. Иначе будут отображены только параметры, входящие в выбранную группу. Для удобства поиска и заполнения параметров предусмотрен фильтр по параметрам. Также можно вывести только заполненные параметры: для этого необходимо нажать на соответствующую кнопку.

Параметр	Значение
Краткое название организации	
Полное название организации	
Должность руководителя	
Ф.И.О. руководителя	
Факс	
Телефон(ы)	
Контактное лицо	
Адрес (юридический)	

Параметры Отчет Карточки HelpDesk CRM Документы

Договор

test_kostya [Фιο: Костя; мьло: ; телефон:] 352700

Период с 30.01.2009 по 03.12.2012

Лицо: Юридическое ОК

Сортировка (дата): ↓

Страница: 1 из 1 [25]

Дата	Пользователь	Лицо
20.01.2010 11:48:41	kostya [kostya]	Юридическое
20.01.2010 11:48:38	kostya [kostya]	Физическое
20.10.2009 11:52:15	kostya [kostya]	Юридическое
17.07.2009 11:53:41	kostya [kostya]	Физическое
07.07.2009 13:12:52	kostya [kostya]	Юридическое
07.07.2009 13:12:44	kostya [kostya]	Физическое
07.07.2009 13:12:41	kostya [kostya]	Юридическое
07.07.2009 13:12:39	kostya [kostya]	Физическое
07.07.2009 13:12:35	kostya [kostya]	Юридическое

Копирование параметров

Для копирования параметров из других договоров в текущий открытый договор необходимо вначале открыть необходимые договоры. Далее в договоре, в который нужно скопировать параметры из других нажать на кнопку **копировать из...**

The screenshot shows a software window titled "Параметры" (Parameters). At the top, there are buttons for "из группы", "все", "только заполненные", "фильтр по типам", "копировать из...", and a "Группа:" dropdown menu. The main area is divided into three sections:

- Договора** (Contracts): A list box containing "test_kostya [ФИО - Васи" and "x0000 [ФИО - Иванов Ив".
- Шаблоны** (Templates): A list box containing "Без шаблона", "Банк (реквизиты)", and "Организация (название,".
- Параметры** (Parameters): A list of parameters with checkboxes. The "только заполненные" radio button is selected. The parameters include:
 - Краткое название организации: *asasdasd ???*
 - Полное название организации: *sd fdsdfsadasd dd*
 - Ф.И.О. руководителя: *Иванов Иван Иванович*
 - Контактное лицо: *d fgd fgd fasd*
 - E-mail: *sda*
 - Адрес (почтовый): *26.03.2005*
 - Комментарии: *комментарий к договору*
 - Адрес(e-mail): *Иван Васильевич ООО "Компания"*
 - Новый адрес: *прарапро*
 - Нужность: *false*
 - Обсл. лицо: *111*
 - Статус договора: *На отключение*
 - Пакетный режим Helpdesk: *false*
 - Телефон (новый): *+7-347-254-44-45; +7-347-239-03-45 [dgd fgd]*
 - Форма собственности: *частная*
 - ОРГ-правовая форма: *ИП*
 - список: *47*
 - Лицевой счет: *test123*

At the bottom of the window, there are buttons for "Все", "Сброс", and "Инверт.". Below the main window, there are three buttons: "Копировать (с заменой)", "Копировать (добавить не заполненные)", and "К списку пара...".

В открывшемся окне необходимо выбрать договор, из которого будут копироваться параметры. Для удобства поиска параметров можно выбрать **все** или **только заполненные** параметры договора, из которого они копируются. Далее необходимо отметить параметры, которые нужно скопировать. Если копирование параметров довольно частое действие и копируются одни и те же параметры, то можно завести шаблон, при выборе которого, параметры, которые указаны в шаблоне, будут автоматически выделяться. Шаблоны настраиваются в [конфигурации сервера](#). Копировать параметры можно в 2х режимах: в режиме замены или режиме дополнения.

Параметры типа "Текст", "Флаг", "Дата", "E-Mail"

Правила редактирования параметров различных типов различны. Для редактирования текстового параметра необходимо выбрать правый столбец, ввести новое значение и нажать **Enter**.

Для редактирования параметра типа **Флаг** достаточно изменить его нажатием мыши.

Обслуживание	Лицо 1: 1
Новый адрес	450105, Уфа, 1 квартал, Айская, д. 66/
Нужность	<input checked="" type="checkbox"/>
Дата включения	03.02.2005
Обсл. лицо	

Для редактирования параметра типа **Дата** необходимо двойным нажатием по строке выбрать календарь и, ввести дату, либо закрыть календарь, выбрав, тем самым, текущую дату, либо нажать **X**, очистив дату.

The screenshot shows a table with parameters and their values. A date selection calendar is overlaid on the table, showing the date 05.07.2007. The calendar has a title bar "5 июль 2007" and navigation buttons. The date 05.07.2007 is highlighted in the calendar grid.

Параметр	Значение
ИДЕНТИФИКАЦИОННЫЙ КОД	NK00001-03
АДРЕС	450105, Уфа, 1 квартал, Айская, д. 66/
ФИО	Равиль Зарипов 829323295, 8-9
Адрес(e-mail)	shamil@bitel.rupingvin@ufanet.ru
Лицо обслуживания	Лицо 1: 1
Новый адрес	450105, Уфа, 1 квартал, Айская, д. 66/
Нужность	<input checked="" type="checkbox"/>
Дата включения	05.07.2007
Лицо обслуживания	
Статус договора	Активен
Дата обнаружения	15.04.2006
Дата начала оказания услуг	23.04.2007
Адрес	asd@aaa.ru

Для редактирования параметра типа **E-Mail** двойным кликом вызовите редактор, указав адреса, каждый с новой строки.

The screenshot shows a parameter table with an email address selected. A dialog box titled "Редактор" is open, showing a list of email addresses and names. The dialog has buttons for "Восстановить", "Ок", and "Отмена".

Параметр	Значение
Краткое название организации	
АДРЕС	
Адрес(e-mail)	Иван Васильевич <ivan@comp.com>; ООО "Компания" <office@comp.com>
Нужность	<input type="checkbox"/>
Дата включения	14.05.2011
Телефон (новый)	+7-347-254-44-45; +7-347-239-03-45 (dgdgfd)
Форма собственности	частная

Редактор

E-Mail Адрес	Имя
ivan@comp.com	Иван Васильевич
office@comp.com	ООО "Компания"

Параметр типа "Адрес"

Для редактирования параметра типа **Адрес** вызывается редактор двойным кликом мыши по строке.

Параметры

из группы: **все** | только заполненные | фильтр по типам | копировать из... | Группа: _____

Тип: **текст** | **адрес** | телефон | e-mail | флаг | дата | список | договор | обл. договора

Параметр	Значение
АДРЕС	450098, г. Уфа, Октябрьский район, Бульвар Давлеткильдеева, д. 5/1, кв. 100, 3 под., 7 эт.
Новый адрес	г. Уфа450098, Российская, д. 159/1, 100 - параметр ср19
Адрес Юридический(АСВТ)	450098, г. Уфа, Октябрьский район, Бульвар Давлеткильдеева, д. 5/1, кв. 100
Адрес_тест	454545, г. Уфа, Ленинский район, 001 юг), Ленина
Адресный параметр для ...	1231231цуаавыап

Редактор

Дом:

Квартира

Номер:

Комната:

Подъезд:

Этаж:

Комментарий | **Дополнительная информация** | **Формат адреса**

Параметр	Значение
квартал	Не указано
Диапазон этажей	1-108/4/1/9
Диапазон подъездов	1-108/36/1
индекс	450098
Дата подключения	23.03.10
район	Октябрьский район

При выборе дома система автоматически высвечивает индекс, район и квартал. Для выбора дома нужно нажать на кнопку с адресом. При этом редактор перейдет в режим поиска дома.

Редактор

Фильтр

Улица:

Дом [дробь]:

Пользовательское значение адреса

Найденные записи:

Дом	ID
Ленина, 13	5 344
Ленина, 22	14 220
Ленина, 32	5 354
Ленина, 32/1	5 355
Ленина, 50	6 523

Для осуществления поиска необходимо начать вводить название улицы в соответствующем элементе управления. При этом, по мере набора, покажется выпадающий список названий улиц (с указанием городов), которые содержат в виде подстроки введенную в поле строку. В этом списке выводится только 25 первых записей. Выделение нужной улицы осуществляется с помощью клавиш "вверх" и "вниз" на клавиатуре. Для выбора выделенной улицы нужно нажать клавишу Enter. При этом автоматически начнется поиск всех домов, находящихся на выбранной улице, с последующим выводом найденного списка в таблице.

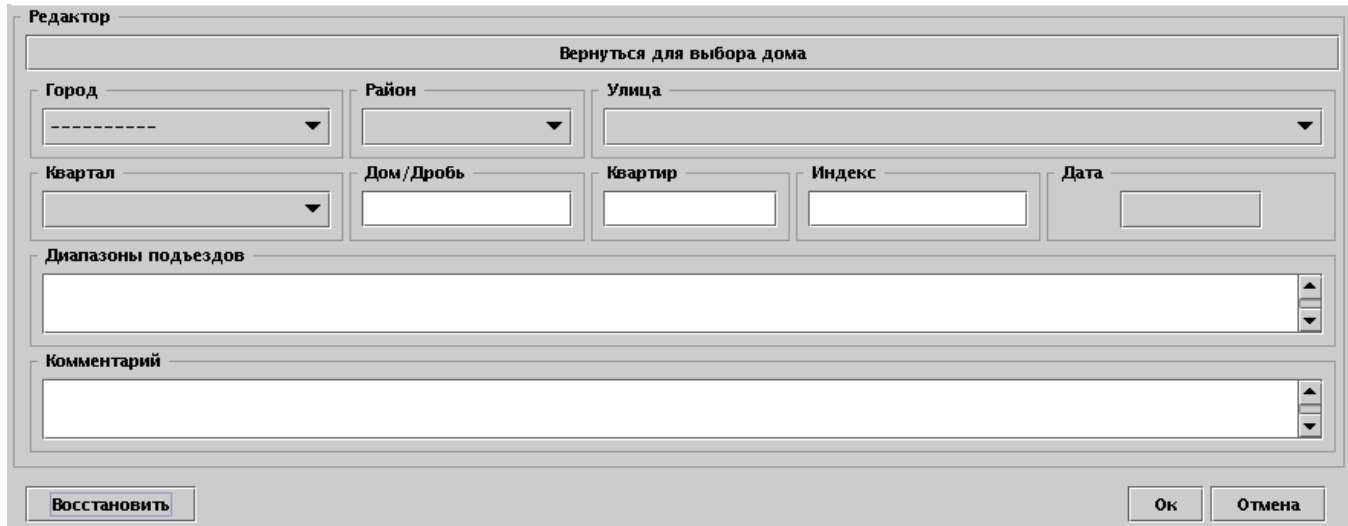
Кроме выбора улицы из выпадающего списка предусмотрен второй вариант. Можно набрать в поле "Улица" название улицы или ее часть и нажать Enter (не выбирая при этом из выпадающего списка). При этом в таблицу будут выведены все найденные улицы, содержащие введенную строку в качестве подстроки. Далее, двойным щелчком по строке таблицы выбирается нужная улица, после чего в таблицу загружаются дома, расположенные на этой улице. Описанный режим поиска улиц полезен, если требуемая улица отсутствует в выпадающем списке (ввиду ограничения в 25 записей).

Для выбора дома нужно дважды щелкнуть на соответствующей строке в списке найденных домов. После этого редактор вернется в исходный режим, где можно указать дополнительную информацию об адресе.

Для уменьшения количества выводимых записей в таблице предусмотрен фильтр по номеру дома. В поле Дом (в него можно перейти нажатием клавиши Tab) нужно ввести номер интересующего дома и нажать Enter. При этом в списке найденных домов останутся только те дома, в номерах которых присутствует подстрока, введенная в поле Дом.

Установкой опции конфигурации сервера **address.create=yes** можно разрешить добавление недостающего дома непосредственно при редактировании параметра. При этом пользователю будет доступна кнопка "**Добавить новый дом...**" в режиме поиска дома. При нажатии на нее редактор перейдет в режим создания нового дома.

Кнопка "**Пользовательское значение адреса**" предназначена для записи адреса в свободной форме операторами биллинга, если необходимая улица не была найдена. По завершении редактирования адреса он попадает в специальный раздел [редактора адресов](#). После чего администратор или оператор, обладающий правами на редактирование адресов, заводит несуществующие адреса в справочник адресов.



При изменении адресного параметра строка адреса форматируется в соответствии с форматом, который можно выбрать на вкладке **Формат адреса** редактора адреса.

В некоторых случаях удобнее и правильнее использовать один формат, тогда как для другого случая данный вариант не подходит. Форматы отображения адреса настраиваются в конфигурации через меню **Сервис->Настройка->Конфигурация**.

Формат адреса настраивается с помощью параметра **addrs.format.pattern.<ID>**, где в качестве **<ID>** может выступать любой уникальный идентификатор: будь то строка или число. Есть также некоторые особенности: если в начале идентификатора указывается **cp** (ContractParameter - параметр договора типа Адрес) или **op** (ObjectParameter - параметр объекта типа Адрес), то после такого идентификатора необходимо указывать pid (из меню **Справочники->Другие->Договоры-параметры**) того адресного параметра, которому необходимо задать формат.

Примеры:

```

addr.format.pattern.1=({index}, )({city})({area})({quarter})({street})({house})({frac})({flat})({room})({pod})({floor})({comment})
addr.format.pattern.2=({index}, )({city})({street})({house})({frac})({flat})({comment})
addr.format.pattern.3=({city})({street})({house})({frac})({flat})({comment})
addr.format.pattern.cp19=({city})({index})({street})({house})({frac})({flat})
addr.format.pattern.op6=({street})({house})({frac})({flat})({room})
addr.format.pattern.cp58=({street})({house})({frac})({flat})

```

Формат адреса - это REGEXP, в котором прописаны определенные переменные. В качестве переменных можно использовать: **\${index}** - индекс; **\${city}** - город; **\${area}** - район; **\${quarter}** - квартал; **\${street}** - улица; **\${house}** - дом; **\${frac}** - дробь; **\${flat}** - квартира; **\${room}** - комната; **\${pod}** - подъезд; **\${floor}** - этаж; **\${comment}** - комментарий параметра.

Для корректной работы форматов адресов необходимо указывать в конфигурации параметр **addr.format.list**, в котором прописать порядок следования форматов друг за другом. В соответствии с этим списком будет заполнена таблица **Формат адреса**. Соответственно, некоторые форматы можно и не включать в список - тогда они не будут отображаться в таблице. Пример:

```
addr.format.list=1;cp19;3;cp58;op6
```

Как видно из примера, формат с идентификатором 2 не включен в список. Помимо настраиваемых форматов в таблице **Формат адреса** на первой строке всегда будет формат по умолчанию:

Редактор

Дом: Очистить поле

Квартала

Номер:

Комната:

Подъезд:

Этаж:

Х	Адрес	ID	...
<input checked="" type="radio"/>	г. Уфа, Ленина, д. 32, дробь /1, 6 подъезд, квартира 6	0	▲
<input type="radio"/>	г. Уфа, Ленина, д. 32/1, 6	1	▼

Если параметр **addr.format.list** не указывается в конфигурации, то в конфигурации ищется параметр **addr.format**, если и он не будет найден, то в таблице будет только 1 формат по умолчанию. Примеры с использованием параметра **addr.format**:

```

addr.format=({city})({street})({house})({frac})({flat})
addr.format.cp58=({city})({street})({house})({frac})({flat})({comment})

```

При редактировании справочников **Города, Улицы, Дома, Районы, Кварталы** все адресные строки, ссылающиеся на изменённые значения переформатируются.

Это свойство можно использовать при изменении формата адреса в конфигурации. Простое пересохранение названия города в справочнике городов переформатирует все адреса, относящиеся к этому городу.

При установке параметра конфигурации сервера **address.unique.check=1** система будет выводить предупреждения если в других договорах есть аналогичный адрес.

Параметр типа "Список"

Для редактирования спискового параметра вызовите редактор двойным кликом по строке.

В редакторе можно выбрать одно из значений ранее заведенные в справочнике "[Договоры - значения списков](#)" или ввести свое значение.

Редактор

Значение справочника

-- не указано --
2
5555
Активен1
впуап
На ожидании пополнения
На отключение
Отключен

Пользовательское значение

Восстановить

Ок Отмена

Выбор **-- не указано --** очистит значение параметра.

Параметр типа "Мультисписок"

Для редактирования мультиспискового параметра вызовите редактор двойным кликом по строке.

В редакторе можно выбрать одно или более значений ранее заведенные в справочнике "[Договоры - значения списков](#)"

The screenshot shows a software window titled "Мультисписок" (Multiselect). At the top, there is a header bar with the text "Мультисписок" and a value "000" next to a dropdown arrow. Below this is a list box containing a single item "000" which is highlighted. Underneath the list box are three buttons: "Все" (All), "Сброс" (Reset), and "Инверт." (Invert). Below these buttons is a checkbox labeled "Пользовательские значения (через ";"):" (User-defined values (via ";")). At the bottom of the window, there are three buttons: "Восстановить" (Restore), "Ок" (OK), and "Отмена" (Cancel).

Также можно добавить через точку с запятой одно или несколько пользовательских значений - значения не из справочника.

Параметр типа "Телефон"

Редактировать параметр типа Телефон можно вызвав редактор двойным щелчком мыши по строке.

Мобильный телефон	7 (347) 256-56-56[Комментарий]; 8 965 656-56-55
Почтовый адрес	
Адрес для корреспонденции	
Обслуживание	

Редактор		Номер	Комментарий
X	<input type="text" value="7 (347) 256-56-56"/>		Комментарий
X	<input type="text" value="8 965 656-56-55"/>		

Параметр договора Телефон может содержать неограниченное кол-во номеров телефонов. Формат результирующей строки (отображается в списке параметров договоров) состоит из формата номера и комментария в квадратных скобках (если комментарий не пустой). Формат номера задается в конфигурации сервера (по желанию), либо используются значение по умолчанию - все цифры телефона без пробелов и иных символов.

Принцип формирования вывода - префикс - последовательность цифр, пробелов, тире и скобок. Произвольные цифры обозначаются символом 'X'. Все цифры идут до первого символа 'X'. Формат вывода номера телефона задается в конфигурации параметром **phones.formats**, в котором задаются через запятую возможные префиксы телефонов.

```
phones.formats=8 (917) xxxxxxxx,7 (347) 2xx-xx-xx,xxx-xx-xx
```

При вводе в поле телефона цифр, будет производиться поиск соответствующего префикса и вывод в соответствии с ним. Поиск производится сравнением первых введенных цифр и первых цифр из префикса. Например, в случае ввода цифр 89 формат вывода телефона будет отформатирован по первому префиксу, т.к. первый префикс подходит для телефонов, начинающихся с комбинаций цифр: 8917, 891, 89, 8. Поиск производится от частного к общему: например, в случае, когда пользователем будут введены 4 цифры 8917 будет производиться поиск префиксов, подходящих для комбинации цифр 8917, затем 891, после 89, потом 8 и в завершении префикс без цифр. Поиск префикса завершается при первом найденном префиксе.

Дополнение разрешенной длины номера телефона

В редакторе телефона разрешенная длина номера равна 11, но разрешенную длину можно дополнить своим значением. Для этого в конфигурации параметрами заведите параметр **phones.customLengthNumber**.

```
phones.customLengthNumber=12
```


С данным параметром в редакторе можно будет набрать номер состоящий из 11 и 12 цифр. Внимание! Максимальная длина номера, даже с этим параметром, не может превышать 14.

Префиксы номеров

В редакторе номеров телефонов в договоре можно подставлять префикс(сы) номеров, например вводя городской номер в будет предложен вариант дополнения номера кодом страны и кодом города.

Префиксы задаются в конфигурации сервер

префикс для всех параметров договора типа телефон


 client.gui.contract.parameter.phone.prefix=7347,73472,7917

префикс для параметра договора с кодом 10 типа телефон

client.gui.contract.parameter.phone.prefix.10=7347

По мере ввода номера в выпадающем списке будут доступны варианты номеров с префиксами (количество вариантов зависит от количества префиксов указанных в параметрах), выбрать нужный вариант можно с помощью стрелок вверх/вниз на клавиатуре, подтвердить выбор клавишей Enter, скрыть список клавишей Esc.

A

Редактор	
	Номер
<input type="checkbox"/>	1 234 5
<input type="checkbox"/>	7 (347) 123-45
<input type="checkbox"/>	7 (347) 212-34-5

История изменения параметра

Для просмотра истории параметров для данного договора необходимо щелкнуть по иконке с лупой (наличие знака "плюс" на иконке означает, что история для данного параметра в текущий момент времени ведется; отсутствие - не ведется) напротив интересующего параметра. В открывшемся окне отобразится история выбранного параметра. При необходимости ее можно очистить, нажав на кнопку **Очистить историю**.

Дата изменения	Установленное значение	Кто изменил
27.05.2009 16:50:05		Кирилл Березин
27.05.2009 16:56:17	значение1	Кирилл Березин
27.05.2009 16:56:23	значение2	Кирилл Березин
27.05.2009 16:56:26	значение3	Кирилл Березин

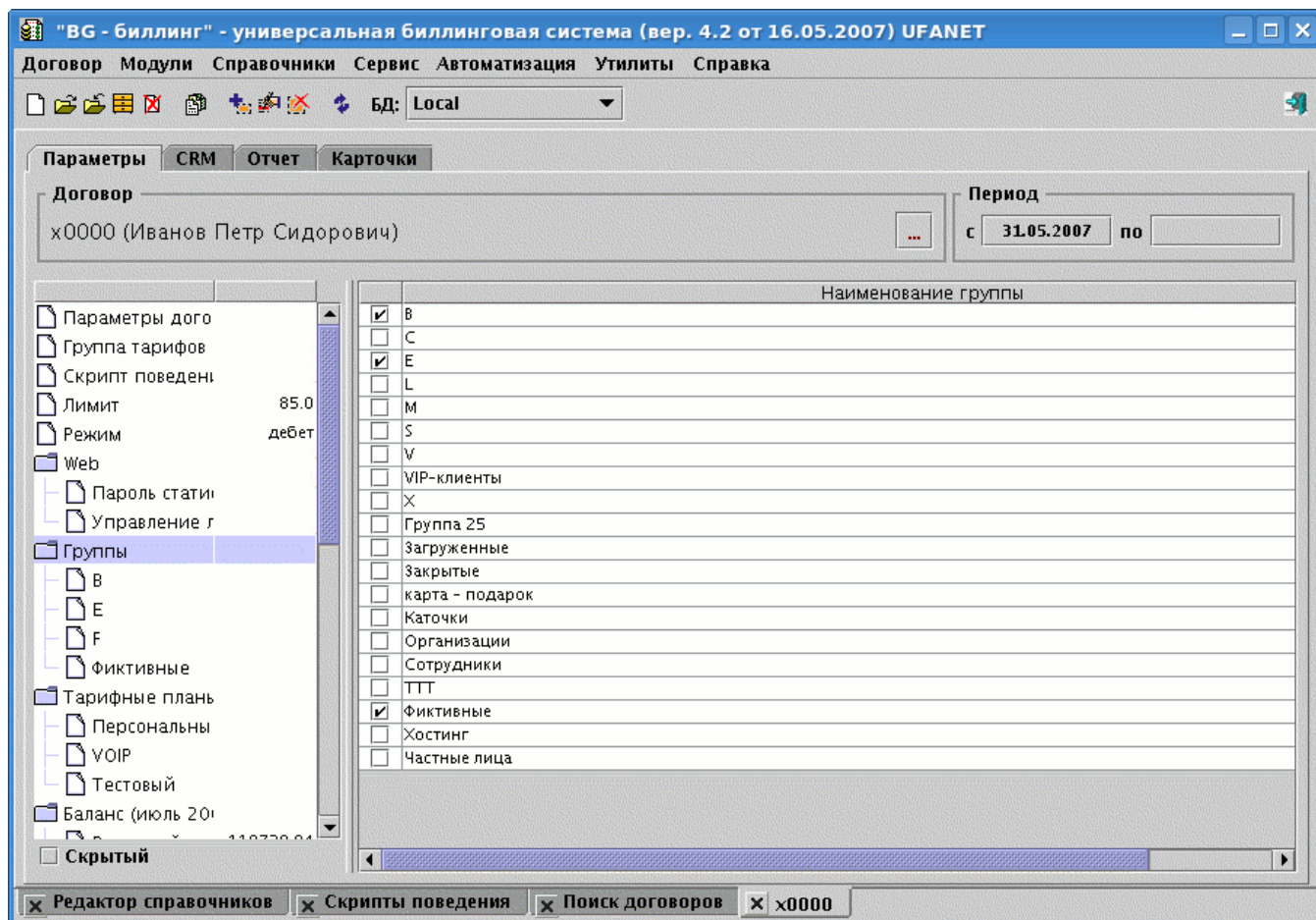
Если текстовый параметр содержит гиперссылку, то она выделится синим и подчеркнется. Редактировать её можно как обычный текстовый параметр. Чтобы "кликнуть" по ней, как по гиперссылке, нужно зажать клавишу Ctrl - ссылка откроется в браузере по умолчанию. Для всех типов параметров вызвать соответствующий редактор можно как двойным кликом мышью, так и клавишей **Enter**.

Группы договоров

Группа - признак, несущий смысловую нагрузку, установленный в договоре. Например: Оператор, Телефония, VIP. Группы устанавливаются битовой маской, в одном договоре могут быть установлены несколько групп. Всего групп 62. Именованье и активация групп производится в справочнике **Справочники=>Другие=>Группы договоров**, см. [ранее](#). Доступны к редактированию в договоре только группы, отмеченные как используемые и которые не помечены как не редактируемые.

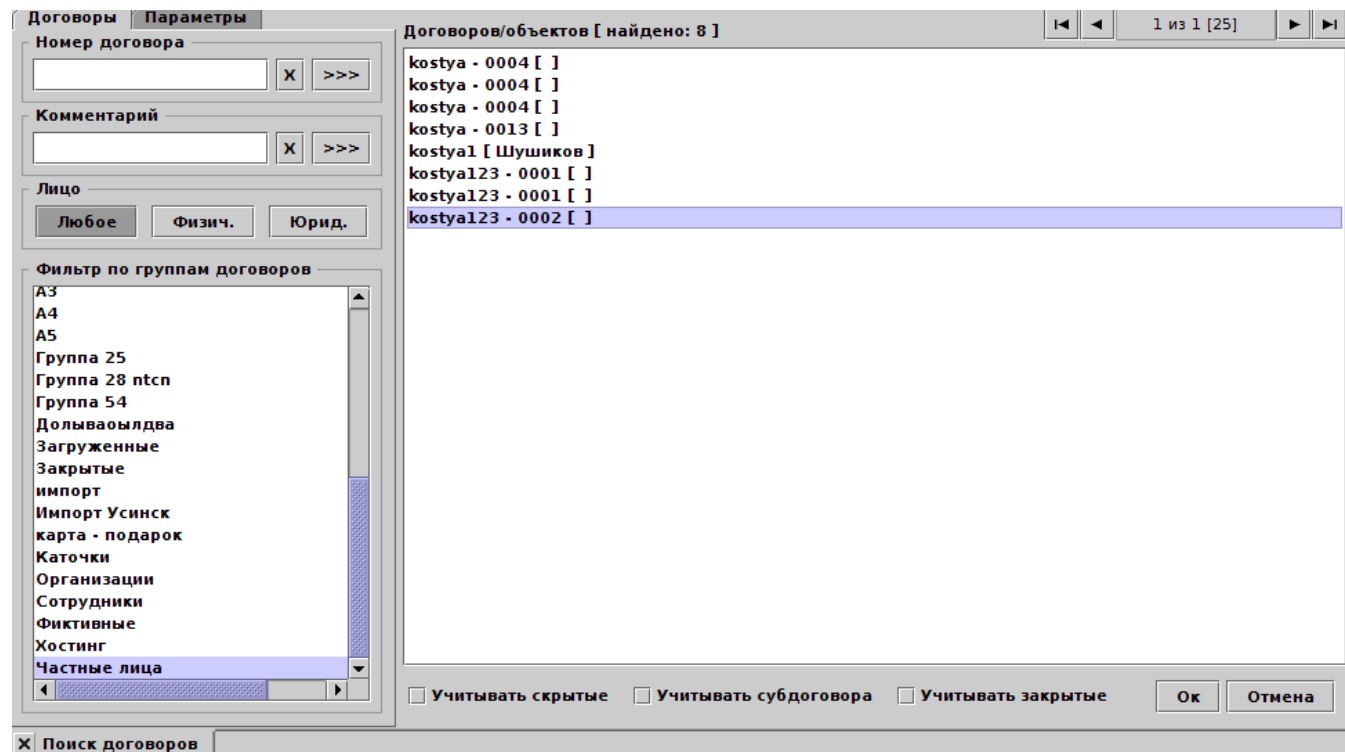
Редактирование групп договора осуществляется на вкладке **Группы** карточки договора простым проставлением, либо снятием галочек. После нажатия кнопки **Обновить** на общей панели инструментов выбранные группы отображаются в дереве.

Группы - основной инструмент деления договоров, доступный в качестве фильтра в большинстве мест, где требуется выбрать несколько договоров.



Поиск договоров

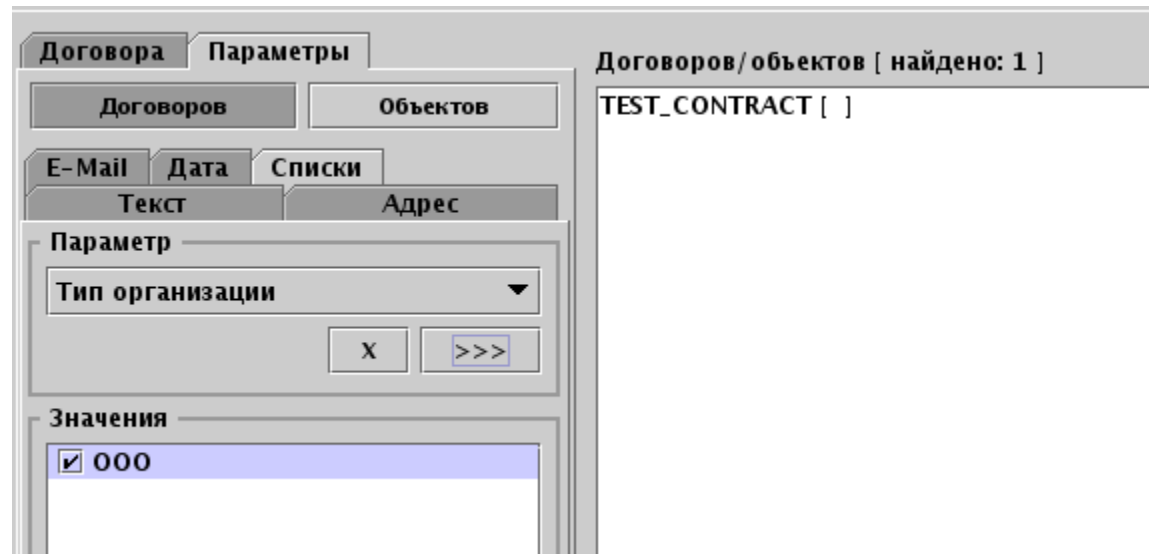
Поиск договоров осуществляется вызовом меню **Договор=>Открыть договор**, либо нажатием аналогичной кнопки на панели инструментов. После этого открывается окно поиска.



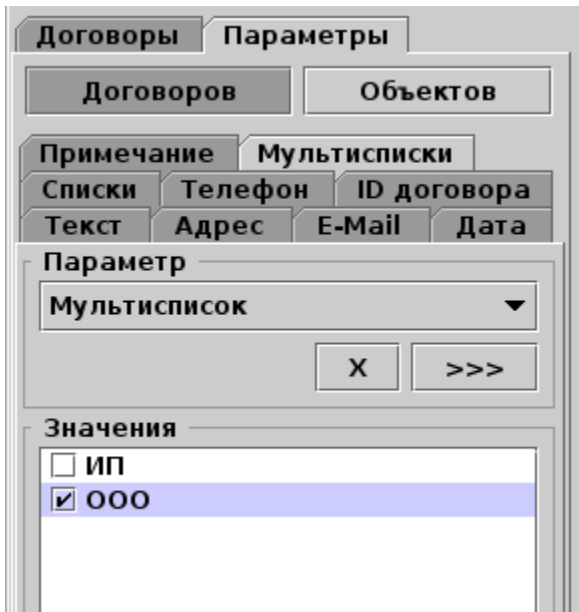
Поиск можно осуществлять по названию и комментарию с фильтром по группам и типу лица, либо по параметрам договора. При поиске по названию договора, либо комментарию следует набрать маску поиска и нажать >>> или клавишу **Enter**. Маска поиска - подстрока, входящая в искомое значение. Кнопка **X** - отмена поиска, сброс значения. >>> - запуск поиска.

Поиск по параметрам осуществляется выбором фильтруемых параметров и далее в зависимости от типа параметра.

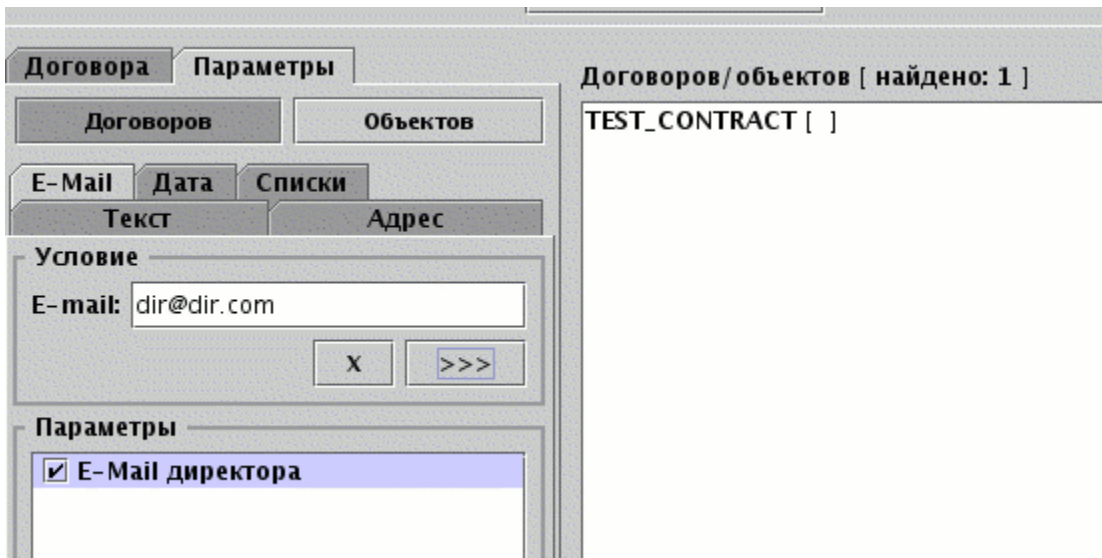
Для спискового параметра выбирается параметр и проставляются галочками, удовлетворяющие условию значения.



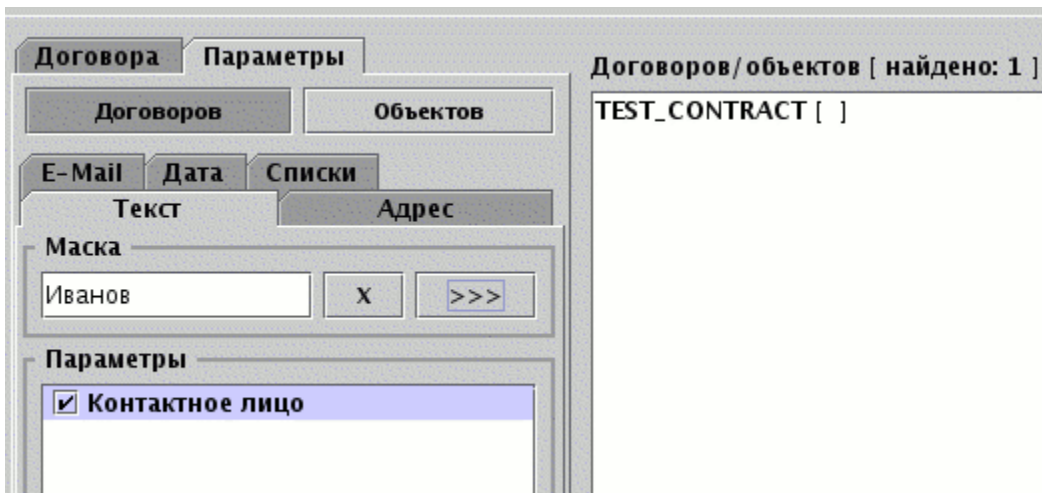
Для мультиспискового параметра выбирается параметр и галочками проставляются удовлетворяющие условию значения.



Для параметра типа E-Mail вводится подстрока адреса и галочками отмечается перечень параметров, в которых анализируется данная подстрока.



Для параметра типа Текст выбирается подстрока и перечень параметров, в которых она ищется.



Для параметра типа Адрес выбираются город, улица, номер дома (двор), квартира и комната, в которых будет произведён поиск. Для того, чтобы искать по городам, в поле **улица** нужно начинать вводить название города начиная с *****(. Например: *(Уфа).

Договор Модули Плагины Справочники Сервис Утилиты Справка

Заданий [просрочено]: 0 [50] +

Договоры Параметры

Договоров Объектов

Списки Телефон ID договора

Текст Адрес E-Mail Дата

Условие

Улица:

Дом [двор]:

Квартира: Комната:

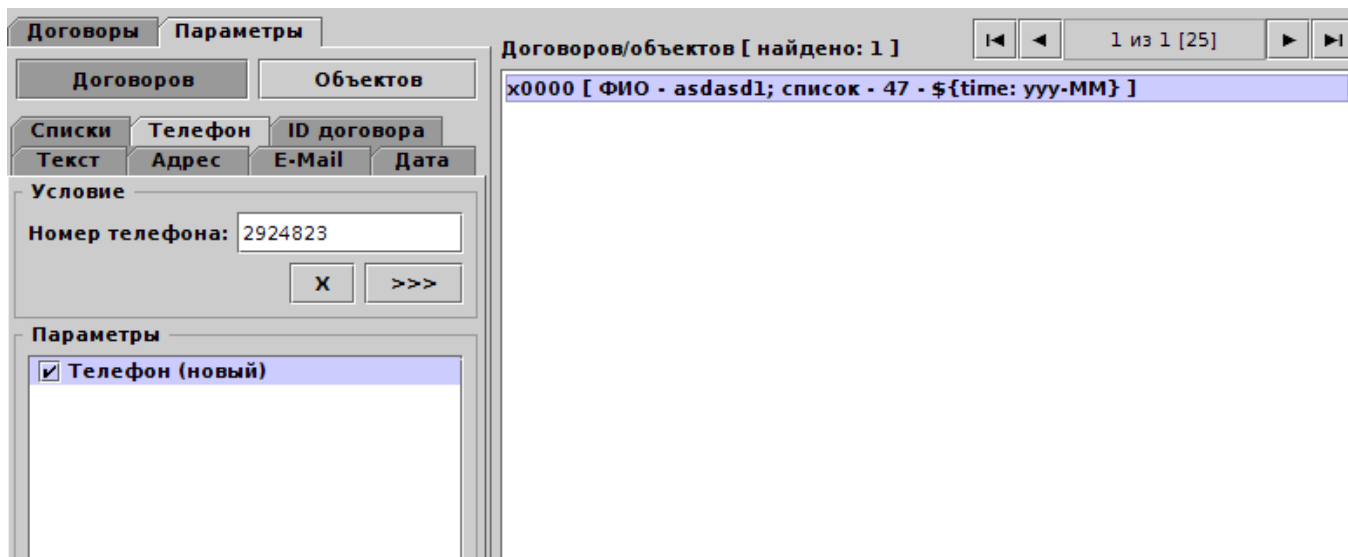
Найти

Параметры

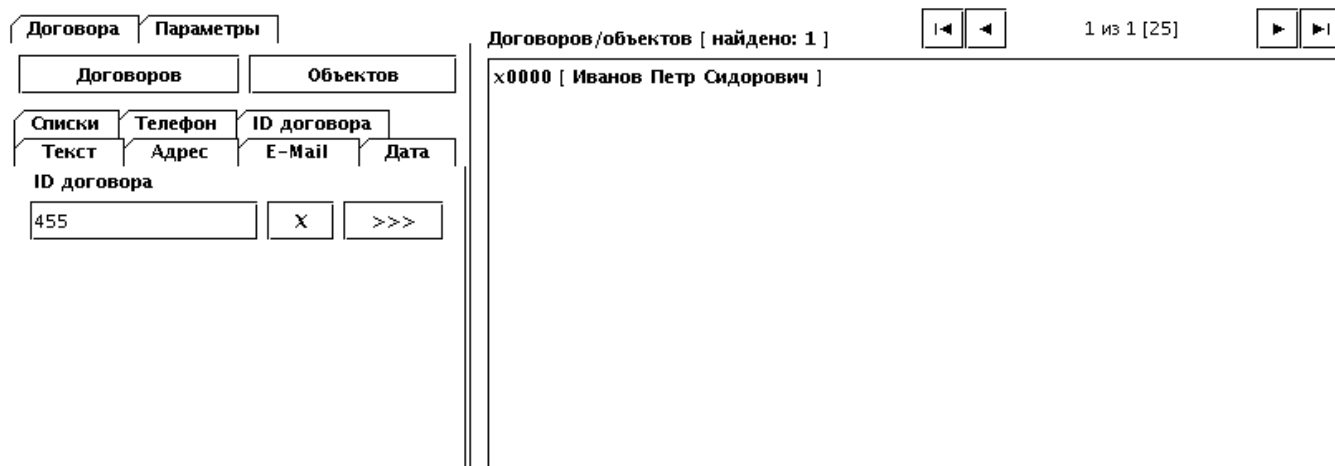
- Адрес (юридический)
- Адрес (прописки по паспорту)
- Адрес (почтовый 1)
- Адрес (почтовый 2)
- Адрес (офиса)

Кроме номера договора в окне результатов поиска отображается сам адрес.

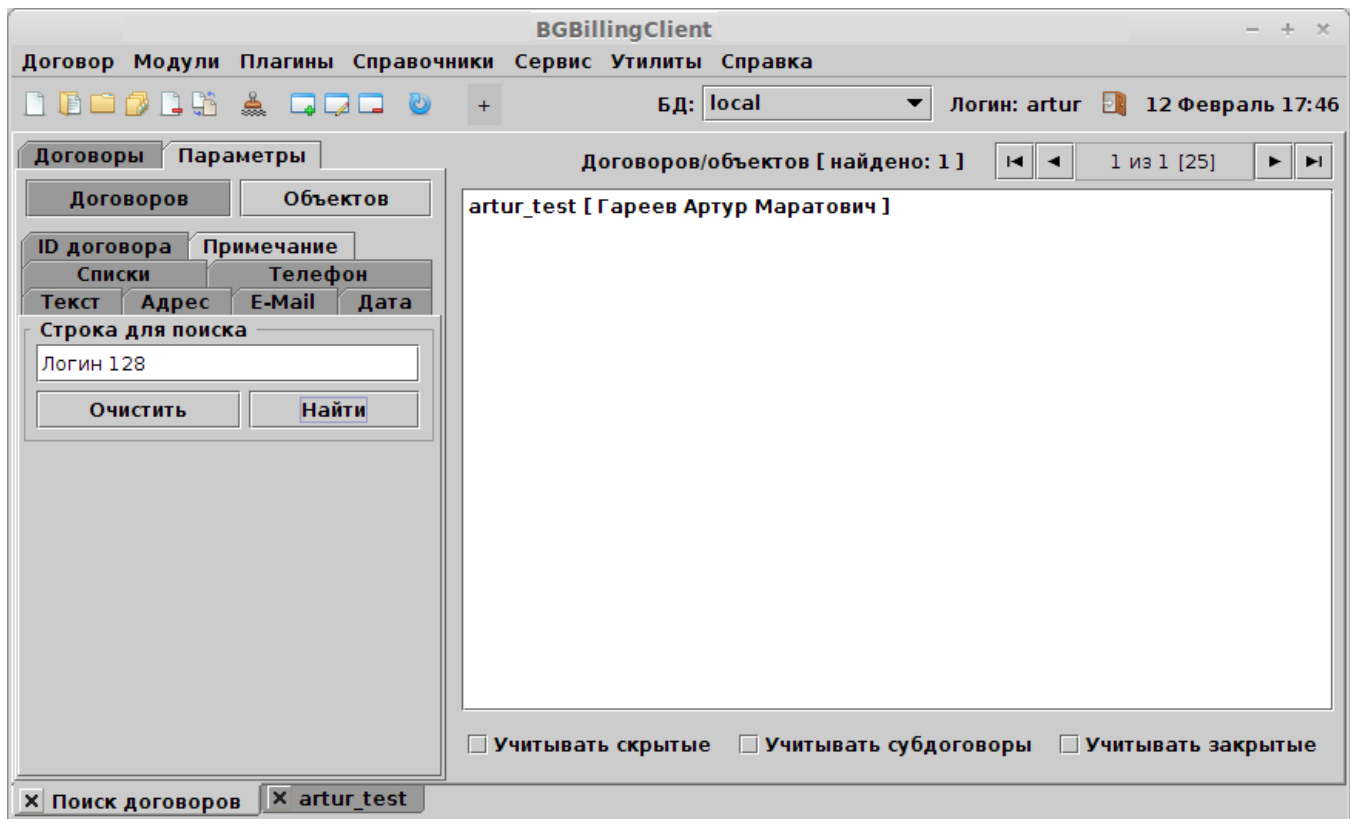
Для поиска по параметру Телефон необходимо указать номер телефона, а также сам параметр типа Телефон, по которому будет производиться поиск.



Для поиска по идентификатору договора (ID договора) необходимо указать интересующий ID и нажать на кнопку >>> или клавишу **Enter**.



Для поиска по примечанию необходимо написать часть текста, содержащегося в примечании договора и нажать на кнопку >>> или клавишу **Enter**.



Также доступны следующие опции:

Учитывать скрытые - позволяет включить в результаты поиска скрытые договоры.

Учитывать субдоговора - позволяет искать субдоговоры при поиске.

Учитывать закрытые - ищет договоры, у которых дата закрытия меньше текущей даты. По умолчанию такие договора не ищутся.

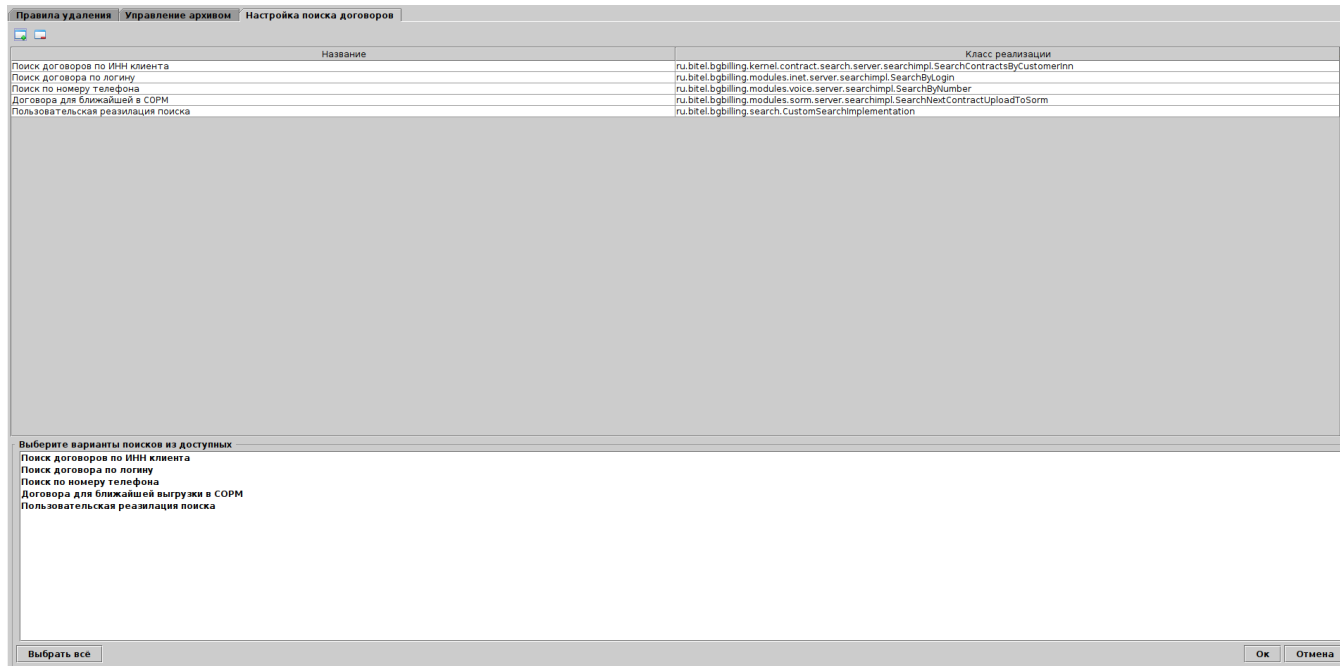
После получения списка договоров выберите интересующие вас и нажмите **OK**. Возможно открытие нескольких договоров, выделив их с помощью кнопок **Shift, Ctrl** и мыши. Открыть договор можно также двойным кликом мышью или клавишей **Enter**.

Создание собственных алгоритмов поиска договоров

Имеется возможность создания собственных поисков договоров с помощью динамического кода.

Так же существуют предустановленные алгоритмы поиска, которые можно включить в пункте меню: **Сервис => Настройка => Менеджер договоров**, далее на вкладке "Настройка поиска договоров".

(После создания собственных реализаций поиска, необходимо будет точно так же включить их)



После выбора одного или нескольких реализаций поиска, они станут активны на экране "Поиск договора". После начала ввода данных в поле "Номер договора(комментарий)", после каждого введенного символа строка с поисковым запросом вызывает метод проверки (check) у каждой реализации поиска.

Алгоритм реализации поиска договоров представляет из себя реализацию интерфейса **ru.bitel.bgbilling.kernel.contract.search.common.ContractSearchable**

ContractSearchable

```
public interface ContractSearchable
{
    //
    String COLOR_RED = "red";
    String COLOR_BLUE = "blue";
    String COLOR_GREEN = "green";

    //
    String CONTRACT_ID = "cid";
    String CONTRACT_TITLE = "contractTitle";
    String CONTRACT_BALANCE = "balance";
    String COMMENT = "comment";
    String COLOR = "color";
    String WIDTH = "width";

    /**
     *
     *
     * @param searchData
     * @return true -
     */
    public boolean check( String searchData );

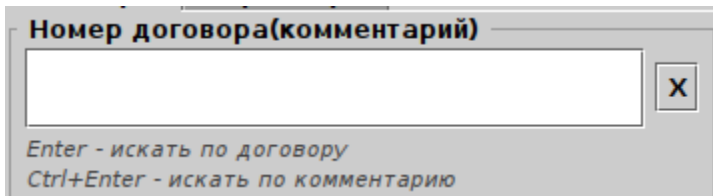
    /**
     *
     * @param searchData -
     * @param moduleId - id . 0,
     * @param filters -
     * @param pageHolder -
     *
     * @return JSON-
     */
    public List<JSONObject> search( String searchData,
                                   int moduleId,
                                   ContractSearchableFilters filters,
                                   Holder<Page> pageHolder )
        throws BGException;

    /**
     *
     *
     *
     * JSONObject      JSONObject      ContractSearchable.TITLES
     *      search()
     *
     * .. search {contractTitle:title}, metadata JSONArray ContractSearchable.TITLES,
     *      JSONObject {contractTitle: }
     * @return
     */
    public JSONObject metadata();

    /**
     *
     * , " IP-"
     * @return
     */
    public String getTitle();
}
```

Метод check()

Данный метод существует для проверки на предмет соответствия вводимых данных с логикой реализации поиска (например, в поле ИНН могут быть только цифры, метод check проверяет это условие и если указанные данные проходят данное условие, то реализация поиска появляется в выпадающем списке под полем для ввода номера договора)



Метод search()

Результирующий список содержит объекты типа JSONObject в котором могут находиться любые данные, но для того, чтобы договор мог быть открыт, каждый JSON должен содержать id договора под ключом cid (константа ContractSearchable.CONTRACT_ID).

Метод metadata()

Данный метод возвращает JSONObject с параметрами, которые могут помочь в корректности отображения результатов поиска.

Для каждой записи есть возможность указать цвет, которым будет отображена запись в результирующем списке или таблице.

Так же есть возможность указать читабельное название каждого поля. Для этого необходимо реализовать метод metadata(), пример его реализации показан ниже.

Исходя из порядка полей в titles будет формироваться очередность выстраивания столбцов или списка.

При выстраивании таблицы есть возможность задать ширину для определённого столбца, для этого в titles для нужного столбца необходимо указать ширину под ключом "WIDTH"(ContractSearchable.WIDTH).

Пример реализации метода getTitle

```
@Override
public JSONObject metadata()
{
    JSONObject metadata = new JSONObject();

    JSONArray titles = new JSONArray();
    titles.put( getTitleContractId( "ID " ) );
    titles.put( getTitleContractTitle( " " ) );
    titles.put( getTitle( SERVICE_LOGIN, " ", 120 ) );
    titles.put( getTitle( SERVICE_STATUS, " ", 50 ) );

    metadata.put( TITLES, metadata );

    return metadata;
}
```

Форматирование результатов поиска (в режиме список)

Для встроенных режимов поиска форматирование результата поиска можно задать в конфигурации сервера, по умолчанию используется следующий

```
client.gui.contract.search.list.layout=[ {text:"${contractTitle}",font:{name:"Monospaced",size:16,style:1},
anchor:10,hAlign:0,insets:{l:5,b:2,r:5,t:2}}, {text:"${comment}",fill:2,weightx:1.0,fColor:6316128,font:{size:14,
style:2},insets:{l:20,b:0,r:20,t:0}}, {text:"ID# ${cid}",fColor:10526880,width:100,anchor:17}]
```

Для пользовательских реализаций форматирование передается через **metadata**

```

JSONObject cell1Font = new JSONObject();
cell1Font.put( "name", "Monospaced" );
cell1Font.put( "size", 16 );
cell1Font.put( "style", 1 ); //

JSONObject cell1Insets = new JSONObject();
cell1Insets.put( "l", 5 ); //
cell1Insets.put( "b", 2 ); //
cell1Insets.put( "r", 5 ); //
cell1Insets.put( "t", 2 ); //

JSONObject cell1 = new JSONObject();
cell1.put( "text", "${contractTitle}" );
cell1.put( "font", cell1Font );
cell1.put( "hAlign", 0 ); //
cell1.put( "anchor", 10 ); //
cell1.put( "insets", cell1Insets );

JSONArray grid = new JSONArray();
grid.put( cell1 );
grid.put( ... );

JSONObject listLayout = new JSONObject();
listLayout.put( "grid", grid );

metadata.put( "listLayout", listLayout );

```

Форматирование задается в виде объектов **JLabel** размещенных на **JPanel** с **GridBagLayout**

для каждого **JLabel** можно указать

text - выводимый текст, возможны вариант, просто текст, пеле из результатов поиска **`\${contractTitle}`** или комбинация текст + поле(я)

font - шрифт с параметрами, **name** название шрифта или семейства, **size** размер шрифта, **style** стиль шрифта (жирный, курсив, обычный или их комбинация)

fColor - цвет текста

hAlign - горизонтальное выравнивание текста в **JLabel**

width - ширина **JLabel**

и задать свойства **GridBagConstraints** задающего размещение **JLabel** на панели

gridx, gridy - столбец и строка

gridwidth, gridheight - количество ячеек занимаемых по ширине и высоте

weightx, weighty - растяжение по ширине и высоте внутри ячейки

anchor - притяжение **JLabel** к сторонам ячейки

fill - заполнение ячейки

insets - отступы **t**(сверху), **l**(слева), **b**(снизу), **r**(справа)

Баланс

Вкладка позволяет отследить движение денег на балансе пользователя. Баланс в системе ведётся помесечно, каждый месяц характеризуется:

1. **входящим остатком** - остатком денежных средств с конца предыдущего месяца;
2. **наработкой** - наработкой по различным услугам, которые потреблял клиент, наработку начисляют модули;
3. **приходом** - поступления денежных средств на баланс пользователя в течении месяца;
4. **расходом** - списанием денежных средств за разовые услуги со счета клиента;
5. **возвратом** - списанием денежных средств, которые были возвращены клиенту;
6. **исходящим остатком** - вычисляемое значение $\text{Исх.ост.} = \text{Вх.ост.} - \text{Расход} + \text{Приход} - \text{Наработка} - \text{Возврат}$, исходящий остаток переходит на входящий остаток следующего месяца.
7. **резервом** - временное списание денежных средств, если включено влияние резервов на баланс.
8. **доступная сумма** - идентично исходящему остатку, но включает еще и резервы, и, соответственно, если у вас включено влияние резервов, то данная сумма будет отлична от исходящего остатка на текущую величину резервов у клиента.

Для того, чтобы включить влияние резервов на баланс договора, необходимо добавить параметр в конфигурактор сервера: **contract.balance.reserve.influenceToBalance=true**. А после перезапустить сервер. Если не включать, то резервы могут нести некий информационный характер.

В дереве вкладок (слева) отображается баланс за тот месяц, за который было зарегистрировано последнее движение по балансу (в целях оптимизации размеров БД). Для просмотра баланса договора за месяц нажмите кнопку **Баланс**.

Баланс отображается в табличном виде в формате : "Месяц, год - Входящий остаток - Приход - Нарботка - Расход - Исходящий остаток".

В верхней области можно выбрать текущий период с точностью до месяца.

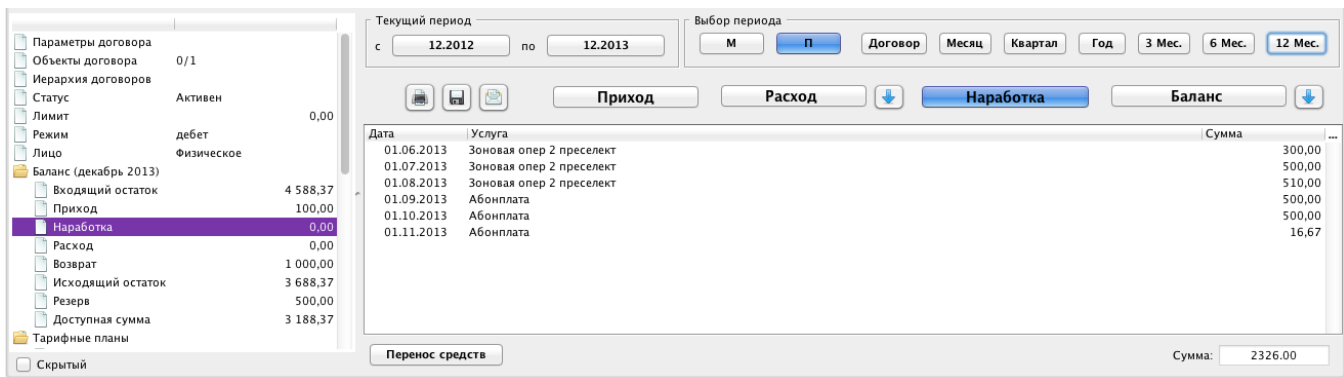
Режим, выбираемый кнопкой **М**, просматривается баланс выбранного месяца. Режим **П** позволяет просмотреть баланс за определенный период путем нажатия на одну из кнопок:

- **"Договор"** - весь период действия договора;
- **"Месяц"** - период, равный текущему месяцу ;
- **"Квартал"** - период, равный от начала текущего квартала до текущего месяца;
- **"Год"** - период, равный от начала текущего года до текущего месяца;
- **"3 Мес."** - период, равный последним 3-м месяцам;
- **"6 Мес"** - период, равный последним 6-м месяцам;
- **"12 Мес"** - период, равный последним 12-м месяцам.

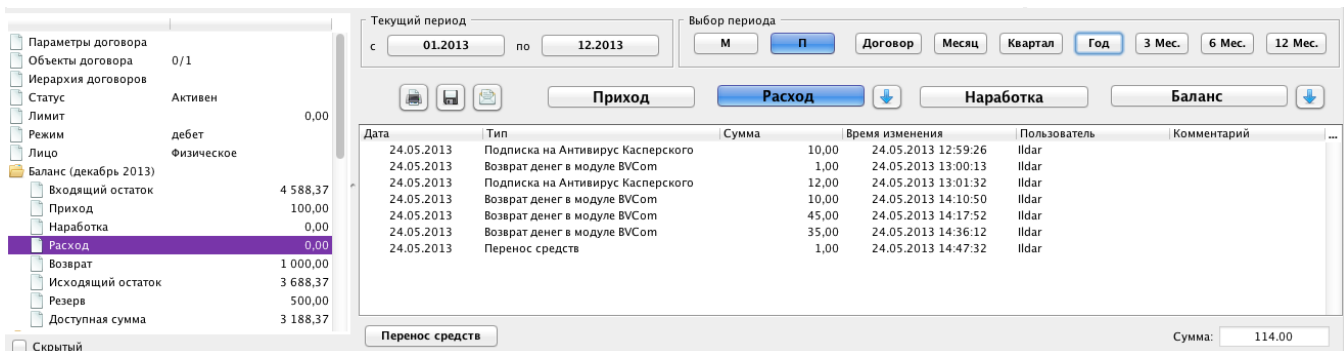
Дата	Входящий остаток	Приход	Наработка	Расход	Исходящий остаток
декабрь 2013	4588.37	100.00	0.00	1000.00	3688.37

Дата	Входящий остаток	Приход	Наработка	Расход	Исходящий остаток
июнь 2013	-115.96	1.00	300.00	0.00	-414.96
июль 2013	-414.96	0.00	500.00	0.00	-914.96
август 2013	-914.96	530.00	510.00	0.00	-894.96
сентябрь 2013	-894.96	0.00	500.00	0.00	-1394.96
октябрь 2013	-1394.96	0.00	500.00	0.00	-1894.96
ноябрь 2013	-1894.96	6500.00	16.67	0.00	4588.37
декабрь 2013	4588.37	100.00	0.00	1000.00	3688.37

При выборе **Наработки** отображается наработка за месяц, разбитая по услугам. Нарботка не редактируется через клиентское приложение, её начисляют модули за различные услуги, потреблённые клиентом.

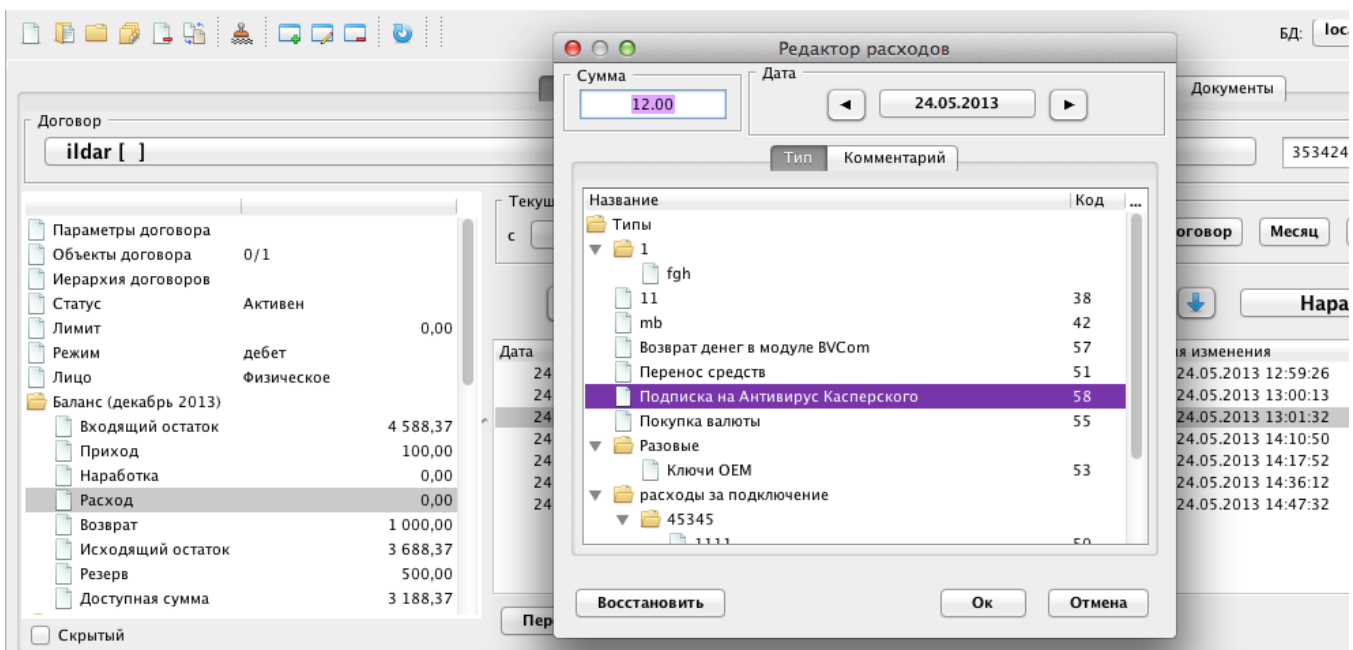


Кнопки **Приход** и **Расход** отображают платежи и списания по договору за месяц. В отличие от наработки, которая ведётся помесячно, платежи и расходы ведутся с указанием даты.



Для добавления платежа нажмите в режиме просмотра платежей кнопку **Новый элемент** на стандартной панели инструментов клиента биллинга. Аналогично вносится расход. При внесении платежей/расходов в списке типов платежей/расходов отображаются только редактируемые типы. Необходимо выбрать сумму, тип платежа/расхода и дату. После внесения платежа/расхода система автоматически обновляет баланс от месяца внесения платежа/расхода до текущего с корректировкой переходящих остатков. При этом за каждый месяц пересчитываются суммарные наработки, приходы и расходы.

Внесение и удаление фиктивного платежа или расхода можно использовать для корректировки сбоев в балансе (когда сумма, отображаемая на вкладке **Баланс**, не соответствует суммам платежей/расходов или наработки)



Для редактирования платежа/расхода нужно дважды кликнуть по нему в таблице, при этом будет вызван редактор, аналогичный редактору внесения платежа/расхода. Платежи/расходы нередатируемых типов править нельзя.

Режим **Баланс дет.** отображает сводную таблицу баланса договора за период с остатками, приходами, расходами и наработкой.

Параметры договора
Объекты договора: 0/1
Иерархия договоров
Статус: Активен
Лимит: 0,00
Режим: дебет
Лицо: Физическое
Баланс (декабрь 2013)
Входящий остаток: 4 588,37
Приход: 100,00
Наработка: 0,00
Расход: 0,00
Возврат: 1 000,00
Исходящий остаток: 3 688,37
Резерв: 500,00
Доступная сумма: 3 188,37

Скрытый

Текущий период: с 10.2013 по 12.2013
Выбор периода: М П Договор Месяц Квартал Год 3 Мес. 6 Мес. 12 Мес.

Приход Расход Наработка **Баланс дет.**

Дата	Договор	Тип	Комментарий	Сумма	Остаток
01.10.2013		Входящий остаток на начало месяца			-1 394,96
31.10.2013	ildar	Наработка "Абонплата"		-500,00	-1 894,96
21.11.2013	ildar	Приход "Виртуальный"		6 000,00	4 105,04
21.11.2013	ildar	Приход "Виртуальный"		500,00	4 605,04
30.11.2013	ildar	Наработка "Абонплата"		-16,67	4 588,37
02.12.2013	ildar	Приход "Банк (Социнвест-агро банк)"		100,00	4 688,37
04.12.2013	ildar	Возврат "По судебному решению"		-1 000,00	3 688,37
01.12.2013		Исходящий остаток на конец месяца			3 688,37

Перенос средств

Сумма: 3688.37

Лимит договора, режимы договора, управление лимитом

Лимит - это минимальный остаток счета договора, при котором он может еще работать. Для редактирования лимита перейдите на вкладку, изображенную на снимке ниже.

Для смены лимита можно использовать как кнопки с предопределенными значениями, так и произвольное значение, указанное в текстовой области. Каждая смена лимита отображается в логе. Есть возможность временного и постоянного изменения лимита.

Временное понижение лимита может использоваться, например, чтобы пользователь смог пополнить счет карточкой. В первом поле вводится значение, на которое изменяется лимит (положительное - для повышения, отрицательное - для понижения), а во втором - на сколько дней. При временном понижении лимита в таблице **Автоматическое изменение лимита** добавляется задание на возврат лимита на измененную сумму (см. снимок ниже). Задание может быть удалено, тогда лимит не будет возвращен в предыдущее состояние.

Возвращение лимита к прежнему значению осуществляется **задачей восстановления лимитов** которую нужно прописать в планировщике.

Если лимит нужно просто установить, второе поле ввода оставляется пустым. В текстовой области перед кнопкой **Ок** можно ввести комментарий, объясняющий изменение лимита.

Если при установке постоянного лимита на текущий момент у договора присутствуют задания на автоматическое изменение лимита, то пользователь будет уведомлен об этом в дополнительном диалоговом окне. Ему будет предложено подтвердить свое решение или же отказаться от него.

В **конфигурации сервера** возможен запрет на установление постоянного лимита в таких случаях, так как это может привести к нежелательным последствиям (например, к ненужному завышению или занижению реального лимита, которое может произойти после отработки задачи восстановления прежде измененных лимитов).

Договор Модули Справочники Сервис Автоматизация Утилиты Справка

Параметры Отчет Карточки HelpDesk CRM Документы

Договор NK00001-03 (20) Период с 01.04.2006 по

Лимит: -100.0 / 3 Обещал оплатить [Ок]

Сортировка (дата) ↓ ↑ Страница 1 из 1 [25]

--1000	--500	--300	--150	--100	--50	--10
= 0	-5/1	-30/1	-100/1	-15/3	-50/3	-100/3

Дата	Пользователь	Лимит	Дней	Комментарий
22.02.2008 01:35:21	Шамиль	-100.0	3	Обещал оплатить

Автоматическое изменение лимита:

Дата	Пользователь	Изменение лимита на
25.02.2008	Шамиль	100.00

Скрытый

Планировщик заданий Поиск договоров x X0000 x K000001 x K000001 x K000010-08 x K03-000010-08 x NK00001-03

Клиенту можно предоставить возможность **самому временно понижать лимит** через страницу Web-статистики (только в режиме **Дебет**).

Режимы баланса договора

В системе существуют два режима работы договора: **дебет** и **кредит**. Их отличие - в способе работы с должниками.

При режиме **дебет** должником признается договор, остаток которого менее текущего лимита счета. Все сервисы должника блокируются, а разблокировка производится как только на баланс договора поступит платеж, достаточный для превышения остатка над лимитом. Данный режим используется для предоставления сервиса по предоплате. Лимит может быть незначительно меньше нуля, либо равен нулю.

Режим **кредит** предназначен для организаций, работающих по кредитовой схеме оплаты. Должником признается договор, сумма платежей которого за текущий месяц не покрывает отрицательный остаток на начало месяца. При этом наработка и расходы за текущий месяц не учитываются. Лимит также является минимально допустимым остатком счета, но носит скорее функцию аварийного блокиратора, предотвращая сильную переработку клиента, например, при вирусной эпидемии. Для договоров в режиме кредит лимит устанавливается отрицательным, равным максимально приемлемой для клиента задолженности.

Отключение должников в кредитовом режиме производится оператором биллинга после выставления счетов и ожидания оплаты в течении нескольких дней. Для отключения используется [система работы с должниками](#).

Статус договора

Статус - это глобальная характеристика договора, общая для всех подключенных услуг. Перечень статусов в системе и их коды задаются переменной **contract.status.list** конфигурации сервера, например, так:

```
contract.status.list=0;;1:;2;;3:;4:;5:
```

Код статуса должен быть уникален. В зависимости от статуса договора различные модули меняют свое поведение. Открывается, либо блокируется доступ к услугам, начисляется, либо не начисляется абонентская плата.

Текущий статус договора отображается отдельным узлом в дереве карточки договора в виде: **Статус <значение>**. При выборе узла дерева в таблице отображаются статусы договора.

The screenshot shows the VCBillingClient application window. The main window title is "BCBillingClient (вер. 5.2 сборка 518 от 20.05.2011 18:10:11) ООО "Бител" (идент. номер: 1494)". The interface includes a menu bar with "Договор", "Модули", "Плагины", "Справочники", "Сервис", "Утилиты", and "Справка". Below the menu is a toolbar with icons for file operations and a search field. The main content area is divided into several sections:

- Параметры:** Includes fields for "Договор" (x0000 [Иванов Петр Сидорович]), "Период" (с 06.02.2005 по), and "Статус" (Активен).
- С даты:** A dropdown menu currently set to "Активен".
- Комментарий:** A text input field.
- История изменения статусов:** A table with columns: Период, Дата, Статус, Пользователь.

Период	Статус	Комментарий
24.08.2008-23.03.2009	Активен	
01.10.2009-31.01.2010	Активен	
01.02.2010-28.02.2010	Приостановлен	fffff
01.03.2010-05.05.2010	Активен	
06.05.2010-01.12.2010	Активен	
02.12.2010-10.05.2011	Активен	Изменено пользователем (отмена смены статуса)
11.05.2011-11.05.2011	Приостановлен	Изменено пользователем
12.05.2011-12.05.2011	Активен	Изменено пользователем
13.05.2011-...	Активен	

Период	Дата	Статус	Пользователь
24.08.2008-...	24.02.2009 16:56:15	Активен	Denis Pimenov
24.03.2009-...	24.02.2009 16:56:32	Активен	Denis Pimenov
24.03.2009-...	24.02.2009 16:56:39	Отключен	Denis Pimenov
08.09.2009-...	08.09.2009 18:27:08	Активен	??? (455)
08.09.2009-...	08.09.2009 18:29:19	Активен	??? (455)
08.09.2009-...	08.09.2009 18:33:28	Активен	??? (455)

Внизу отображается таблица с историей изменения статусов.

В этом же окне возможно изменение статусов для договоров с указанием периодов и комментария. При смене статуса оператор указывает период и новый статус. В комментарии может быть указана причина изменения. Например: "По гарантийному письму XXXX". Период может быть указан с пустой датой окончания, в этом случае период считается не ограниченным сверху.

Вновь установленный статус перетирает все статусы с пересекающимися периодами. В один день возможен только один статус договора.

Не все статусы доступны для ручной установки, перечень запрещённых задаётся переменной конфигурации сервера **contract.status.no.manual.set**. Данные статусы могут устанавливаться, например, скриптами.

Если какой-либо статус более не используется, то его код должен быть указан в переменной конфигурации сервера **contract.status.deprecated**, коды указываются через запятую. Статус останется в истории смен, но будет недоступен в фильтрации и установке.

Если устанавливаемый статус договора включает текущую дату, то действующий статус договора меняется сразу. Однако возможна ситуация, когда меняется текущий статус и остается задание на изменение статуса договора в дальнейшем. Такие задания выполняет стандартная задача планировщика [Установка статусов договоров](#).

При смене статуса супердоговора изменяются статусы его зависимых [субдоговоров](#). Включить смену статусов независимых договоров можно опцией конфигурации сервера **independ.subcontract.status.change=1**.

Настройка активных статусов в модулях

Статусы экземпляра модуля, в которых сервис доступен, указываются в переменной конфигурации модуля **contract.status.active.codes**. Коды статусов указываются через запятую. Например.


```
contract.status.active.codes=0
```

Статусы, в которых сервис считается приостановленным указываются в переменной **contract.status.suspend.codes**. Например.

```
contract.status.suspend.codes=3,4
```

Для модуля NРау данная переменная означает статусы, при которых не снимается абонплата. Для модулей с потреблением услуг влияет, например, на узлы-диапазоны в тарифных планах. Значение в узлах уменьшается в зависимости от времени, которое сервис был приостановлен. Более подробно это описано в документации модулей.

Роль статуса в дебетовых договорах

Для договоров в режиме **Дебет** статусы договора устанавливаются только администратором, перевод статуса договора в любой из неактивных делает модули заблокированными. Поступление денег на счет ситуацию не меняет. При нормальной работе дебетовый договор постоянно находится в активных статусах, сервисы при этом блокируются модулями, если баланс опускается меньше лимита и автоматически разблокируются после поступления денег на счет.

Роль статуса в кредитовых договорах

Для кредитовых договоров статус тесно завязан с системой работы с кредитовыми должниками. Он может изменяться в результате оплаты договора. Активный статус для кредитовых договоров задаётся переменной конфигурации сервера **credit.contract.active.status**. Например.

```
credit.contract.active.status=0
```

В активном статусе работает аналогичная дебетовым договорам аварийная блокировка/разблокировка сервисов в модулях по балансу и лимиту договора.

Основная рабочая область оператора для отключения кредитовых должников вкладка **Сервис=>Администрирование=>Монитор статуса**. Перед началом работы необходимо сделать срез балансов, нажав соответствующую кнопку. Все описанные далее алгоритмы применимы только к кредитовым договорам.

Монитор выводит информацию по состоянию кредитовых договоров, их статусам и сальдо (входящий остаток на начало месяца плюс платежи за месяц). Таблицу с результатами выборки договоров можно выгрузить в csv формате.

Все фильтры соединяются по условию **И**. Возможна фильтрация по:

1. группе договора + исключение определенных групп договоров;
2. минимальному объему наработки по указанным услугам за прошлый и текущий месяц;
3. режиму договора: кредит/дебет;
4. статусу договора;
5. периоду, который договор был в данном состоянии в днях или месяцах;
6. сумме сальдо (указывается диапазон от и до);
7. соотношению текущего баланса или баланса на начало месяца с лимитом;
8. размеру превышения модуля отрицательного сальдо начала месяца над наработкой какого-то количества предыдущих месяцев.

Изменения статусов договоров производится идентично смене статуса одного договора (см. выше), необходимые договоры выбираются в таблице с использованием клавиш Ctrl и Shift. Двойной клик по строке таблицы открывает договор.

В определенный момент времени оператор выбирает все договоры с отрицательным сальдо, находящиеся в активном статусе и, выбрав строки таблицы, переводит их в отключенное состояние. Отключенных статусов может быть несколько, их состав и назначение настраивается администратором биллинга.

В случае, если отключенный статус попадает в перечень из переменной конфигурации **credit.contract.open.by.payment.status**, договор может быть возвращён в активный статус по приходу платежа, если сальдо станет положительным.

По звонку отключенного клиента с обещанием оплаты оператор может перевести его в активный статус на несколько дней. При этом период активного статуса указывается с датой открытия и закрытия. Если до окончания периода клиент не оплатит, отключенный статус станет действующим и клиента заблокирует. В случае оплаты вновь установленный активный статус перекроет будущий отключенный. При этом перекрытие производится только для статусов, которые указаны в переменной конфигурации **credit.contract.override.future.to.active.status**.

При неоплате клиента в течении какого-то времени возможна его выборка монитором статуса и перевод в другой отключенный статус, из которого уже не выводит платёж.

Если по каким-либо причинам такое поведение статусов нежелательно, то можно отключить автоматическую активацию договора при изменении сальдо на положительное. Для этого в конфигурации сервера необходимо указать флаг **do.not.open.contract.on.payment=1**. Отсутствие этой записи или установка флага в 0 оставит поведение по умолчанию.

Если же возникла необходимость отключить данное поведение не для всех договоров, а только для каких-либо групп договоров, то необходимо установить флаг.

```
do.not.open.groups.on.payment=X,Y,Z,...
```

Где X, Y, и Z - это номера групп договоров, для которых стандартное поведение нежелательно.

Изменение стандартной логики перетирания статусов

Имеется возможность задать свою любую логику перетирания статусов при установке. Стандартная логика заключается в полном перекрытии лежащих уже в договоре отрезков статуса. Иногда требуется при разных условиях выполнять разные действия. Для этого предназначено событие BGBS **Задание логики перетирания статусов**. В скрипте (обработчике этого события) можно отслеживать откуда выполняется попытка установить статус (сервер, клиент, web) и полностью изменить алгоритм.

На Wiki-странице "[Изменение стандартной логики перетирания статусов](#)" приведены дополнительные подробности и прилагается полный рабочий скрипт, представляющий собой в точности реализованную стандартную логику. Его также можно использовать для изучения алгоритма. Там же расположено описание алгоритма установки статусов и примеры скриптов.

Возможность включается установкой в конфигурации сервера флага

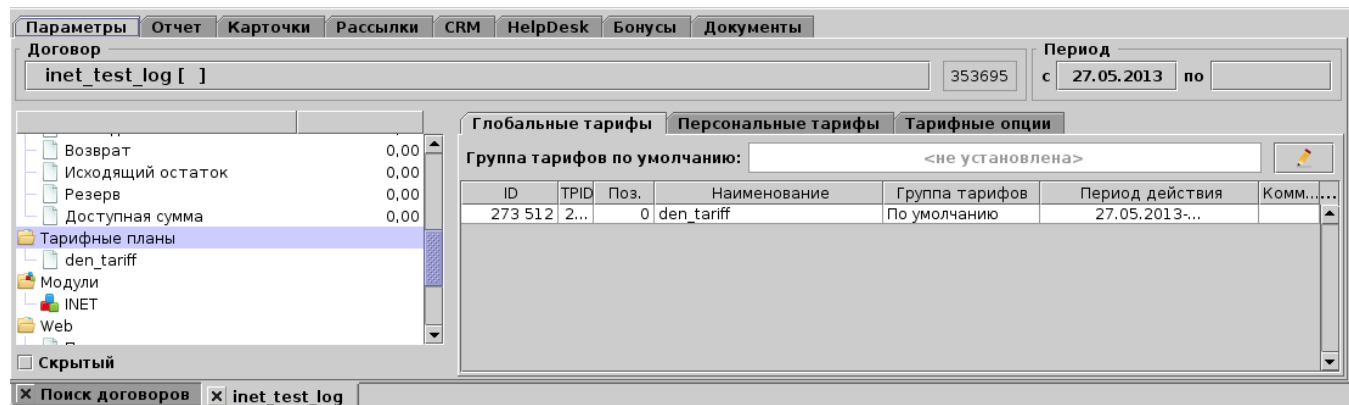
```
# / ( )  
use.event.set.status.logic=1
```

Тариф и группа тарифов

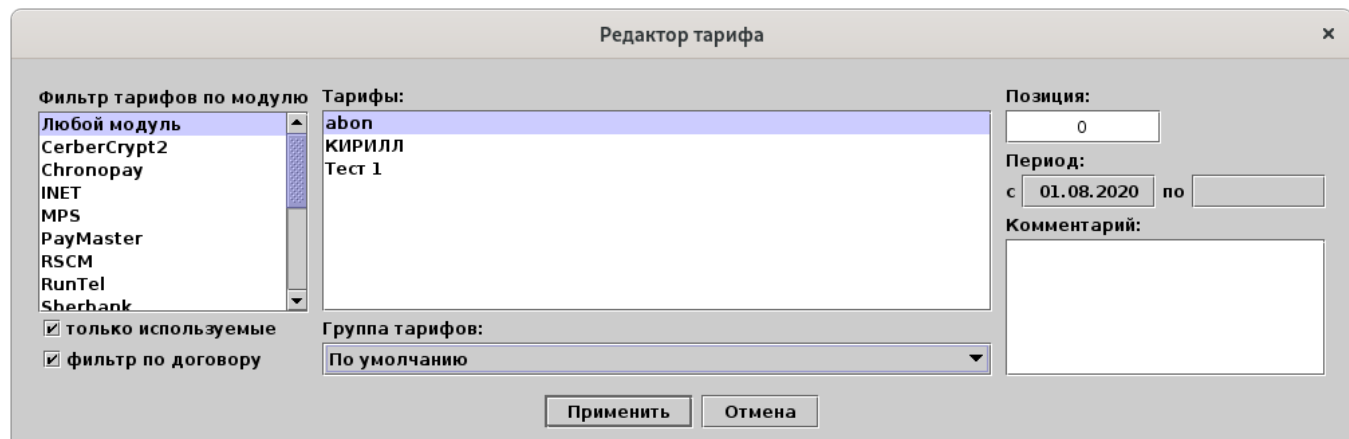
Тарифный план позволяет определять параметры тарификации для абонента. К параметрам тарификации можно отнести стоимость услуг, различные опции, параметры округления, зависимость тарификации от различных условий, которые могут отличаться у абонентов.

Для работы с тарифными планами в карточке договора предусмотрен отдельный раздел **Тарифные планы**. На договоре одновременно может действовать несколько разных тарифных планов. Их названия будут отображаться списком прямо в карточке договора. Выбрав пункт **Тарифные планы**, справа отобразится ряд вкладок для работы с тарифами договора. В частности, на вкладке **Глобальные тарифы** отображается список подключенных тарифов из [Справочника тарифов](#). Вкладка **Персональные тарифы** позволяет настроить индивидуальное тарифное дерево на договоре. Вкладка **Тарифные опции** позволяет управлять подключением/отключением **тарифных опций** договора. Вкладка **Группа тарифов** заведует управлением группами тарифов договора.

Поле **TRID** в таблице показывает id тарифного плана из справочника тарифов.



Для добавления глобального, либо персонального тарифного плана используйте кнопку **Новый элемент**.



Откроется редактор тарифного плана, в котором тарифы из справочника тарифов можно фильтровать по признаку используемости, по модулю, по договору, по группе тарифов. Галочка **Только используемые** позволяет отобразить только те тарифы из справочника тарифов, у которых установлена аналогичная галочка. Галочка **Фильтр по договору** позволяет вывести только те тарифы, параметрам которых соответствуют сам договор (группы договора, тип договора (юр.лицо/физ. лицо)). Галочка **Фильтр по группе тарифа** отображает только тот список тарифов, которые добавлены в соответствующие группы тарифов, которые подключены в договоре.

В один момент у договора может быть несколько тарифов, в каждом из которых устанавливаются цены услуг в разных модулях.

Позиция тарифного плана - это число, задающее порядок просмотра тарифа при поиске цены. Позиция действует лишь для некоторых модулей. Более подробно о [позициях](#) и тарифных планах [вообще](#) можно прочитать в соответствующих разделах.

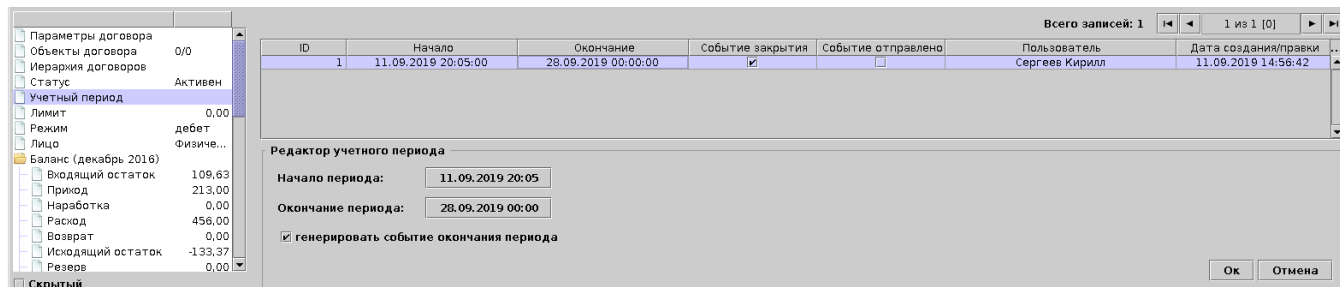
Группа тарифов нужна исключительно для ограничения тех тарифов, на которые клиент может переключиться через Web-интерфейс. Если для тарифа выставлена Группа тарифов **<не установлена>** или на тарифе выбрана **По умолчанию**, а на договоре **<не установлена>** смена данного тарифа в ЛК запрещена.

Учетные периоды договора

Если в конфиге сервера есть параметр `client.gui.contract.tree.order` его требуется удалить (закомментировать) или в него добавить `period`
`client.gui.contract.tree.order=parameters objects hierarchy status limit period mode face balance tariff modules groups web tariffGroup script addAction memo`

если параметр был или его правила, требуется перезапустить клиента биллинга

в дереве договора появится пункт **Учетный период**



при создании, редактирование, удаление **учетного периода** генерируются события

- **Создание/редактирование учетного периода** (ru.bitel.bgbilling.kernel.contract.period.server.event.ContractPeriodUpdateEvent)
- **Удаление учетного периода** (ru.bitel.bgbilling.kernel.contract.period.server.event.ContractPeriodDeleteEvent)

при окончании **учетного периода** может генерироваться событие (если при создании/редактировании **учетного периода** был выставлен соответствующий флаг и настроен соответствующий режим, см. ниже)

- **Окончание учетного периода** (ru.bitel.bgbilling.kernel.contract.period.server.event.ContractPeriodCloseEvent)

возможно программное создание **учетного периода**

```
// ContractPeriod
ContractPeriod contractPeriod = new ContractPeriod();
// ID
contractPeriod.setContractId( 111 );
//
PeriodWithTime period = new PeriodWithTime();
// 2019-09-01 01:02:03
period.setLocalDateTimeFrom( java.time.LocalDateTime.of( 2019, 9, 1, 1, 2, 3 ) );
// 2019-09-10 04:05:59
period.setLocalDateTimeTo( java.time.LocalDateTime.of( 2019, 9, 10, 4, 5, 59 ) );
//
contractPeriod.setPeriod( period );
// ( true - , false - )
contractPeriod.setGenerateCloseEvent( true );
//
ServerContext.get( ServerContext.class ).getService( ContractPeriodServiceImpl.class, 0 ).updateContractPeriod(
contractPeriod );
```

если необходимо генерирование события окончания периода, для этого есть два режима

1) задача планировщика

Задача

Модуль:	Ядро
Название:	Генерации событий окончания учетного периода
Приоритет:	нормальный
Состояние:	выключена

она при запуске будет выбирать все учетные периоды у которых

- установлен флаг необходимости генерации события окончания периода

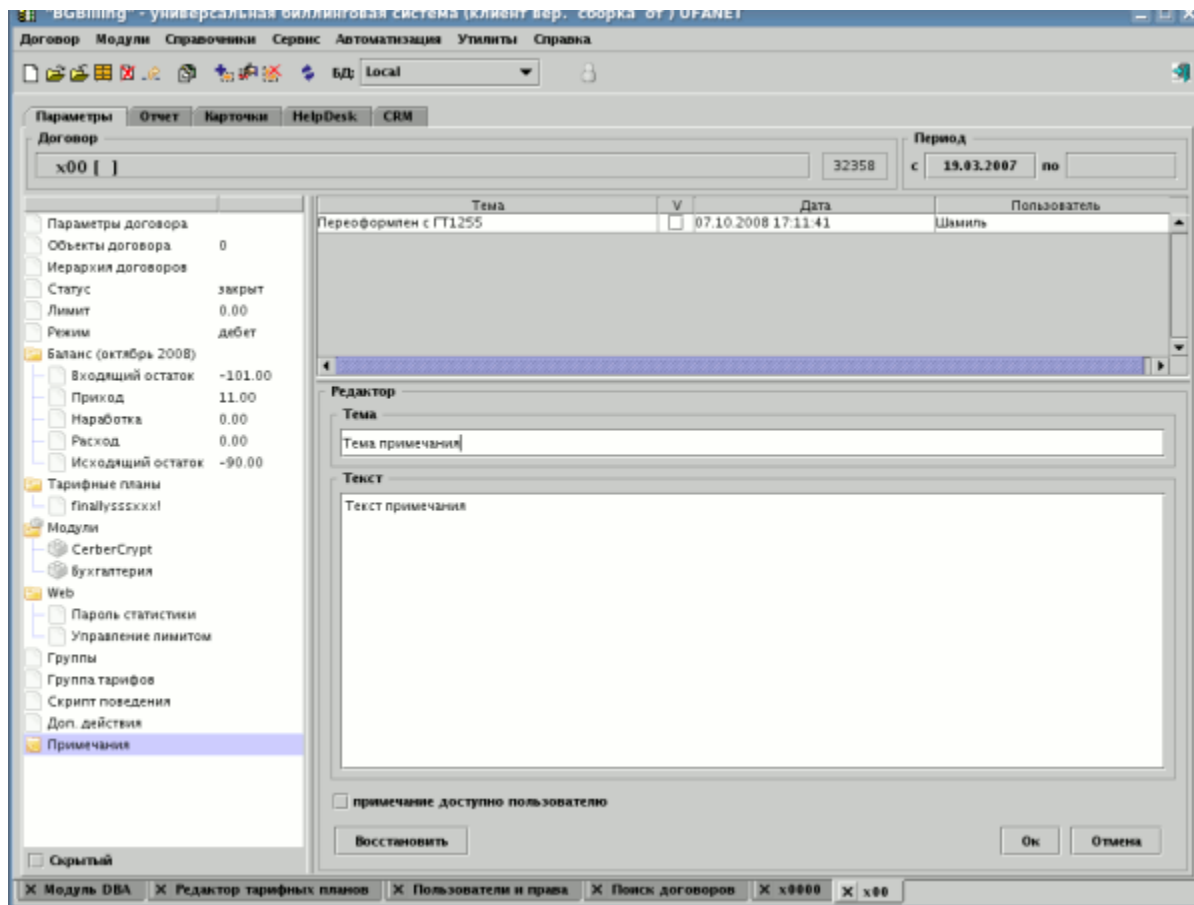
- событие не было отправлено ранее

- дата окончания периода ранее **now + period**, где **now** - текущее время, **period = 900** (по умолчанию) сек. (значение **period** может быть задано в параметрах запуска задачи планировщика в параметре **period=900**)

2) с помощью встроенной в биллинг подсистемы JobScheduler. Для включения этого режима в конфиге сервера необходимо прописать параметр **contract.period.job.close.event.enable=true**

Примечания

В примечаниях могут быть заведены произвольные заметки по договору, примечание состоит из темы и текста.



Галочка **примечание доступно пользователю** отображает примечания в Web-интерфейсе клиента в меню **Примечания**.

Дополнительные действия

Дополнительные действия в договоре - это возможность непосредственного вызова скрипта BGBS для данного договора через АРМ оператора биллинга, либо из Web-статистики. Список дополнительных действий формируется скриптом по событию **Получение списка доп. действий для договора**, либо **Получение списка доп. действий для Web**, пример в [Wiki](#).

Далее оператор биллинга, либо клиент на Web-статистике вызывает действие, снова вызывая скрипт, но уже с событием **Обработка доп. действия для договора**. Скрипт выполняет какие-то действия с договором, возвращая сообщения.

Сообщения о выполненных действиях возможно создавать в виде html, при этом содержимое должно быть валидным xml-фрагментом, а корневым элементом должен быть элемент <html>:

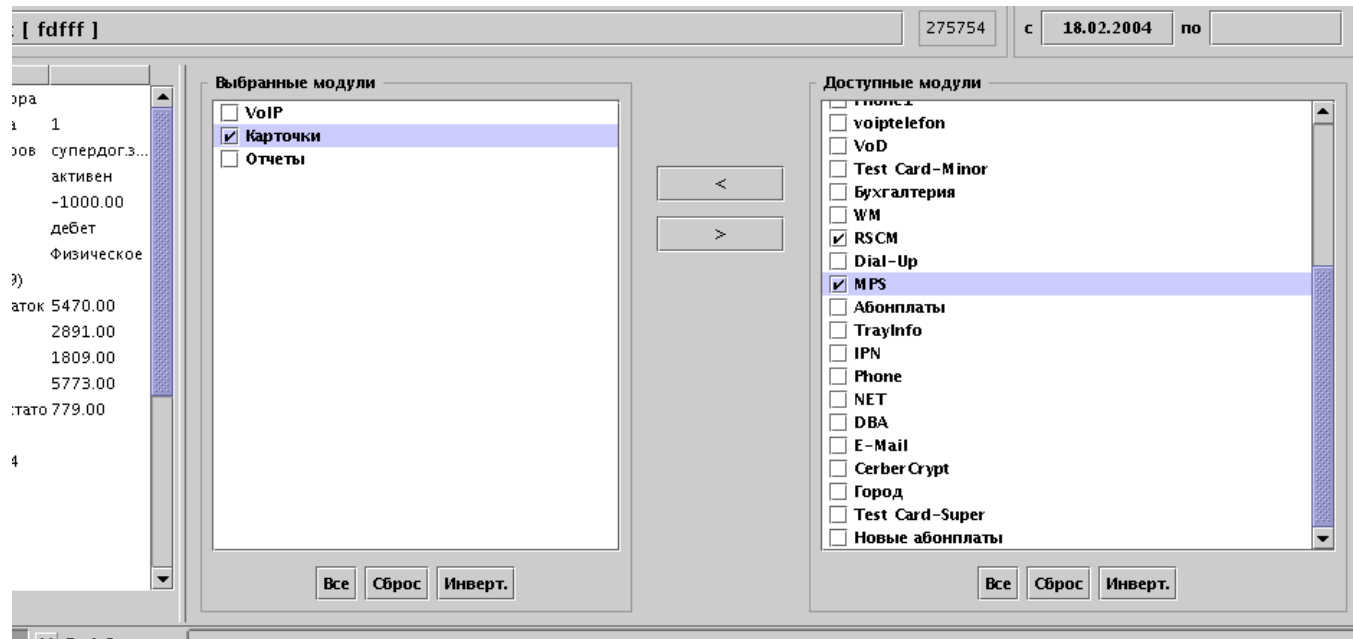
```
event.addReport( "<html><table cellpadding=\"1\"><thead><tr><td></td></tr></thead><tbody>"  
+ "<tr><td class=\"comment\">, !!!</td></tr></tbody></table></html>" );
```

Подключение модулей и их услуг к договору

При выборе узла **Модули** отображаются как потребляемые клиентом модули, так и неиспользуемые. В качестве дочерних узлов выступают подключенные модули.

На открывшейся панели возможно манипулировать подключенными к договору модулями. В левом списке перечислены модули, которые уже подключены к договору. В правом списке, соответственно, указаны модули, которые не подключены. Для того, чтобы подключить к договору новые модули, выберите их из списка права, отметив галочками, и нажмите на кнопку <. Все выбранные модули будут подключены к договору. Соответственно, для отключения модулей из договора выберите их в списке слева и нажмите на кнопку >.

При удалении модулей из договора сохранность выделенных сущностей, оказанных услуг и прочей информации данного модуля, связанной с данными договором, **НЕ ГАРАНТИРУЕТСЯ**.



Добавленный модуль должен появиться в дереве. Далее его можно выбрать и изменить пользовательскую настройку этого модуля для контракта (для DialUp - это список логинов входа), а также добавить разрешенные услуги для модулей, где они необходимы.

Договор Период

х0000 [Иванов Петр Сидорович] 455 с 06.02.2005 по

Параметры договора:

- Объекты договора 1
- Иерархия договоров
- Статус активен
- Лимит 960.00
- Режим кредит
- Баланс (март 2009)
 - Входящий остаток -159.60
 - Приход 10.00
 - Наработка 0.00
 - Расход 100.00
 - Исходящий остаток -249.60
- Тарифные планы
 - IPN
- Модули
 - Assist.Ru
 - CerberCrypt
 - Dial-Up**
 - E-Mail
 - HD
 - IPN открыт
 - Phone
 - TrayInfo
 - VoIP
 - Абонплаты
 - Бухгалтерия
 - Карточки
 - Новые абонплата
- Группы
- Скрытый

Свойства **Разрешенные услуги**

Логины **Учетные периоды**

Логин	Алиас(ы)	Период	Сессии	Ограничение доступа
3681		20.03.20...	не ограничено	разрешен

В некоторых модулях конфигурация услуг данного модуля, добавленная в договор, также имеет значение. В модуле DialUP возможен режим, когда при авторизации проверяется наличие в договоре всех требуемых для данной сессии услуг.

Разрешенные для модуля услуги настраиваются на соответствующей вкладке в настройках договора. Чтобы добавить новый модуль нажмите кнопку **Добавить** на стандартной панели инструментов.

Договор Период

х0000 [Иванов Петр Сидорович] 455 с **06.02.2005** по

Параметры договора:

- Объекты договора 1
- Иерархия договоров
- Статус активен
- Лимит 960.00
- Режим кредит
- Баланс (март 2009)
 - Входящий остаток -159.60
 - Приход 10.00
 - Наработка 0.00
 - Расход 100.00
 - Исходящий остаток -249.60
- Тарифные планы
 - IPN
- Модули
 - Assist.Ru
 - CerberCrypt
 - Dial-Up
 - E-Mail
 - HD
 - IPN** открыт
 - Phone
 - TrayInfo
 - VoIP
 - Абонплаты
 - Бухгалтерия
 - Карточки
 - Новые абонплата
- Группы
- Скрытый**

Свойства **Разрешенные услуги**

Наименование	Период действия
Максимальный IPN	13.01.2009-09.02.2009

Выбор услуги

Услуги

Только используемые

Внешний вх.

Исходящий внутренний

Локальный вх.

Локальный исх.

Максимальный IPN

Период действия

с по

Комментарий

Карты регистрации договора

FOR-карточки

Вид карточки:

Параметры Отчет **Карточки** HelpDesk CRM Документы

Карточки **Полная карта**

AA 1 из 1 [25] Размер: 100

Карта регистрации к договору x0000 (Иванов Петр Сидорович)

ФИО:
Телефон(ы):
Адрес:
E-mail (рассылка):
Тарифный план
Пароль доступа к статистике: 12345

	LOGIN	PASS
Internet		
Email	11111	11111

Примечание: _____

Оператор Абонент
Директор направления Интернет
ОАО «Уфанет»
_____/А.Р.Мустафина _____
«__» _____ 2006 г. «__» _____ 2006 г.
МП. МП.

**Карта регистрации к договору x0000 (Иванов Петр Сидорович)
(для абонента)**

Договор №: x0000
ФИО:
Тарифный план
Пароль доступа к статистике: 12345

	LOGIN	PASS
Internet		
Email	11111	11111

Электронный адрес (e-mail)

Техническая поддержка (круглосуточно): 900-405, mastera@ufanet.ru

Информация о наработке и состоянии Вашего лицевого счета, статистика подключений: <http://www.ufanet.ru/>, раздел «Клиентам», раздел «СТАТИСТИКА», подраздел «Выделенный канал»

Оператор Абонент
Директор направления Интернет
ОАО «Уфанет» _____

XML [Print] [Save] [Email]

Карточки генерируются по XSLT:FO-шаблону, указанному в основной конфигурации:

```
contractcard.1=card_inet.xsl:  
contractcard.2=card_inet.xsl:
```

С версии 8.2107 дополнительно поддерживаются параметры

```
contractcard.1=card_inet.xsl:
contractcard.1.withParams=true|false //
contractcard.1.addSub=true|false //
contractcard.1.addSuper=true|false //
contractcard.1.addCustomer=true|false //
#
# 1 - (e-mail,, ..)
# 2 - ( , , , , , , ..)
# 3 -
# 4 -
# 5 -
# 6 -
# 7 -
# 8 - contract_comment
# 9 -
contractcard.1.xmlContractConsist=3,5,8,1 // ID
contractcard.1.xmlModulesConsist=4,3,8 // ID
contractcard.1.xmlPluginConsist=1,5 // ID
```

ID,

`contractcard.withParams=true`

Возможно указание групп договоров, для которых отображается данная карта. Группы указываются после указания карты в конфигурации, через двоеточие:

```
contractcard.2=card_inet.xsl: :0,4,5
```

В приведенном выше примере карточка отображается только для групп договоров с кодами 0, 4 и 5.

XSLT-шаблон `card_inet.xsl` можно использовать как пример, поправив его под ваши модули. Шаблон необходимо переименовать, чтобы избежать его перетирания при [обновлении](#).

Если существуют какие-либо более сложные условия для добавления той или иной карточки к данному договору, то можно воспользоваться скриптом поведения на [событие](#) **Получение списка карточек договора**.

Возможно создание собственных карточек: при генерации карточки в XML-документ собирается информация по договору, затем документ трансформируется по выбранному шаблону. Содержимое XML-документа можно посмотреть, если нажать на кнопку **XML**:

Параметры Отчет Карточки HelpDesk CRM Документы

Карточки Полная карта

AA 1 из 1 [25] Размер: 100

```
<?xml version="1.0" encoding="windows-1251"?>
<data secret="01E98A33DA1C85A9F332CE405F694F39" status="ok" xslt="contract.xsl">
<contract balance_rest="0" cid="455" comment="Иванов Петр Сидорович"
date1="06.02.2005" date2="" fc="0" firm="0" gr="0"
limit="960.00" mode="0" param_group="2" pswd="12345" status="0"
status_date="24.08.2008" time="24.02.2009 18:10" title="x0000"/>
<contract_comment
comment="Логин 19 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с да
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 18 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с да
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 17 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с да
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 8 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 7 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 6 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 5 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 3 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 2 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 1 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 0 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с дат
dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 18 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с да
dt="2008-07-10 18:47:59.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 20 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-$(CARD_LOGIN) (Иванов Петр Сидорович) с да
dt="2008-07-10 18:48:00.0" subject="Перенесен логин с даты" uid="0"/>
<contract_comment
comment="Логин 3595 перенесен из договора x00 0 в договор x0000 (Иванов Петр Сидорович) с даты 20.11.2008"
dt="2008-11-26 13:25:37.0" subject="Перенесен логин с даты" uid="1"/>
<contract_comment
comment="Логин 3595 перенесен из договора x00 0 в договор x0000 (Иванов Петр Сидорович) с даты 20.11.2008"
dt="2008-11-26 13:28:32.0" subject="Перенесен логин с даты" uid="1"/>
```

XML [Print] [Save] [Email]

С помощью кнопок управления внизу окна карточка может быть распечатана, сохранена в PDF-файл, отправлена на почту. Карточки доступны к загрузке в [Web-интерфейсе клиента](#).

Полная карта договора

Карта имеет следующий вид:

The screenshot shows a web application interface with a navigation bar at the top containing tabs: "Параметры", "Отчет", "Карточки", "HelpDesk", "CRM", and "Документы". Below this, there are sub-tabs: "Карточки" and "Полная карта". The main content area displays contract information for "Иванов Петр Сидорович" on "24.02.2009 18:11".

Key information displayed includes:

- Режим: КРЕДИТ
- Остаток на счете: 0
- Лимит: 960.00
- Пароль доступа к статистике: 12345
- Лицо: ФИЗИЧЕСКОЕ

Parameters of the contract (Параметры договора):

Параметр	Значение
Агент	
Телефон (новый)	+8-902-343-72-89; +8-917-758-48-48

Tariff plans (Тарифные планы):

List of consumed services (Перечень потребляемых услуг):

- E-Mail

E-Mail	Пересылки	Пароль	Квота (КБ)
11111@freemail.irk.ru		11111	1024

- Карточки

Card type, allowed for use for account replenishment (Тип карты, разрешенной к использованию для пополнения счета):

- VoIP

Логин	Телефоны безпарольного доступа	Пароль	Состояние
-------	--------------------------------	--------	-----------

- Dial-Up

Логин	Алиасы	Пароль	Состояние
-------	--------	--------	-----------

- Абонплаты

Charged subscriptions (Начисляемые абонплаты):

- IPN

Начало диапазона	Конец диапазона
55.55.55.55	55.55.55.55
192.168.184.35	192.168.184.35
192.168.184.35	192.168.184.35

- Бухгалтерия
- Phone

At the bottom right of the interface, there are buttons for "XML", a printer icon, a save icon, and an email icon.

Карточка генерируется по XSLT-шаблону **contract.xsl**. При генерации карточки в XML-документ собирается информация по договору, затем документ трансформируется в HTML. Содержимое XML-документа можно посмотреть если нажать на кнопку **XML**.

Содержимое **XML** задается параметрами (описание параметров см. на странице [FOP-карточки](#))




Параметры Отчет Карточки HelpDesk CRM Документы

Карточки Полная карта

```

<?xml version="1.0" encoding="windows-1251"?>
<data secret="C4E57CC3E7033CAAC6C16A135EB90D61" status="ok" xsi:type="contract.xml">
  <contract balance_rest="0" cid="455" comment="Иванов Петр Сидорович"
    date1="06.02.2005" date2="" fc="0" firm="0" gr="0"
    limit="960.00" mode="0" param_group="2" pswd="12345" status="0"
    status_date="24.08.2008" time="24.02.2009 18:12" title="x0000"/>
  <contract_comment
    comment="Логин 19 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 18 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 17 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 8 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 7 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 6 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 5 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:31.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 3 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 2 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 1 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 0 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-09 17:03:30.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 18 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-10 18:47:59.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 20 перенесен из договора x0000 (Иванов Петр Сидорович) в договор RK001-${CARD_LOGIN} (Иванов Петр Сидорович) с
    dt="2008-07-10 18:48:00.0" subject="Перенесен логин с даты" uid="0"/>
  <contract_comment
    comment="Логин 3595 перенесен из договора x00 () в договор x0000 (Иванов Петр Сидорович) с даты 20.11.2008"
    dt="2008-11-26 13:25:37.0" subject="Перенесен логин с даты" uid="1"/>
  <contract_comment
    comment="Логин 3595 перенесен из договора x00 () в договор x0000 (Иванов Петр Сидорович) с даты 20.11.2008"
    dt="2008-11-26 13:28:32.0" subject="Перенесен логин с даты" uid="1"/>
  <contract_comment
    comment="Логин 3595 перенесен из договора x00 () в договор x0000 (Иванов Петр Сидорович) с даты 20.11.2008"
  </contract_comment>
</data>

```

XML   

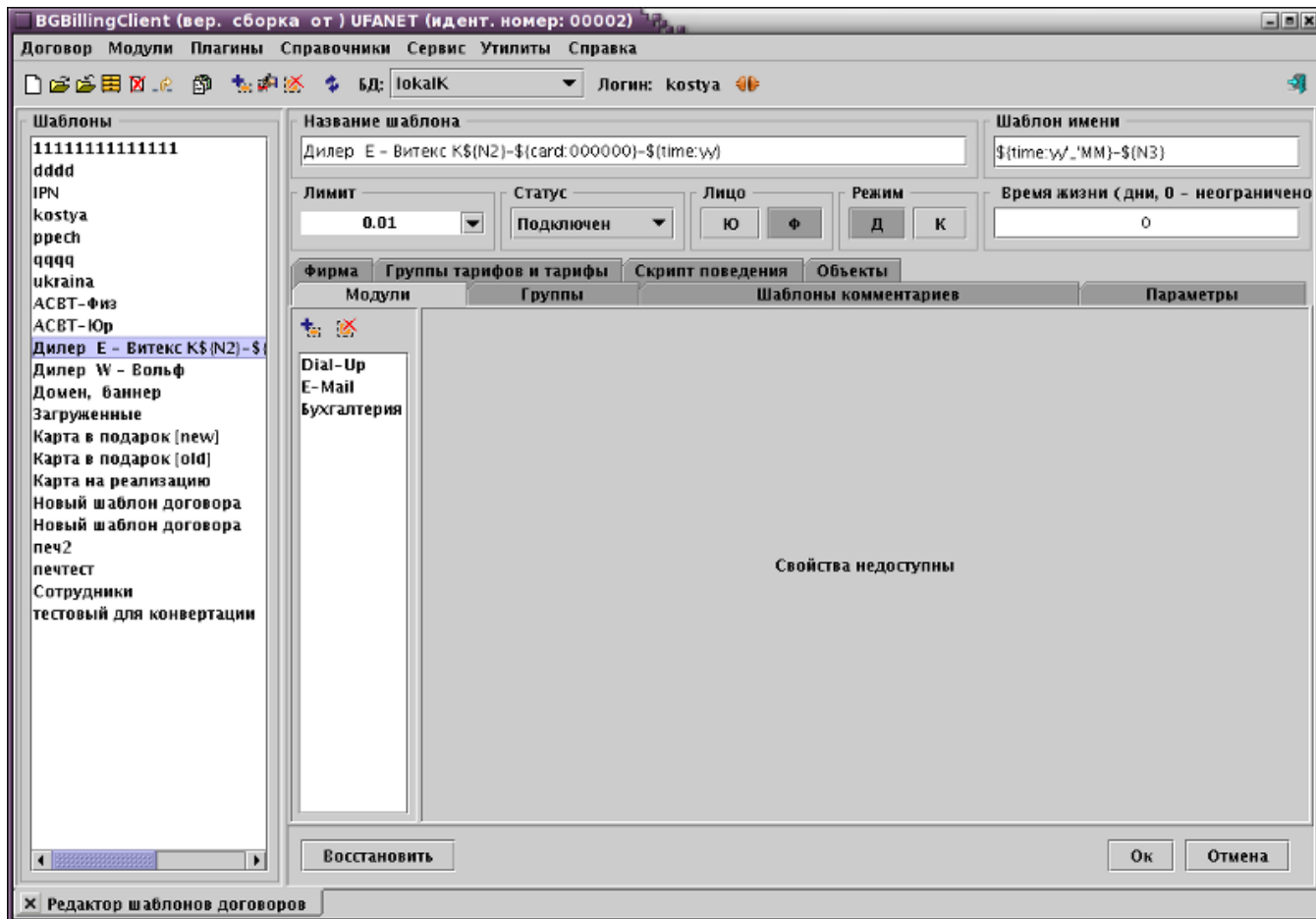
С помощью кнопок управления внизу окна карточка может быть распечатана, сохранена в HTML-файл, отправлена на почту.

Шаблоны договоров

Шаблоны договоров созданы для двух целей:

- 1. они позволяют избежать рутинного заполнения параметров при частом создании однотипных договоров, достаточно указать шаблон и параметры будут указаны автоматически;
- 2. для автоматического создания договоров; например, при первом звонке клиента с интернет-картой, ему создается договор по шаблону, который указан для этой карты.

Для создания или редактирования шаблонов зайдите в **Договор=>Шаблоны**. Редактирование осуществляется с помощью общей панели инструментов. Для создания шаблона, ему необходимо дать имя и указать параметры.



Рассмотрим подробно, какие параметры для создаваемого договора можно задать в шаблоне.

Шаблон имени

Шаблон имени задает имя договора сразу после создания, при пустом поле сразу после создания договор называется **New contract**.

Шаблон имени может включать буквы, символы и следующие подстановки:

- $\${NX}$ - порядковый номер договора, **X** - цифра. Подстановка будет заменена порядковым номером договоров такого типа, дополненным слева нулями до длины **X**;
- $\${Y2}$ - две последние цифры года создания договора;
- $\${Y4}$ - четыре последние цифры года создания договора;
- $\${time:<format>}$ - время создания договора, вместо **<format>** может быть строка макроса с yy - две последние цифры года, yyyy - четыре цифры года, MM - месяц, dd - день месяца. Полное описание допустимых макросов доступно [здесь](#);
- $\${NRX}$ - относительный порядковый номер договора, где **X** - число разрядов в номере (аналогично $\${NX}$ -подстановке). Относительный порядковый номер формируется следующим образом: сначала выполняются все прочие подстановки (например, текущая дата), затем находится договор в базе с таким "шаблоном" имени договора, берется последний относительный номер среди подобных договоров и увеличивается на единицу, после чего подставляется непосредственно в имя текущего создаваемого договора. Например, если шаблон имени определен как "D $\${Y4}\${time:MM}\${time:dd}$ -\${NR4}", то при создании за текущие сутки (например, 01.01.2009) двух договор получим номера, соответственно, D20090101-0001 и D20090101-0002, а при создании нового договора по этому же шаблону на следующие сутки получим номер D20090102-0001.

Для модуля карт (создание договора по карте) доступны следующие макросы:

- $\${card:00000}$ - логин карты, количество нулей может быть любым и задает дополнение логина нулями слева до определенной длины;
- $\${card_series:00000}$ - серийный номер карты, количество нулей может быть любым и задает дополнение логина нулями слева до определенной длины.

При создании договора выбираются все договоры, шаблон имени которых совпадает с данным, далее определяется последующий порядковый номер и создается имя договора. Имя договора может и не содержать номера.

Возможно использовать **Именованные номера шаблонов** (Шаблоны задаются в **Справочники -> Другие Договоры - именованные номера шаблонов**),

Название	Комментарий	Индекс	Кол-во цифр	...
seq1		1	5	▲
seq2		100	4	

в шаблоне имени договора указываем **NSseq2** (нумерация начнется с 100 и будет иметь 4 цифр, первый номер 0100)

Лимит, лицо, режим, время жизни, статус

Второй ряд элементов управления позволяет установить параметры:

- лицо (юридическое и физическое) - тип договора;
- режим (дебет и кредит) - режим обсчета;
- **лимит** - минимальный остаток на счете;
- время жизни - количество дней, в течении которых будет существовать договор. При этом 0 означает неограниченный срок (пока не будет закрыт вручную). Договор будет создан с указанной датой закрытия. Данный параметр можно использовать для создания карт с ограниченным сроком действия. Т.к. после активации карточки создается договор, он будет создан с проставленной датой закрытия;
- статус (подключен, отключен, закрыт, приостановлен) - выбор статуса при создании договора.

Модули2

На вкладке **Модули** указываются подключаемые к договору **модули**. Каждый подключаемый модуль должен быть добавлен в список, после чего для него возможно указание выполнения некоторых дополнительных действий. Как-то: добавление логинов, типов счетов и т.п. Также возможно указание разрешенных услуг для модулей, в которых они используются.

Шаблоны 1111111111111111 dddd IPN kostya rrech qqqq ukraina АСЕТ-Физ АСЕТ-Юр Дилер Е - Витекс К\${N2}-\${card:000000}-\${time:yy} Дилер W - Вольф Домен, баннер Загруженные Карта в подарок [new] Карта в подарок [old] Карта на реализацию Новый шаблон договора Новый шаблон договора печ2 печтест Сотрудники тестовый для конвертации	Название шаблона Дилер Е - Витекс К\${N2}-\${card:000000}-\${time:yy}	Шаблон имени \${time:yy}_MM-\${N3}		
	Лимит 0.01	Статус Подключен	Лицо Ю Ф	Режим Д К
Фирма Группы тарифов и тарифы Скрипт поведения Объекты				
Модули Группы Шаблоны комментариев Параметры				
Dial-Up E-Mail Бухгалтерия				
Свойства недоступны				
Восстановить		Ок Отмена		

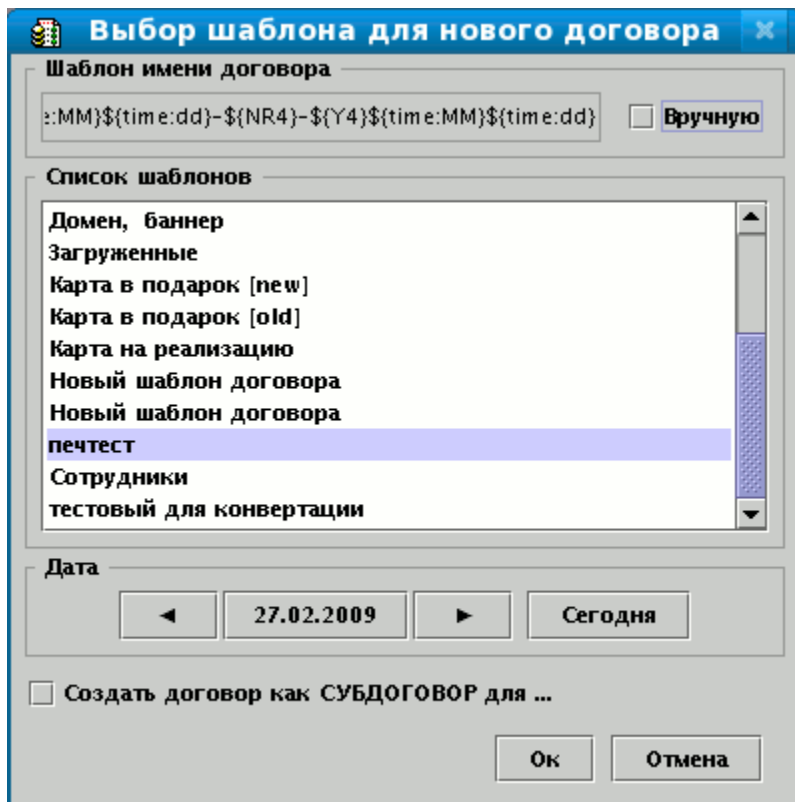
Прочие параметры

Прочие параметры, устанавливаемые на вкладках правее вкладки **Модули**:

- **Группы** - [группы](#), в которые будет входить договор;
- **Шаблон комментариев** - устанавливаемый договору [шаблон комментария](#);
- **Параметры** - [группа параметров договора](#);
- **Группы тарифов и тариф** - один или несколько [тарифных планов договора с указанием их позиций](#), а также [группы тарифных планов](#);
- **Скрипт поведения** - один или несколько [скриптов поведения](#), сопоставляемых создаваемому договору;
- **Объекты** - при необходимости создания в создаваемом договоре [объектов](#), с указанием типов и количества.

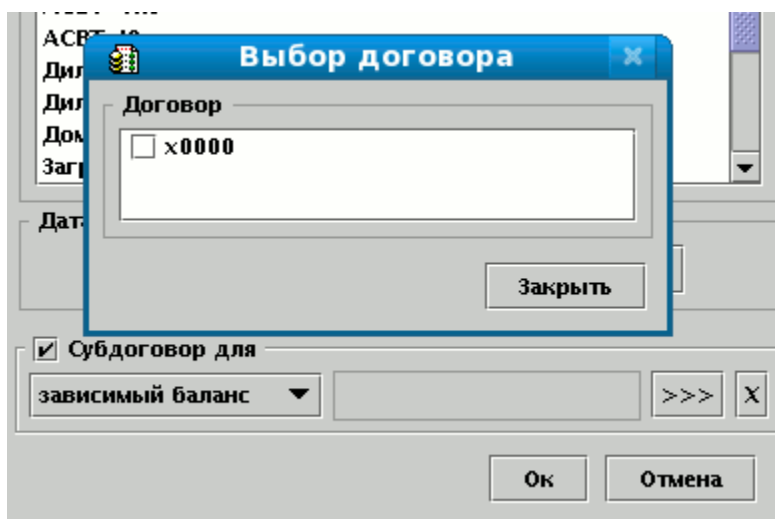
Создание договора по шаблону

Для создания нового договора выберите пункт меню **Договор=>Новый договор**. В появившемся окошке выберите базовый шаблон и дату заключения договора.



При выборе некоторого шаблона договора можно вручную изменить шаблон его имени, отметив поле **"Вручную"**. Причем вводить можно не только конкретное имя нового договора, но и изменять текущие макроподстановки (фактически создавать новый шаблон имени). Это удобно в том случае, если имя договора не генерируется системой, а по тем или иным причинам заранее известно, однако необходимо сохранить внутренние настройки шаблона договора.

Также договор можно сразу создать как **субдоговор** для какого-либо супердоговора. Для этого следует отметить поле **"Создать договор как СУБДОГОВОР для ..."**



... выбрать тип баланса для данного договора, а также непосредственно сам супердоговор (нажав на >>>), вкладка которого должна быть предварительно открыта.

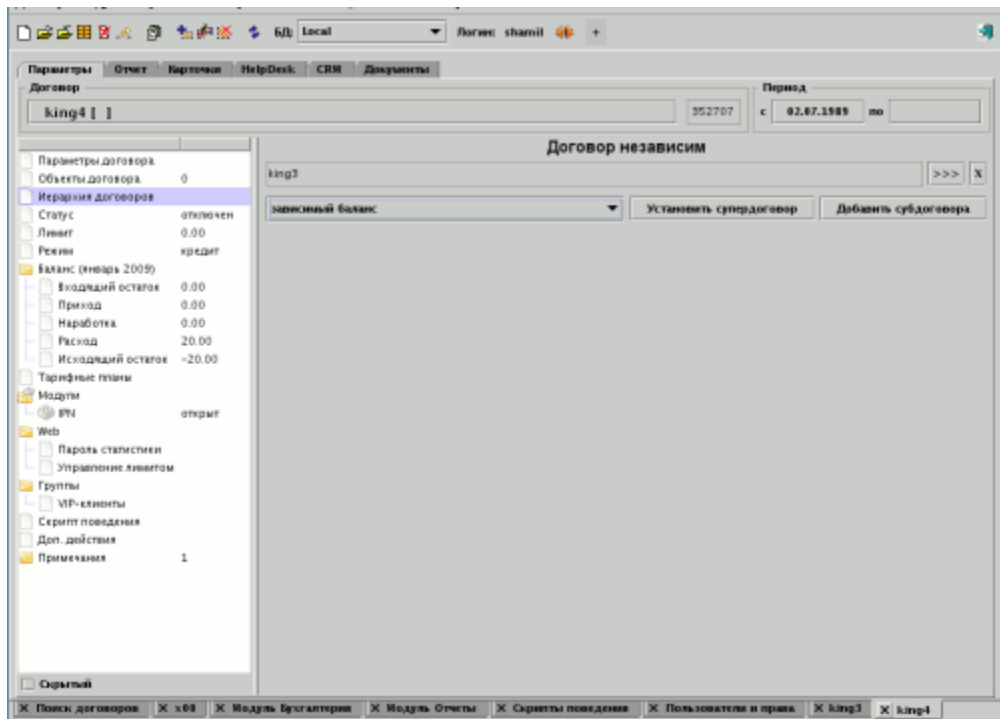
Субдоговоры

Биллинг позволяет выстраивать договоры в иерархии. При этом в вершине иерархии устанавливается супердоговор, а подчиненными к нему - субдоговоры. Каждый субдоговор является полноценным договором во всех остальных отношениях: к нему могут быть подключены модули, назначен тарифный план, установлены параметры.

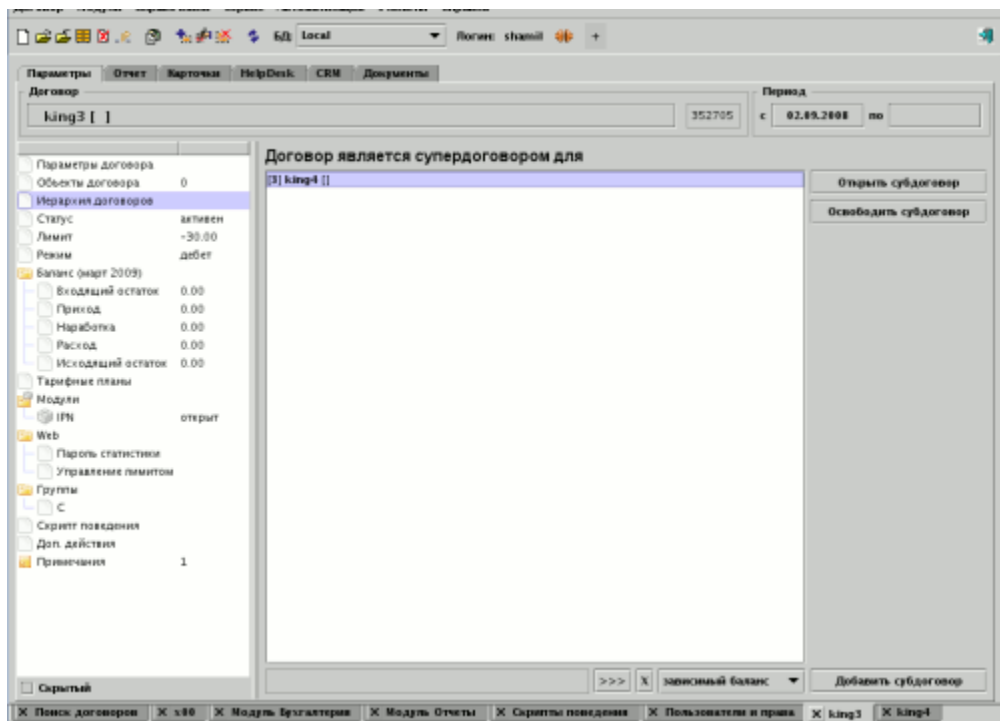
Субдоговора различаются на два типа: с зависимым балансом (**зависимые субдоговора**) и независимым балансом (**независимые субдоговора**). Выбор режима субдоговора производится при его подключении к супердоговору.

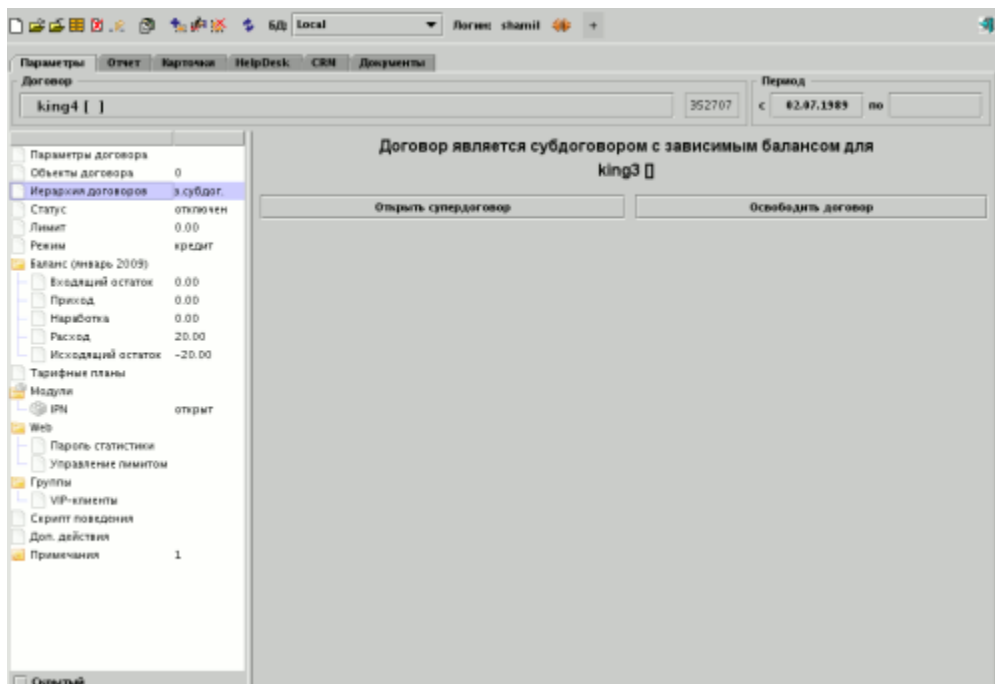
Добавление субдоговоров

Для выстраивания иерархии договоров необходимо либо выбрать для добавления субдоговор в супердоговоре, либо в будущем субдоговоре выбрать супердоговор (см. снимки экрана ниже). Супер и субдоговор должны быть открыты во вкладках. При выстраивании иерархии необходимо выбрать тип отношения - зависимый, либо независимый баланс.



При этом вкладка **Иерархия договоров** меняется следующим образом для супердоговора и субдоговора. Обратите внимание, что тип зависимости указывается буквой рядом с именем субдоговора в списке субдоговоров супердоговора и в дереве субдоговора строкой **з.субдог** или **н.субдог**.





При присоединении/отсоединении договора его баланс пересчитывается в соответствии с текущим статусом. Таким образом любые два договора в любой момент можно выстроить в иерархические отношения, либо вернуть в независимое состояние. Для быстрого перехода в субдоговор из супердоговора можно использовать двойной клик в списке субдоговоров.

Зависимые субдоговоры

Все зависимые субдоговоры супердоговора имеют единый баланс со своим супердоговором.

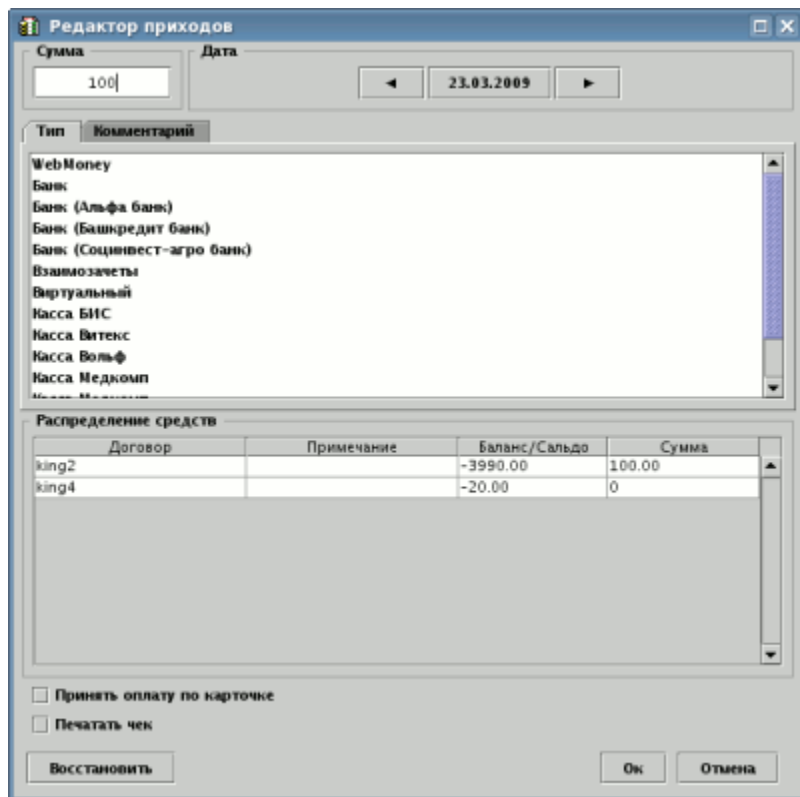
В балансе субдоговора входящий остаток на начало месяца всегда равен нулю. Приход единого баланса супердоговора и субдоговоров равен сумме доходов супердоговора и его субдоговоров. Расход единого баланса равен сумме расходов супердоговора и его субдоговоров. Нарботка единого баланса равна сумме наработок супердоговора и всех субдоговоров.

При принятии решении о доступе пользователя к услуге по субдоговору используется остаток на едином балансе и лимит субдоговора.

При изменении [статуса](#) супердоговора изменяются статусы его зависимых субдоговоров.

Независимые субдоговора

Независимый субдоговор имеет свой собственный баланс, его приходы/расходы/наработка отделены от супердоговора. Субдоговоры с независимым балансом позволяют разделять несколько видов сервиса для одного клиента, каждый из которых блокируется отдельно и имеет свой счет. При занесении прихода в супердоговор с независимыми субдоговорами возможно произведение распределения прихода по супердоговору и независимым субдоговорам. По умолчанию сначала гасится задолженность супердоговора, затем поочередно субдоговоров. Распределение суммы может быть скорректировано в таблице.



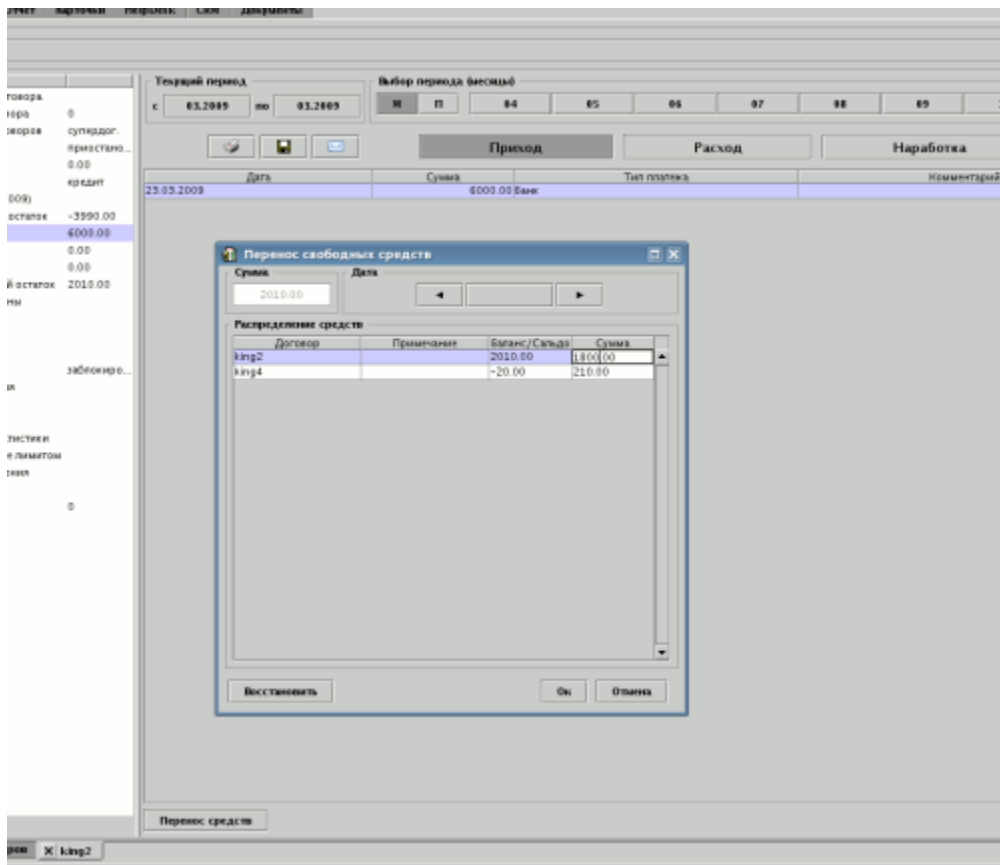
В зависимости от [режима баланса](#) супердоговора и независимых субдоговоров при оценке задолженности берется либо баланс, либо сальдо. Распределение наработки может быть произведено и после занесения прихода оператором биллинга путем нажатия кнопки **Перенос средств** в любой из панелей баланса договора.

Для управления функцией распределения суммы между договорами при добавлении платежа существует параметр `client.gui.payment.show.balance.invisible.subcontracts.mask` (в конфигурации сервера). В качестве значения он должен содержать регулярное выражение. Если название независимого договора(-ов) удовлетворяет данному регулярному выражению, то он(они) будет убран из распределения. Например если в качестве значения установить `.+`, то распределение фактически будет выключено. После внесения/изменения параметра необходимо перезапустить клиентское приложение (сервер не нужно), для того чтобы изменения вступили в силу.

```
# ..
client.gui.payment.show.balance.invisible.subcontracts.mask=.+
```

Для вывода баланса вместо сальдо для независимых договоров в режиме кредит в конфигурации сервера необходимо указать флаг:

```
# , ,
client.gui.payment.show.balance.for.credit.contract=1
```



После выбора распределения средств между супердоговором и субдоговорами в супердоговор заносится расход, а в субдоговоры - приходы. Типы расхода и прихода должны быть определены в справочниках как не редактируемые и указаны в конфигурации сервера следующим образом:

```
transfer.payment.type=< , >
transfer.charge.type=< , >
```

Перенос средств доступен также через Web-интерфейс пользователя.

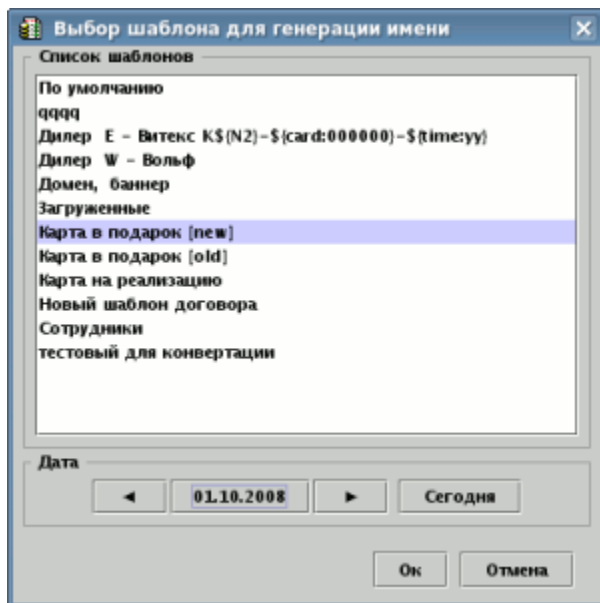
В модуле телефонии (Phone) субдоговоры с независимым балансом несут дополнительную функцию организации тарификации по агентской схеме, когда звонки и наработка по одному и тому же поинту, добавленному в супердоговор, относятся либо к супердоговору, либо к одному из его независимых субдоговоров.

При изменении **статуса** супердоговора статусы его независимых субдоговоров не изменяются, однако это поведение может быть изменено опцией конфигурации сервера биллинга **independ.subcontract.status.change=1**.

Переоформление договоров

Типовая операция, подразумевающая закрытие старого договора определенной датой с переносом всех его свойств на новый договор следующим днем от той даты. При этом в новом договоре указывается новое название организации, реквизиты.

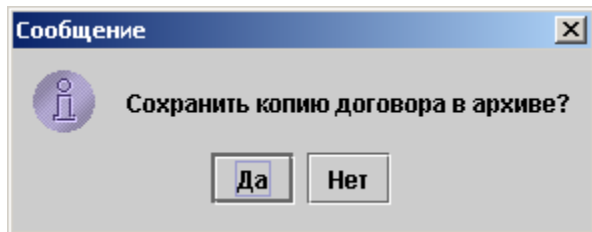
Для переоформления договора откройте карточку договора, выберите кнопку **Переоформить договор** в общей панели инструментов либо в меню **Договор**. В открывшемся диалоговом окне выберите дату, с которой будет открыт новый договор и шаблон, на основании макроса имени которого будет сгенерировано имя нового договора.



После нажатия **Ок** старый договор будет закрыт, новый открыт и на него будут перенесены все опции старого с новой даты. Как то: логины, адреса, услуги, абонплаты и пр.

Удаление договоров, архив договоров

Для удаления договора следует его открыть и нажать на кнопку **Удалить договор**. После чего нужно ответить на вопрос: удалять его безвозвратно или в архив.



После удаления в архив договор будет сериализован в формат XML, сжат в архив ZIP и выложен в папку **..BGBillingServer/archive**. После чего данные из базы будут удалены.

Для автоматического удаления договоров зайдите в **Сервис=>Настройка=>Менеджер договоров** и на вкладке **Правила для удаления** укажите правила, по которым будет удаляться договор. Правила делятся на два типа:

- по времени - устанавливается параметр срок - период в месяцах, прошедший с момента закрытия договора(это вторая дата у периода действия договора). Также можно установить фильтр по группам;
- по сумме - устанавливается сумма - минимальный остаток на счете, после которого договор будет удален и срок , в течении которого не было движения по счету. Движением по счету считается наличие прихода, наработки или расхода(с точностью до месяца или месяцев, даже если указано кол-во дней).

Очень желательно поделить все ваши договоры по группам: например Карточки, Организации и т.д. Это позволит вам более гибко устанавливать правила удаления, не рискуя удалить не те договоры.

После добавления нужных правил добавьте задачу удаления в планировщике заданий "Удаление старых договоров". Установите временные критерии запуска задачи.

Рекомендуется ставить эту задачу на ночное время, чтобы днем не загружать БД. В параметрах задачи укажите значения:

```
max_balance=20
max_closed=10
email=bill@bill.ru
#
#folder=card
```

Значение параметров **max_balance** - максимальное количество договоров, удаляемое за один раз по правилам по сумме, **max_closed** - по времени, **email** - адрес, на который будет послан отчет с перечнем удаленных договоров, если таковые имелись.

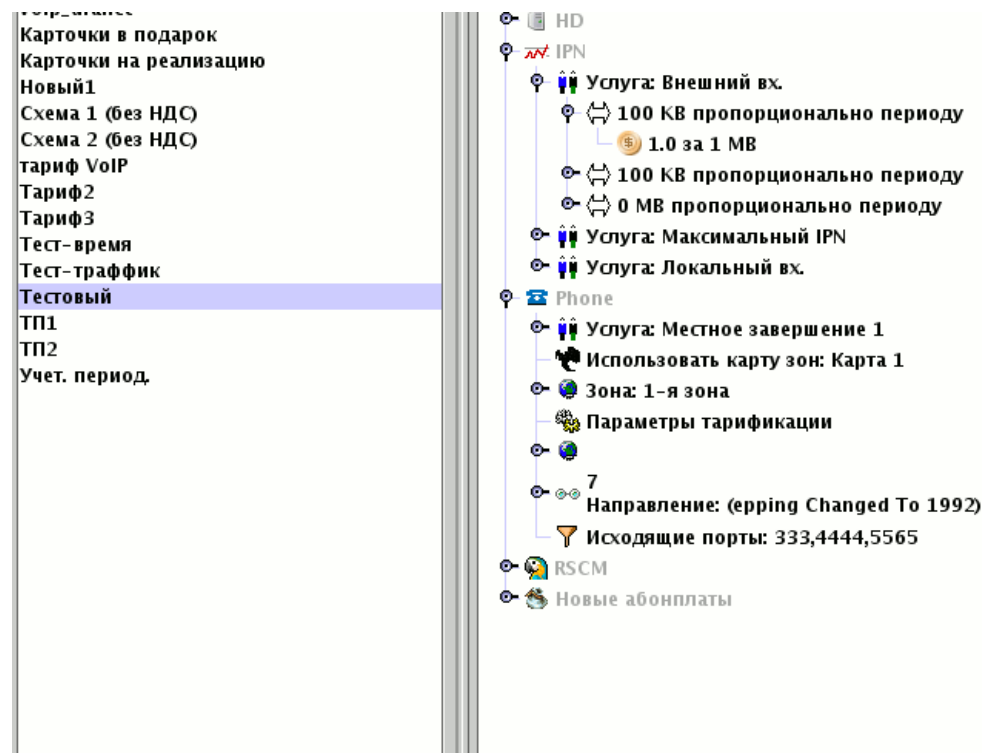
Тема входящего письма: "These contracts were deleted!".

Для того, чтобы планировщик смог отсылать письма не забудьте указать в настройках сервера параметры E-Mail. Все автоматически удаляемые договоры помещаются в архив договоров. Для просмотра архива используйте вкладку **Сервис=>Настройка=>Менеджер договоров=>Управление архивом**.

С помощью фильтра можно выбирать время создания файла. Выбрав строку и нажав кнопку **Восстановить договор из архива**, вы можете восстановить договор из файла в базу данных. Если восстановление прошло успешно, файл будет удален из архива.

Тарифные планы

Тарифный план задает стоимость услуги в какой-либо момент времени. Кроме того, в нём могут быть указаны параметры сервиса, опции тарификации. Тарифный план представляет собой дерево, в котором определяются одно или несколько модульных поддеревьев, задающих правила тарификации в экземплярах модулей.



Следует понимать, что тарификатор каждого модуля реагирует только на свое "поддерево" в тарифе, установленном для договора и игнорирует поддеревья других модулей. В зависимости от удобства использования модульные поддеревья могут собираться как в едином тарифе, так и разбиваться по нескольким.

Например, если у вас в системе есть линейка DialUP-тарифов с разной абонплатой и стоимостью часа, вы можете объединить модульные поддеревья DialUP и NPay-модулей, указывая в договоре только один тариф. Но, в то же время, те же договоры могут быть подписаны на разные Voip-тарифы, для которых целесообразно создать отдельную линейку и указывать в договоре 2 тарифа: DialUP+Абонплата и Voip.

Тарифные планы могут быть глобальные (созданы в справочнике тарифных планов и могут использоваться всеми договорами) и персональные (созданы в конкретном договоре) .

Для того, чтобы создать глобальный тарифный план, зайдите в меню **Справочники=>Тарифные планы**. Перед вами откроется окно со списком всех тарифных планов в системе, который можно отфильтровать по ряду критериев:

- 1) По названию;
- 2) по используемости;
- 3) по модулям;
- 4) по меткам.

Также можно видеть ряд статистических данных по тарифам: сколько договоров использует каждый тариф, сколько из них активных, сколько тарифов-наследников. В случае, если имеется сложный тариф с множеством услуг и условий и необходимо создать похожий, отличающийся незначительными изменениями, то можно воспользоваться кнопкой копирования выбранного тарифа. Она расположена над таблицей со списком тарифов.

Фильтр
 Название тарифа:
 Флаг "используется": не учитывать

Модули:
 CerberCrypt2
 CerberCryptTest
 Dial-Up
 DrWeb
 DrWebN
 HD2
 INET
 Internet
 IPN
 Phone

Все Сброс Инверт.

Метки: Ф Р

- Корневой элемент
 - Абонплаты [1]
 - Подневные
 - Помесячные
 - Акции [1]
 - Сочи 2014 [2]
 - Неиспользуемые тарифы [1]
 - Старые тарифы [1]
 - Тарифы по услугам
 - Разовые услуги

Тарифные планы
 Всего записей: 86 1 из 2 [50]

ID	Название	Комментарий	Наследовано ТП	Договоров (активных)	Договоров (всего)
60	111111:test<~>		6	0	0
56	abon		0	6	6
144	abonka		0	8	8
210	airat_plan		4	0	0
190	amir		4	18	21
279	artur_tariff		0	2	4
141	boris		1	30	34
281	boris copy		0	0	0
290	cerbercrypt test dimon		0	1	3
282	den_tariff		0	1	1
271	den1		0	0	0
135	den4	12345	0	0	0
131	dialup		3	0	0
47	DialUp - Test delayed		0	0	0
289	inet		0	3	3
137	IPN		0	0	0
147	IPN_RULE		0	1	1
192	ipn_rule2		0	1	1
193	ipn_test		1	0	0
138	IPNIPN		0	11	12
181	my_voip		0	1	2
287	New tariff plan		0	0	0
293	New tariff plan		0	0	0
302	New tariff plan		0	0	0
309	New tariff plan		0	0	0
310	New tariff plan		0	0	0
283	New tariff plan1		0	0	0
75	nnp		2	0	1
70	no limit гтс с апус для асвт...		0	0	176
189	npay		0	0	0
261	Pech45		1	0	0
142	PechPlan		1	0	0
295	phone простой		0	0	0
294	phone_cdr_test		0	1	1

Х Редактор тарифных планов

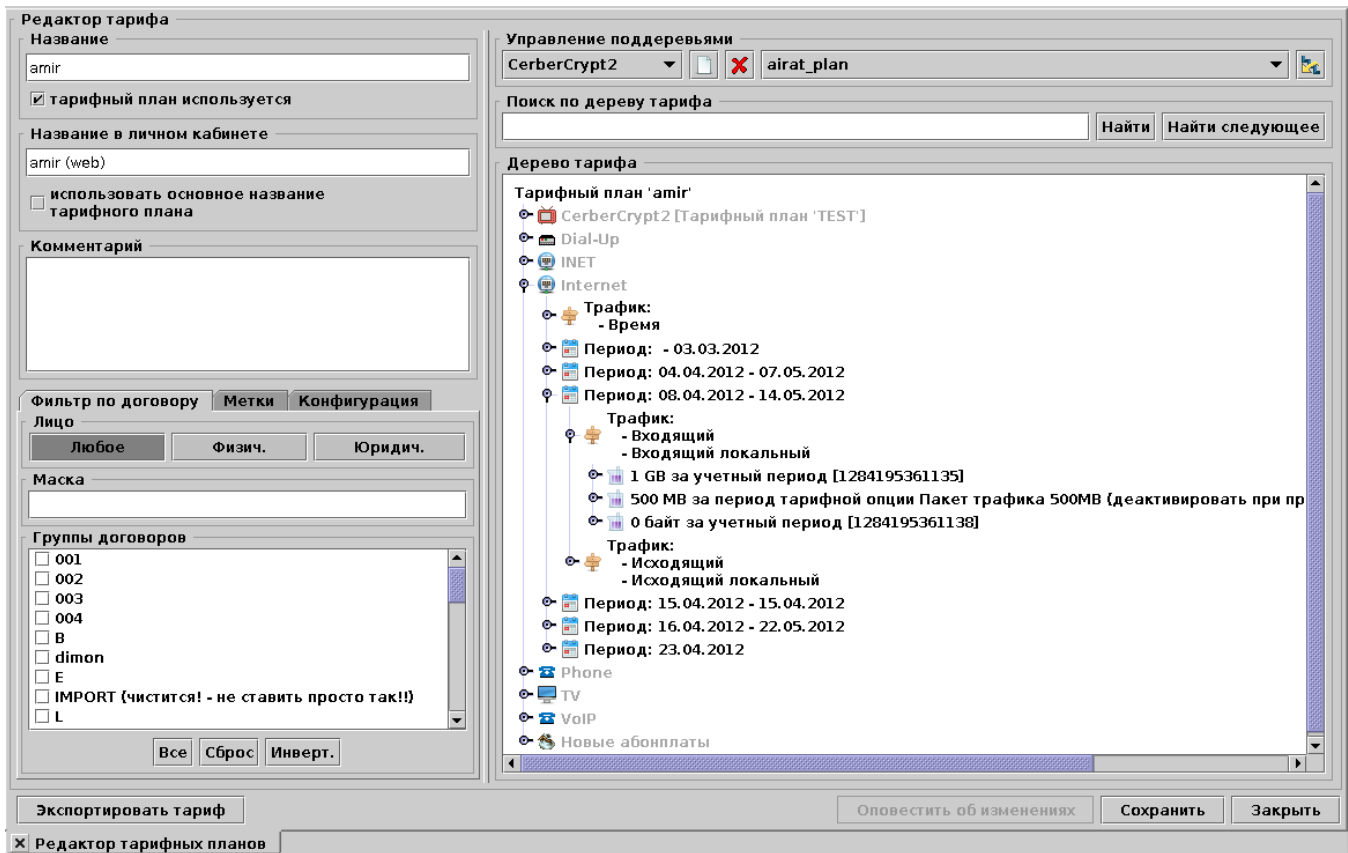
Метки предназначены для удобной каталогизации имеющихся тарифов, которых со временем может скопиться огромное множество. Редактор меток доступен по нажатию кнопки **Р** в фильтре по меткам.

Метки: Ф Р

- Корневой элемент
 - Абонплаты
 - Подневные
 - Помесячные
 - Акции
 - Сочи 2014
 - Неиспользуемые тарифы
 - Старые тарифы
 - Тарифы по услугам
 - Разовые услуги

Добавление/изменение/удаление меток осуществляется через контекстное меню. Изменение положения меток осуществляется перетаскиванием мышью.

Для добавления нового тарифного плана нажмите кнопку **Новый элемент** на стандартной панели инструментов.

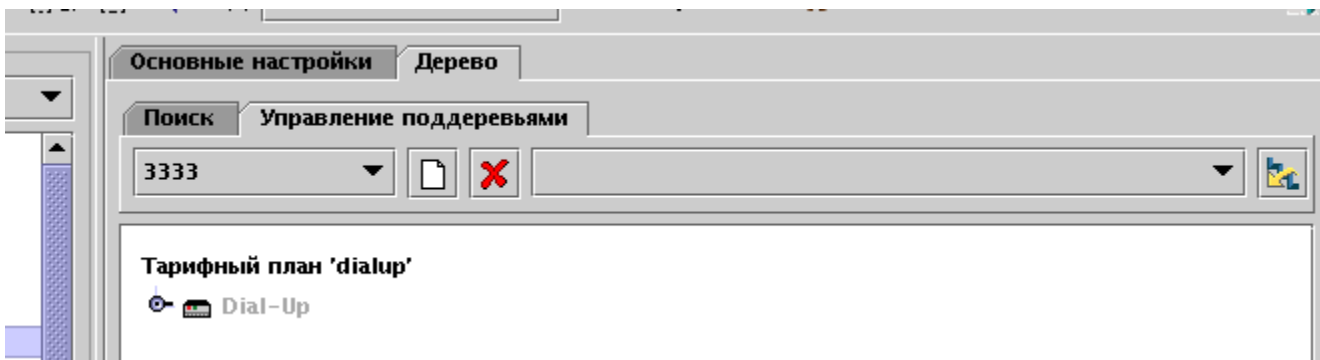


Новый тарифный план создается с пустым деревом. Во вкладке **Основные настройки** находятся основные параметры тарифа. Для изменения названия тарифного плана введите новое имя в поле сверху. Также есть галочка, обозначающая используется тариф или нет.

Фильтр по договору используется при добавлении тарифного плана для некоторого договора (см. [здесь](#)). Тариф будет отображаться в списке возможных для добавления тарифов к данному договору только в том случае, если удовлетворяет выбранному фильтру. **Лицо** - тариф будет доступен, соответственно, для любых договоров, только для физических лиц или только для юридических лиц. **Маска** - тариф будет доступен только, если имя договора удовлетворяет заданной маске договора. **Группы** - тариф будет доступен, если договор входит хотя бы в одну из выбранных групп.

В разделе **Метки** можно назначить тарифу одну и более меток, по которым его в дальнейшем возможно отфильтровать.

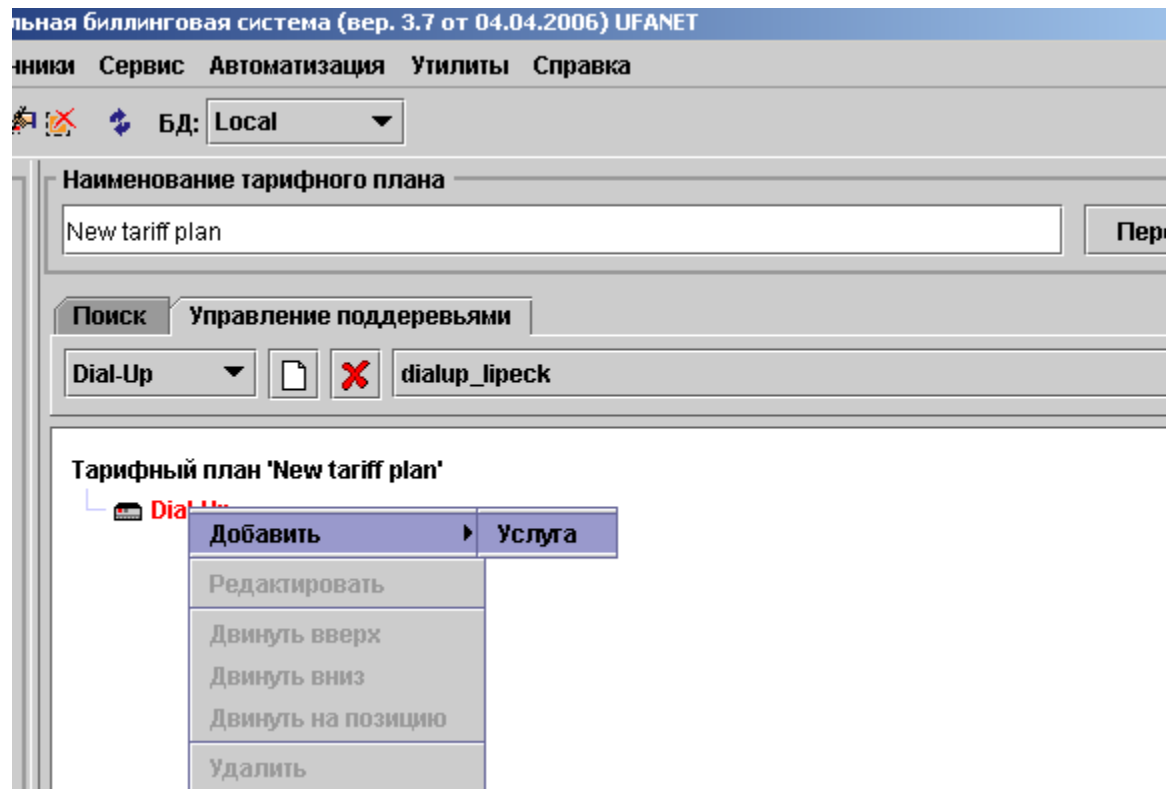
Для добавления в тариф возможности тарификации услуг определенных модулей необходимо добавить модульные поддеревья, перейдя на вкладку **Дерево** -> **Управление поддеревьями**.



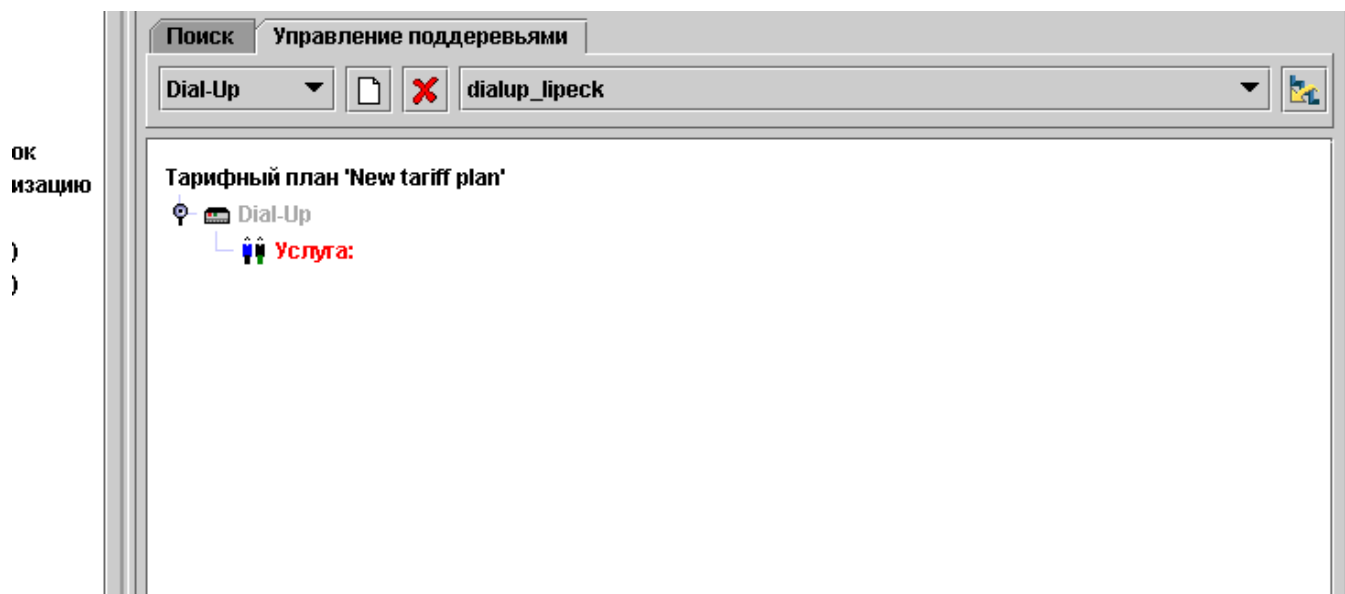
Слева расположен выпадающий список экземпляров модулей, правее - кнопки создания поддерева для модуля, либо удаление. Далее располагается список тарифных планов, у которых уже есть поддеревья для выбранного модуля, которое можно расширить.

Редактирование тарифного поддерева

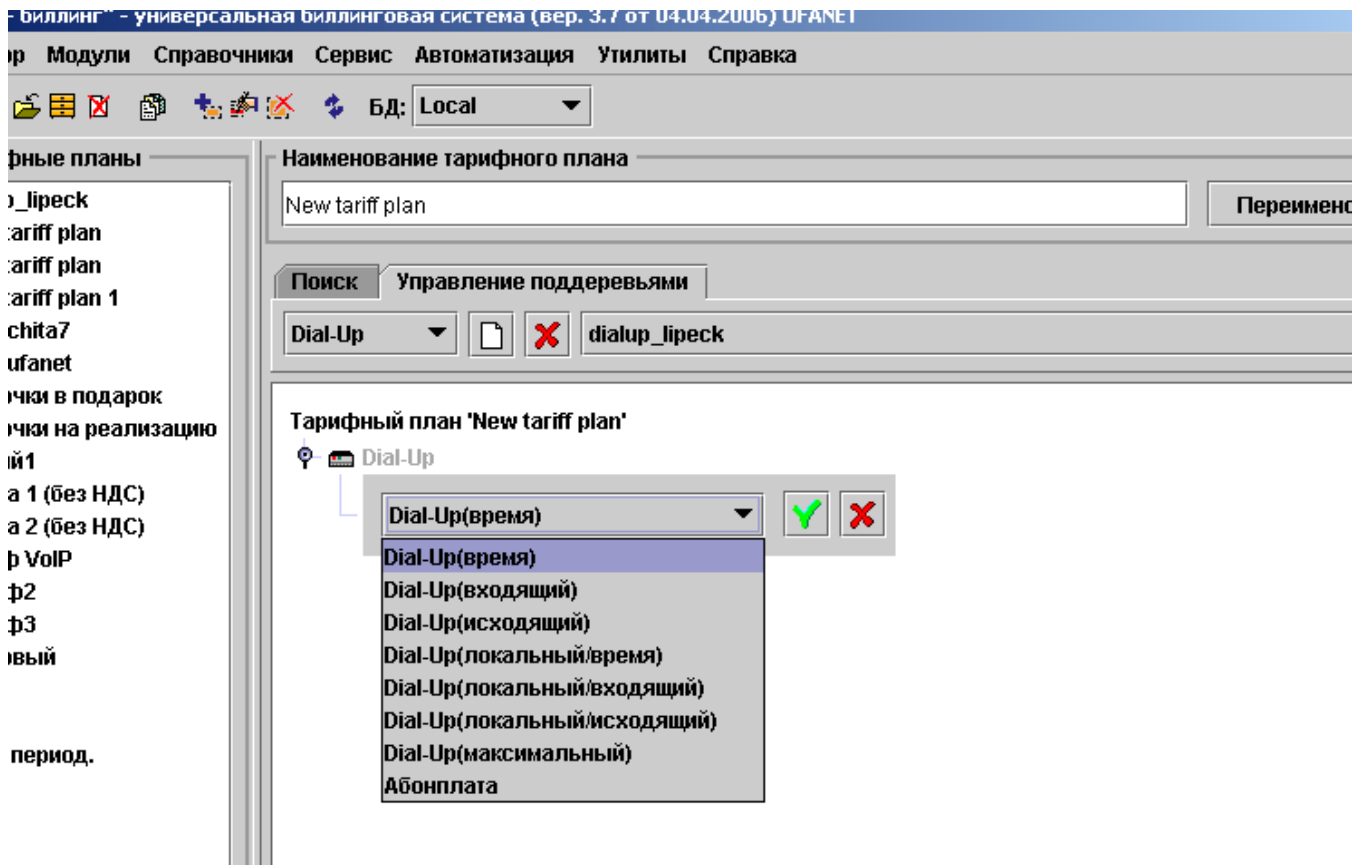
Для добавления в узел дерева дочерних элементов нажмите по нему правой клавишей мышки, выберите **Добавить** и выберите нужный тип узла. На рисунке показано, как в модуль DialUp добавляется узел **Услуга**.



А вот вид дерева после добавления узла:



Для того, чтобы редактировать узел нажмите правой кнопкой и выберите в меню **Редактировать**. Для узла отобразится специфичный для него редактор. Ниже изображен редактор для узла типа "Услуга". После завершения редактирования (в нашем случае, это выбор услуги) необходимо нажать кнопку с галочкой для сохранения результата, либо с красным крестиком, если результат не нужно сохранять. В некоторых узлах вместо данных кнопок реализованы кнопки **Ок** и **Отмена**.



Во многих узлах позиция дочернего узла не имеет значения. Но иногда она важна. Для смены положения узла в родителе нужно в меню узла выбрать пункты **Двинуть вверх** или **Двинуть вниз**.

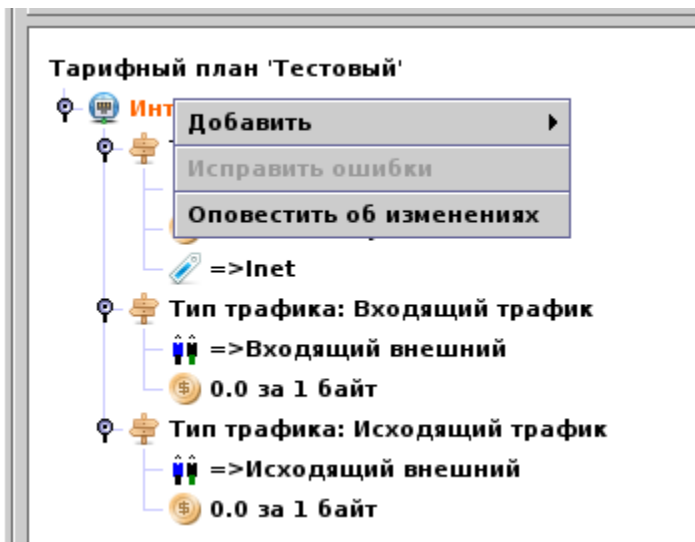
Положение узла значимо, например, для временных фильтров. Если цена будет найдена в одном фильтре, следующий уже не будет просмотрен. Поэтому можно располагать самые жесткие ограничения (праздники, например) первыми, а остальные после них.

Для удаления узла выберите **Удалить узел** в контекстном меню узла.

Тарифные деревья поддерживают возможность поиска по отображению. Для поиска воспользуйтесь текстовой областью над деревом, кнопками **Найти** и **Найти следующее**. Найденные узлы подсвечиваются синим цветом. Если вхождений больше нет, дерево свернется.

В ходе правки тарифное дерево может принимать неконсистентные состояния. Например, вы удалили узел с какой-то стоимостью и ещё не добавили новый. При этом в этот момент дерево не пригодно для тарификации.

В модулях Inet, NPray кэшированное в памяти тарифицирующего приложения дерево не перечитывается до тех пор, пока не будет передано событие. Событие передаётся выбором пункта **Оповестить об изменениях** в корневом узле модульного поддерева. О внесённых в поддерево изменениях сигнализирует оранжевый цвет корневого узла.



Это новая технология, на которую постепенно будут переведены все модули. Она позволяет пересчитывать тарифы даже для соединений, тарифицирующихся в реальном времени.

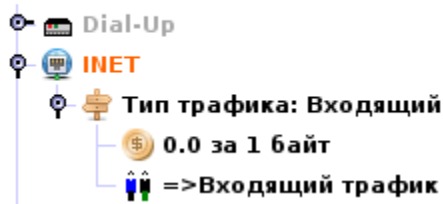
В оставшихся модулях считывание производится при первом обращении к дереву после его правки. Т.е. при правке дерева необходимо стараться уменьшить время, когда оно пребывает в неконсистентном состоянии. Отчасти проблема сглаживается тем, что для тарификации реального времени (DialUp-модуль) дерево кэшируется в момент авторизации соединения и далее не изменяется даже, если будет поправлено. Т.е. правка дерева может вызвать только временные проблемы с авторизацией соединений. Для периодической тарификации (IPN, RSCM и т.п.) дерево также загружается в момент начала обчёта и далее используется. Т.е. проблема может возникнуть только, если на начало периодической тарификации дерево некорректно, повторный переобсчёт проблему устраняет.

Оранжевый цвет корневого дерева также может информировать об ошибках в структуре его хранения. В этом случае вам следует выбрать в контекстном меню корневого узла пункт **Исправить ошибки**, после чего закрыть тариф, открыть снова и проверить порядок узлов.

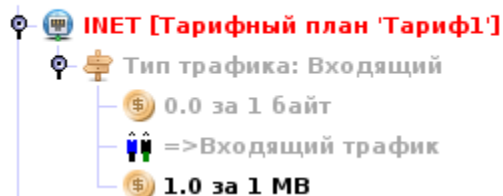
Расширение тарифных планов

Механизм расширения позволяет создать модульное поддерево на основании существующего поддерева аналогичного модуля из другого тарифа, добавив в него новые узлы. Например, с его помощью можно переопределять цены. Рассмотрим для примера "Тариф1" и "Тариф2", модульное поддерево экземпляра модуля "INET" которого получено путём расширения модульного поддерева первого тарифа.

Тарифный план 'Тариф1'



Тарифный план 'Тариф2'



Для расширения поддерева необходимо создавать с помощью кнопки расширения, слева от которой расположен список тарифов, содержащих поддерево экземпляра модуля. Название тарифа с поддеревом - предком отображается в квадратных скобках в корневом узле.

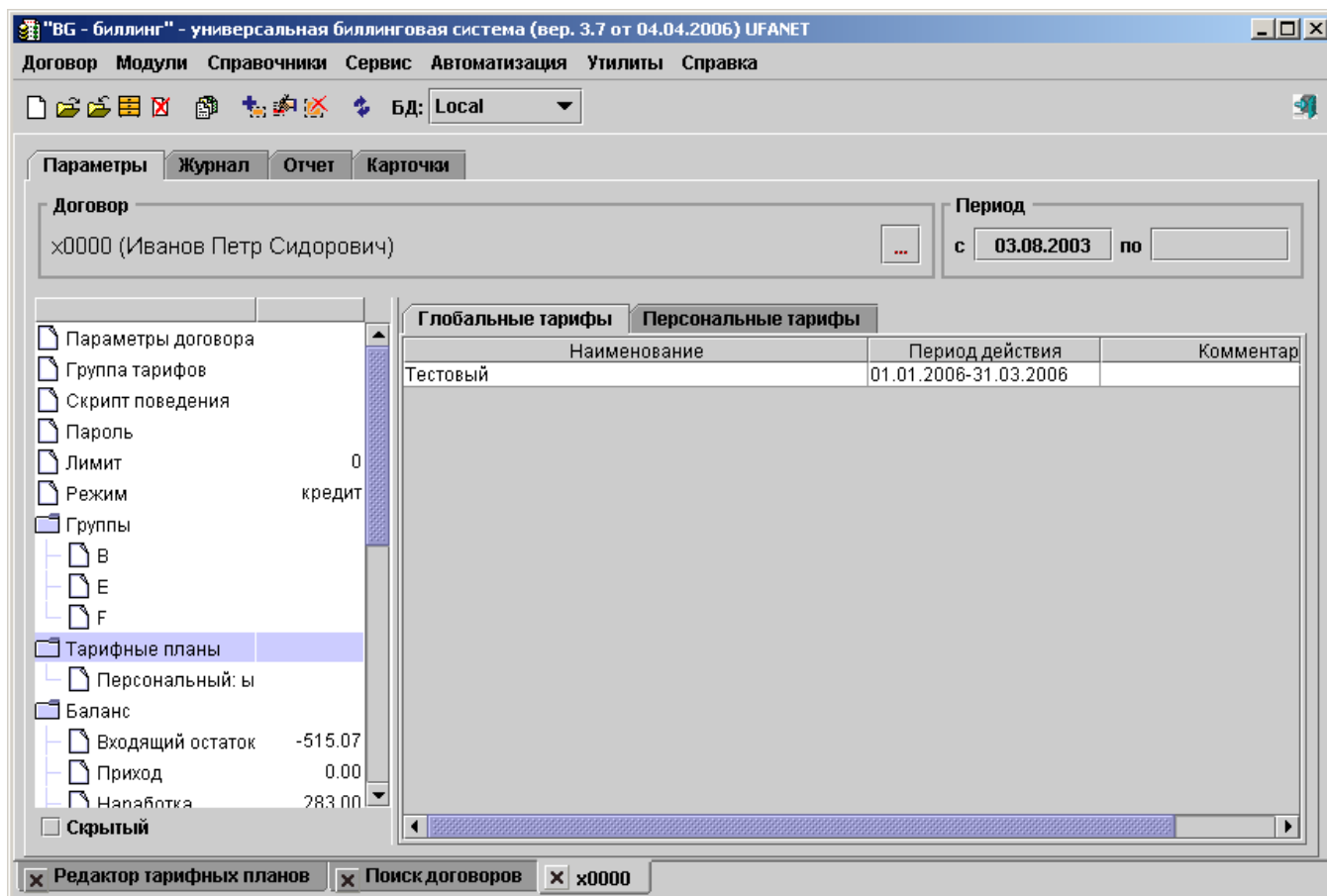


Узлы дерева-предка всегда выделены серым цветом и недоступны для редактирования. Узлы дерева-потомка всегда располагаются после узлов предка. Так же они обрабатывают позже тарифный запрос. При этом они могут перетирать значения параметров ответа тарифного запроса, установленные предыдущим узлом предком. Например, в предыдущем примере при использовании тарифа "Тариф2" стоимость мегабайта типа трафика "Входящий" будет стоить 1 рубль.

В базовом поддереве могут быть не определены цены, можно использовать его просто как каркас для построения конечных тарифов. В этом примере в тарифе "Тариф1" можно было не размещать узел типа "Стоимость".

Персональные тарифные планы

В договоре могут быть определены собственные (персональные) тарифные планы.



Для создания персонального тарифа необходимо перейти на вкладку **Персональные тарифы** и нажать кнопку **Добавить** в единой панели инструментов сверху.

тр Сидорович)

период
... с 03.08.2003 по

Глобальные тарифы		Персональные тарифы	
Название		Период	
ываыва		30.03.2006 -	

0
кредит

ы

жк -515.07

0.00

283.00

0.00

Редактор

Название: Персональный VOIP с 05.04.2006 по

Ok Отмена

Далее для редактирования дерева тарифа строку с тарифом нужно выбрать в таблице персональных тарифов и нажать кнопку **Открыть дерево** в нижнем правом углу. Правила редактирования дерева персонального тарифа не отличаются от редактирования дерева глобального тарифа. Персональный тариф может быть расширен от глобального, либо создан с нуля.

Отчет Карточки

Сидорович) ...

Период с 03.08.2003 по

Глобальные тарифы **Персональные тарифы**

Поиск Управление поддеревьями

VoIP [X] VoIP

0 кредит

-515.07
0.00
283.00
0.00

Тариф 'Персональный VOIP' для договора 'x0000'

- VoIP [Тарифный план 'VOIP']
 - Услуга: IP Телефония через Cisco
 - Направление: AFGHANISTAN /АФГАНИСТАН/
 - +7 123
 - Минута исходящего: 2.0
 - Минута исходящего: 1.0**

Закреть дерево

Для закрытия персонального тарифа - кнопка **Закреть дерево** в правом нижнем углу.

Порядок просмотра тарифных планов

В различных модулях используется различный алгоритм порядка просмотра тарифных планов.

Алгоритм 1:

- 1. ищется тариф, содержащий модульное поддерево экземпляра модуля среди персональных тарифов договора, используется фильтр по дате, сортировка по позиции тарифа; если найдено - то 3;
- 2. ищется тариф, содержащий модульное поддерево среди глобальных тарифов договора, используется фильтр по дате, сортировка по позиции тарифа;
- 3. в найденном модульном поддереве ищется цена на все услуги, которые нужно тарифицировать; если цена какой-то услуги там не найдена, то ошибка тарификации.

При такой схеме в один день в договоре может активно действовать только один тариф, содержащий модульное поддерево экземпляра модуля. Если таких тарифов окажется несколько, то будет выбран первый из них в соответствии с алгоритмом. Если в найденном тарифе не окажется цен на все требуемые услуги - получится ошибка тарификации, другие тарифы просматриваться не будут.

Алгоритм 2:

- 1. выбирается список тарифов, содержащих модульное поддерево экземпляра модуля среди персональных тарифов договора, используется фильтр по дате, сортировка по позиции тарифа;
- 2. выбирается список тарифов, содержащих модульное поддерево экземпляра модуля среди глобальных тарифов договора, используется фильтр по дате, сортировка по позиции тарифа.

Для каждой из тарифицируемых сущностей (услуга, тип трафика) ищется последовательно цена в списках 1 и 2. Как только находится тариф с ценой - поиск прекращается.

В ряде модулей (Phone, NPay) алгоритм поиска тарифа отличается от перечисленных здесь и описан отдельно.

Стандартные узлы тарифных планов

Принцип работы дерева заключается в передаче запроса, содержащего определенные параметры, внутрь дерева. Параметры запроса отличаются для разных модулей. Типичными параметрами являются: время, код услуги.

Каждый узел дерева получает запрос и обрабатывает его, помещая результаты обработки в ответ и модифицируя запрос. В результате обработки узел может как пропустить запрос внутрь для всех своих подузлов, так и не пропускать его. Запретить обработку запроса следующим за ним узлом узел не может.

Вид поддереьев и состав используемых узлов различен для каждого модуля и описан в его документации. Некоторые узлы специфичны для модуля, мы рассмотрим здесь универсальные узлы, встречающиеся в тарифах всех модулей. Ниже перечислены основные типы узлов, встречающиеся во всех модулях:

Услуга

Редактор узла и узел в тарифе выглядят так:

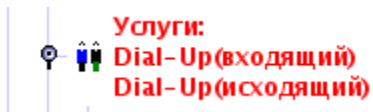
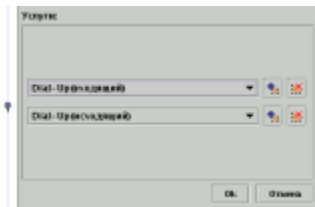
Dial-Up ▼	<input type="checkbox"/>	<input type="checkbox"/>	X
-----------	--------------------------	--------------------------	---

⚙️ **Услуга: Dial-Up**

Узел выступает фильтром по услуге и передает запрос каждому из потомков только, если услуга из тарифного запроса совпадает с указанной в узле.

Мультиуслуга

Редактор узла и узел в тарифе выглядят так:



Узел выступает фильтром по услуге и передает запрос каждому из потомков только, если услуга из тарифного запроса совпадает с одной из указанных в узле. Действие узла полностью идентично узлу Услуга за исключением возможности указания нескольких услуг в одном узле. Эта возможность может быть удобна при необходимости указания одинаковой цены на несколько услуг, либо выдачи единой квоты по трафику (IPN, DialUP модули).

Период

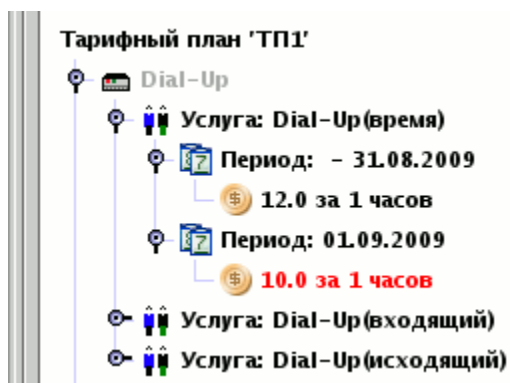
Редактор узла и узел в тарифе выглядят так:

с по

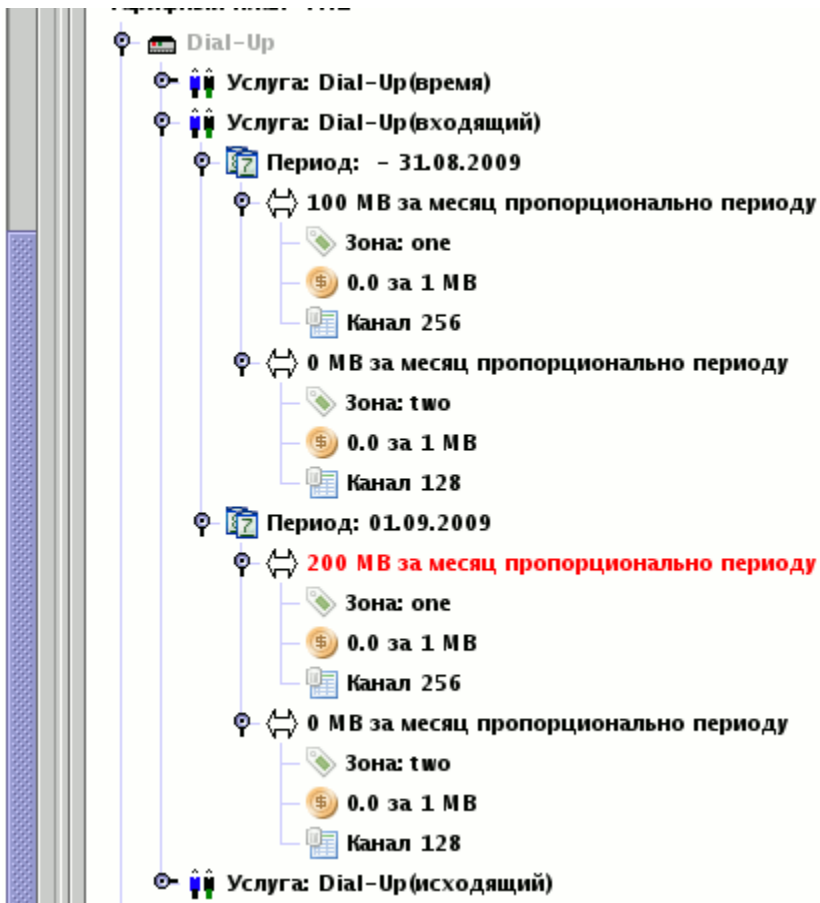
 Период: 01.05.2004 - 13.05.2004

Параметр "время" из тарифного запроса попадает в указанный период. Если одна из дат не указана, то она берется бесконечной (в прошлое или будущее). Узел период с неуказанными обеими датами просматривает дочерние узлы в любом случае.

Основное назначение узла - изменение стоимости в тарифном плане с какой-либо даты. Ниже приведен пример, как изменить стоимость услуги **Dial-Up(время)** с 1 сентября 2009 года.



В период можно помещать также секции тарифного дерева, регулируя, когда обрабатывает тот или иной блок. В приведенном ниже примере с 1 сентября увеличивается объем предоплаченного трафика в тарифе модуля DialUp.



Фильтр по времени

Редактор узла и узел в тарифе выглядят так:

Часы => 0 - 23; Месяцы => 1 - 2

Добавить	Удалить	Редактировать
Ок	Отмена	Справка

Набор ограничений:

Часы => 0 - 23; Месяцы => 1 - 2

Для редактирования условия нужно вызвать редактор, сделав двойной клик по строке условия в редакторе, либо нажав кнопку **Добавить**. Набор масок дней, часов, дней недели и месяцев, перечисленных в условии, должен выполняться весь, т.е. чтобы сработало условие, нужно совпадение всех масок. При вводе масок условия часы могут принимать значения от 0 до 23, дни недели от 1 до 7, дни месяца от 1 до 31, месяцы от 1 - 12. При вводе можно использовать символы "-" (интервал со входящими концами) и "," (перечисление). Для удобства ввода можно использовать записи * (будут выведены все значения), а также */n (n - целое число, все что делится на n, например */2 - 0, 2, 4, 6...), либо *\n (n - целое число, все что не делится на n).

Узел выступает фильтром тарифного запроса, запрос пропускается к обработке в узлах-потомках только, если параметр "время" в запросе совпадает хотя бы с одним набором условий узла. Фильтр позволяет определять в тарифах различную стоимость услуги по времени суток, дням недели, месяца. Пустой набор ограничений означает отсутствие фильтра по времени. Узел пропускает в себя запрос, только если перед ним в том же узле-предке не стоял другой фильтр по времени, либо **Фильтр по типу времени**, уже пропустивший запрос в себя (принявший запрос). Данный принцип позволяет делать наборы ограничений по умолчанию. Например, так можно определить стоимость в будние дни и праздники для модуля DialUp.

Услуга: Dial-Up (входящий)
Набор ограничений:
Дни недели => 6 - 7
1.0 за 1 MB
Набор ограничений:
1.5 за 1 MB

В приведенном выше примере во втором наборе ограничений не указаны явно дни с 1 по 5, узел обрабатывает запросы, не попавшие в первый набор ограничений. Для данного случая метод не дает ощутимой выгоды, однако при необходимости указывать многочисленные праздничные дни он удобен. Обратите внимание, что следующий пример уже не будет работать аналогичным образом. Т.к. у узлов разный предок.


Услуга: Dial-Up (входящий)
Набор ограничений:
Дни недели => 6 - 7
1.0 за 1 MB
Период: 02.11.2009
Набор ограничений:
1.5 за 1 MB

Во втором примере все запросы будет принимать второй пустой набор ограничений. Т.к. у узлов разный узел-предок. Данную ситуацию (желание задать цены на будние дни с периодом) корректно было бы разрешить так:

  **Услуга: Dial-Up (входящий)**

 **Набор ограничений:**

Дни недели => 6 - 7

 **1.0 за 1 МВ**

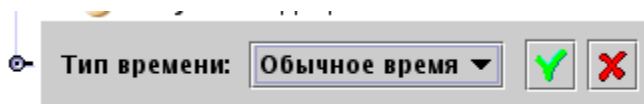
 **Набор ограничений:**

  **Период: 02.11.2009**

 **1.5 за 1 МВ**

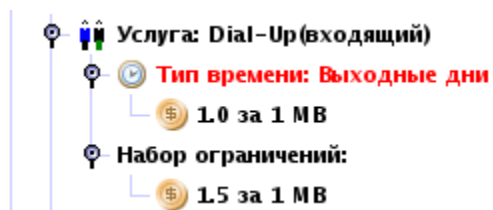
Фильтр по типу времени

Редактор узла и узел в тарифе выглядят так:

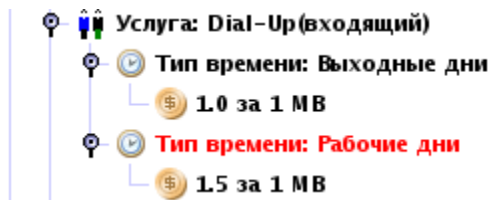


Тип времени: Рабочее время

Действие данного узла абсолютно идентично **Фильтру по времени**, за исключением того, что сами маски определяются в [справочнике типов времени](#). Это более удобно при большом количестве фильтров в тарифах разных модулей. Узлы данных типов абсолютно равноправны и могут быть использованы комбинированно. Например, описанный в предыдущей секции пример с исключаящими масками может быть разрешен так:



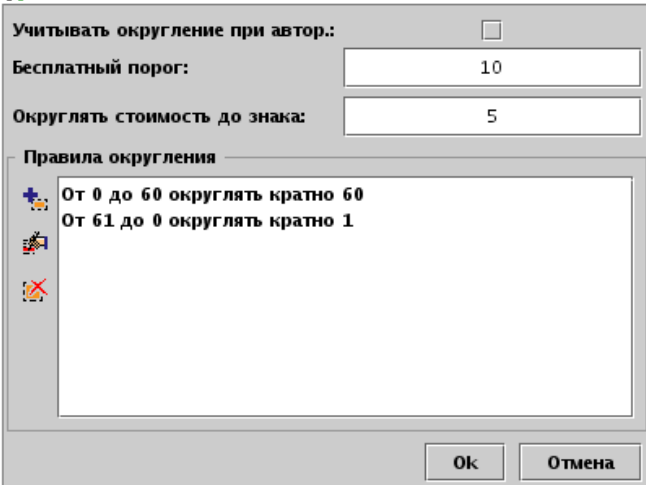
Либо так:



При этом в типе времени **Рабочие дни** не указываются никакие маски.

Параметры тарификации

Редактор узла и узел в тарифе выглядят так:



Учитывать округление при автор.:

Бесплатный порог:

Округлять стоимость до знака:

Правила округления

- От 0 до 60 округлять кратно 60
- От 61 до 0 округлять кратно 1

Ок Отмена

Параметры тарификации

Узел передает в ответной части тарифного запроса параметры тарификации:

- максимальную длительность бесплатного звонка в секундах;
- количество десятичных знаков после запятой в стоимости звонка (не имеет смысл устанавливать более, чем разрядность поля таблицы БД);
- правила округления длительности звонка.

Правила задаются в виде диапазонов длительностей звонка в секундах и кванта, до которого звонок округляется в данном диапазоне в большую сторону. На приведенном снимке звонки до минуты округляются до целой минуты, а далее идет посекундная тарификация. 0 секунд в верхней части диапазона означает отсутствие ограничений сверху.

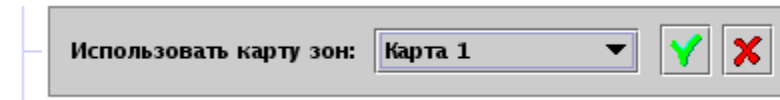
Звонки до 10 секунд включительно считаются нулевой длительности. Стоимость звонка округляется до пятого знака, т.е., например 4.32344.

Параметр **Учитывать округление при автор.** действует только для Voip-модуля и позволяет выдавать клиенту длительность разговора с учетом округления длительности в большую сторону. Т.е., если сессии от 20 до 120 секунд округляются кратно 20, а человеку хватает денег на 83 секунды, то после авторизации RADIUS вышлет 80 секунд разрешенного времени.

Узел может быть указан на разных уровнях тарифного дерева, устанавливая правила тарификации разных направлений.

Использовать карту зон

Редактор узла и узел в тарифе выглядят так:



The screenshot shows a form element with a label "Использовать карту зон:" followed by a dropdown menu containing "Карта 1" and a downward arrow. To the right of the dropdown are two buttons: a green checkmark and a red X.

 **Использовать карту зон: Карта 1**

Узел получает вызываемый номер из тарифного запроса, определяет по нему направление и зону и помещает их в тарифный запрос.

Зона

Редактор узла и узел в тарифе выглядят так:



The image shows two parts of a software interface. The top part is a node editor window with a grey border. It contains a label 'Зона:' followed by a dropdown menu showing '1-я зона'. To the right of the dropdown is an empty text input field, a '+' button, a green checkmark button, and a red 'X' button. The bottom part shows a node in a tariff, represented by a globe icon and the text 'Зона: 1-я зона' in red.

Узел пропускает запрос внутрь только если установленная в запросе зона равна указанной в узле. Поле ввода текста в правой части и кнопка "+" позволяют быстро добавить отсутствующую зону в справочник. В запрос зона устанавливается узлом **Использовать карту зон**.

Часть префикса и Диапазон префиксов

Редактор узла и узел **Часть префикса** в тарифе выглядят так:

Часть префикса: 7
Название ветки: Россия
Направление: Россия, СНГ
Зона: -- нет --
OK Отмена

7 Россия
Направление: Россия, СНГ

либо так для **Диапазона префиксов**:

Диапазоны префиксов: 3472-3473
Название ветки:
Направление: Республика Башкортостан, Уфа
Зона: -- нет --
OK Отмена

3472-3473
Направление: Республика Башкортостан, Уфа

Отличие двух узлов - в способе задания маски номера.

В **Части префикса** задается REGEXP-выражение, которое должно совпасть с начальной частью "остатка" номера.

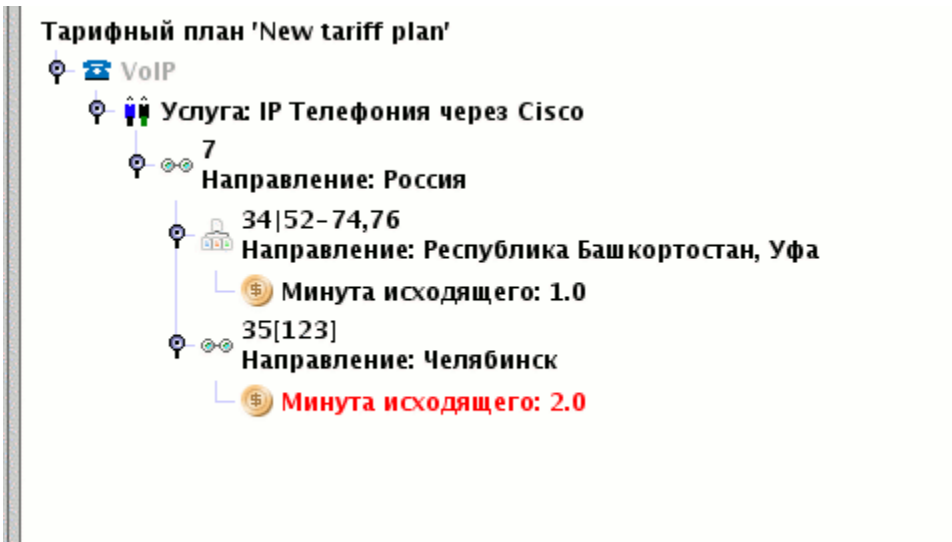
В **Диапазоне префиксов** задается выражение вида <общий префикс><диапазоны>. Например в указанный выше диапазон можно переписать в виде: 34|72-73, где 34 будет являться общим префиксом. Общий префикс может быть и не указан. Если он указан, то как бы подразумевается для всех диапазонов после него. Начальная часть остатка номеров должна попасть в один из указанных диапазонов. При этом все диапазоны префиксов в узле должны быть одной длины.

Оба узла работают с параметром "номер", переданным в тарифном запросе. При этом разбор номера идет по частям, начиная с начальных символов. Каждый узел, в который попадает телефон, сравнивает начало номера с шаблоном в узле.

Если совпадение произошло, совпавшее начало номера как бы "откусывается" и узел передает остаток номера на обработку внутрь себя, где процесс "откусывания" продолжается до нахождения цены звонка и направления. После совпадения части префикса с номером, в следующих, соседних в этом же предке, узлах запрос будет пропущен.

У каждого узла может быть установлено направление и зона. Если в совпавшем узле указано направление, либо зона, они подставляются в ответ тарифного запроса.

Рассмотрим для примера, как происходит разбор телефона 7347255555 в указанном ниже тарифе. В нем используется комбинация этих двух узлов, т.к. они являются взаимозаменяемыми.



1) Ветка 7 Россия - начало совпадает с 7 => 7 отбрасывается и далее передается 3472555555

2) аналогично ветки 3 и 4

3) в ветке 72 определяется направление звонка

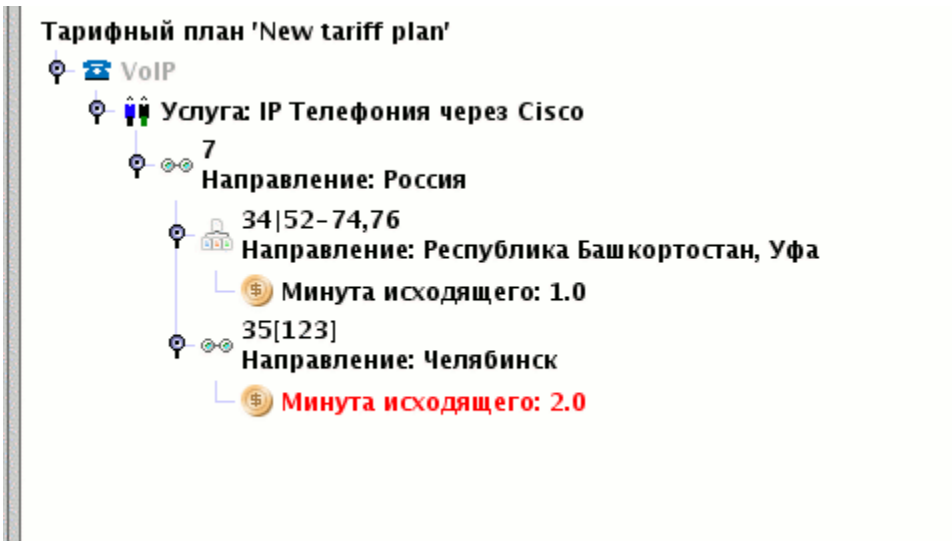
4) в ветке Минута исходящего - цена минуты разговора.

Ветки 3 и 4 просто прописывают данные в проходящий через них тарифный запрос.

После выхода из ветки 7 звонок будет помечен как обработанный после прохождения его через все дочерние узлы ветки 7. Все ветки после 7 не пропустят запрос внутрь себя.

Таким образом достигается высокая эффективность обработки запросов.

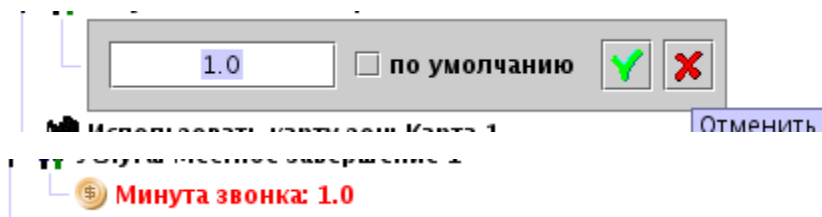
Рассмотрим еще один пример использования узлов:



В приведенном выше примере префиксы 73472-73474 и 73476 будут отнесены к Уфе, а префиксы 7351-7353 к Челябинску.

Стоимость минуты

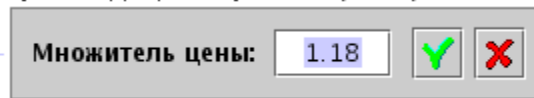
Редактор узла и узел **Стоимость минуты** в дереве выглядят так:



Узел используется в голосовых модулях. При прохождении запроса через узел в ответной части узел размещает стоимость, указанную в нем. При установке опции **по умолчанию** цена переустанавливается только, если в запросе еще не установлена цена, либо она установлена также узлом со свойством по умолчанию в дереве-предке (тариф расширен). Если опция не установлена, цена будет переопределена при прохождении тарифного запроса через узел в любом случае.

Множитель цены

Редактор узла и узел **Множитель цены** в дереве выглядят так:



↘ **Множитель цены: 1.18**

Узел используется для умножения цены, установленной узлом **Стоимость минуты** на определенный коэффициент. Например, используя данный узел, можно быстро создать тариф с НДС для физ. лиц, расширив базовый тариф и добавив в конце дерева узел с умножением.

Тарифный план 'Телефония Совинтел преселект с НДС'

☒ Phone [Тарифный план 'Телефония Совинтел преселект']

Набор ограничений:

Часы => 0 - 7

>>>> Льготный

Набор ограничений:

Часы => 8 - 23

>>>> Бизнес

☑ 1 США

☑ 2

☑ 3

☑ 4

☑ 5

☑ 6

☑ 7 Россия и СНГ

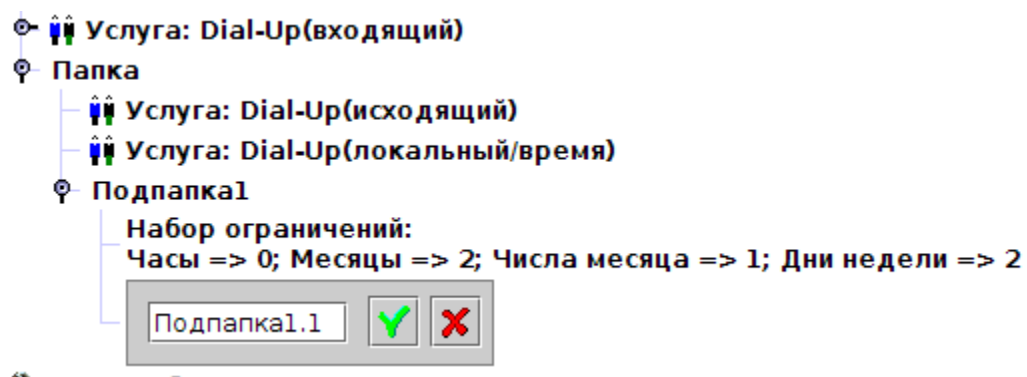
☑ 8

☑ 9

↘ **Множитель цены: 1.18**

Элемент каталога

Редактор узла и его внешний вид в дереве представлен на рисунке ниже.



Используется для группировки различных узлов по определенным критериям. Суть узла аналогична понятию каталог в файловой системе.

Тарифные опции

Доступны в Справочники => Тарифные опции.

Тарифные опции позволяют расширить логику тарифа. Их можно обозначить как дополнительные услуги, которые клиент может подключить из Web-интерфейса на определенный период за определенный расход. Тарифные опции можно также активировать и из клиента биллинга.

На первой закладке указываются тарифные планы, при которых можно будет активировать опцию.

Период	Название	Деактивация	Реактивация	Тип расхода	Стоимость
18.05.2012-...	11111 (3 дня)			Ticket	5,00
26.02.2015-...	Льготный	моментально	запрещена	Ticket	0,00
18.01.2022-...	1 минута			AV-Desk	0,00

На третьей - указываются режимы активации опции.

Период	Название	Деактивация	Реактивация	Тип расхода	Стоимость
18.05.2012-...	11111 (3 дня)			Ticket	5,00
26.02.2015-...	Льготный	моментально	запрещена	Ticket	0,00
18.01.2022-...	1 минута			AV-Desk	0,00

Редактирование режима ✕

Название:

Период:
с по

Активация:

Начало

на

до

Деактивация: Реактивация (отмена деактивации):

Удаление тарифной опции: Удаление расхода:

Стоимость активации тарифной опции (расход):

Период режима активации - это период, когда возможно активировать опцию. Опция активируется на определенный период. Количество 0 означает что опция будет активирована бесконечно. При этом появляется возможность деактивировать опцию.

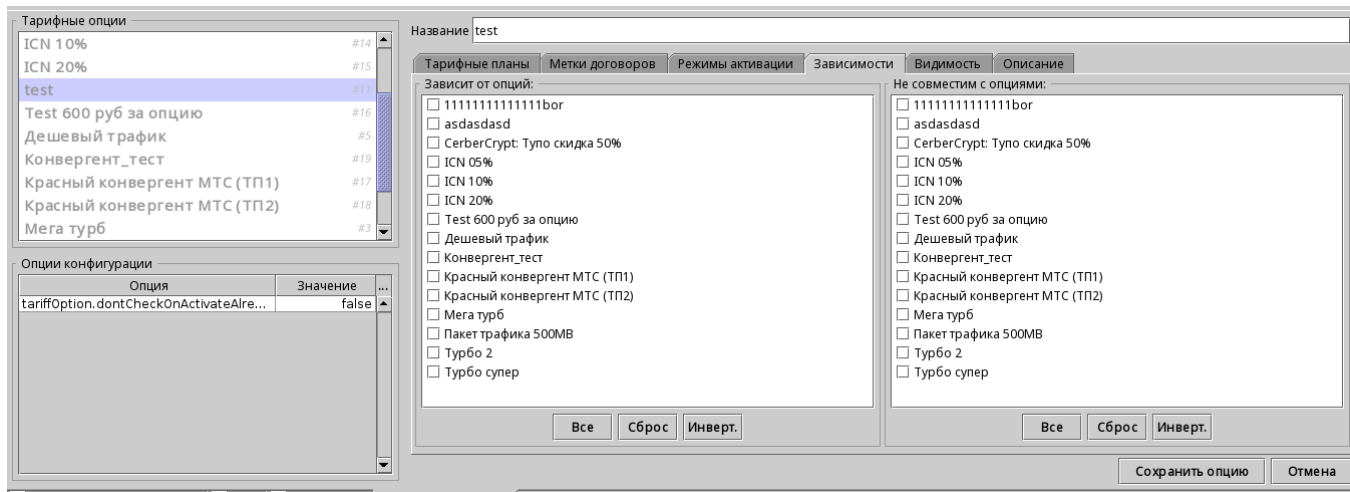
Режимы периода: **часов** - опция активируется с текущего момента на указанное количество часов; **часов со следующего** - с 00:00 следующего часа на указанное количество часов; **часов с текущего** - с 00:00 текущего часа на указанное количество часов; **дней** - с текущего момента на указанное количество дней; **дней со следующего** - с 00:00:00 следующего дня до (не включая) 00:00:00 через указанное количество дней; **дней с текущего** - с 00:00:00 текущего дня. Аналогично для месяцев и недель.

Следует учитывать, что в модулях DialUp и IPN наработка хранится по часам, поэтому, если в зависимости от тарифной опции меняется цена, то активация тарифной опции не должна быть в режиме "часов (с текущего момента)" (как исключение, для модуля Dialup можно установить разрыв соединения), а также для модуля Dialup не должна быть в режиме "с текущего часа/дня/недели/месяца", чтобы цена прошедшего времени не изменялась. Если же в модуле Dialup в зависимости от тарифной опции выполняется только посылка CoA, то режим "часов (с текущего момента)" может быть активирован.

Если количество часов (дней/месяцев/недель) указано 0, то опцию можно будет деактивировать. Режимы деактивации: **моментально** - с момента отработки заброса; **до конца дня** - до (не включая) 00:00:00 следующего дня; **до конца недели** - до (не включая) 00:00:00 понедельника следующей недели; **до конца месяца** - до (не включая) 00:00:00 первого числа следующего месяца. В этом случае также можно указать возможность **реактивации**: если режим деактивации установлен не "моментально" и опция была деактивирована, а время окончания еще не наступило, то клиент сможет **реактивировать(отменить деактивацию)** тарифную опцию.

Если сумма в поле Снять расход больше нуля, то при активации опции проверяется, баланс - сумма \geq лимита. Если средств не достаточно, опция не активируется, иначе - перед активацией снимается расход указанного типа. Тип расхода может быть только **не редактируемым**.

На следующей закладке указываются зависимости редактируемой опции от других.



Если опция зависит от других опций, то все эти опции должны быть активированы в тот период, на который активируется опция.

Если опция не совместима с какими-либо другими опциями, то ни одна из этих опций не должна быть активирована в период активации редактируемой опции.

Периоды пересечений сравниваются с точностью до секунды.

В Web-интерфейсе клиент выбирает опцию и режим активации для нее.

Тарифные опции

Текущие опции [История опций](#)

Опция	Начало	Окончание	Стоимость
Турбо	04.02.2010 19:58:31	Деактивировать	10.00

Активировать опцию: [Активировать](#)

На отдельной странице можно увидеть опции, период действия которых истек.

Тарифные опции

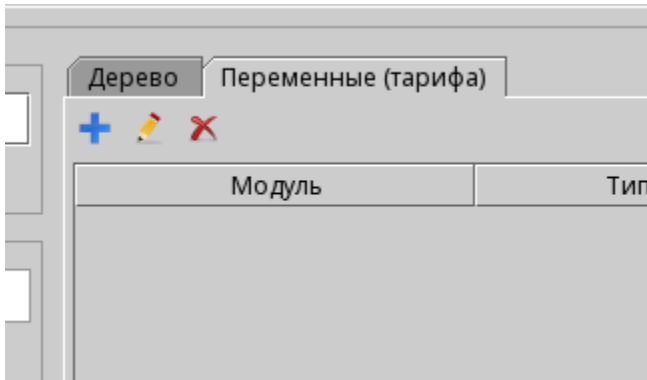
Текущие опции [История опций](#)

Опция	Начало	Окончание	Стоимость
МегаТурбо	04.02.2010 19:58:45	04.02.2010 20:58:45	0.00

Тарифные переменные

С версии 9.2207+ добавлена поддержка тарифных переменных

В редакторе тарифных планов доступна закладка "Тарифные переменные", на которой можно добавить тарифные переменные, а также задать значение по умолчанию



Новая тарифная переменная

Модуль:

Тип:

Имя:

Значение по умолчанию:

Модуль	Тип	Название	Значение по умолчанию	...
Новые абонплаты	Цена	Абонплата ТВ	400	▲

После добавления тарифной переменной ее можно использовать в редакторе тарифа

Новые абонплаты

Абонплата помесичный режим снятия [дата списания: 01]

стоимость из константы

стоимость из переменной

Режим начисления:

Новые абонплаты

Абонплата помесячный режим снятия [дата списания: 01]

Стоимость из переменной: < Абонплата ТВ [400] >; Режим списания: пропорц. периоду

Далее при добавлении данного тарифа на договор клиента можно переопределить значение переменной для данного договора

Глобальные тарифы						
ID	ТП#ID	Поз.	Наименование	Группа тарифов	Период действия	Комментарий
750 190	162	0	TEST	По умолчанию	01.01.2023-...	

Переменные тарифа: все глобальные тарифные только переопределенные значения

Тип переменной	Глобальная	Название	Значение
Цена	<input type="checkbox"/>	Абонплата ТВ	400

меняем значение переменной

Глобальные тарифы						
ID	ТП#ID	Поз.	Наименование	Группа тарифов	Период действия	Комментарий
750 190	162	0	TEST	По умолчанию	01.01.2023-...	

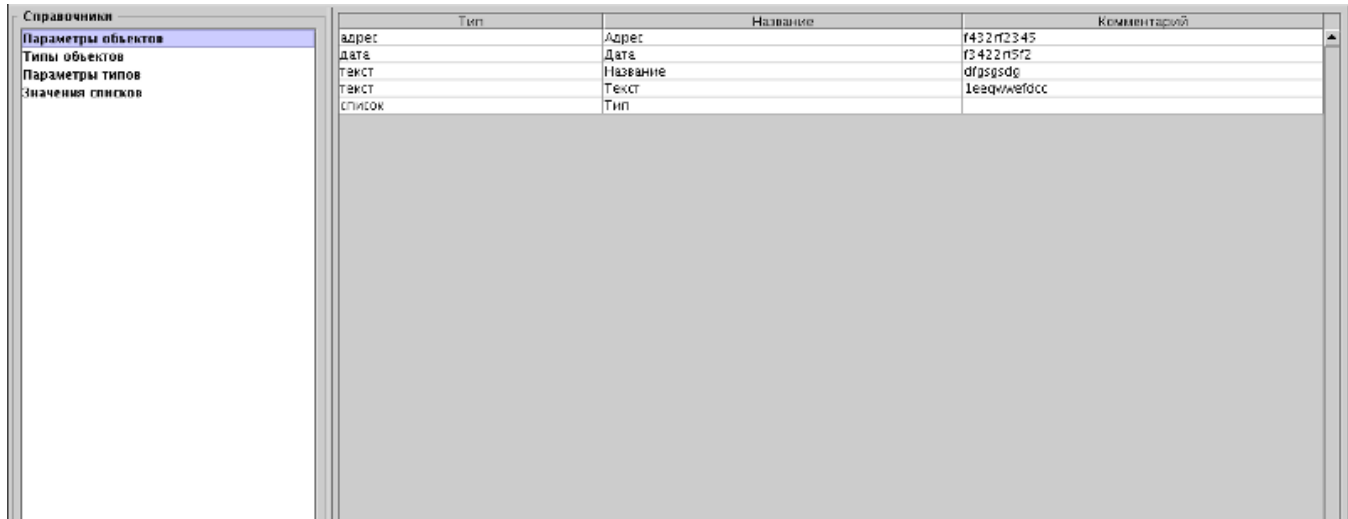
Переменные тарифа: все глобальные тарифные только переопределенные значения

Тип переменной	Глобальная	Название	Значение
Цена	<input type="checkbox"/>	Абонплата ТВ	300

Данный механизм выступает альтернативой созданию персональных тарифных планов.

Объекты

Понятие объекта введено в систему для упрощения учета сложных мультисервисных договоров. Объекты никак не влияют на обсчет договоров. Объект представляет из себя именованную сущность с набором параметров и привязанных сущностей из различных модулей. Например, возможно заведение в договоре нескольких точек подключений, к каждой из которых будет привязан свой диапазон адресов из IPN-модуля, установлен собственный адрес и тип точки. Использованию объектов предшествует заполнение справочников (меню **Справочники=>Объекты**). Параметры объектов могут быть следующих типов: Адрес, Текст, Список, Дата.

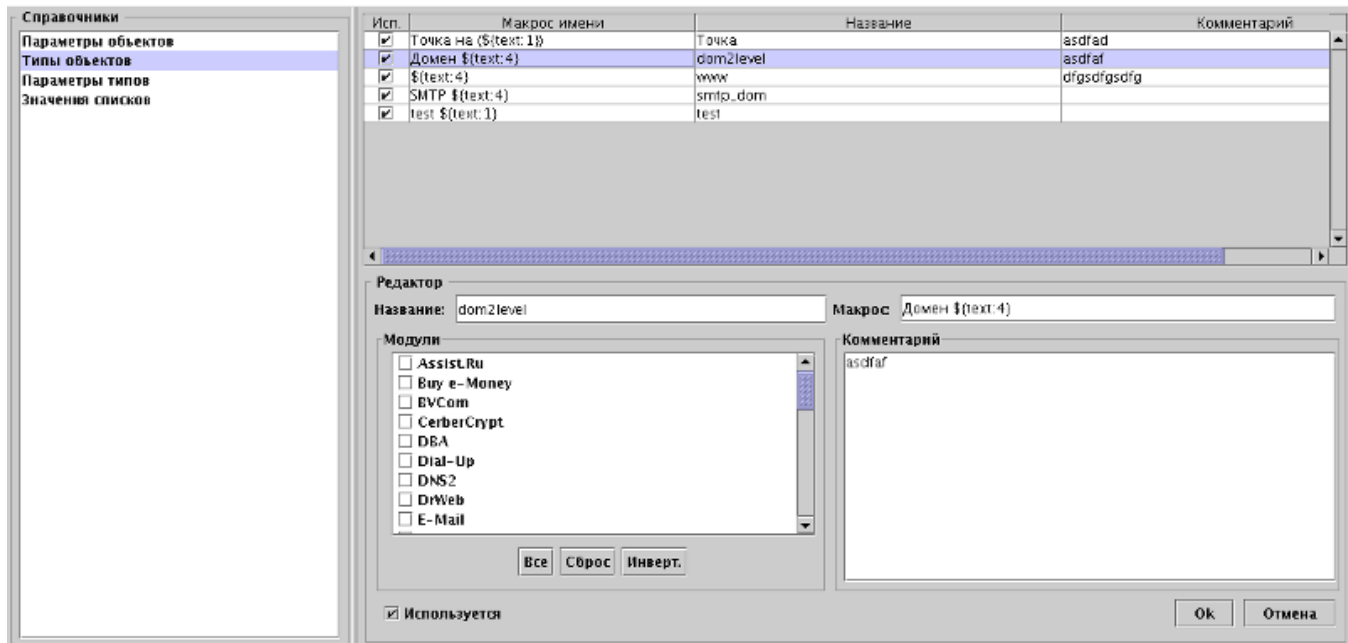


Тип	Адрес	Название	Комментарий
адрес	r432r2345		
дата	r3422r5r2		
текст	dfgsdgdg		
текст	1e2qWwefccc		
список			

В меню **Значения списков** должны быть определены допустимые значения для каждого спискового параметра.

В типах объектов определяются существующие в системе типы объектов и макрос их именования. В макросе именования могут быть подставлены следующие значения: `$(text:<код параметра>)` `$(address:<код параметра>)` `$(list:<код параметра>)` `$(date:<код параметра>)`.

Здесь же определяются привязанные к типу объекта модули, их свойства будут доступны в редакторе объекта в договоре.



Исп.	Макрос имени	Название	Комментарий
<input checked="" type="checkbox"/>	Точка на \$(text:1)	Точка	asdfad
<input checked="" type="checkbox"/>	Домен \$(text:4)	dom2level	asdfaf
<input checked="" type="checkbox"/>	\$(text:4)	www	dfgsdfgsdfg
<input checked="" type="checkbox"/>	SMTP \$(text:4)	smtp_dom	
<input checked="" type="checkbox"/>	test \$(text:1)	test	

Редактор
Название: dom2level Макрос: Домен \$(text:4)

Модули

- Assist.Ru
- Buy e-Money
- BVCom
- CerberCrypt
- DBA
- Dial-Up
- DNS2
- DrWeb
- E-Mail

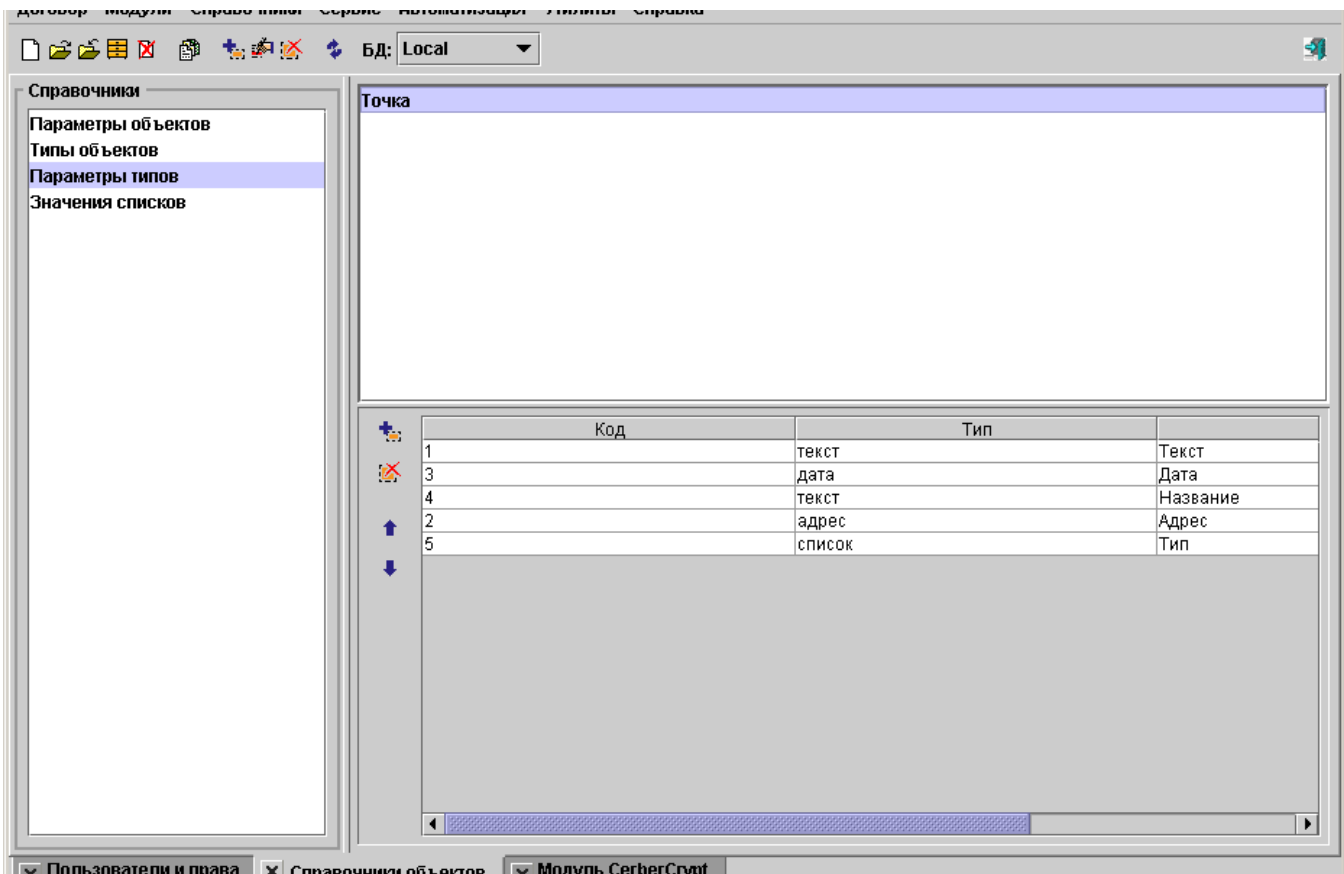
Все Сброс Инверт.

Используется

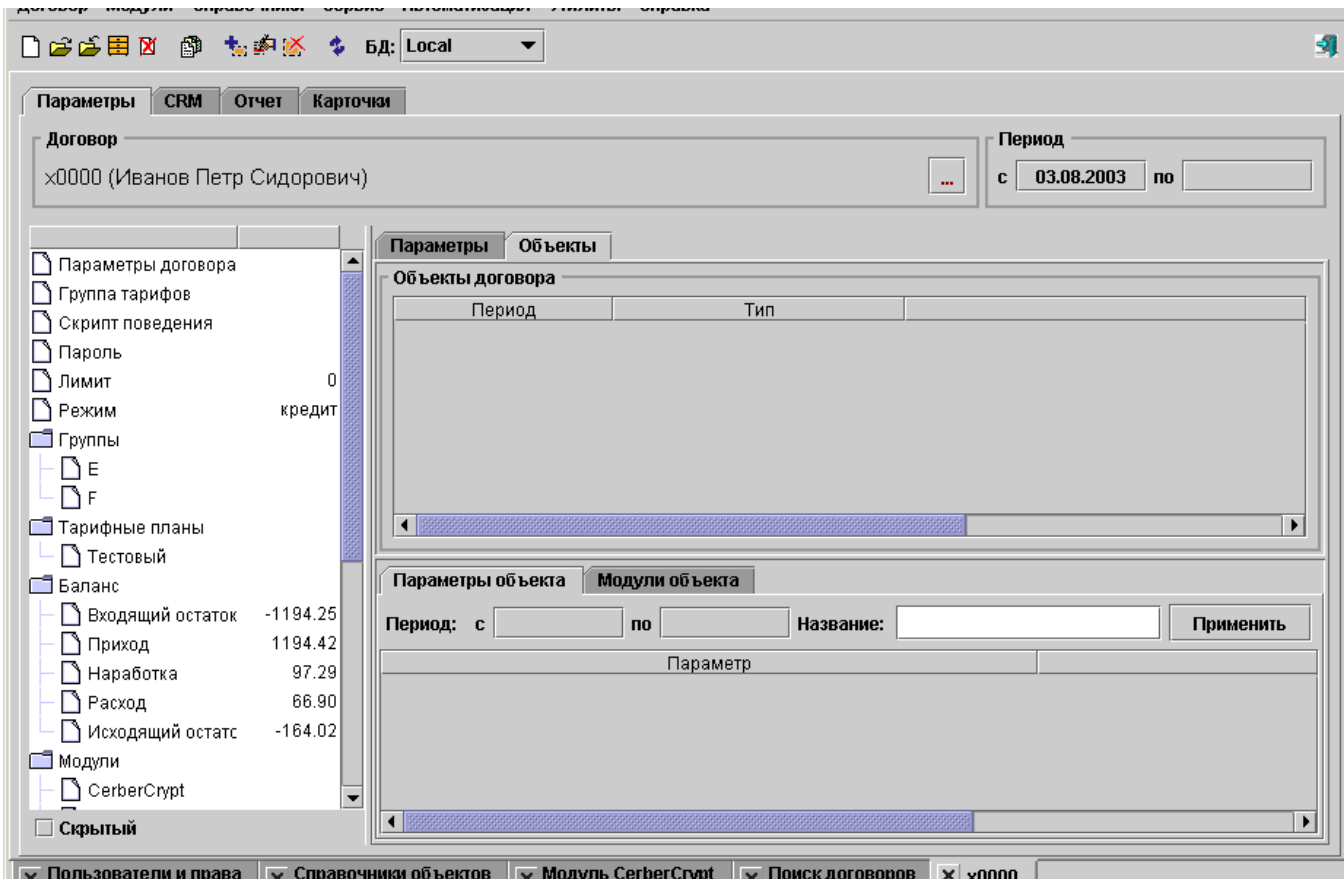
Комментарий: asdfaf

Ok Отмена

После определения типов и параметров необходимо произвести привязку параметров к типам.



Для добавления объекта в договор перейдите на вкладку **Параметры=>Объекты** в договоре.



Для добавления объекта выберите **Добавить** в верхней панели инструментов. Далее выбирается тип объекта и открывается его редактор.



После чего откроется редактор для занесения параметров объекта.

ич)

Период
... с 03.08.2003 по

Параметры | Объекты

Объекты договора

Период	Тип	Названи
--------	-----	---------

Параметры объекта | Модули объекта

Период: с по Название: Применить

Параметр	Значение
Текст	
Дата	23.05.2006
Название	Первый объект
Адрес	
Тип	

вочники объектов | x Модуль CerberСрут | x Поиск договоров | x x0000

Далее вводится название объекта, либо поле оставляется пустым и нажимается **Применить**, после чего имя объекта формируется на основании макроса.

дорович)

Период

... с 03.08.2003 по

0
цит

Параметры Объекты

Объекты договора

Период	Тип	Название
-	Точка	Точка на , Уфа, Ленинский, 1 квартал, Айская, д. 64, др. 3, кв. 49 , под., эт. н

Параметры объекта Модули объекта

Период: с по Название: Точка на , Уфа, Ленинский, 1 квартал, Айская, д. 64, др. 3, Применить

Параметр	Значение
Текст	
Дата	23.05.2006
Название	Первый объект
Адрес	, Уфа, Ленинский, 1 квартал, Айская, д. 64, др. 3, кв. 49 , под., эт.
Тип	fdgdfg

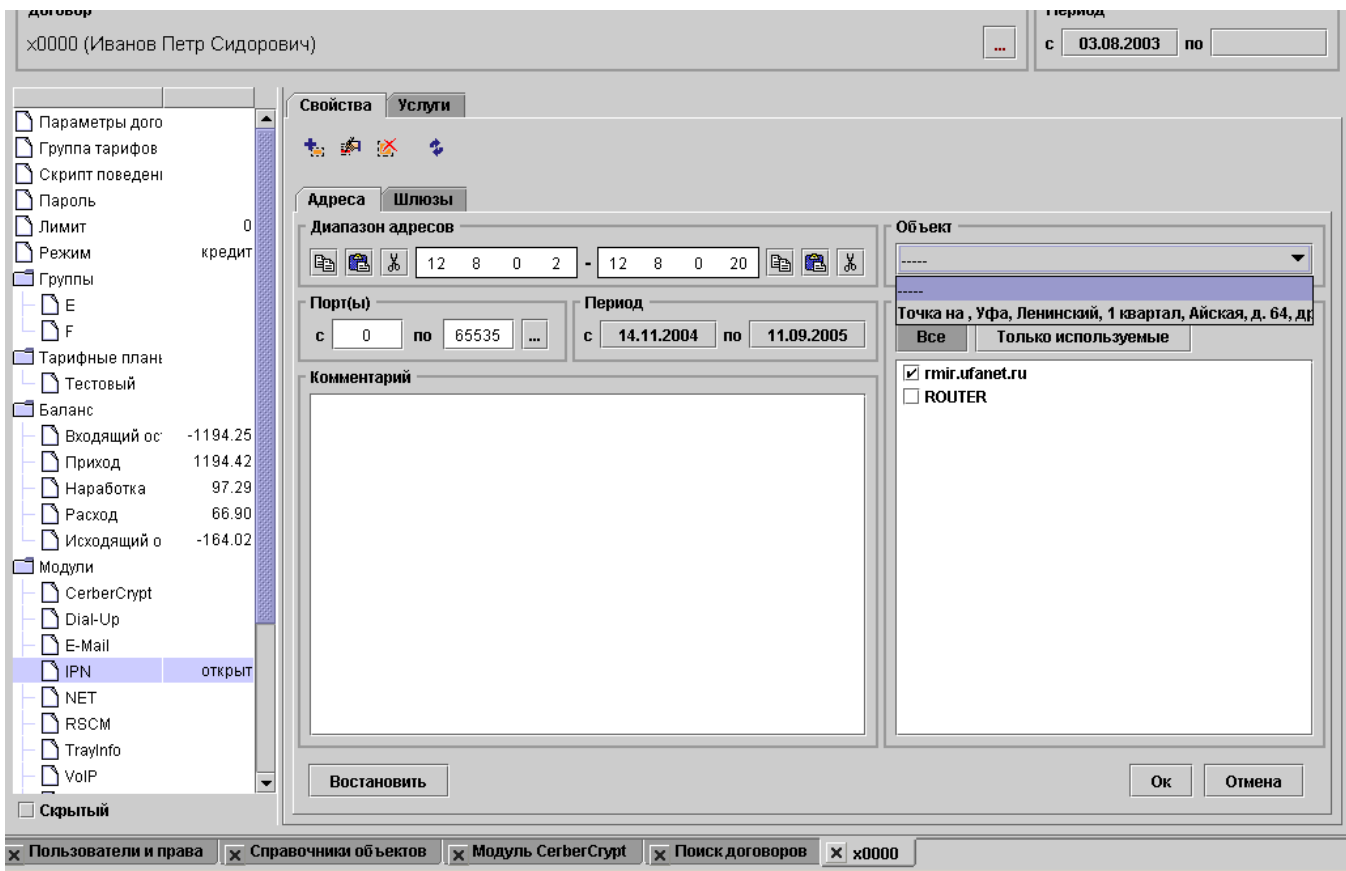
25
42
29
90
02

ыт

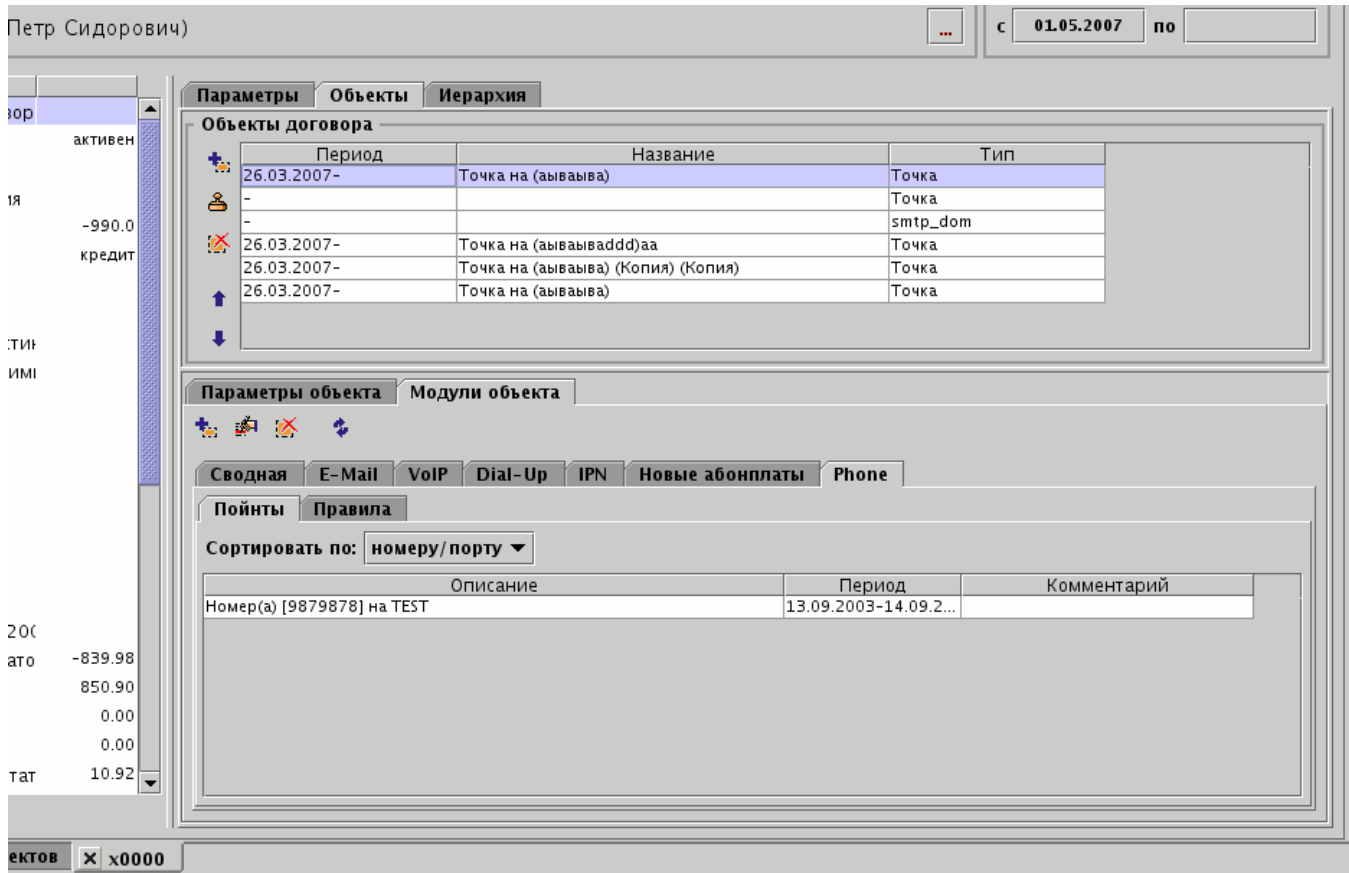
Справочники объектов | Модуль CerberCrypt | Поиск договоров | x лппп

Привязку сущности модуля к объекту можно осуществить двумя способами:

- 1) Для сопоставления объекту сущностей модуля в каждом модуле предусмотрено доп. поле.



2) В редакторе объекта доступны редакторы сущностей модулей, перечень которых указан в свойствах объекта



Каждая добавленная на закладке сущность модуля будет автоматически отнесена к выбранному объекту.

На вкладке **Сводная** отображается сводная таблица по всем подключенным к объекту сущностям модулей.

Модуль	Сущность	Период	Комментарий
Новые абонплаты	Абонплата: Абонплата	22.02.2008-	
Phone	Номера: 9879878	13.09.2003-14.09.2003	

Возможно перенести объект с текущего договора на другой открытый. Для этого необходимо выбрать объект в таблице, вызвать всплывающее контекстное меню кликом правой кнопки мыши и активировать пункт **Перенести на другой договор**.

В текущей версии при переносе объекта производится перенос только самого объекта и его параметров. Сущности модулей отвязываются от объекта, перенос наработки также не производится. В последующем функционал будет дореализован.

Поиск договора может осуществляться по параметрам привязанных к нему объектов.

договор модуль справочники сервис автоматизация тарифы справка

БД: Local

Договора | Параметры

Договоров | Объектов

Текст | Адрес

Условие

Улица: Айская

Дом: 64 / 3

Квартира: ком.

Х >>>

Параметры

Адрес

Договоров/объектов [найдено: 1] 1 из 1

x0000 [Иванов Петр Сидорович] Точка на , Уфа, Ленинский, 1 квартал, Айская, д. 64, др. 3, кв. 49 , под., эт. на

Web-интерфейс пользователя

Общие сведения

Web-страница пользователя предоставляет клиентам полную информацию о состоянии счета, детальные отчеты по всем модулям договора. Через персональный кабинет пользователь может также изменять пароли, получать и выставлять себе счета, производить оплату через различные платежные системы и т.п. в зависимости от состава установленных модулей и плагинов.

По умолчанию страница статистики доступна по адресу: <http://<адрес сервера биллинга>:8080/bgbilling/webexecuter>. Путь контекста (/bgbilling) может быть исправлен в файле конфигурации `BGBillingServer/data/data.properties` переменная `context.path`, в т.ч. установлен пустым (`context.path=`).

Нельзя ставить `context.path=`

При входе клиента на эту страницу происходит его перенаправление на страницу авторизации.



Авторизация

Номер договора:

Пароль:

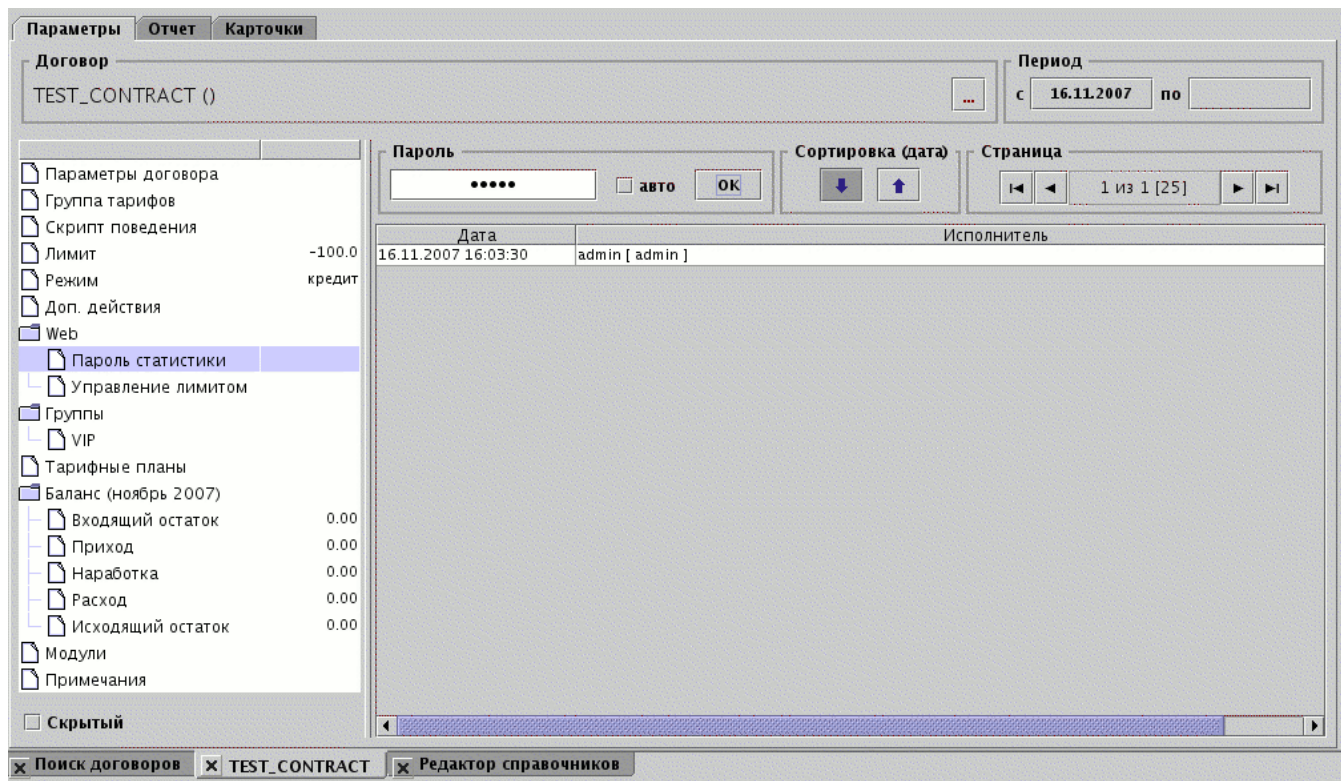
[Забыли пароль?](#)

© 2007-2021 ООО "Бител"

Настройка доступа к статистике

По умолчанию клиентам разрешена авторизация по номеру договора и паролю статистики, пароль статистики изменяется на вкладке **Web=>Пароль статистики** договора. Для смены пароля наберите его в текстовой области, либо установите галочку **авто** и нажмите **ОК**.


В таблице отображается лог смен пароля, его можно сортировать по дате по убыванию и возрастанию.




Текущий пароль статистики, установленный после создания договора, можно посмотреть в полной карте договора (вкладка **Карточки=>Полная карта** в договоре).

По умолчанию отображается примерно следующая страница пользователя. Содержимое меню может пополняться по мере подключения модулей и плагинов к системе.


ГЛАВНАЯ / Договор

Баланс**Остаток на 05.12.2022, 15:11****3 499,50 ₺**Лимит: 0,00  Статус: Активен**Платежи** 

08.11.2022	Банк	1 000,00 ₺
01.11.2022	Банк	2 161,50 ₺
05.07.2022	Сбербанк (карты)	103,00 ₺
Платеж через платежную систему CloudPayments (#9)		

Списания 

22.10.2022	Перенос средств	6 500,00 ₺
25.04.2022	Ticket	841,60 ₺
Активация тарифной опции test		
22.02.2022	Ticket	1 052,00 ₺
Активация тарифной опции test		

Наработка 

декабрь 2022	Вызов мастера	40,00 ₺
--------------	---------------	---------

Параметры авторизации на странице статистики задается в [конфигурации ядра биллинга](#).

```
web.auth.modes=0:1
```

Данное значение обозначает, что разрешена авторизация на странице статистики по номеру договора + паролю (вкладка **Пароль** в карточке договора).

Также есть режим авторизации, который использует параметр договора в качестве логина (работают разные типы параметров, трактуется как текст):

```
web.auth.modes=0:2  
# ,  
web.auth.contract.text.parameter=< >
```

Страница авторизации пользователя формируется шаблоном **login.xml**. В стандартной поставке шаблон содержит только форму авторизации по номеру договора (либо текстовому параметру договора) и паролю.

При правке этого шаблона не забудьте, что он перетирается при обновлении системы.

Клиент работает на странице статистики до нажатия пункта меню **Выйти**, либо до истечения таймаута. При выходе со страницы статистики клиент перенаправляется на страницу, задаваемую опцией.

```
web.exit.redirect=about:blank
```

По умолчанию это пустая страница, но вы можете разместить здесь URL страницы провайдера.

Сервер определяет тип авторизации по передаваемому в форме авторизации параметру **midAuth**=<коду модуля>. Если параметр отсутствует - авторизация идет по номеру договора (либо текстовому параметру договора) + паролю доступа к статистике.

Предположим что в системе существует модуль DialUP с кодом 21. Для того чтобы разрешить авторизацию через его логины с паролями нужно добавить опцию.

```
web.auth.modes=0:1;21:1
```

Теперь раскомментируйте вторую форму авторизации в **login.xml**.

```
<form method="post" action="webexecuter">
<input type="hidden" name="midAuth" value="21" />
<table align="center" width="300px" class="filter" style="margin-top:20px;">
  <tr>
    <th align="left"> DialUP:</th>
    <td><input class="logon" type="text" name="user" SIZE="15" /></td>
  </tr>
  <tr>
    <th align="left"> DialUP:</th>
    <td><input class="logon" type="password" name="pswd" SIZE="15" /></td>
  </tr>
  <tr>
    <td></td>
    <td><input type="submit" value="" style="width: 100%; border: 1px solid #404040;" /></td>
  </tr>
</table>
</form>
```

Теперь клиенту будет предлагаться два режима авторизации.



Номер договора:	<input type="text" value="x0000"/>
Пароль:	<input type="password" value="*****"/>
	<input type="button" value="Вход"/>
	Забыли пароль?
Логин DialUP:	<input type="text"/>
Пароль DialUP:	<input type="password"/>
	<input type="button" value="Вход"/>

Аналогично можно добавить авторизацию по логину и паролю Voip-модуля. Возможно в **midAuth**-параметре формы передавать несколько модулей через запятую, 0 - код модуля ядра. При этом будет осуществлен последовательный поиск в указанных модулях. Модуль должен быть разрешен в **web.auth.modes**.

Так же для модуля Dialup есть отдельный режим авторизации по ip-адресу сессии. Пусть 21 - это код модуля Dialup, тогда этот режим настраивается так

```
web.auth.modes=0:1;21:2
```

Где 2- означает авторизацию по ip-адресу сессии. При этом сравнивается http-заголовок с ip-адресом переданный в http-запросе и Ip-адрес сессии абонента.

Имя http-заголовка, в котором хранится ip адрес, задается параметром

```
header.name.remote.addr=
```

в конфигурации сервера.

При ошибке авторизации отображается страница, задаваемая шаблоном **error.xml**. Скорректируйте в данном шаблоне номер телефона вашей техподдержки. Если вы используете дополнительные режимы авторизации кроме договора, скорректируйте выводимый текст.

ОШИБКА: Неправильный пароль

Вы ввели не правильный пароль для договора - x0000

Возможные причины:

- Ошибка при вводе пароля
- Ошибка в номере договора
- Проверьте состояние клавиши Caps Lock и языковую раскладку клавиатуры
- Внимание пароль для доступа к серверу статистики ОТЛИЧАЕТСЯ от пароля для доступа по VPN. Пароль для доступа к серверу статистики указывается в карте регистрации выдаваемой при заключении договора.
- Попробуйте воспользоваться [системой восстановления пароля](#)
- Другие причины, для выяснения свяжитесь с расчетной службой по тел. +7 (342) 290-04-05

Попробуйте [войти еще раз](#)

Восстановление пароля

С помощью системы восстановления пароля пользователь может произвести отправку пароля на E-Mail.

Для восстановления пароля введите номер Вашего договора и email (указанный в договоре для функции восстановления пароля). Дальнейшие инструкции по восстановлению пароля будут высланы на указанный email.

Номер договора:

Номер договора состоит из нескольких букв в начале, затем несколько цифр, затем тире "-" и две цифры (последние две цифры от года, когда был заключен договор). Номер вводится слитно без пробелов. Например: NK00001-05, КФ123456-06 и т.д.

Email:

Email указанный в договоре для функции восстановления пароля.

Применить

В поле **Номер договора** пользователь должен ввести номер своего договора, а в поле E-mail - почтовый адрес, который должен совпасть со значением текстового параметра договора, содержащим E-Mail-адрес для восстановления пароля. Код этого параметра указывается в конфигурации сервера биллинга **contract.password.forgot.email.param.id**=<числовой код параметра>.

При использовании функции восстановления пароля высылается письмо со ссылкой, перейдя на которую пользователь попадает на страницу смены пароля Web-статистики. Текст, тема письма и URL-сервера статистики задаются в конфигурации сервера биллинга.

Защита от подбора пароля

Если пользователь несколько раз в течение короткого времени введет неверный пароль, доступ временно блокируется с указанием времени, до которого вход запрещен. Это мера безопасности от подбора пароля Web-статистики.

SYNTEGA

Ошибка при авторизации

ОШИБКА: Доступ заблокирован

Доступ к статистике по договору x0000 заблокирован до 22.02.2008 13:35:59

Возможные причины:

- Для защиты от подбора пароля к серверу статистики методом перебора, после каждой попытки войти на сервер статистики с неправильным паролем, доступ к серверу блокируется на некоторое время. После истечения этого времени дается возможность провести еще одну попытку войти на сервер, если будет вновь введен не правильный пароль доступ снова будет заблокирован на более длительный промежуток времени и т.д. После нескольких последовательных попыток входа на сервер с неправильным паролем, доступ на сервер блокируется на несколько часов.
- Доступ заблокирован администратором, свяжитесь с службой поддержки по тел. +7 (342) 290-04-05
- Другие причины, для выяснения свяжитесь с расчетной службой по тел. +7 (342) 290-04-05

После окончания времени блокировки Вы можете попробовать войти еще раз. Предварительно убедитесь, в правильности вводимых данных.

Обратите внимание на:

- Состояние клавиши Caps Lock и языковую раскладку клавиатуры
- Пароль для доступа к серверу статистики ОТЛИЧАЕТСЯ от пароля для доступа по VPN. Пароль для доступа к серверу статистики указывается в карте регистрации выдаваемой при заключении договора.
- Если не уверены в корректности вводимого пароля, попробуйте воспользоваться [системой восстановления пароля](#)

[Войти](#)

Временные характеристики защиты могут быть настроены в конфигурации сервера. Переменная **logon.counter.max** задает максимальное количество неудачных попыток авторизации для договора подряд. После каждой попытки может быть установлен таймаут, базовый размер которого определяется переменной **logon.timeout.period**. После первой неудачи таймаут равен периоду, после второй - двум периодам, затем трем и т.п. Алгоритм увеличения периода задается переменной **logon.timeout.action**. После исчерпания **logon.counter.max** попыток авторизации договор блокируется на время **logon.timeout.lock** секунд. После изменения этих параметров сервер, возможно, потребует перезагрузить.

Оператору биллинга на вкладке Web договора доступны функции мониторинга доступа к Web-статистике.

Договор: x0000 (Иванов Петр Сидорович) Период: с 01.05.2007 по

Последний вход: Статус ОШИБКА (2раз подряд) последняя в 22.02.2008 13:35:27 с 127.0.0.1 Разблокировать

Счетчик обращений: 0 Лимит обращений: Общий (20) Отключен Персональный: Применить

Входы Ошибки

Фильтр: с по Обновить 1 из 2 [25]

Дата и время	IP адрес	Логин	Ошибка
15.01.2008 16:06:10	127.0.0.1	x0000	[1] pswd = 123456
15.01.2008 19:12:27	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:13:04	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:13:54	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:14:36	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:32:23	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:32:28	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:32:41	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:35:16	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:36:35	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:39:21	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:39:41	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:40:20	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:40:23	127.0.0.1	x0000	[1] pswd =
15.01.2008 19:42:27	127.0.0.1	x0000	[1] pswd =
16.01.2008 12:30:18	127.0.0.1	x0000	[1] pswd =
16.01.2008 13:08:19	127.0.0.1	x0000	[1] pswd = 123451
16.01.2008 13:09:57	127.0.0.1	x0000	[1] pswd = 123451

Справочники объектов: x0000 Поиск договоров: K000001

На вкладке **Входы** отображаются входы на Web-статистику. На вкладке **Ошибки** - ошибки авторизации с отображением введенных логина и пароля. Кнопка **Разблокировать** позволяет досрочно открыть закрытый из-за превышения количества неудачных попыток доступ к статистике.

Счетчик обращений и **лимит обращений** - это защита от автоматизации пользователями получения информации о балансе (написание различных скриптов, запрашивающих статистику, что ведет к сильной загрузке сервера), возможно установление лимита на количество обращений к статистике HTTP-запросов параметром **web.max.day.request.count** в конфигурации сервера биллинга.

Кнопкой **Сброс** можно сбросить значение счетчика, также можно установить для каждого договора пользовательское значение, выбрав опцию **Персональный**, введя число в поле ввода и нажав **Применить**.

Обращением считается каждая перегрузка страницы статистики (переход по ссылке в меню, вывод данных отчета). При превышении количества обращений пользователь будет перенаправлен на страницу, формируемую шаблоном **limit_error.xml**.

Для продолжения работы введите символы, показанные на картинке.
Буквы вводятся без учета регистра.



Введя код, пользователь сбрасывает счетчик обращений и может продолжать работу на странице статистики. Введение кода подтверждает, что это не бот и не программа.

Настройка страницы статистики

Для смены логотипа следует заменить файл **BGBillingServer/webroot/img/logo.gif** нужным логотипом. По умолчанию в дистрибутиве идет логотип оператора Synterra в качестве примера. Цветовую гамму Web-интерфейса можно поменять в стилевой таблице **style.css**.

Страницы Web-интерфейса пользователя собираются из **XML**-документа с использованием **XSLT**-шаблона. Каждый модуль и плагин помещает свои XSLT-шаблоны в каталог **BGBillingServer/webroot/xsl**. Модифицируя их, вы можете настраивать оформление страницы пользователя. Главный шаблон - **main.xsl**.

При каждом обновлении модуля и ядра XSLT-шаблоны [перетираются](#).

Достоинством первого метода является снижение нагрузки на сервер, а недостатком - не все браузеры смогут обработать такой ответ. Также подобный режим можно использовать для интеграции со сторонними приложениями

Параметр **web.xslt** конфигурации сервера - URL папки, где будут находиться ваши XSLT-шаблоны. При использовании режима **xml** адрес **127.0.0.1** следует заменить на адрес машины с сервером биллинга, доступный пользователям из внешней сети.

Название пунктов меню в каждом модуле задаются в его конфигурации, например для ядра:

```
web.menuItem1=  
web.menuItem2=  
web.menuItem3=  
web.menuItem4=  
web.menuItem5=  
web.menuItem6=  
web.menuItem7=  
web.menuItem8=  
web.menuItem9=  
web.menuItem10=  
web.menuItem11=
```

Перечень пунктов вы можете найти в документациях к модулям. Если название пункта меню не указано, берётся значение по умолчанию. Для того, чтобы заблокировать пункт меню нужно указать его название **none**. Например:

```
web.menuItem1=none
```

XSLT-процессор получает XML-данные от сервера и на основании шаблонов создаёт XML-страницу. Чтобы просмотреть XML-документ, на основании которого создаётся страница, установите режим работы Web-статистики в **xml**, либо добавьте к URL в браузере строку **&ct=xml**, далее вызовите страницу и сделайте просмотр её исходного кода.

```
Исходный код: http://127.0.0.1:8080/bgbilling/webexecuter?ct=xml - Mozilla Firefox
Файл  Правка  Вид  Справка

<?xml version="1.0" encoding="windows-1251"?>
<?xml-stylesheet type="text/xsl" href="http://127.0.0.1:8080/bgbilling/xsl/main.xsl"?>
<data action="menu" balance_mode="0" comment="Иванов Петр Сидорович"
contract="x0000" fc="0" id="455" mid="contract" module="contract" title="x0000">
  <menu>
    <item action="Account" mid="1" module="email" title="Управление ящиками"/>
    <item action="Forward" mid="1" module="email" title="Пересылки"/>
    <item action="Password" mid="1" module="email" title="Смена пароля на ящики"/>
    <item action="MPS" mid="70" module="mps" title="Платежные системы"/>
    <item action="ShowStat" mid="33" module="ipn" title="IP-статистика (IPN)"/>
    <item action="GateControl" mid="33" module="ipn" title="Управление шлюзом (IPN)"/>
    <item action="Card" mid="5" module="card" title="Пополнение счета с помощью Интернет-карт"/>
    <item action="TrayInfo" mid="39" module="trayinfo" title="Активация TrayInfo"/>
    <item action="Bill" mid="65" module="bill" title="Счета"/>
    <item action="Invoice" mid="65" module="bill" title="Счета-фактуры"/>
    <item action="ShowSessions" mid="6" module="voiceip" title="Просмотр сессий VoiceIp"/>
    <item action="ChangeLoginPassword" mid="6" module="voiceip" title="Смена пароля на логины VoiceIp"/>
    <item action="LoginDirect" mid="6" module="voiceip" title="Наработка логина VoiceIp по направлениям"/>
    <item action="Ticket" mid="110" module="ticket" title="Ковшечек"/>
    <item action="Subscription" mid="43" module="cerbercrypt" title="Управление подпиской CerberCrypt"/>
    <item action="VirtualCinema" mid="43" module="cerbercrypt" title="Виртуальный кинотеатр CerberCrypt"/>
    <item action="ReportSession" mid="73" module="phone" title="Просмотр сессий Телефонии"/>
    <item action="ReportAmount" mid="73" module="phone" title="Наработка по Телефонии"/>
    <item action="ReportDirect" mid="73" module="phone" title="Наработка по направлениям Телефонии"/>
    <item action="ReportService" mid="73" module="phone" title="Наработка по услугам Телефонии"/>
    <item action="ReportServiceSession" mid="73" module="phone" title="Детализация по Телефонии"/>
    <item action="ShowSessions" mid="21" module="dialup" title="Просмотр сессий DialUp"/>
    <item action="ShowLoginsBalance" mid="21" module="dialup" title="Наработка по логинам DialUp"/>
    <item action="ChangeLoginPassword" mid="21" module="dialup" title="Смена пароля на логины DialUp"/>
    <item action="ShowPeriods" mid="21" module="dialup" title="Учетные периоды"/>
    <item action="WM" mid="54" module="wm" title="Оплата через WebMoney"/>
    <item action="ContractService" mid="60" module="rscm" title="Разовые начисления"/>
    <item action="HelpDeskTable" mid="0"
      module="ru.bitel.bgbilling.plugins.helpdesk" title="HelpDesk2"/>
    <item action="DocumentsTable" mid="0"
      module="ru.bitel.bgbilling.plugins.documents" title="Документы2"/>
  </menu>
  <news>
    <item date="16.05.2007" title="2"/>
    <item date="16.05.2007" title="1"/>
    <item date="07.12.2006" title="Дешевый трафик с UfaPhoto.Ru">
      <data>
        <p>
          style="margin:0px;padding:0px;text-indent:20px;">C <b>1 сентября</b> трафик с уфимского фотопортала UfaPhoto.RU для абонентов Ufanet считается
          href="http://www.ufaphoto.ru" target="_blank" title="Откроется в новом окне">UfaPhoto.RU</a>
        </p>
      </data>
    </item>
  </news>
</data>
```

Возможно добавление в XML-документ дополнительных данных по договору установкой опции в конфигурации сервера.

```
web.add.contract=1
```

Это создаёт дополнительную нагрузку на сервер биллинга, но позволяет разместить на странице пользователя дополнительную информацию, отсутствующую в стандартном дереве.


```

dt="2007-05-07 23:53:14.0" subject="Перенесен логин" uid="0"/>
<contract_comment
comment="Логин 1 перенесен из договора x0000 (Иванов Петр Сидорович) в договор x0002 (Aoi7ie oaoioaue)"
dt="2007-05-08 00:37:30.0" subject="Перенесен логин" uid="0"/>
<contract_comment
comment="Логин 1 перенесен из договора x0000 (Иванов Петр Сидорович) в договор x0002 (Aoi7ie oaoioaue)"
dt="2007-05-08 00:39:57.0" subject="Перенесен логин" uid="0"/>
<contract_comment
comment="Логин 2 перенесен из договора x0000 (Иванов Петр Сидорович) в договор x0002 (Aoi7ie oaoioaue)"
dt="2007-05-08 00:48:15.0" subject="Перенесен логин" uid="0"/>
<contract_comment
comment="Логин 2 перенесен из договора x0000 (Иванов Петр Сидорович) в договор x0002 (Aoi7ie oaoioaue)"
dt="2007-05-08 00:55:57.0" subject="Перенесен логин" uid="0"/>
<contract_comment comment="фыв" dt="2007-05-10 15:52:59.0"
subject="фыв" uid="0"/>
<contract_comment
comment="Логин 2 перенесен из договора x0002 (Aoi7ie oaoioaue) в договор x0000 (Иванов Петр Сидорович)"
dt="2007-05-14 16:33:02.0" subject="Перенесен логин" uid="0"/>
<contract_comment comment="sdfsd" dt="2007-08-06 12:02:44.0"
subject="sdf" uid="0"/>
<contract_comment comment="ghh" dt="2007-08-06 12:03:47.0"
subject="gf" uid="1"/>
<contract_comment
comment="Email sdfsd@ufamail.ru перенесен из договора x0000 (Иванов Петр Сидорович) в договор x00001 ()"
dt="2007-08-24 13:28:46.0" subject="Перенесен email" uid="1"/>
<contract_comment
comment="Email sdfsd@ufamail.ru перенесен из договора x0000 (Иванов Петр Сидорович) в договор x00001 ()"
dt="2007-08-24 13:30:09.0" subject="Перенесен email" uid="1"/>
<parameters>
<parameter pid="1" pt="1"
title="Краткое название организации" value="СЛОБОДЧИКОВ ВЛАДИМИР иванович"/>
<parameter pid="2" pt="1"
title="Полное название организации" value="sdsdfffffasdasghghjasad"/>
<parameter pid="3" pt="1" title="Должность руководителя" value="Директор"/>
<parameter pid="4" pt="1" title="Ф.И.О. руководителя" value="asdsadasd"/>
<parameter pid="5" pt="1" title="Факс" value="asd"/>
<parameter pid="6" pt="1" title="Телефон(ы)" value="15"/>
<parameter pid="7" pt="1" title="Контактное лицо" value="20"/>
<parameter pid="9" pt="1" title="Адрес (юридический)" value="sdsdsaddsasd"/>
<parameter pid="8" pt="1" title="E-mail" value="sef"/>
<parameter pid="10" pt="1" title="Адрес (почтовый)" value="sd"/>
<parameter pid="11" pt="1" title="Адрес (физический)" value=""/>
<parameter pid="12" pt="1"
title="Адрес для рассылки о долге" value="shamil@itel.ru"/>

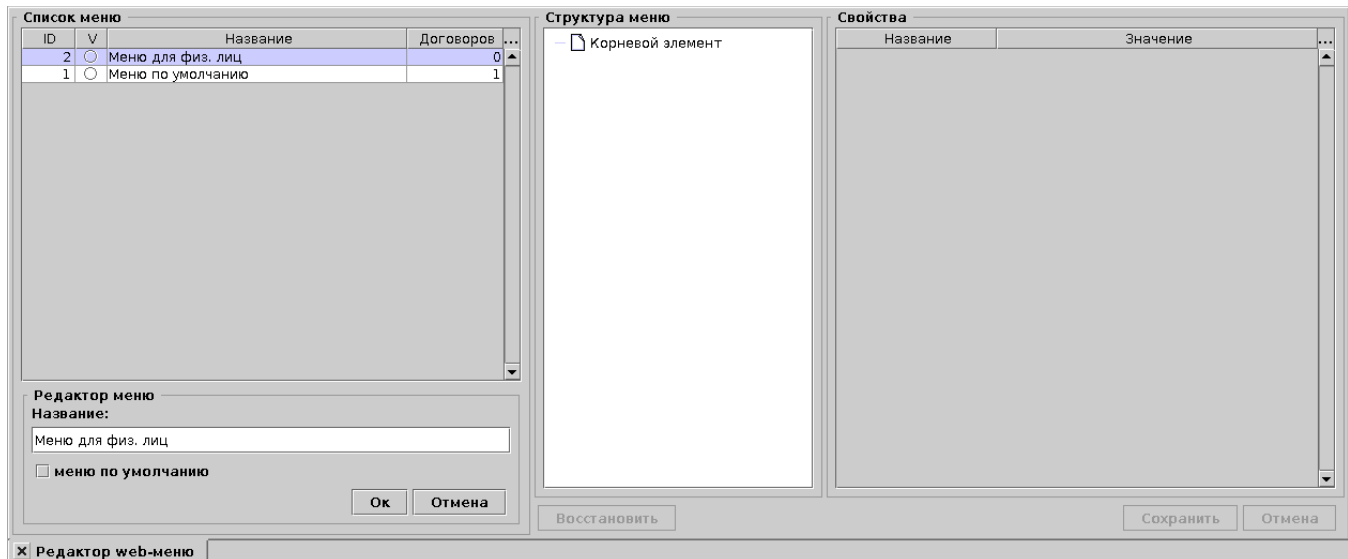
```

рока 69, Символ 58

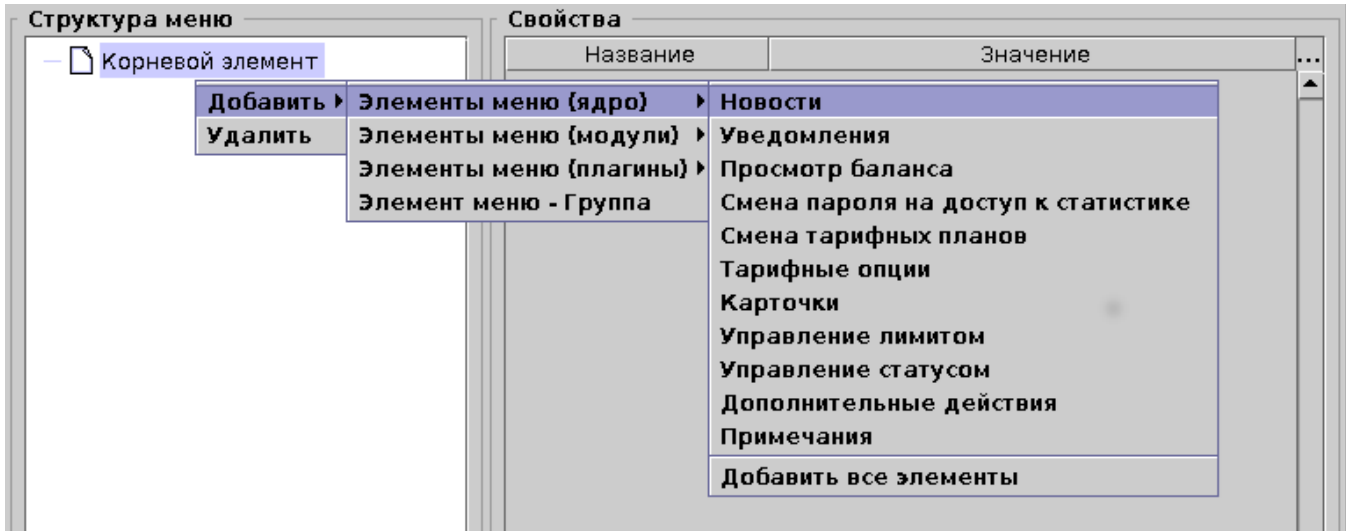
Таким образом на странице статистики можно разместить, например, некоторые параметры договора.

Начиная с версии 6.0 в системе доступен редактор web-меню, доступный из меню Сервис = Настройка = Редактор web-меню. Можно создать несколько меню, назначить созданные меню на договора и указать в шаблонах. Одно из созданных меню можно назначить как меню "по умолчанию". Если в редакторе не создано ни одного меню или ни одно из них не указано как меню "по умолчанию", то меню будет генерироваться автоматически, так же как это было в предыдущих версиях.

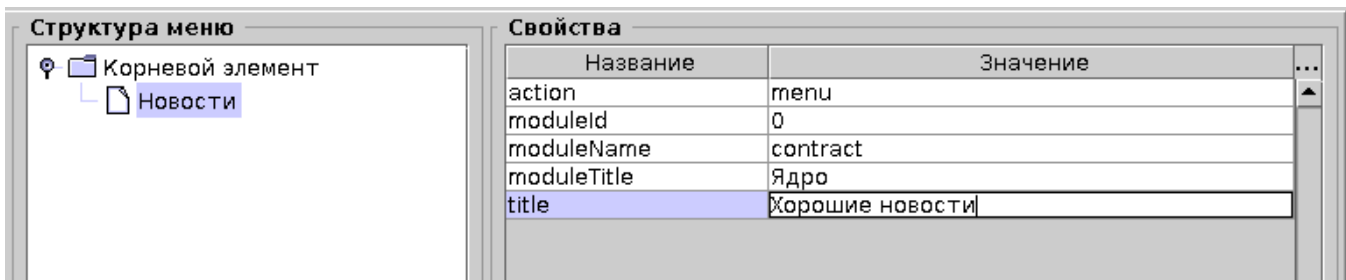
Редактор web-меню.



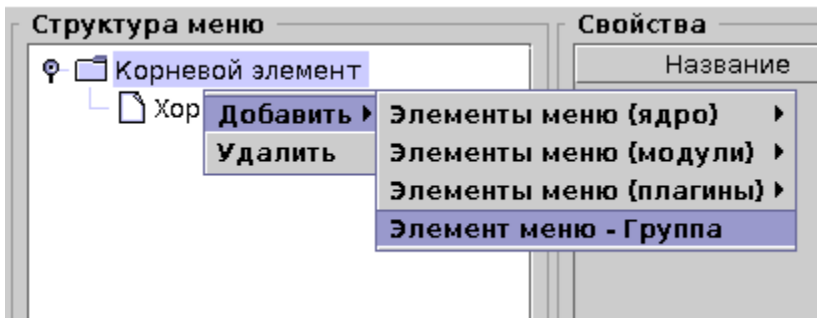
Добавление пункта web-меню.



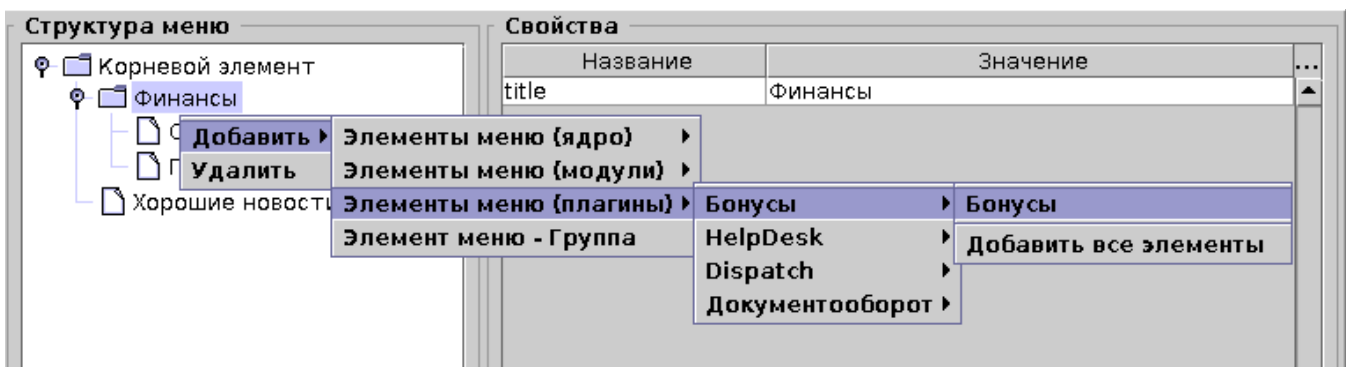
Переименование пунктов web-меню.



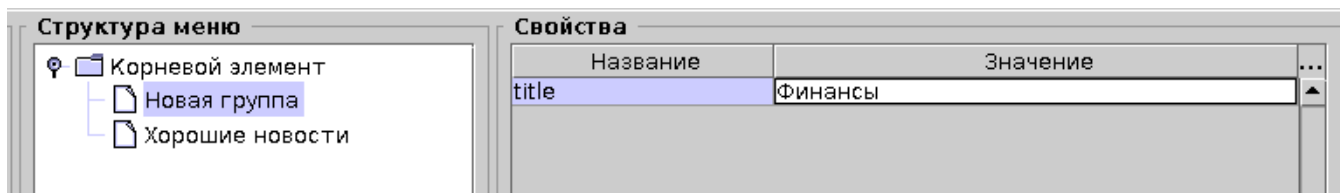
Добавление группы.



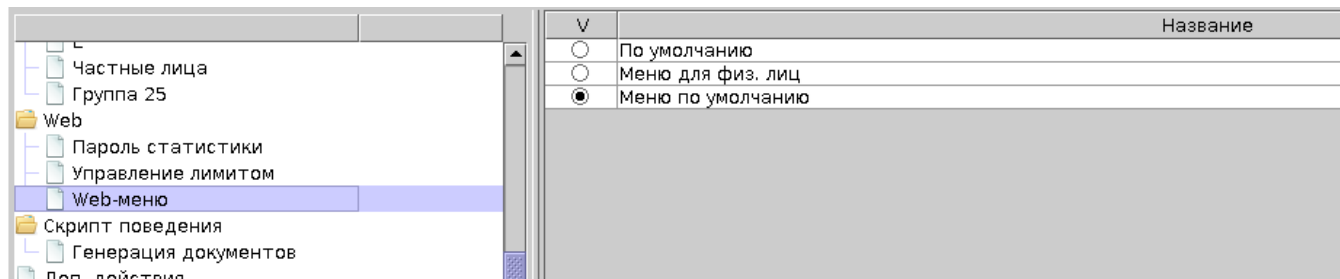
Добавление плагинного пункта web-меню.



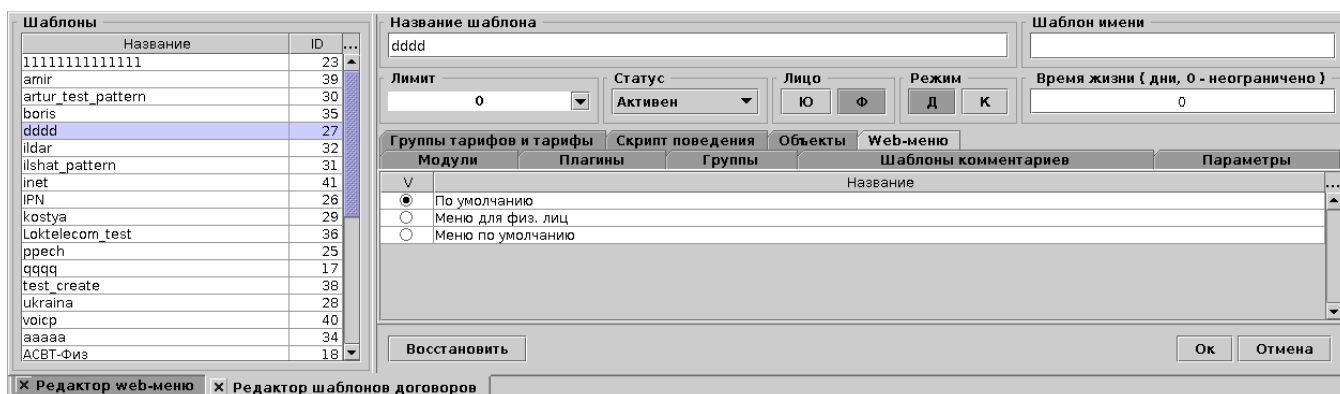
Переименование группы web-меню.



Назначение меню на договор.



Задание меню в шаблоне договора.



Обзор Web-интерфейса пользователя

Новости

Отображаются на первой странице кабинета пользователя.

Для редактирования новостей зайдите в **Сервис=>Настройка=>Редактор новостей**. Нажмите кнопку **Новый элемент**, введите тему и текст новости. При установленном фильтре групп, новость будет отображена всем пользователям, иначе только выбранным группам.

Список новостей

Заголовок: Торжественный пуск нового биллинга

Дата: 19.11.2007

Редактор: <data> Поздравляем всех с ПРАЗДНИКОМ !!!
 Желаем счастья и т.д и т.п </data>

Группы договоров: VIP

Восстановить

Все Сброс Инверт. Ок Отмена

Поиск договоров TEST_CONTRACT Редактор новостей

Чтобы работали html-теги необходимо заключить новость в xml-теги <data></data>. В этом случае содержащаяся внутри информация должна быть правильным xml-документом, т.е все теги должны быть закрыты, все атрибуты должны быть в ""

```
<data>
  <b></b>!!!<br/>
  .
  .
</data>
```

После того как клиент зайдёт на свою страницу статистики он увидит справа от меню вашу новость.

Меню

- »» Новости
- »» Просмотр баланса
- »» Смена пароля на доступ к статистике
- »» Подписка на рассылки
- »» Смена тарифных планов
- »» Управление лимитом
- »» Дополнительные действия
- »» Выход

НОВОСТИ

19.11.2007 Торжественный пуск нового биллинга
 Поздравляем всех с ПРАЗДНИКОМ!!!
 Желаем счастья и т.д и т.п

Просмотр баланса, истории платежей, расходов и наработки

Меню **Просмотр баланса** отображает на выбранный месяц входящий и исходящий остатки на счету пользователя, приходы, расходы и наработку договора. Приходы и расходы отображаются с комментариями.

Меню

- »» Новости
- »» Просмотр баланса
- »» Смена пароля на доступ к статистике
- »» Подписка на рассылки
- »» Смена тарифных планов
- »» Управление лимитом
- »» Дополнительные действия
- »» Выход

Отчет по платежам и расходам

Месяц: Год:

Описание	Сумма, руб.
Входящий остаток на начало месяца	0.00
Приход за месяц (всего)	100.00
16.11.2007 Платеж в кассу	100.00
Расход за месяц (всего)	-300.00
16.11.2007 Подключение	-300.00
Наработка за месяц (всего)	0.00
Исходящий остаток на конец месяца	-200.00
Лимит	-200.00

Смена пароля на доступ к статистике

Позволяет клиенту изменить пароль договора для доступа к статистике.



Меню	
>>	Новости
>>	Просмотр баланса
>>	Смена пароля на доступ к статистике
>>	Подписка на рассылки
>>	Смена тарифных планов
>>	Управление лимитом
>>	Дополнительные действия
>>	Выход

Смена пароля

Новый пароль:	<input type="text"/>
Повторите пароль:	<input type="text"/>
<input type="button" value="Сохранить"/>	

Тарифные опции

Страница тарифных опций позволяют абоненту просматривать активные тарифные опции и активировать новые, тем самым, например, меняя подключенные услуги, скорость соединения.

Карточки

У абонентов есть возможность загружать на свой компьютер карточки своего договора в формате PDF.

Договор № artur_test

Меню	
>>	Новости
>>	Просмотр баланса
>>	Смена пароля на доступ к статистике
>>	Подписка на рассылки
>>	Смена тарифных планов
>>	Тарифные опции
>>	Карточки
>>	Управление статусом
>>	Дополнительные действия
>>	Примечания
>>	Оплата через BVCom
>>	IP-статистика (IPN)
>>	Управление правилами шлюзо(IPN)
>>	HelpDesk
>>	Документы
>>	Выход

Карточки

Карточка	Ссылка для загрузки
AA	Скачать
Карта Договора	Скачать
ЗАЯВЛЕНИЕ	Скачать
Дополнительное соглашение	Скачать
АСВТ(договор)	Скачать
DDDDDD	Скачать
test	Скачать
TEST	Скачать

Загрузка карточки происходит после нажатия на ссылку **Скачать**. Для просмотра сохраненной карточки необходимо установить программу для просмотра PDF-документов (например: Adobe Acrobat Reader, STDUViewer и др.).

Дополнительные действия

В данном пункте возможен вывод с помощью скриптов BGBS [дополнительных действий](#) для вызова пользователем.

Параметра договора

При необходимости можно вывести в Web-интерфейсе параметры договора. Для включения данного пункта меню необходимо в конфиге прописать **web.menuItem12=Параметры договора** или настроить в **Редакторе web-меню**

Какие параметры должны отображаться в Web-интерфейсе, задается в редакторе параметров выбором двух флагов **Чтение в ЛК** и **Правка в ЛК**

Справочники		Параметры						
Адрес - страны		ID	Тип поля	Наименование поля	Чтение в ЛК	Правка в ЛК	История	...
Адрес - города		1	Текст	Краткое название организации	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Адрес - районы		2	Текст	Полное название организации	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Адрес - улицы		3	Текст	Должность руководителя	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Адрес - кварталы		4	Текст	Ф.И.О. руководителя	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Адрес - дома		5	Текст	Факс	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Договоры - параметры		6	Текст	Телефон(ы)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Договоры - группы параметров		7	Текст	Контактное лицо	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Договоры - группы								

В зависимости от выбранных флагов, параметр в Web-интерфейсе не будет отображаться вообще, не будет отображаться, но его можно будет редактировать, будет выводиться, но без возможности редактирования и будет отображаться и доступен для редактирования

Меню	Параметры договора		
» Параметры договора	Параметр	Значение	
» Новости	Краткое название организации	ООО "Бител"	[правка]
» Управление лимитом	АДРЕС	< СКРЫТО >	[правка]
» Управление статусом	Адрес(e-mail)	ООО "Бител" <info@bitel.ru>	[правка]
» Смена тарифных планов	Нужность	нет	
» Уведомления	Дата включения	12.02.2013	
» Тарифные опции	Телефон (новый)	+7-347-292-48-23 [тест]	[правка]
» Дополнительные действия			
» Карточки			
» HelpDesk			
» Параметры			

Примечания

В данном меню отображаются разрешённые к показу пользователю [примечания](#) договора.

Смена тарифных планов

Биллинг поддерживает возможность смены тарифов пользователями через Web-интерфейс пользователя.

Процесс смены тарифа может быть изменён с штатной логики на любую другую путём обработки событий **Смена тарифа по заданию пользователя** и **Запрос дат, с которых разрешена смена тарифа через Web**. Примеры обработки данных событий вы можете посмотреть в [Wiki](#).

Для предоставления пользователю смены тарифов необходимо установить для договора группу (группы) тарифных планов. Если хотя бы одна группа тарифных планов не указана, то смена их через Web-интерфейс невозможна. Группа определяет набор взаимозаменяемых тарифов договора. Обычно это линейка тарифов на одну услугу.

Рассмотрим случай наличия в системе 3х тарифных планов Тариф1..Тариф3. При этом переход между ними необходимо разрешить только в начале месяца. Предполагается что тарифы уже созданы в редакторе тарифных планов.

В **Справочники=>Группы тарифов** создадим группу с названием "Первая" и установим следующие параметры.

Название	Начало периода	Конец периода
Тариф1		
Тариф2	08.11.2009	
Тариф3	10.11.2009	13.11.2009

Кроме перечня разрешённых тарифов и момента, когда можно менять тариф, необходимо установить:

1. Позиция, которую тарифные планы данной группы занимают в договоре;
2. Количество дней, следующих после текущей даты, в которые может быть добавлено задание на переход. Например, если сейчас 20-е число месяца, в котором 30 дней и в данной опции стоит, что задания можно добавлять вперёд на 30 дней, то будут просмотрены даты от 20-го числа текущего месяца до 20 числа последующего. При этом будут выбраны даты разрешённых переходов. В данном случае подойдёт только 1-ое число последующего месяца. Если данный параметр установить в 60, то пользователь сможет генерировать задания за 2 месяца.

Также обратите внимание, что для каждого тарифа может устанавливаться начальная и/или конечная даты периода, когда эти тарифы видны в Web-интерфейсе пользователя и доступны для выбора. Это дополнительное средство обеспечения гибкости. Можно, например, сделать доступными некоторые из тарифов группы через некоторое время. Но это не имеет никакого отношения к возможности установки на какое-то число начала действия тарифа. Это только период, когда тарифы видны в списке выбора тарифов для смены.

Теперь необходимо установить в договор группу тарифов и скрипт поведения. Сделать это следует, открыв договор и щёлкнув на дереве **Группа тарифов**. Значение добавляется с периодом. На страничке Web-статистики пользователя должно появиться меню **Смена тарифных планов**. В верхней таблице приведена история смены тарифов.

Тарифный план	С даты	По дате	Сменить
TEST1	01.08.2006		
Тариф1	01.10.2008		

Перейти на тариф: Тариф2 с даты: 01.11.2008 Сменить тариф

Обратите внимание, что система предлагает к смене только тарифы, входящие в установленные на договоре группы тарифов. Не разрешается переходить на неиспользуемые тарифные планы. После смены тарифа страничка примет следующий вид.

Управление тарифными планами			
Тарифный план	С даты	По дате	Сменить
TEST1	01.08.2008		
Тариф1	01.10.2008	31.10.2008	
Тариф2	01.11.2008		

Также для пользователей имеется возможность отменить выбор тарифа для перехода будущим числом. Таким образом, можно отменить ошибочный переход. Обратите внимание, что если на событие смены тарифа выполняются какие-либо действия, то в событии отмены перехода (см. ниже) нужно это предусмотреть и корректно обработать. Как известно, событие смены тарифа отрабатывает в момент выбора тарифа (даже, если тариф начнёт действовать только в следующем месяце). Алгоритм обработки этой ситуации работает следующим образом:

1. при смене тарифа сохраняется информация с какого тарифа договора происходит переход (id строки из contract_tariff);
2. на будущих тарифах в Web-интерфейсе есть кнопка отмены перехода на тариф (и указано для удобства на какой тариф откатится, ведь мы знаем предыдущий тариф);
3. при нажатии на неё (после проверок на хаки) генерируется синхронное событие **Отмены перехода на тариф**;
4. там можно обработать что угодно, имея ContractTariff "до" и "после";
5. там же можно установить processed (аналогично смене тарифа), чтобы штатно не обрабатывалось;
6. штатная обработка состоит в следующем: удаляем второй тариф (новый, который отменяем) и обновляем предыдущий, продляя дату до бесконечности (то есть "окрывая" его);
7. после этого генерируются сразу два асинхронных события ContractTariffUpdateEvent и ContractTariffDeleteEvent для соответствующих потревоженных тарифов договора.

Управление статусом

У пользователя имеется возможность самостоятельно изменить статус договора.

Управление статусом

Период	Статус	Комментарий
04.12.2009-08.12.2009	активен	
09.12.2009-23.01.2010	активен	Автоматически после оплаты
24.01.2010-24.01.2010	приостановлен	
25.01.2010-25.01.2010	активен	
26.01.2010-03.02.2010	приостановлен	
04.02.2010-04.02.2010	активен	Изменено пользователем
05.02.2010-08.02.2010	приостановлен	Изменено пользователем
09.02.2010-	активен	Изменено пользователем
Текущий статус: активен		

Здесь можно приостановить или активировать договор. Приостановить договор можно не ранее завтрашнего числа. Активировать можно начиная с сегодняшнего.

Заблокировать договор с

Имеются некоторые ограничения. Пользователь может менять только со статуса "активен" на статус "приостановлен" и наоборот. Другие статусы ему недоступны. Нельзя приостановить раньше, чем завтра. Нельзя активировать раньше, чем сегодня.

Если статус договора не "активен" и не "приостановлен" - тоже не разрешена смена.

Если **closed.date.enable d**, то проверятся, что устанавливаемый статус не пересекается с закрытым периодом.

Смена статусов сопровождается событиями, как и при обычной смене статусов, не из Web. Только в событии `ContractStatusChangingEvent` устанавливается флаг `isweb=true`. Событие `ContractStatusChangedEvent` выполняется точно так же.

Список дат, которые будут отображаться, регулируются событием `GetContractStatusChangeDatesEvent`. Если вернули список, то будет отображён список дат. Если не обработано событие, то будет дан выбор дня, месяца, года. Если же будет возвращен пустой список, то это означает, что смена статуса запрещена, о чём сообщится пользователю вместо выбора даты.

Управление лимитом

Данная функция аналогична функции "Обещанного платежа" других биллинговых систем.

Управление лимитом доступно только договорам с режимом **Дебет**

Для активации необходимо добавить в [конфигурацию](#) биллинга один или несколько блоков записей вида.

```
#      ,      , ' ' ( - id )
contract.limit.<n>.groups=
#      ( ) ,
#      , 0
#
contract.limit.<n>.maxnotpayoffed=
#      ,
#      (0-1,
#      )
contract.limit.<n>.maxpartialpayoffed=
#      ,
#      , 0 -
contract.limit.<n>.maxexpiredforblock=
#
contract.limit.<n>.mindays=
contract.limit.<n>.maxdays=
#
contract.limit.<n>.minsumm=
contract.limit.<n>.maxsumm=
#      ( -100),
# .
contract.limit.<n>.minlimit=
```

Где: **<n>** - целое число, начинающееся с 1, определяющее блок конфигурации для определённых групп договоров.

Например, настройка понижения лимита для групп договоров с кодами 1 или 2:

```
#      , ' ' ( - id )
contract.limit.1.groups=1,2
#      ( ) ,
#      , 0
#
contract.limit.1.maxnotpayoffed=0
#      ,
#      (0-1,
#      )
contract.limit.1.maxpartialpayoffed=0
#
#      , 0 -
contract.limit.1.maxexpiredforblock=1
#
contract.limit.1.mindays=1
contract.limit.1.maxdays=4
#
contract.limit.1.minsumm=100
contract.limit.1.maxsumm=200
#      ( -100),
# .
contract.limit.1.minlimit=-400
```

Используя блоки, настройки управления лимитом можно задать различными для разных групп договоров. Разумеется, группы блоков не должны пересекаться.

Для управления лимитом используется меню **Управление лимитом** на странице статистики. Для понижения лимита выбирается сумма и на сколько дней необходимо понижение. В случае, если договор не входит хотя бы в одну из групп, указанных в конфигурации, для которых доступно понижение лимита или если режим договора не **Дебет**, меню **Управление лимитом** не отображается.

Меню
Новости
Просмотр баланса
Смена пароля на доступ к статистике
Подписка на рассылки
Смена тарифных планов
Управление лимитом
Дополнительные действия
Выход

Управление лимитом

Месяц: Год:

Сумма	Дата понижения	Дата восстановления	Состояние
100.00	16.11.2007	19.11.2007	не погашено

Текущий лимит: -200.00

Лимит будет восстановлен при поступлении суммы равной или большей суммы понижения, а понижение лимита помечено как погашено. Если понижение лимита по истечении срока не погашено, оно помечается как просрочено и лимит восстанавливается к прежнему значению. Сумма понижения может быть от 1.00 до 100.00. Лимит может быть понижен до -500.00.

Понизить лимит на сумму: сроком на

Пониженный лимит **всегда** возвращается в исходное состояние. Это происходит либо при внесении на счёт суммы, большей или равной сумме понижения до наступления даты восстановления, либо по наступлению даты восстановления задачей "Восстановление лимитов" планировщика.

В случае, если лимит возвращается в исходное состояние задачей, понижение отмечается как **просроченное**, максимально допустимое количество просроченных понижений устанавливается в конфигурации.

При каждом платеже, поступившем на договор, система погашает **не погашенные** (активные) понижения лимита. При этом понижение может быть погашено частично, становясь **частично погашенным**. Лимит при этом не изменяется. Если до **даты восстановления** будут произведены ещё платежи на счёт, понижение при достаточной сумме может быть признано **погашенным**, лимит возвращается в исходное состояние досрочно (разумеется, можно погасить понижение и одним платежом). Если до **даты восстановления** платежей больше не будет, понижение так и останется **частично погашенным**.

Если платежа хватает на погашение понижения с избытком, система ищет следующий непогашенный платёж, расходуя на него остаток платежа, и т.д. Все понижения могут быть погашены как одним так и серией платежей.

Понижение лимита доступно только при следующих условиях:

- количество уже сделанных не погашенных понижений меньше либо равно переменной **contract.limit.<n>.maxnotpayoffed** конфигурации блока;
- количество просроченных понижений меньше или равно переменной **contract.limit.<n>.maxexpiredforblock** конфигурации блока;
- количество частично оплаченных платежей меньше или равно переменной **contract.limit.<n>.maxpartialpayoffed** конфигурации блока.

В случае, если понижение недоступно, пользователь увидит страницу следующего вида.

Управление лимитом

Месяц: Год:

Сумма	Дата понижения	Дата восстановления	Состояние
100.00	14.02.2007	15.02.2007	просрочено
100.00	14.02.2007	15.02.2007	просрочено
100.00	14.02.2007	15.02.2007	просрочено
100.00	15.02.2007	16.02.2007	просрочено
100.00	14.02.2007	14.02.2007	погашено

Текущий лимит: 0.00

Лимит будет восстановлен при поступлении суммы равной или большей суммы понижения, а понижение лимита помечено как погашено. Если понижение лимита по истечении срока не погашено, оно помечается как просрочено и лимит восстанавливается к прежнему значению.

Возможность понижения лимита заблокирована

В момент понижения система также контролирует следующие условия:

- количество дней на которое понижается лимит должно быть от **contract.limit.<n>.mindays** до **contract.limit.<n>.maxdays**;

- сумма понижения должна быть от **contract.limit.<n>.minsumm** до **contract.limit.<n>.maxsumm**;
- в результате понижения лимит не должен стать менее, чем **contract.limit.<n>.minlimit**.

Оператор биллинга может индивидуально управлять доступом договора к сервису временного понижения лимита через вкладку **Web=>Управление лимитом** договора.

На вкладке **Управление** возможно включение/отключение данной возможности для конкретного договора. По умолчанию возможность включается для всех групп договоров, указанных в конфигурации (см. выше).

Параметры Отчет Карточки HelpDesk CRM Документы

Договор
x0000 (Иванов Петр Сидорович)

Период
с 01.05.2007 по

Управление Статистика

Режим
Разблокировано Заблокировано

Сортировка (дата)

Страница
1 из 2 [25]

Дата	Пользователь	Новый режим
16.02.2007 18:40:09	shamil [Шамиль]	разблокировано
16.02.2007 18:40:08	shamil [Шамиль]	заблокировано
16.02.2007 13:11:55	shamil [Шамиль]	разблокировано
15.02.2007 12:37:33	shamil [Шамиль]	разблокировано
15.02.2007 12:37:30	shamil [Шамиль]	разблокировано
14.02.2007 17:52:19	shamil [Шамиль]	разблокировано
14.02.2007 17:52:00	Сервер	заблокировано
14.02.2007 17:51:17	shamil [Шамиль]	разблокировано
14.02.2007 17:51:10	shamil [Шамиль]	заблокировано
14.02.2007 17:51:01	shamil [Шамиль]	разблокировано
14.02.2007 17:51:00	shamil [Шамиль]	разблокировано
14.02.2007 17:50:48	Сервер	заблокировано
14.02.2007 17:47:37	Сервер	заблокировано
14.02.2007 17:47:33	shamil [Шамиль]	разблокировано
14.02.2007 17:46:42	Сервер	заблокировано
14.02.2007 17:37:05	Сервер	заблокировано
14.02.2007 17:37:03	Сервер	заблокировано
14.02.2007 17:36:47	Сервер	заблокировано
14.02.2007 17:35:40	Сервер	заблокировано
14.02.2007 17:33:23	Сервер	заблокировано
14.02.2007 16:11:35	shamil [Шамиль]	разблокировано
14.02.2007 16:10:27	shamil [Шамиль]	заблокировано
14.02.2007 16:10:26	shamil [Шамиль]	разблокировано
14.02.2007 16:10:25	shamil [Шамиль]	заблокировано
14.02.2007 16:10:24	shamil [Шамиль]	разблокировано

Параметры договора
Статус: активен
Группа тарифов
Скрипт поведения
Лимит: -990.0
Режим: кредит
Доп. действия
Web
Пароль статистики
Управление лимитом
Группы
В
Е
F
Фиктивные
Тарифные планы
УОИР
Тестовый
Баланс (февраль 2008)
Входящий остаток: -839.98
Приход: 50.90
Наработка: 0.00
Расход: 0.00
Исходящий остаток: -789.08
Исходящее saldo: -789.08

Скрытый

Редактор шаблонов договоров Поиск договоров x0000

На вкладке **Статистика** отображается история понижений лимита. Количество просроченных платежей сбрасывается каждый раз при активации возможности понижения лимита для договора. Таким образом, если возможность блокируется для договора системой, повторная активация администратором сбрасывает счётчики.

Договор
x0000 (Иванов Петр Сидорович)

Период с 01.05.2007 по

- Параметры договора
- Статус активен
- Группа тарифов
- Скрипт поведения
- Лимит -990.0
- Режим кредит
- Доп. действия
- Web
 - Пароль статистики
 - Управление лимитом
- Группы
 - В
 - Е
 - Ф
 - Фиктивные
- Тарифные планы
 - VOIP
 - Тестовый
- Баланс (февраль 2008)
 - Входящий остаток -839.98
 - Приход 50.90
 - Наработка 0.00
 - Расход 0.00
 - Исходящий остаток -789.08
 - Исходящее сальдо -789.08
- Скрытый

Управление Статистика

Период с 01.02.2008 по Вывести

Сумма	Дата понижения	Дата восстановления	Статус
10.00	26.12.2007	15.02.2008	погашено

Разграничение прав доступа

Пользователи, группы, права

С версии 9.2204 разграничение прав реализовано с помощью ролей.

Права доступа к биллингу распределяются по пользователям. Пользователи также могут принадлежать одной или нескольким группам, у которых также настраиваются права на выполнение тех или иных действий. Настройка пользователей, групп и принадлежности пользователей группам выполняется в меню **Сервис => Администрирование => Пользователи и права**.

Первым шагом является заведение группы действий. Для этого перейдите на вкладку **Группы** и нажмите **Добавить** на стандартной панели инструментов.


Дважды кликните по узлу **Все действия** для его раскрытия. После этого проставьте галочки на всех разрешённых группе действий. Далее выберите группы договоров, с которыми сможет работать данная группа, используя разрешённые ей действия. (Например, есть две группы, у одной разрешены действия с DialUp и группа договоров DLP, у другой разрешены действия с Email, и группа договоров EML, если назначить эти группы пользователю, то он не сможет работать с договорами EML используя действия DialUp)

В каждой группе помимо действий могут быть установлены фильтры по группам договоров и **Запрещённые параметры** - параметры договоров, которые данная группа изменять не может. Настройка действий, групп договоров и запрещённых параметров также может быть индивидуальной для каждого пользователя (см. далее).

При проверке каждого действия, сервер проверяет все группы, в которые входит пользователь (включая индивидуальные права) и ищет разрешение данного действия. Если действие в некоторой группе разрешено, то оцениваются фильтры по группе договоров и запрещённым параметрам (если это необходимо). В случае удовлетворения фильтров всем условиям действие разрешено. В противном случае - запрещено.

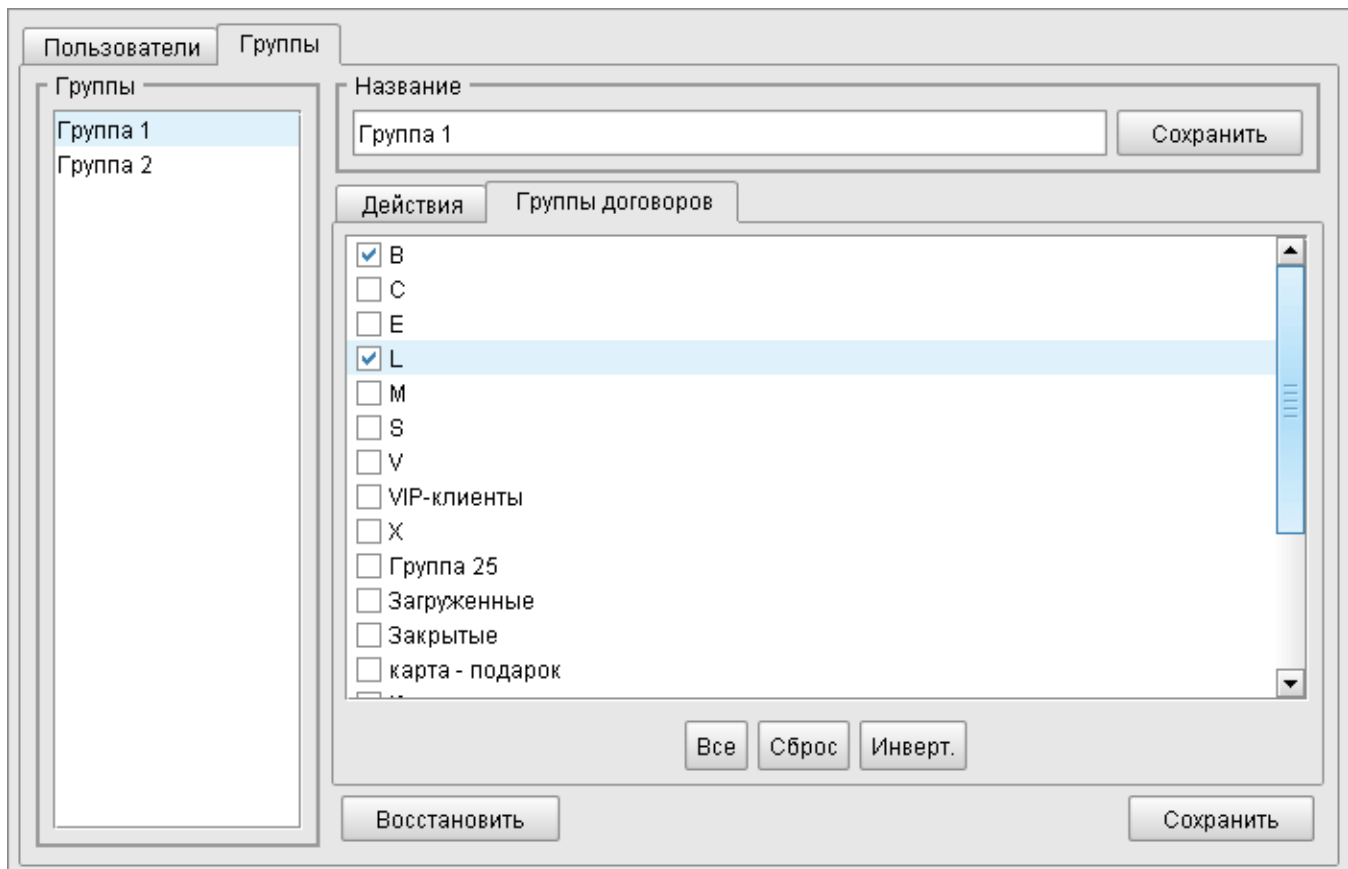
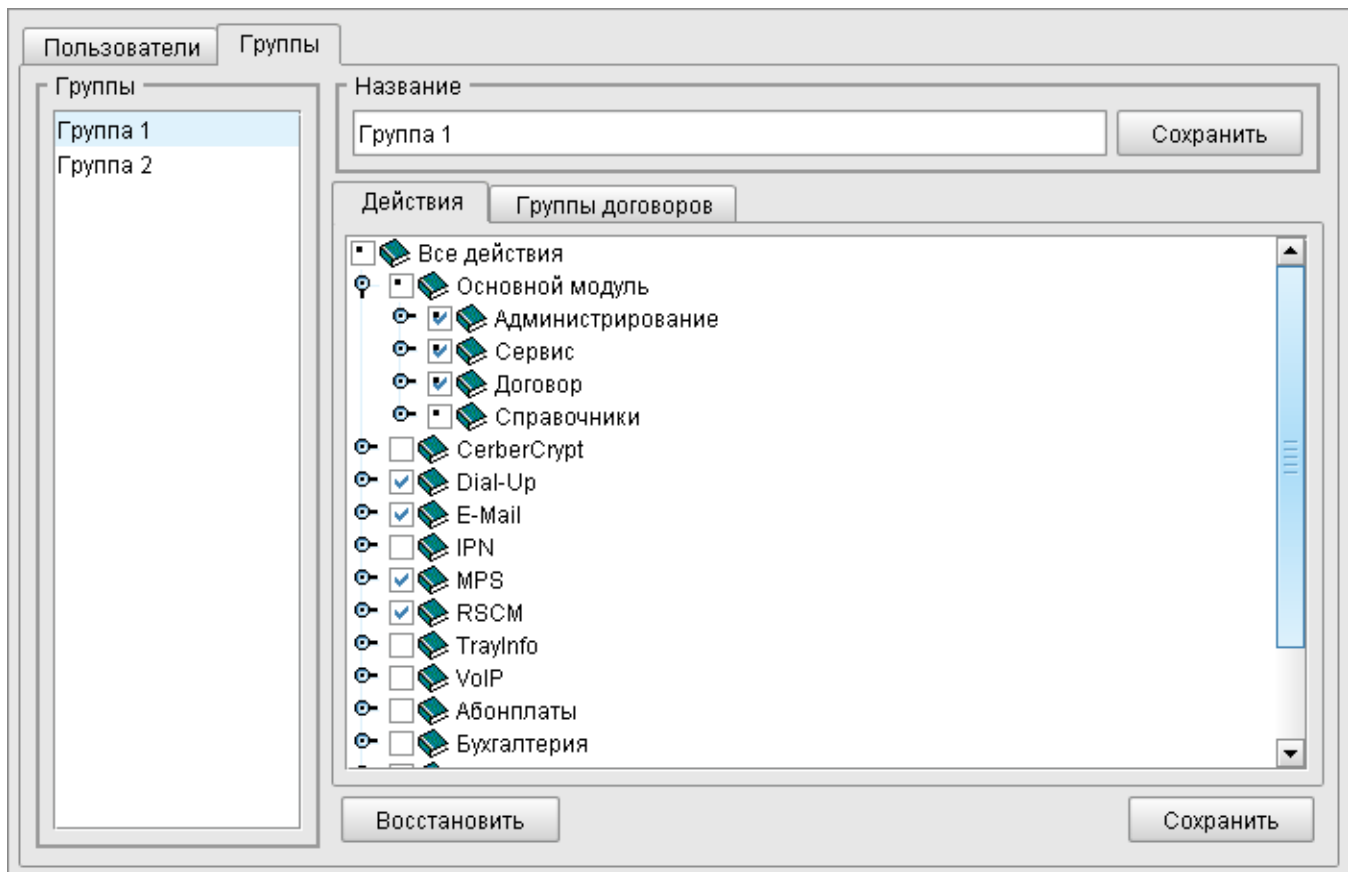
В связи с этим необходимо учитывать, что пустой фильтр групп договоров (не всегда, см. далее) и пустой список запрещённых параметров разрешает работу с любыми договорами и параметрами договоров. И если, например, действие изменения договоров разрешено в свойствах пользователя и не установлен фильтр запрещённых параметров, а в свойствах некой группы, куда входит пользователь, такой список присутствует, то пользователю будет разрешено изменение любых параметров.

То же относится и к группам договоров - разрешён будет максимум из присвоенных клиенту групп пользователей. Если в какой-нибудь из групп пользователей не будет фильтра по группам договоров (и будет установлен параметр **ИЛИ**, см.далее) - будет разрешена работа со всеми договорами.

Если в группе пользователей или у пользователя не будет фильтра по группам договоров (не будет отмечено ни одной группы), то в этом случае  ведение системы будет зависеть от параметра **И/ИЛИ**:

- При установленном параметре **ИЛИ** будет разрешена работа со всеми договорами;
- При параметре **И** будет разрешена работа только с договорами, не входящими ни в одну группу.

Таким образом, если необходимо не учитывать разрешения группы или пользователя к группам договоров, то уберите все галочки и установите параметр **И**. А при включенном параметре **ИЛИ**, будет разрешено все, вне зависимости от других групп пользователей или пользователя.



На вкладке **Пользователи** заводятся пользователи системы, там же определяется принадлежность пользователя группам. Индивидуальные разрешения на действия, фильтры запрещённых параметров и групп договоров можно рассматривать, как уже упоминалось ранее, как добавление ещё одной группы.

Указание групп договоров непосредственно в пользователе скрывает неуказанные типы договоров при поиске. **Режим совпадения** и/или для **Групп договоров** означает, что пользователь видит договор либо при наличии всех определённых для него групп в договоре, либо хотя бы одной из определённых для него групп.

The screenshot shows a web-based user management interface. At the top, there are tabs for 'Пользователи' (Users) and 'Группы' (Groups). Below the tabs is a search filter section with a text input for 'Имя пользователя', a dropdown for 'Статус' (Status) set to 'Все', and a 'Поиск' (Search) button. To the right, it shows 'Всего записей: 19' and pagination controls for '1 из 1 [25]'. Below the search is a table listing users with columns: 'Полное имя', 'Статус', 'Логин', 'Группы', 'E-mail', 'Действ.' (Actions), and 'Описание'. The user 'Amir' is highlighted in blue. Below the table is a 'Редактор' (Editor) window for the user 'Amir'. The editor has several sections: 'Информация о пользователе' (User Information) with fields for 'Пользователь', 'Полное имя', 'E-mail', 'Описание', and 'Статус'; 'Пароль пользователя' (User Password) with 'Пароль' and 'Подтверждение' fields; 'Привязка к договору' (Contract Linking) with a 'Параметр' dropdown and a 'Договор' dropdown; and 'Запрещенные параметры' (Forbidden Parameters) with sub-tabs for 'Группы', 'Действия', and 'Группы договоров'. The 'Группы' sub-tab is active, showing a list of groups with checkboxes: 'WiFi', 'Группа 1', 'Группа 2', 'Запрещены группы договоров', 'Контроллеры', 'может всё, или почти всё', 'некая группа', 'тест 1', and 'тест 2'. At the bottom of the editor are buttons for 'Восстановить', 'Все', 'Сброс', 'Инверт.', 'Ок', and 'Отмена'.

В поле **E-mail** может быть задан адрес электронной почты пользователя, который автоматически проставляется по умолчанию в полях, где требуется ввод почты.

В общем случае следует сначала сопоставить пользователю группы действий, пароль и прочие опции, исключая **Действия**. После этого сохранить пользователя. Если требуется добавление персональных действий, не включённых в указанные группы, открыть редактор пользователя снова и проставить на вкладке **Действия** требуемые разрешения. Выделенные серым блокированные галочки уже указаны в группах пользователя.

This screenshot shows the 'Действия' (Actions) tab of the user editor. The left sidebar shows the user information for 'Борис'. The main area displays a tree view of actions with checkboxes. The 'Все действия' (All actions) checkbox is checked. Under 'Администрирование' (Administration), 'Конфигурация модуля' (Module configuration) is checked, and its sub-items 'Просмотр конфигураций' (View configurations) and 'Обновление конфигурации' (Update configuration) are also checked. Under 'Управление NAS ами' (NAS management), 'Сервис' (Service) is checked. The 'Договор' (Contract) checkbox is unchecked. At the bottom, there are buttons for 'Восстановить' and a taskbar showing 'Редактор тарифных планов' and 'Пользователи и права'.

Поле **Привязка к договору** в свойствах пользователя позволяет заводить в системе агентов. При этом все агентские договоры должны иметь параметр типа **Ссылка на договор (Агент** в примере) и значение этого параметра должно быть равным агентскому договору (**NK00001-03** для приведённого выше снимка экрана). Данному пользователю системы будет разрешена работа только со своими договорами. Они же будут выведены в результатах поиска. В договоре агента (NK00001-03 в примере) вы можете вести взаимозачёты с агентом. Правка параметра **Агент** должна быть запрещена для пользователя-агента!

В **Конфигурации** пользователя может быть установлены следующие опции:

- **user.actions.permit.show_errors=0** - сокрытие ошибок **Доступ запрещён** от пользователя, при этом сервер просто игнорирует запрещённые действия.

Включить/выключить проверку прав можно в **конфигурации сервера**. Права пользователя, чей код в базе равен 1, не проверяются, он может делать все. По умолчанию это **admin**, созданный установочным скриптом.

В дополнение к системе разграничения доступа система предоставляет возможность аудита действий пользователей. **Журнал запросов** доступен в меню **Сервис=>Журналы=>Журнал запросов** .

При просмотре списка пользователей возможна фильтрация по имени пользователя и статусу.

Настройка дерева действий

Права на выполнение различных действий внутри системы контролируются подсистемой BGSecure. Все действия в биллинге собраны в древовидную структуру. При установке модуля вместе с ним устанавливаются его действия. Действия модуля прописаны в файле, расположенном в каталоге **BGBillingServer/actions**.

При необходимости вы можете корректировать файл, добавляя свои собственные действия. Например, можно определить редактирование каждого параметра договора отдельным действием, либо просмотр каждого отчёта. Т.к. действия просматриваются по порядку, то свои действия необходимо добавлять перед существующими

Запросы вы можете посмотреть в файле log, запустив клиент биллинга в DEBUG-режиме скриптом **bgbilling_debug.bat (.sh)**, либо воспользуйтесь **Сервис=>Журналы=>Журнал запросов**. О различиях в видах запросов к серверу вы можете посмотреть [здесь](#).

Файлы с конфигурациями действий перетираются при каждой установке/обновлении модуля, не забывайте заново корректировать их, если вы добавили в них собственные действия. Вы также можете создать **orig**-файлы конфигурации действий, что позволит корректировать файлы только, если они действительно изменились в дистрибутиве. Нумеруйте свои действия номерами больше 10000, чтобы не наступил конфликт ваших действий с добавляемыми стандартными.

HTTP Action

Действия определяются в следующем виде:

```
<group title=" ">
  <action id="14" mask="module=service;action=GetServices" title=" "/>
  <action id="15" mask="module=service;action=UpdateService" title=" "/>
  <action id="16" mask="module=service;action=DeleteService" title=" "/>
</group>
```

Группы предназначены исключительно для их визуального разделения в графическом дереве. Каждое действие должно иметь уникальный в пределах файла код **id**. Маска **mask** действия определяет набор параметров HTTP-запроса, по которым он будет отнесён к данному действию. В параметре запроса может быть указано и **REGEXP**-выражение, это позволяет выделять некоторые типы действий. Например, в модуле dialup можно выделить в отдельное действие переобсчёт сессий по конкретному договору:

```
"module=dialup;action=RecalculateSessions;contract=R:\d+"
```

Как видно из примера, REGEXP указывается после строки **R**:

Web-сервис

Действия определяются в следующем виде:

```
<group title="">
  <service id="89" name="PaymentService" operation="paymentList" title=" "/>
  <service id="90" name="PaymentService" operation="paymentUpdate" title=" " expression="payment.getId() gt 0"
/>
  <service id="101" name="PaymentService" operation="paymentUpdate" title=" "/>
</group>
```

Первичным ключом действия выступает имя Web-сервиса и операция. С помощью **JEXL** выражения в **expression** возможно дополнительная фильтрация параметров метода (см. пример). Все параметры метода передаются в JEXL контекст и у них могут быть вызваны функции. В приведённом выше примере по признаку неположительного идентификатора платежа отделены добавление и изменение платежа.

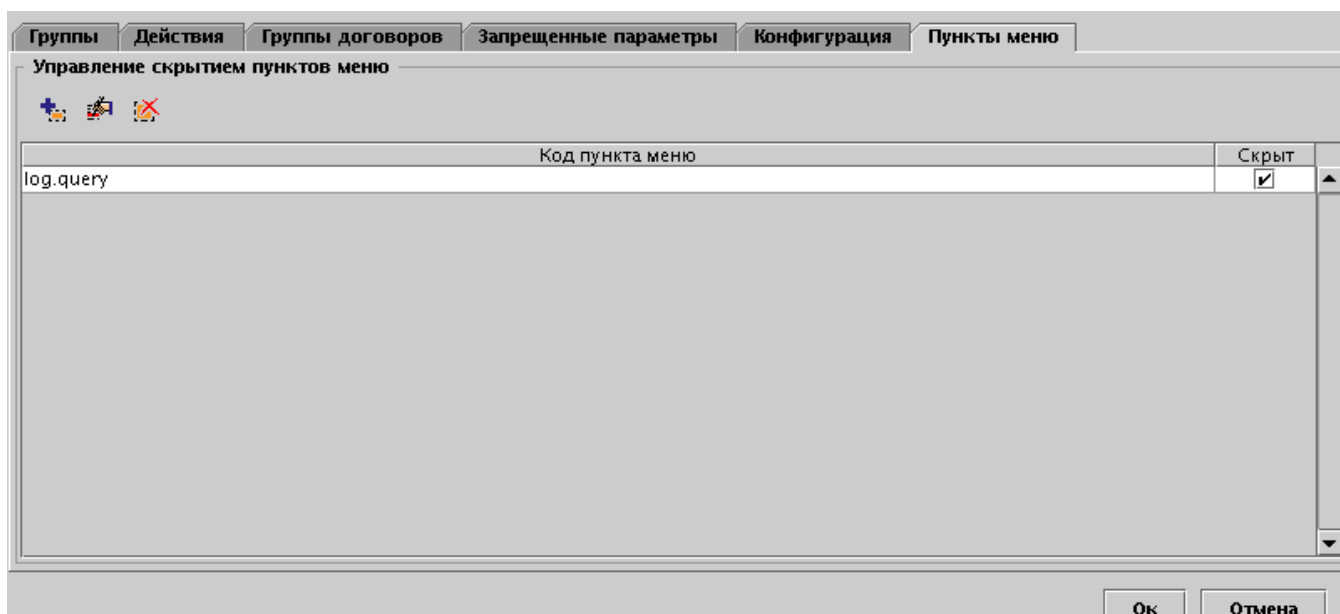
Доступность пунктов меню в клиенте BGBillingClient

Для каждого пользователя, либо для группы пользователей, ко всему прочему, можно настроить также и возможность скрытия ненужных по каким-либо причинам пунктов меню в клиенте BGBillingClient. Само описание меню хранится в файле `BGBillingServer/data/menu.xml` в виде дерева. Например:

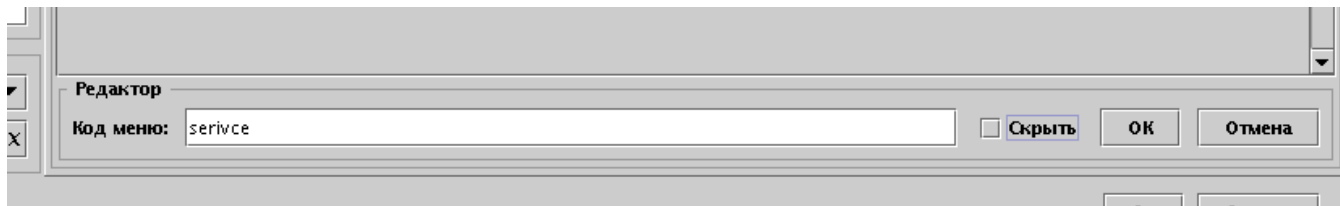
```
...
<menu title="" id="service">
  <menu title="" id="log">
    <menuItem id="query" className="bitel.billing.module.admin.bgsecure.ActionQueryLogViewer" title="" />
    ...
  </menu>
</menu>
...
```

Каждый элемент меню однозначно характеризуется связкой **<код меню>.<код пункта меню>**. При наличии вложенности нескольких меню для определения конкретного пункта меню (т.е. листа дерева), используется только последний код меню, непосредственно включающего данный пункт. Например, пункту меню **Сервис => Журналы => Журнал запросов** соответствует ключ `log.query` (см. код выше). Пункты меню, соответствующие конкретному модулю имеют код вида **<код модуля>**. Например, для модуля Inet с кодом 12 код пункта меню будет равен **12**. Пункты меню, соответствующие плагинам, прописаны в файле `plugin.xml`, который находится в корне jar-архива плагина. Данный архив находится в папке `<BGBilling_server_path>/lib/app`.

Для того, чтобы скрыть от пользователей или их групп часть меню или элементов меню, необходимо при редактировании конкретного пользователя или конкретной группы перейти на вкладку **Пункты меню**. Для добавления, удаления или редактирования правила на скрытие (или, напротив, отображения) какого-либо пункта меню или целого меню необходимо нажать на соответствующие кнопки над таблицей с правилами.



Для добавления нового правила необходимо ввести код меню (или пункта меню), которого оно касается, а также указать показывается он или является скрытым. Для скрытия целого меню используется правило вида **<код меню>**, а для скрытия конкретного подпункта - **<код меню>.<код пункта меню>**.



Важно заметить, что правила для конкретного пользователя имеют более высокий приоритет, чем групповые. Например, для группы "Операторы" можно скрыть все меню Модули, но для оператора Васи сделано исключение, и оно будет отображаться.

При наличии же нескольких групп у пользователя сложение правил видимости действует по принципу "сложения по видимости". Т.е. если некий пользователь Петров принадлежит к двум группам (группе А и группе Б), причем в группе А действует правило невидимости пункта меню Модули, а у группы Б - пункта меню Плагины, то у Петрова все равно будут видны оба этих пункта. Потому что ни у одной из этих двух групп нет общих правил невидимости пунктов меню. Если же Петров будет принадлежать либо к группе А, либо группе Б (но не одновременно к обоим), то у него не будет виден пункт либо, соответственно, Модули, либо - Плагины.

Скрытие пунктов меню для пользователей не относится к системе BGSecure и носит чисто визуальный характер! При использовании альтернативных клиентов права действия "внутри" пунктов меню необходимо разграничивать при помощи системы BGSecure.

Пользовательские правила

С версии 8.2107 поддерживается определение пользовательских правил и переопределение существующих. Для этого необходимо создать файл с названием `<module>.custom.xml` и разместить его в каталоге `BGBillingServer/actions/`

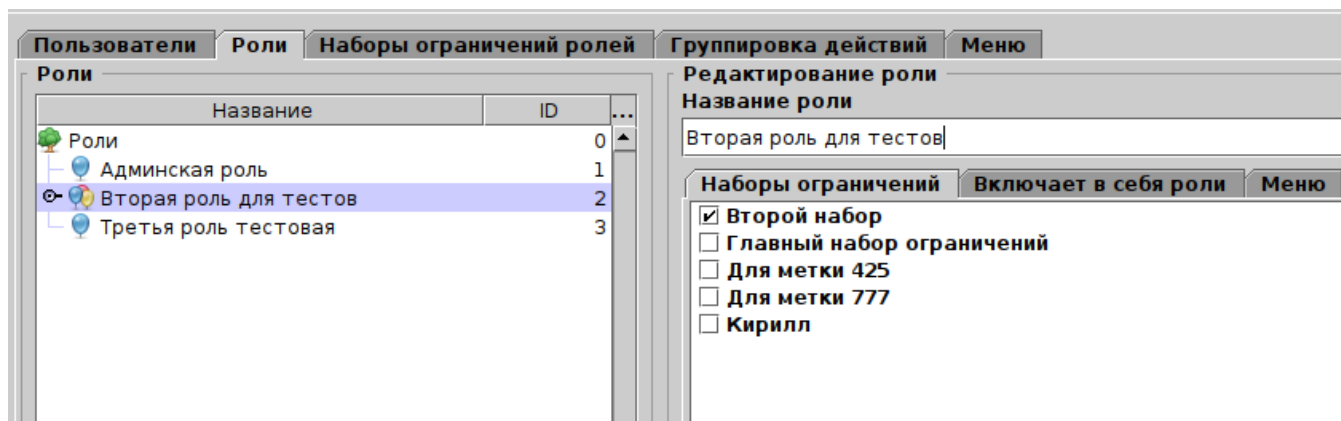
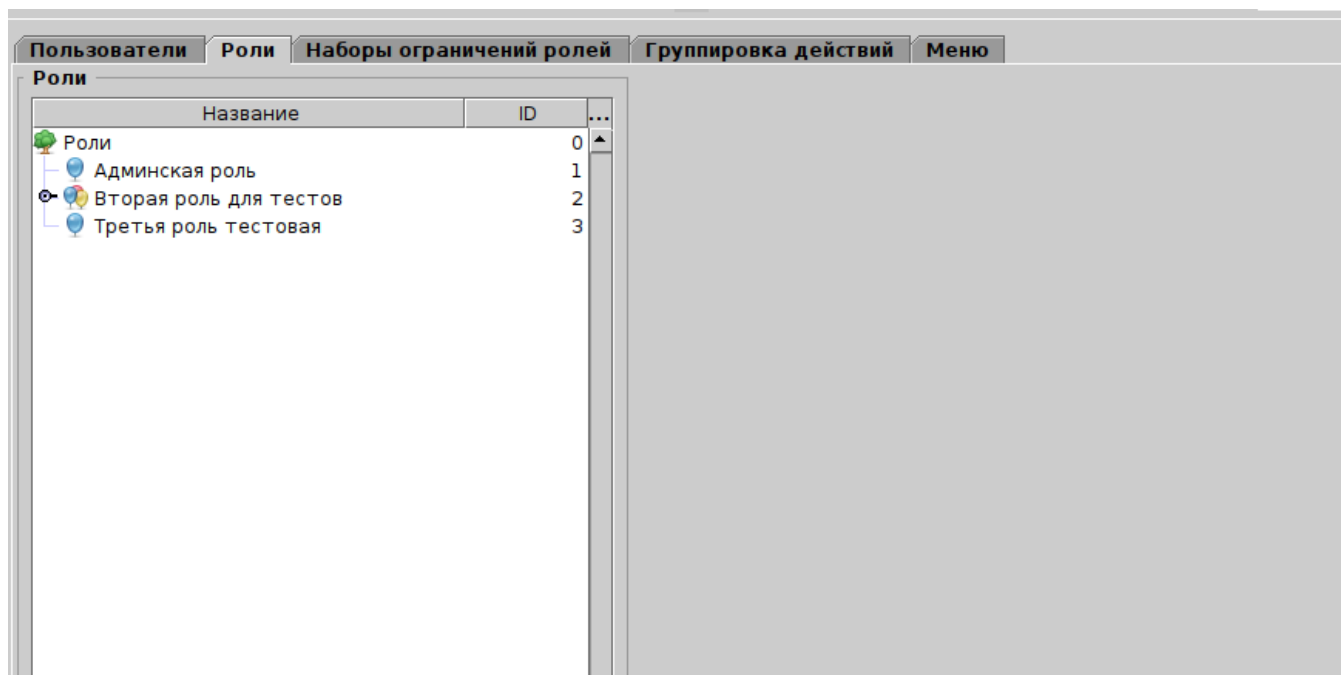
Пример пользовательского файла

```
<actions>
  <!-- , ID , -->
  <service id="1000" name="ConfigService" operation="getModuleConfigList" title=" " />

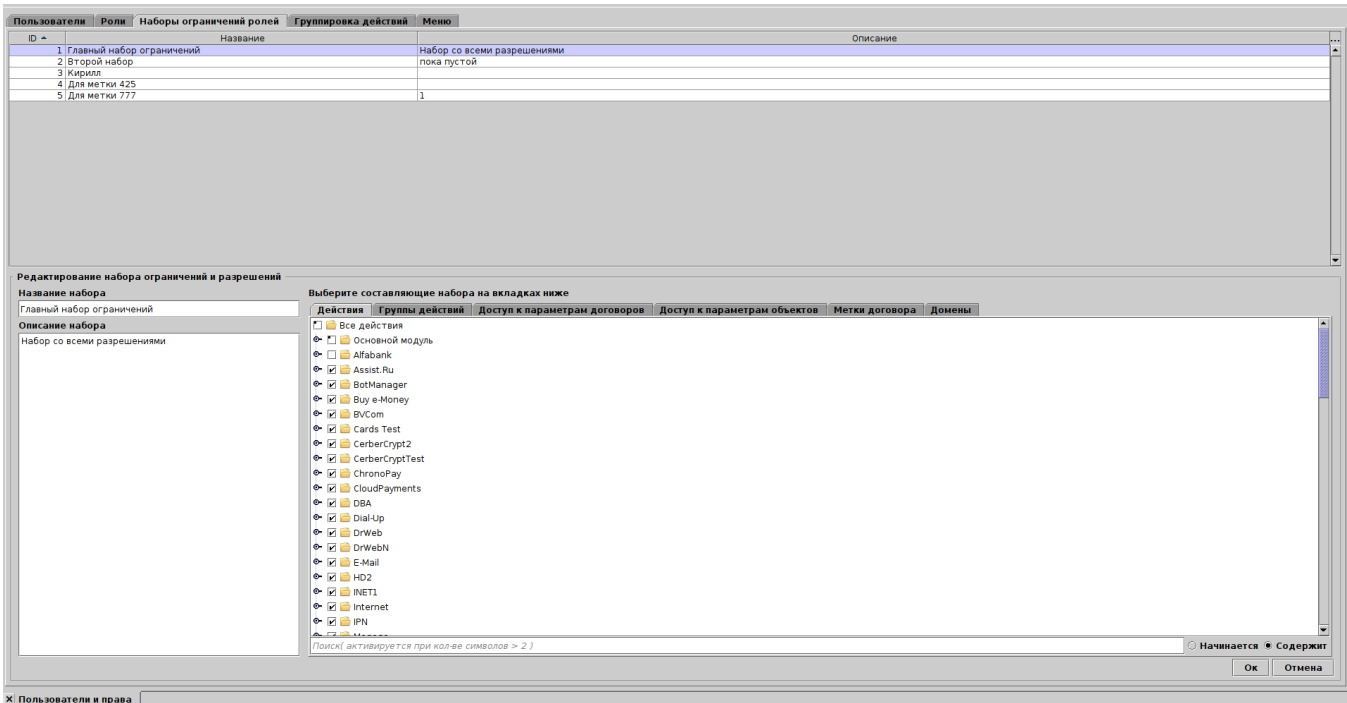
  <!-- , ID , title -->
  <action id="1" mask="module=assist;action=ContractPayList" title=" ( )"/>
</actions>
```

Настройка ролей пользователей(с версии 9.2204)

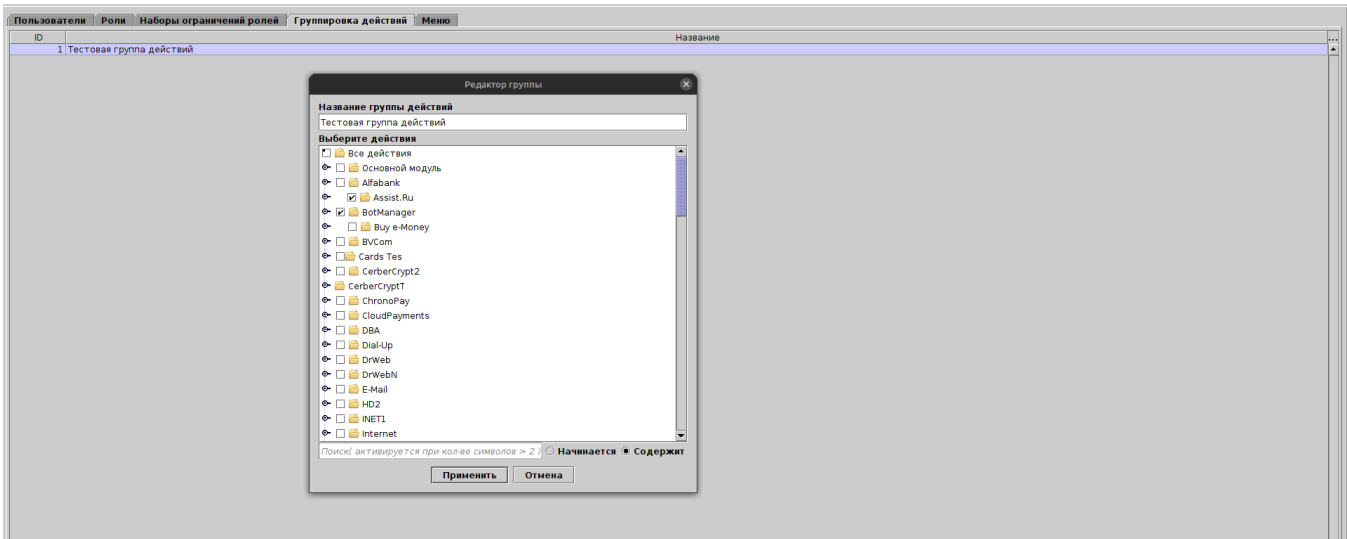
Роль может включать в себя один или несколько "наборов ограничений", меню, другие роли.



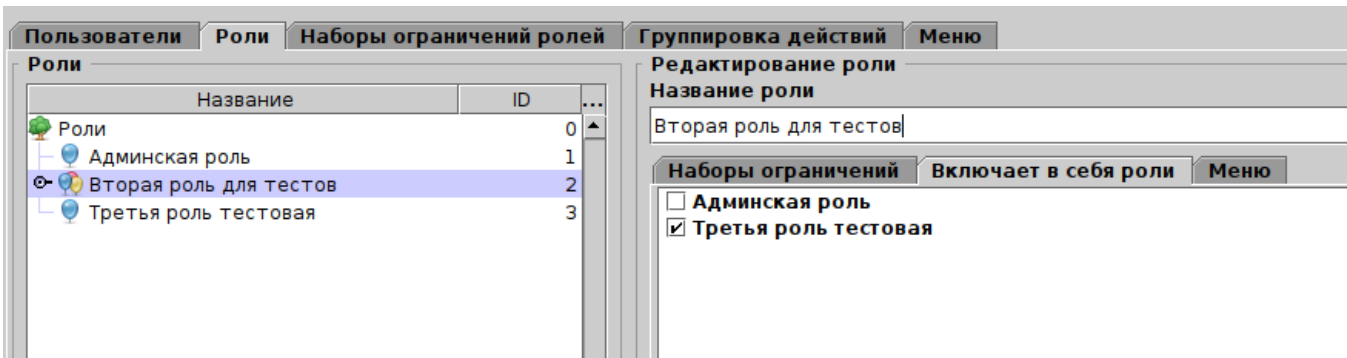
Набор ограничений - аналог группы пользователей в ранних версиях. Включает в себя действия, группы действий, ограничения доступа к: параметрам договора, к объектам, ограничения по меткам договора и доменам.



Группа действий - сгруппированные действия для более удобного и гибкого разграничения доступа.



Каждая роль может включать в себя одну или несколько других ролей, соответственно пользователь, который получил роль с несколькими дочерними ролями, получает права настроенные во всех ролях.



Для того, чтобы назначить роль пользователю, необходимо перейти на вкладку "Пользователи" и открыть в редакторе нужного пользователя. Затем в выпадающем списке "Назначенная роль" выбрать нужную роль и сохранить изменения.

Сервис

Общие сведения

Большинство пунктов меню **Сервис** описано в документации в контексте подсистемы, к которой относится пункт меню. Описание прочих пунктов меню приведено ниже.

Журналы

Ошибки обработки логов

Ошибки обработки логов возникают в модулях, которые используют систему обработки логов с источниками. Каждая ошибка привязана к модулю, источнику и часу. Каждая запись в таблице содержит в себе ошибки обработки часа целиком. При повторной обработке логов за этот же час все ошибки удаляются, т.е. если ошибок вновь не возникнет, то за этот час ошибок в базе не будет.

Для просмотра ошибок обработки логов выберите соответствующую вкладку в меню **Сервис=>Журнал ошибок**. Для просмотра ошибок необходимо выбрать интересующий модуль, источник, а также указать период. Далее необходимо нажать кнопку с галочкой. Выбрав интересующую запись из таблицы, можно просмотреть конкретные сведения об этих ошибках.

"BGBilling" - универсальная биллинговая система (клиент вер. сборка от) UFANET

Договор Модули Справочники Сервис Автоматизация Утилиты Справка

БД: 111 Login: pechenie

Ошибки обработки логов Ошибки периодических процессов

Модуль: IPN

Источники: localhost

Период: Февраль 2009

дни: с до

ID	Дата	Описание
528	2009-02-11 12:00	Не найдена услуга
526	2009-02-11 12:00	Не найдена услуга
527	2009-02-11 12:00	Не найдена услуга
1245	2009-02-11 13:00	Не найдена услуга
1244	2009-02-11 13:00	Не найдена услуга
1243	2009-02-11 13:00	Не найдена услуга
1242	2009-02-11 13:00	Не найдена услуга
1241	2009-02-11 13:00	Не найдена услуга
1240	2009-02-11 13:00	Не найдена услуга
1227	2009-02-11 13:00	Не найдена услуга
1228	2009-02-11 13:00	Не найдена услуга
1237	2009-02-11 13:00	Не найдена услуга
1229	2009-02-11 13:00	Не найдена услуга
1236	2009-02-11 13:00	Не найдена услуга
1235	2009-02-11 13:00	Не найдена услуга
1233	2009-02-11 13:00	Не найдена услуга

Сведения о ошибке

Входящий пакет: Flow 0:192.168.184.254:38768 / 0:192.168.184.35:2001 15528 ToS/DiffServ: 0 nextHop: 0.0.0.0; код источника 255

Журнал ошибок

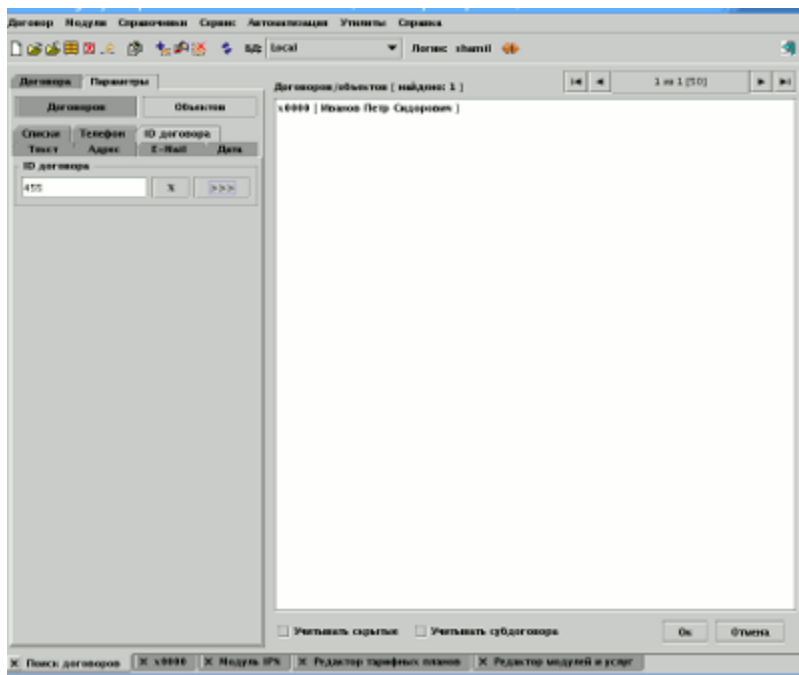
Ошибки периодических процессов

Ошибки периодических процессов возникают при периодических (или запущенных самостоятельно) переобсчётах. Например, при переобсчёте в модуле NPray. Ошибки периодических процессов можно удалять вручную, воспользовавшись кнопкой **Удалить** в тулбаре. Также ошибки периодических процессов за данный месяц и по данному коду модуля удаляются в случае, если запущен полный переобсчёт по всем договорам. И если в этом случае ошибки опять возникают, то они вновь добавляются в журнал.

Для просмотра выявленных ошибок откройте вкладку **Ошибки периодических процессов** в меню **Сервис=>Журнал ошибок**. Обязательным является указание месяца, за который производился переобсчёт. Далее возможно указание подстроки, содержащейся в тексте ошибки (например, строки "npray", если необходимо найти все ошибки переобсчётов по модулю Npray), а также периода, в течение которого производился переобсчёт по выбранному месяцу. Для вывода ошибок необходимо нажать на кнопку **Найти**. По двойному щелчку на выбранной ошибке в таблице откроется полный текст ошибки.

The screenshot shows the "BGBilling" application window with the "Ошибки периодических процессов" (Periodic Process Errors) tab selected. The interface includes a menu bar with options like "Договор", "Модули", "Справочники", "Сервис", "Автоматизация", "Утилиты", and "Справка". Below the menu is a toolbar with various icons and a search bar containing "111". The main area is divided into search filters and a table of errors. The search filters include "Месяц обсчета" (Billing Month) set to "Март" (March) for "2009", and a "Найти" (Find) button. The table has three columns: "Дата ошибки" (Error Date), "Месяц обсчета" (Billing Month), and "Лог ошибок при переобсчете модуля Npray" (Error Log for Npray module recalculation). Two error entries are visible, both dated "17.03.2009" and "03.2009". Below the table, the full text of the selected error is displayed, including details like "CalcMode = null", "Модуль npray. Номер модуля: 66", and "Возникла ошибка при переобсчете (договор 455): CalcMode = null." The interface also shows a "Журнал ошибок" (Error Log) button at the bottom left.

Обратите внимание, что идентификаторы в сообщениях об ошибках передаются в виде числовых кодов. Для поиска договора по коду воспользуйтесь вкладкой **Параметры=>ID договора** окна поиска договоров.



Сопоставить услугу с ее кодом вы можете в **Модули=>Редактор модулей и услуг**, открыв перечень услуг конкретного модуля.

Журнал запросов

Данный журнал предоставляет возможность аудита действий пользователей и доступен в меню **Сервис=>Журналы=>Журнал запросов**. Для удобства поиска различных действий доступны следующие фильтры: по периоду, по действиям, по пользователю биллинга, по договору и по строке запроса. Фильтр по строке запроса поддерживает поиск по **MySQL REGEXP** в полях **Параметр** и **Значение**, для этого нужно отметить **REGEXP**. Если не использовать выражения с **regexp**, то необходимо убрать выделение **REGEXP**, но при этом можно указать символы **%** - любое кол-во символов и **_** - один любой символ также в обоих полях строки запроса. **Regexp** и замена спец. символов производится по полям, и автоматически добавляет между ними =>. Пример поиска со спец. символами **%** и **_** можно видеть на скриншоте ниже. Под таблицей с запросами выводится дополнительная информация по выделенному запросу.

Период: Март 2012

Договор: X >>>

Пользователи: Любой

Фильтр по строке запроса

Параметр: module = Значение: admin_% REGEXP

Вывести Очистить Всего записей: 405 5 из 17 [25]

Время	Пользователь	IP	Номер договора
07.03.2012 14:03:15	kostya	127.0.0.1	
07.03.2012 14:03:16	kostya	127.0.0.1	
07.03.2012 14:03:16	kostya	127.0.0.1	
07.03.2012 14:15:01	Ильшат Латыпов	127.0.0.1	
07.03.2012 15:03:01	Ильшат Латыпов	127.0.0.1	
07.03.2012 15:08:03	Ильшат Латыпов	127.0.0.1	
07.03.2012 15:40:16	Ильшат Латыпов	127.0.0.1	
07.03.2012 16:04:20	Ильшат Латыпов	127.0.0.1	
07.03.2012 16:13:24	Ильшат Латыпов	127.0.0.1	
07.03.2012 16:19:39	Ильшат Латыпов	127.0.0.1	
07.03.2012 16:38:56	Ильшат Латыпов	127.0.0.1	

Дополнительная информация по запросу

```
contracts =>
BGBillingSecret => OH8CZ3L8ZHPKpBp1dBX0EMeH
pvalue =>
module => admin.bgsecure
pageIndex => 2
isregexp => 0
action => QueryLogTable
options =>
```

Включить/выключить журналирование действий можно в [конфигурации сервера](#).

Журнал Web-запросов

Данный журнал предоставляет возможность аудита действий клиентов, а также просмотр удачных/неудачных авторизаций в личном кабинете, и доступен в меню **Сервис=>Журналы=>Журнал Web-запросов**. Для удобства поиска различных действий доступны следующие фильтры: по периоду, по запросу, по договору. В фильтре по запросу можно использовать спец. символы оператора **LIKE** в MySQL: **%** - любое количество символов; **_** - один любой символ. Под таблицей с запросами выводится дополнительная информация по выделенному запросу.

Период: Март 2012 Контракт:

дни: с до Фильтр по запросу:

Web-запросы | **Web-авторизации (успешные)** | **Web-авторизации (безуспешные)**

Дата	Договор	IP адрес
01.03.2012 21:34:21	x0000	127.0.0.1
05.03.2012 14:31:27	x0000	127.0.0.1
05.03.2012 15:21:15	x0000	127.0.0.1
06.03.2012 18:14:25	x0000	127.0.0.1
06.03.2012 18:14:34	x0000	127.0.0.1
06.03.2012 18:14:35	x0000	127.0.0.1
06.03.2012 18:15:05	x0000	127.0.0.1
06.03.2012 18:15:13	x0000	127.0.0.1
06.03.2012 18:15:16	x0000	127.0.0.1
06.03.2012 18:15:17	x0000	127.0.0.1
06.03.2012 18:15:19	x0000	127.0.0.1
07.03.2012 14:22:15	x0000	127.0.0.1
07.03.2012 14:25:15	x0000	127.0.0.1
07.03.2012 15:32:59	x0000	127.0.0.1
07.03.2012 16:14:29	x0000	127.0.0.1
07.03.2012 19:17:44	x0000	127.0.0.1
08.03.2012 02:21:14	x0000	127.0.0.1
11.03.2012 14:11:30	x0000	127.0.0.1
13.03.2012 17:54:13	x0000	127.0.0.1
13.03.2012 18:04:07	x0000	127.0.0.1
15.03.2012 17:27:19	x0000	127.0.0.1
16.03.2012 09:47:11	x0000	127.0.0.1
19.03.2012 12:48:12	x0000	127.0.0.1

Дополнительная информация по запросу
GUID1984817449095750 => ; pswd => 123456; midAuth => 0; user => x0000;

Загрузка платежей и расходов из файла

Может использоваться как протокол обмена между внешними программами по приёму платежей и сервером биллинга. Загрузка производится из текстового, либо DBF-файла. Менеджер загрузки расположен в меню **Сервис=>Автоматизация=>Загрузка платежей и расходов**.

Загрузка платежей и расходов может производиться из файлов различных форматов. Для каждого формата файла должен быть определён шаблон, шаблоны загрузки платежа (расхода) описываются в конфигурации сервера биллинга (меню **Сервис=>Настройка=>Конфигурация**).

Для загрузки строки с платежом (расходом) из файла программа выполняет следующие шаги:

1. Разбор строки на части (только для TXT-файлов). При этом из сплошной строки образуется **запись**, содержащая **позиции** (подстроки), нумерующиеся с единицы. Очевидно, что для DBF файлов данный этап не требуется, т.к. файл изначально состоит из записей;
2. Вычленение из записи сумм платежей (расходов), идентификатора договора, комментария платежа (расхода), даты платежа/расхода (опционально);
3. Сопоставление записи договору в биллинге;
4. Зачисление (снятие) суммы на договор.

Каждый шаблон обозначается уникальным числовым идентификатором (<id>), он должен присутствовать в каждой строке, описывающей шаблон. Шаблоны следует нумеровать **строго** по порядку. Приведённые ниже примеры описывают шаблон с кодом 1.

Все дальнейшие примеры будут описывать шаблоны загрузки только реестров платежей. Шаблоны реестров расходов идентичны шаблонам платежей с точностью до слова **payment** в начале каждой строки (для расходов следует писать **charge**, например **charge.load.pattern.1=Новый шаблон**) и до указания типа **payment_type** (следует писать **charge_type**). Нумерация шаблонов расходов независима от нумерации шаблонов платежей (т.е. может быть как шаблон платежа, так и шаблон расхода с номером 1).

В каждом шаблоне должны быть обязательно определены параметры:

```
payment.load.pattern.<id>=<>
payment.load.pattern.<id>.type=<>
payment.load.pattern.<id>.encoding=<>
payment.load.pattern.<id>.payment_type=< >
payment.load.pattern.<id>.position_sum=< >
```

Где:

- **Название** - текстовая строка, обозначающая шаблон в выпадающем списке менеджера загрузки платежей
- **Тип** - число, определяющее тип файла. 1 - для текстового файла, 2 - DBF.
- **Кодировка** - кодировка файла (Cp1251, UTF-8, Cp866).
- **Типы платежей** - коды типов платежа, которыми вносятся платежи в договор. Лучше, если это будет не редактируемый тип. Код вы можете посмотреть в справочнике типов платежей **Справочники=>Другие=>Типы платежей**.
- **Позиции сумм** - из какой позиции записи брать суммы платежей.

Начиная с версии 4.6 появилась возможность указывать в одной строке одновременно несколько сумм для разных типов платежей. Для этого в полях **payment_type** и **position_sum** нужно указывать типы платежей и позиции сумм через запятую, в правильном, соответственном порядке. Т.е. если указать **payment_type=1,2,3** и **position_sum=2,3,4**, то это будет означать, что на 2ой позиции расположена сумма с платежом типа 1, на 3ей позиции - типа 2 и т.п.

В случае если тип формата определён как 1 (текстовый файл), то должен быть определён **REGEXP** разбиения строки файла на позиции записи.

```
payment.load.pattern.<id>.regexp=<REGEXP>
```

Дополнительно для каждого шаблона могут быть определены следующие необязательные параметры:

```
payment.load.pattern.<id>.summa.replace=< >
payment.load.pattern.<id>.position_id=< >
payment.load.pattern.<id>.position_date=< >
payment.load.pattern.<id>.date_format=< >
payment.load.pattern.<id>.position_comment=< >
payment.load.pattern.<id>.comment_delimiter=< >
payment.load.pattern.<id>.handler_class=< ru.bitel.bgbilling.kernel.integration.server.bean.RegisterHandler
( )>
```

Где:

- **Строка замен** - несколько записей вида **{REGEXP что}=>{На что}**, разделённые вертикальной чертой;
- **Позиции комментария платежа** - из какой позиции записи брать комментарий платежа, если параметр не указан, то комментарий - пустая строка; если необходимо собрать комментарий из нескольких полей, то можно указать номера полей через запятую;
- **Разделитель полей комментария** - любая строка, которая будет вставлена между полями комментария. По умолчанию используется пробел.
- **Позиция уникального идентификатора** - из какой позиции записи брать уникальный строковый идентификатор платежа. При указании данного поля загрузчик будет считать ошибочными платёж, если в уже загруженных реестрах с датой реестра в том же месяце, что и загружаемый реестр, уже есть платёж с таким идентификатором;
- **Позиция даты и Формат даты платежа** - из какой позиции записи брать дату платежа и в каком формате она указана. Если параметры не указаны, то дата платежа равна дате реестра.

Макрос замен используется для приведения произвольного формата суммы к виду сплошной записи с десятичной точкой в качестве разделителя, например: 545454.55. Для приведения записи 545 454-55 к этому формату необходимо указать следующий макрос:

```
payment.load.pattern.<id>.summa.replace=\-=>.\|s=>
```

Последнее, что необходимо определить для каждого шаблона - метод сопоставления записи с договором биллинга. Для этого могут быть определены один или несколько методов поиска. По умолчанию результаты поиска по всем методам складываются, но можно соединить их условием **логическое И**, установив опцию:

```
payment.load.pattern.<id>.search.mode=and
```

В системе установлено жёсткое ограничение, по которому каждый тип поиска возвращает не более 10 результатов. Это реализовано в MySQL опцией LIMIT. Режимы поиска работают с учётом даты платежа, она сравнивается с периодом действия договора, логина, телефона и пр. В результаты поиска не попадают договоры, помеченные как "скрытые".

Каждый метод поиска имеет свой подход (<s_id>) в пределах шаблона. В методе поиска должны быть определены обязательно параметры:

```
payment.load.pattern.<id>.search.<s_id>.type=< >
payment.load.pattern.<id>.search.<s_id>.pos=<>
payment.load.pattern.<id>.search.<s_id>.regime=< >
```

Где:

- **Тип метода** - может принимать значения: **cid** (код договора), **contract** (название договора), **comment** (комментарий договора), **parameter** (текстовый параметр договора), **parameter_email** (параметр договора типа E-Mail), **cerbercrypt** (номер карты в модуле CerberCrypt), **login** (логин, либо алиас модуля DialUp/VoiceIP), **phone** (номер телефона модуля phone);
- **Позиция** - из какой позиции записи брать идентификатор для данного метода поиска;
- **Метод сопоставления** - какое MySQL-выражение использовать для запроса поиска. 1 - "=", точное совпадение идентификатора с записью в базе; 2 - LIKE; 3 - REGEXP.

При типах метода parameter, parameter_email необходимо указание кода параметра:

```
payment.load.pattern.<id>.search.<s_id>.pid=< >
```

При типах поиска, связанных с конкретным модулем (login, phone, cerbercrypt), необходимо указать код экземпляра модуля:

```
payment.load.pattern.<id>.search.<s_id>.mid=< >
```

Дополнительно для каждого метода поиска могут быть указаны следующие необязательные параметры:

```
payment.load.pattern.<id>.search.<s_id>.groups=< >
payment.load.pattern.<id>.search.<s_id>.pattern=<REGEXP->
payment.load.pattern.<id>.search.<s_id>.replace=<REGEXP >
#
payment.load.pattern.<id>.search.<s_id>.no.contract.period.check=1

# . :
# 5- . , bankTitle-,
payment.load.pattern.<id>.extra=5:bankTitle;6:bankAccount;
```

Где:

- **Группы договоров** - список через запятую кодов групп договоров, разрешённых к поиску данным методом. Группы можно посмотреть в справочнике **Справочники=>Другие=>Группы договоров**;
- **REGEXP фильтр** - ограничение искомых договоров по их названию, следует учитывать, что используется **MySQL REGEXP**;
- **REGEXP замены** - преобразование идентификатора договора с помощью **REGEXP**-выражения. Возможно указание нескольких замен, разделённых двойной вертикальной чертой (|). Каждая замена состоит из REGEXP-шаблона, далее строка => и на что необходимо заменять шаблон. В строке-замене возможно использование макросов \$0-\$5, соответствующих номерам групп из REGEXP-выражения. Например при методе сопоставления LIKE (см. ранее) можно добавлять в конец строки-идентификатора "%", указав в replace-парамetre: **..replace=.*=>\$0%**.

Формат строки в исходном файле (разделители полей - точка с запятой). Первым идёт название договора, далее сумма платежа и комментарий. Даты платежа нет, в этом случае она равна дате реестра.

```
x0000;13.4;sfdsd
```

Описание шаблона в конфигурации. Идентификация клиента проводится по номеру договора.

```
payment.load.pattern.1=1
payment.load.pattern.1.type=1
payment.load.pattern.1.encoding=UTF-8
payment.load.pattern.1.payment_type=2
payment.load.pattern.1.position_sum=2
payment.load.pattern.1.regex=(\w+);([\d\.]+);(\w+)
payment.load.pattern.1.position_comment=3
payment.load.pattern.1.search.1.type=contract
payment.load.pattern.1.search.1.pos=1
# 1 - eq, 2 - LIKE, 3 -REGEXP
payment.load.pattern.1.search.1.regime=1
```

Формат строки в исходном файле (разделители полей - табуляторы). На второй позиции идёт номер телефона, на четверной - сумма платежа, на пятой - комментарий договора. Дата платежа из файла не используется.

```
2 2775102 20.07.07 147.88 . , .144 .17
```

Описание шаблона в конфигурации. Идентификация клиента производится по номеру телефона + ФИО.

```
payment.load.pattern.1=
payment.load.pattern.1.regex=(\d+)\t(\d+)\t([\d\.]+)\t([\d\.]+)\t([-- \w+])\t(.+)
# ( )
payment.load.pattern.1.payment_type=2
payment.load.pattern.1.position_sum=4
payment.load.pattern.1.position_comment=1
#
payment.load.pattern.1.search.1.type=phone
#
payment.load.pattern.1.search.1.mid=1
# REGEXP
payment.load.pattern.1.search.1.pos=2
# E.164
payment.load.pattern.1.search.1.replace=.*=>7846$0
#
payment.load.pattern.1.search.2.type=comment
#
payment.load.pattern.1.search.2.pos=5
# +
payment.load.pattern.1.search.mode=and
```

Формат строки в исходном файле (разделители полей - точка с запятой). Первым идёт номер договора, далее сумма, комментарий договора и дата платежа. Обратите внимание, что комментарий договора может быть указан не полностью, только фамилия.

```
x0000;13-4; ;04.07.2008
```

Описание шаблона в конфигурации. Идентификация клиента производится по номеру договора + началу ФИО.

```
payment.load.pattern.2=2
payment.load.pattern.2.type=1
payment.load.pattern.2.encoding=UTF-8
payment.load.pattern.2.payment_type=2
payment.load.pattern.2.regex=(\w+);([\d\.\.\s-]+);([--\s]+);([\d\.]*)
payment.load.pattern.2.position_sum=2
#
payment.load.pattern.2.summa.replace=\->.\s=>
#
payment.load.pattern.2.position_date=4
payment.load.pattern.2.date_format=dd.MM.yyyy
payment.load.pattern.2.search.mode=and
payment.load.pattern.2.search.1.type=contract
payment.load.pattern.2.search.1.pos=1
payment.load.pattern.2.search.1.regime=1
payment.load.pattern.2.search.2.type=comment
payment.load.pattern.2.search.2.pos=3
payment.load.pattern.2.search.2.regime=2
payment.load.pattern.2.search.2.replace=\s*([--]+)\s*=>%$1%
#
```

Загрузка производится из DBF-файла, второй столбец - дата платежа, третий - сумма, шестой - номер договора. Описание шаблона в конфигурации. Идентификация клиента производится по номеру договора.

```
payment.load.pattern.3=3
payment.load.pattern.3.type=2
payment.load.pattern.3.encoding=Cp866
payment.load.pattern.3.payment_type=2
payment.load.pattern.3.position_sum=3
payment.load.pattern.3.position_date=2
payment.load.pattern.3.date_format=yyyy-MM-dd
payment.load.pattern.3.search.1.type=contract
payment.load.pattern.3.search.1.pos=6
payment.load.pattern.3.search.1.regime=1
```

Обработка платежей осуществляется на основании реестров следующим образом:

1. загрузка файла в реестр с проверкой ошибок формата, сопоставлением платежей договорам;
2. проведение реестра с начислением платежей в баланс;
3. откат реестра при необходимости.

Фильтр

Период: с по **08.09.2008** >>

Управление реестром

Провести реестр Откатить реестр

Реестры

Код	Дата	Тип платежа	Заголовок	Статус
132	21.08.2008	Банк (Башкредит Банк)	TNE0301.DBF	проведен
136	04.08.2008	Банк (Башкредит Банк)	TNE20080804.DBF	проведен
135	04.08.2008	Банк (Башкредит Банк)	TNE20080804.DBF	не проведен
134	04.08.2008	Банк (Башкредит Банк)	TNE20080804.DBF	проведен
133	04.08.2008	Банк (Башкредит Банк)	TNE20080804.DBF	проведен

Загрузка реестра

Файл: ... Дата: Шаблон: Шаблон1 Название: Загрузить

Лог загрузки / просмотр реестра

Лог загрузки Платежи

К Редактор групп тарифов X Настройка... X Редактор справочников X Загрузка платежей

Каждый реестр характеризуется датой и названием. Дата реестра устанавливается в дату платежа в случае, если дата не берётся из записи. Проведение реестра - процесс занесения приходов по разобранному реестру в договоры. Откат реестра - процесс удаления платежей, может быть полезен при ошибочном реестре.

Для загрузки реестра необходимо выбрать файл, дату реестра и шаблон для разбора. Если название реестра не указано, оно принимается равным имени файлу. После нажатия кнопки **Загрузить** реестр отображается в таблице (фильтр по дате должен включать дату реестра) и в нижнем окне выводится лог загрузки реестра и выделенные платежи.

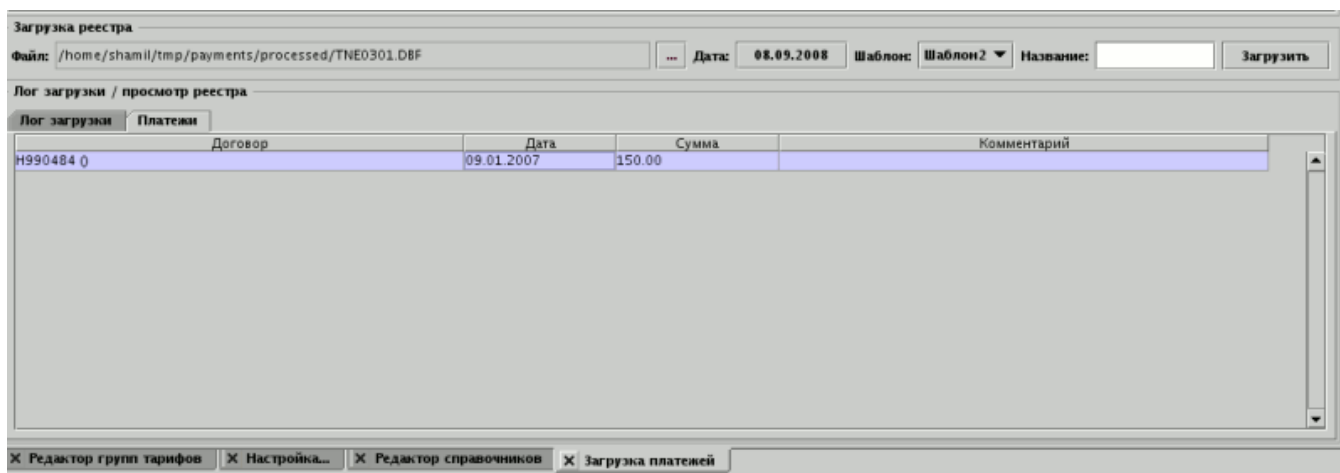
Загрузка реестра

Файл: /home/shamill/tmp/payments/processed/TNE0301.DBF ... Дата: **08.09.2008** Шаблон: Шаблон2 Название: Загрузить

Лог загрузки / просмотр реестра

Лог загрузки Платежи

Лог загрузки	Платежи
(1) 403443	2007-01-09 150.00 пубова' тия Аль' ... Ташки ... д.204 кв.100 H111
(2) 403443	2007-01-09 150.00 ашкин' идмир ...
(3) 403443	2007-01-09 150.00 сонов' атьяна ...
(4) 403443	2007-01-09 600.00 злова' тлана Г. ...
(5) 403443	2007-01-09 150.00 зедев' ем Вален ...
(7) 403443	2007-01-09 180.00 дьяко' горь Ген ...
(8) 403443	2007-01-09 75.00 дьякс' горь Ген ...
(9) 403443	2007-01-09 300.00 нина' ия Алекса ...
(10) 403443	2007-01-09 50.00 пник' Елена Фед ...
(11) 403443	2007-01-09 150.00 афиул' на Майя Ис ...
(12) 403443	2007-01-09 180.00 ерем' а Ирина Д ...
(13) 403443	2007-01-09 100.00 зькин' инаида Ал ...
(14) 403443	2007-01-09 150.00 емиг' ов Геннад ...
(15) 403443	2007-01-09 150.00 дина' етлана Ал ...
(16) 403443	2007-01-09 150.00 елух' Владимир ...
(17) 403443	2007-01-09 300.00 лосо' дий Миха ...



Для проведения реестра - выбрать строку в таблице и нажать кнопку **Провести реестр**. Для отката - также выбор строки и кнопка **Откат реестра**.

Автоматическая загрузка реестров платежей

Для настроенных шаблонов [реестров платежей](#) возможна автоматическая загрузка по планировщику.

Работает **ТОЛЬКО** для загрузки платежей.

Для автоматической загрузки реестров необходимо настроить задачу планировщика **Загрузка платежей из файлов**. Папка, из которой будут загружаться файлы и в которую будут переноситься в результате обработки, **должны быть в локальной файловой системе**. В конфигурации задачи указываются:

```
report.mail=<E-Mail , >
load.from.dir=< , >
processed.to.dir=< , >
```

Все возможные типы файлов-реестров должны быть описаны следующим образом, <id> - уникальный для каждого типа файла код.

```
filetype.<id>.load_pattern=< >
filetype.<id>.name_pattern=<REGEXP >
```

Дополнительно, могут быть указаны параметры для извлечения даты реестра из имени файла. Если данные параметры не указаны, то дата реестра принимается равной дате последней модификации файла.

```
filetype.<id>.date_format=< >
filetype.<id>.position_date=< REGEXP , >
```

В данном примере имена реестров представляют из себя файлы вида TNEyyyMMdd.DBF:

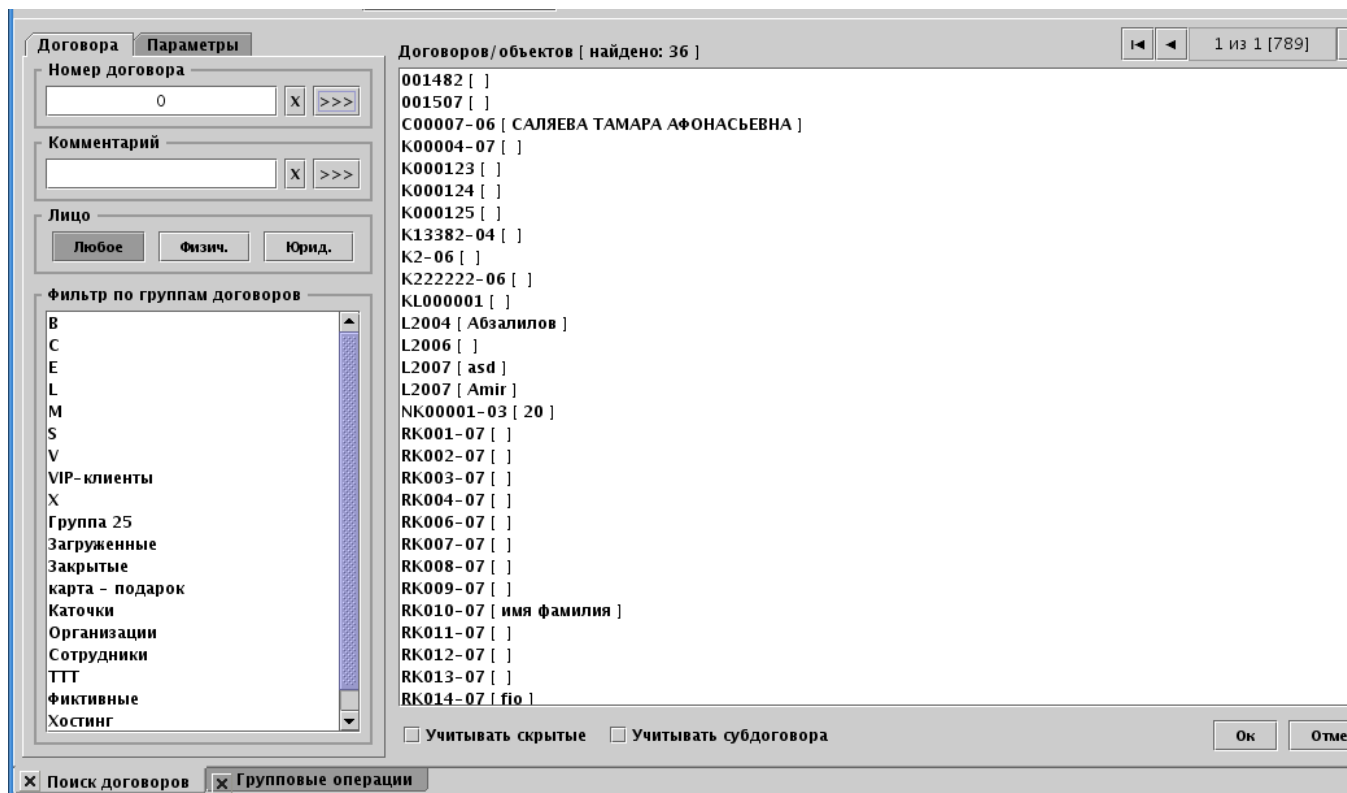
```
report.mail=shamil@bitel.ru
load.from.dir=/home/shamil/tmp/payments
processed.to.dir=/home/shamil/tmp/payments/processed
filetype.1.load_pattern=3
filetype.1.name_pattern=TNE(\d{8})\DBF
filetype.1.date_format=yyyymmdd
filetype.1.position_date=1
```


Групповые операции над договорами

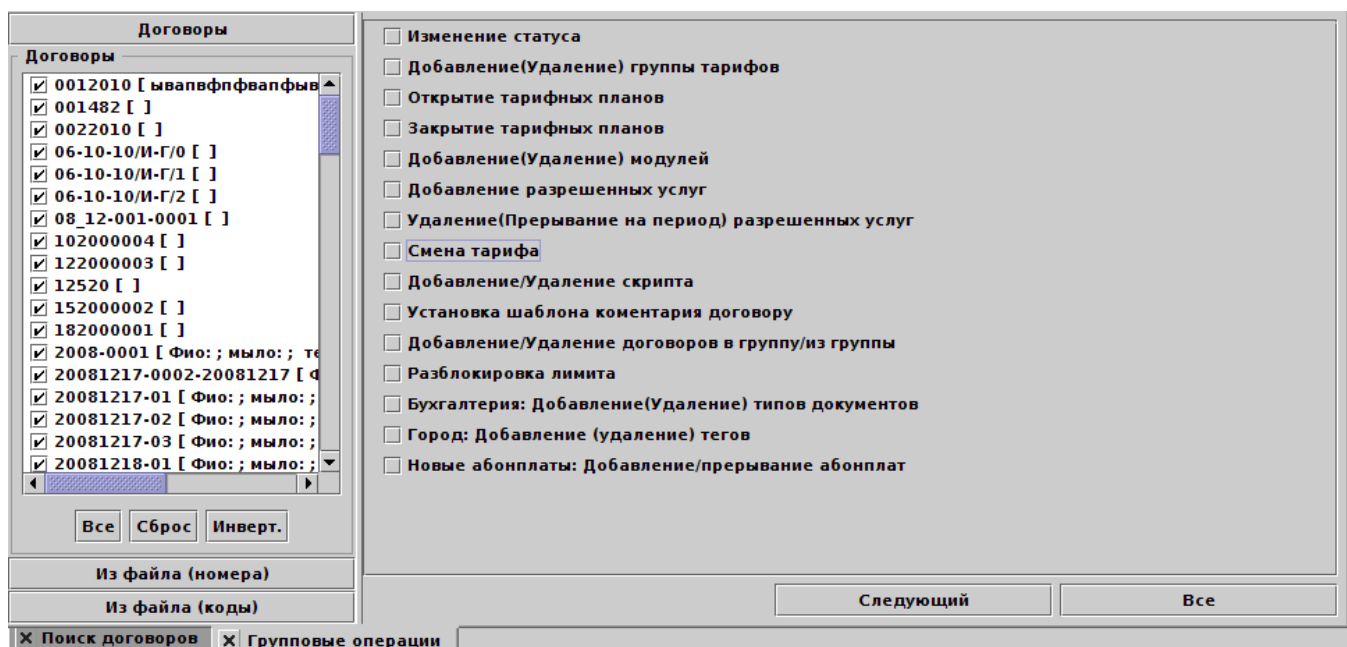
Общие сведения

Групповые операции доступны в меню **Сервис=>Администрирование=>Групповые операции** и позволяют автоматизировать типичные манипуляции над большим количеством договоров.

Перед применением операции договоры должны быть отфильтрованы в окне поиска договоров.



Далее открывается вкладка **Групповые операции** и нажатием кнопки **Договоры** выбранные договоры помещаются в список. В списке можно убрать пометки с ненужных договоров.



Далее выбирается галочкой **Выполнить** одна или несколько модификаций. Доступны следующие модификации.

1. Установка услуг в выбранные договоры с указанного периода. Данная опция необходима при введении новых модулей, услуги которых следует добавить всем договорам, например WM. Услуги разбиты по модулям и представлены древовидным списком;
2. Прерывание услуг на период (абонплат). Данная модификация необходима при временной недоступности услуг для абонентов для компенсации абонплаты;
3. Для модуля Bill доступна модификация по установлению типов счетов и фактур в договоры.

Могут быть выбраны одновременно несколько модификаций. Кнопка **Следующий** проводит модификацию над следующим по списку договором и снимает галочку, данная функция полезна при отладке. При правильном функционировании кнопка **Все** применяет модификацию ко всем оставшимся договорам. Также список договоров можно получить не только на вкладке Поиск, но и из файла, для этого доступны кнопки

Из файла(номера) - загрузить номера договоров из файла, каждый номер на отдельной строке.

Из файла(коды) - загрузить коды (id) договоров из файла, каждый код на отдельной строке.

Операция "Изменение статуса"

Смена статуса договора. При этом происходит корректное перекрытие существующих на моменты времени статусов с тем, чтобы на каждый день была активна только одна запись о статусе.

<input checked="" type="checkbox"/> Изменение статуса		
Изменяются таблицы contract_status		
С даты:	<input type="text"/>	по дату: <input type="text"/> Подключить ▼
Комментарий:	<input type="text"/>	

Операция "Добавление группы тарифов"

Добавление групп тарифов в договоры.

<input checked="" type="checkbox"/> Добавление группы тарифов	
Изменяется таблица contract_tariff_group	
<input checked="" type="checkbox"/> Добавлять только отсутствующие	Группы тарифов
	<input type="checkbox"/> Вторая <input type="checkbox"/> Первая
Период	Комментарий
с <input type="text" value="18.03.2009"/> по <input type="text"/>	
<input type="button" value="Все"/> <input type="button" value="Сброс"/> <input type="button" value="Инверт."/>	

Операция "Открытие тарифных планов"

Открытие тарифных планов в договорах с определённым периодом.

Открытие тарифных планов

Изменяются таблицы contract_tariff и last_tariff_change

Период с по

Тарифы

- hlp
- IPNIPN
- New tariff plan 1
- nnp
- VOIP
- Новый1
- Тариф1

Показывать только используемые

Комментарий

Операция "Закрытие тарифных планов"

Закрытие тарифных планов в договорах с определённой даты.

Закрытие тарифных планов

Изменяются таблицы contract_tariff и last_tariff_change

Дата:

Тарифы

- hlp
- IPNIPN
- New tariff plan 1
- pnp
- VOIP
- Новый1
- Тариф1

Показывать только используемые

Операция "Добавление (Удаление) модулей"

С версии 4.6 появилась привязка договоров к модулям. Данная операция позволяет добавлять(удалять) модули в договоры. В договор будут добавлены только отсутствующие в нем модули.

The screenshot shows a dialog box titled "Добавление(Удаление) модулей" (Add/Remove modules). At the top, there is a checked checkbox for the title. Below it are two radio buttons: "Добавить" (Add) is selected, and "Удалить" (Remove) is unselected. A list box labeled "Модули" (Modules) contains the following items, each with an unchecked checkbox: "VoIP", "222222222222222222222222222222", "Карточки" (Cards), "Супер Карты" (Super Cards), "IPN", "3333", "1111111111111111111111111111", and "NET". At the bottom of the dialog, there are three buttons: "Все" (All), "Сброс" (Reset), and "Инверт." (Invert).

Операция "Добавление разрешённых услуг"

С версии 4.6 появилась привязка договоров к модулям. Для некоторых модулей необходимо добавить разрешённые услуги. Данная операция позволяет добавлять разрешённые услуги для модулей, в которых это имеет смысл. Услуги будут добавлены только в том случае, когда модуль подключён к договору.

Добавление разрешенных услуг

Модуль:

Период: с по

Услуги

Только используемые

- Dial-Up(время)
- Dial-Up(входящий)
- Dial-Up(исходящий)
- Dial-Up(локальный/время)
- Dial-Up(локальный/входящий)
- Dial-Up(локальный/исходящий)
- Dial-UP(максимальный)
- Dial-Up(превалирующий)
- Dial-Up(суммарный)

Комментарий

Операция "Удаление(Прерывание на период) разрешённых услуг"

Данная операция позволяет удалять, либо прерывать на период разрешённые услуги для модулей.

Удаление(Прерывание на период) разрешенных услуг

Модуль
Карточки

Период
с 24.03.2009 по

Действие
 Прервать Удалить

Услуги
 Только используемые

- Агентское вознаграждение
- Активация DialUp карт
- Карточки
- Платежи

Все Сброс Инvert.

Операция "Смена тарифа"

Операция производит закрытие одного тарифа и открытие другого. Операция выполняется только в случае, когда закрываемый тариф открыт на дату закрытия.

<input checked="" type="checkbox"/> Смена тарифа			
Изменяются таблицы contract_tariff и last_tariff_change			
По дате:	<input type="text"/>	закреть тариф	---
С даты:	<input type="text"/>	открыть тариф	---
<input checked="" type="checkbox"/> Показывать только используемые			
Комментарий:	<input type="text"/>		

Операция "Добавление скрипта"

Операция добавляет скрипт в договоры.

<input checked="" type="checkbox"/> Добавление скрипта
Изменяется таблица contract_script
<input checked="" type="checkbox"/> Добавлять только отсутствующие
Скрипт: <input type="text" value="1111"/> с <input type="text" value="18.03.2009"/> по <input type="text" value="18.04.2009"/>
Комментарий: <input type="text"/>

Операция "Установка шаблона комментария договору"

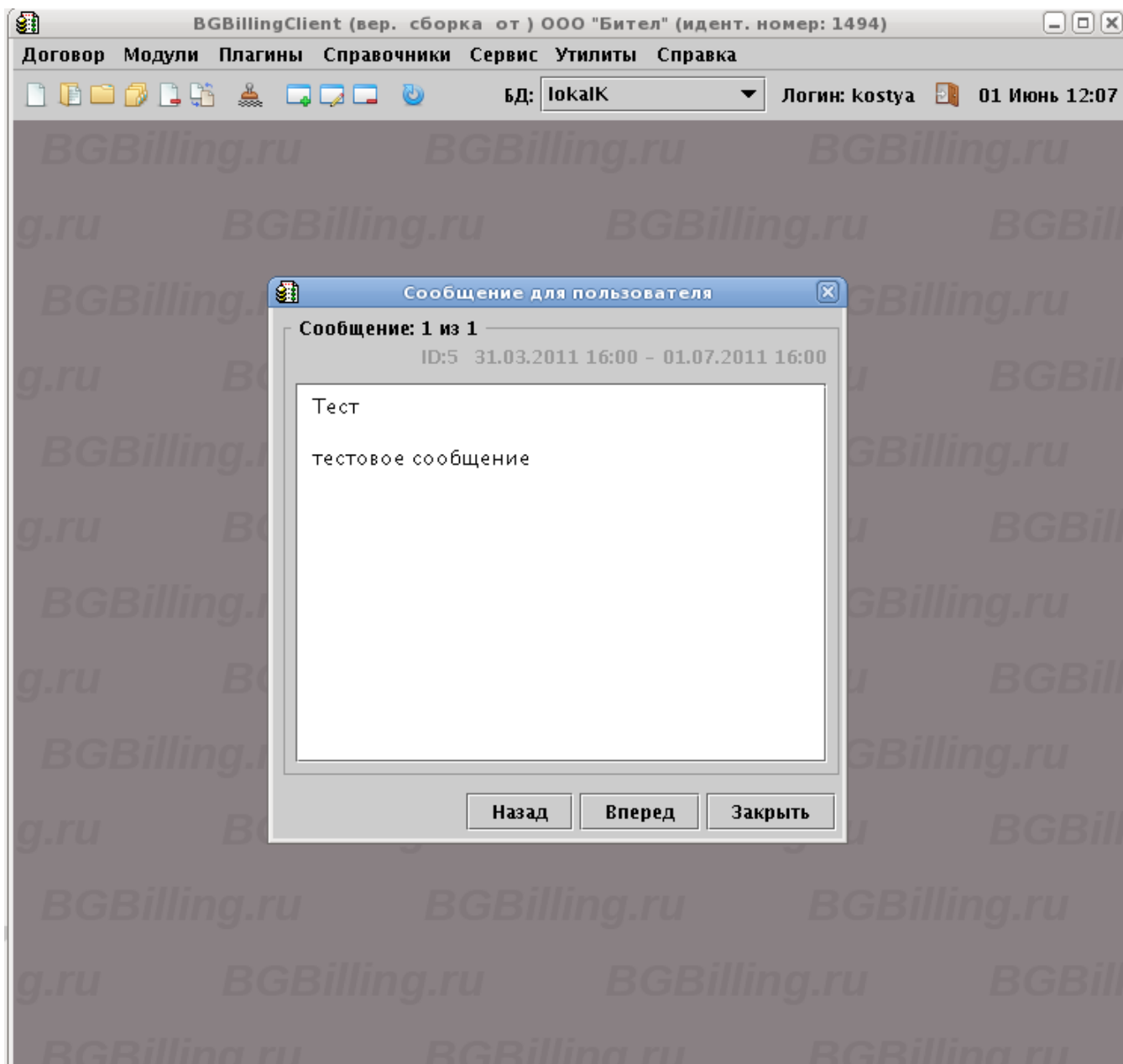
Операция устанавливает шаблон комментария договорам

<input checked="" type="checkbox"/> Установка шаблона комментария договору	
Шаблоны:	Тестовый шаблон

Сообщения пользователям

Данная подсистема позволяет отправлять сообщения пользователям биллинга и доступна в меню **Сервис=>Настройка=>Сообщения пользователям**. Сообщения отображаются после авторизации пользователя в биллинге или во время работы, блокируя весь интерфейс, что не позволяет пользователю проигнорировать их. Для проверки новых сообщений пользователю используется таймер, промежуток времени срабатывания которого можно указать в конфигурации сервера в миллисекундах. Если параметр не задан, то значение его равно одному часу.

```
client.gui.time.of.check.messages4users=3600000
```



При создании сообщения необходимо указать заголовок, тело сообщения, пользователей и период(период не может быть бесконечным, т.е. обязательно должна быть дата по какое число сообщение является актуальным), в течение которого оно будет отображаться пользователям, если не прочитано.

Для выбора пользователей, которым необходимо отправить сообщение, можно использовать группы пользователей или же в ручном режиме на закладке "**Список пользователей**" можно выбрать нужных пользователей или дополнить список тех, которые были отмечены автоматически при выборе группы.



Список сообщений

- 4 [31.03.2011 00:00 - 27.05.2011 00:00]
- 5 [31.03.2011 16:00 - 01.07.2011 16:00] Тест
- 6 [08.03.2011 00:00 - 30.06.2011 00:00]
- 7 [15.03.2011 00:00 - 18.05.2011 00:00]
- 8 [07.06.2011 15:00 -] Тест 2
- 9 [07.06.2011 15:00 - 05.02.2012 00:00] Тест3
- 10 [25.01.2012 16:00 - 01.04.2012 16:00] тест 5

Заголовок

тест 5

Период действия сообщения

с: 25.01.2012 16:00 по: 01.04.2012 16:00

Редактор

тест 5

перенос строк

Список групп

Список пользователей

- WiFi
- Группа 1
- Группа 2
- Запрещены группы договоров
- Контроллеры
- может всё, или почти всё
- некая группа
- тест 1
- тест 2

Все

Сброс

Инверт.

OK

Отмена

Индикатор лицензии

Индикатор лицензии вызывается из меню **Справка=>Индикатор лицензии**. В нём отображается состояние лицензии для каждого модуля и плагина. Отражены как временные, так и количественные характеристики.

Каждый строка отображает максимально разрешённое и текущее состояние по количеству и времени. Параметры которые превысили максимально допустимое значение выделяются красным цветом.

С версии 7.2 на закладке лицензий кроме просмотра состава текущей лицензии доступен список лицензий из ЛК договора на биллинг (для отображения списка доступных лицензий необходимо в конфигурации сервера прописать параметры **license.contract.title=<номер договора, номер ВХХХХ-УУ>** и **license.contract.password=<пароль от ЛК>**)

Доступные лицензии:						Состав лицензии:			
ID	Активна	Дата создания	Компонентов	Хранилище	Хэш	Наименование модул...	Количество лицензий	Действует до	...
5 830	<input checked="" type="checkbox"/>	20.11.2018	76	база	FF6BC52B789B5B6886E976B2C...	agr	0 из неограничено (...)	30.12.2050	▲
5 721	<input type="checkbox"/>	30.08.2018	75	сервер	DCEDF81BB046898D06B8324F0...	assist	0 из неограничено (...)	30.12.2050	
5 650	<input type="checkbox"/>	10.07.2018	74	сервер	58D93054FC776CE5FEA8E26C3...	bill	1773 из неограниче...	30.12.2050	
5 649	<input type="checkbox"/>	10.07.2018	74	сервер	CCAC93B2E905C7053B67FB936...	buymoney	0 из неограничено (...)	30.12.2050	
5 514	<input type="checkbox"/>	07.03.2018	73	сервер	D2A023C5EB6FF78B05A735B6D...	bvcom	0 из неограничено (...)	30.12.2050	
5 466	<input type="checkbox"/>	01.02.2018	72	сервер	E9732024ED1594463924289C2...	card	136 из неограничен...	30.12.2050	
5 437	<input type="checkbox"/>	11.01.2018	71	база	7F7C732CR366AD1DD68BE19E7...	cerbercrypt	0 из неограничено (...)	30.12.2050	
5 163	<input type="checkbox"/>	22.06.2017	70	база	624762EC92AE36F3B9781978A...	dba	0 из неограничено (...)	30.12.2050	
5 162	<input type="checkbox"/>	22.06.2017	68	сервер	15D1A3F8D070043C1E74DB5FA...	dialup	0 из неограничено (...)	07.04.2050	
4 730	<input type="checkbox"/>	26.09.2016	69	сервер	AD96A5ED8075F3A9DE4DF35BB...	dialup.wifi	0 из неограничено (...)	07.04.2050	
4 687	<input type="checkbox"/>	23.08.2016	68	база	C4FE8904B1AB74CB80CCC34DB...	drweb	0 из неограничено (...)	30.12.2050	
4 686	<input type="checkbox"/>	22.08.2016	67	сервер	2A62683C15DE705CBD699EA8E...	drwebn	0 из неограничено (...)	30.12.2050	
4 683	<input type="checkbox"/>	22.08.2016	66	база	B40696BEB35EF61ED0832C4CF...	email	0 из неограничено (...)	30.12.2050	
4 600	<input type="checkbox"/>	27.06.2016	66	база	62A6340D0BFFB7AE8CE5E4AC7...	enaza	0 из неограничено (...)	30.12.2050	
4 583	<input type="checkbox"/>	20.06.2016	65	сервер	BEEE381CFF2EE7908244B9249...	forpost	0 из неограничено (...)	30.12.2050	
4 582	<input type="checkbox"/>	20.06.2016	64	база	72B94145A7D267FEB78EB9B71...	gazprombank	0 из неограничено (...)	30.12.2050	
4 471	<input type="checkbox"/>	14.04.2016	63	база	BCB8AD7EAF14F46E5DC22F40...	gorod	134 из неограничен...	30.12.2050	
4 451	<input type="checkbox"/>	08.04.2016	64	база	FEC4CEC3BE20CADC209CEF5C9...	hd	501 из неограничен...	30.12.2050	
4 446	<input type="checkbox"/>	06.04.2016	64	база	08C2B38FE69B8D3D5CEFC440...	inet	0 из неограничено (...)	30.12.2050	
4 445	<input type="checkbox"/>	06.04.2016	64	сервер	B06CBCB11FC600A95B3577E71...	inet.wifi	0 из неограничено (...)	30.12.2050	
4 444	<input type="checkbox"/>	06.04.2016	63	сервер	7A0FA111C302815F1DC038512...	ipn	0 из неограничено (...)	07.04.2050	
4 440	<input type="checkbox"/>	05.04.2016	60	сервер	314304B5C820921839564088B...	license	854 из неограничен...	30.12.2050	
4 437	<input type="checkbox"/>	05.04.2016	60	база	E471A1083D92DC4C4DAD1CF66...	megogo	0 из неограничено (...)	30.12.2050	
						mobicash	0 из неограничено (...)	30.12.2050	
						mobile	0 из неограничено (...)	30.12.2050	
						mobimoney	0 из неограничено (...)	30.12.2050	

X Просмотр лицензий

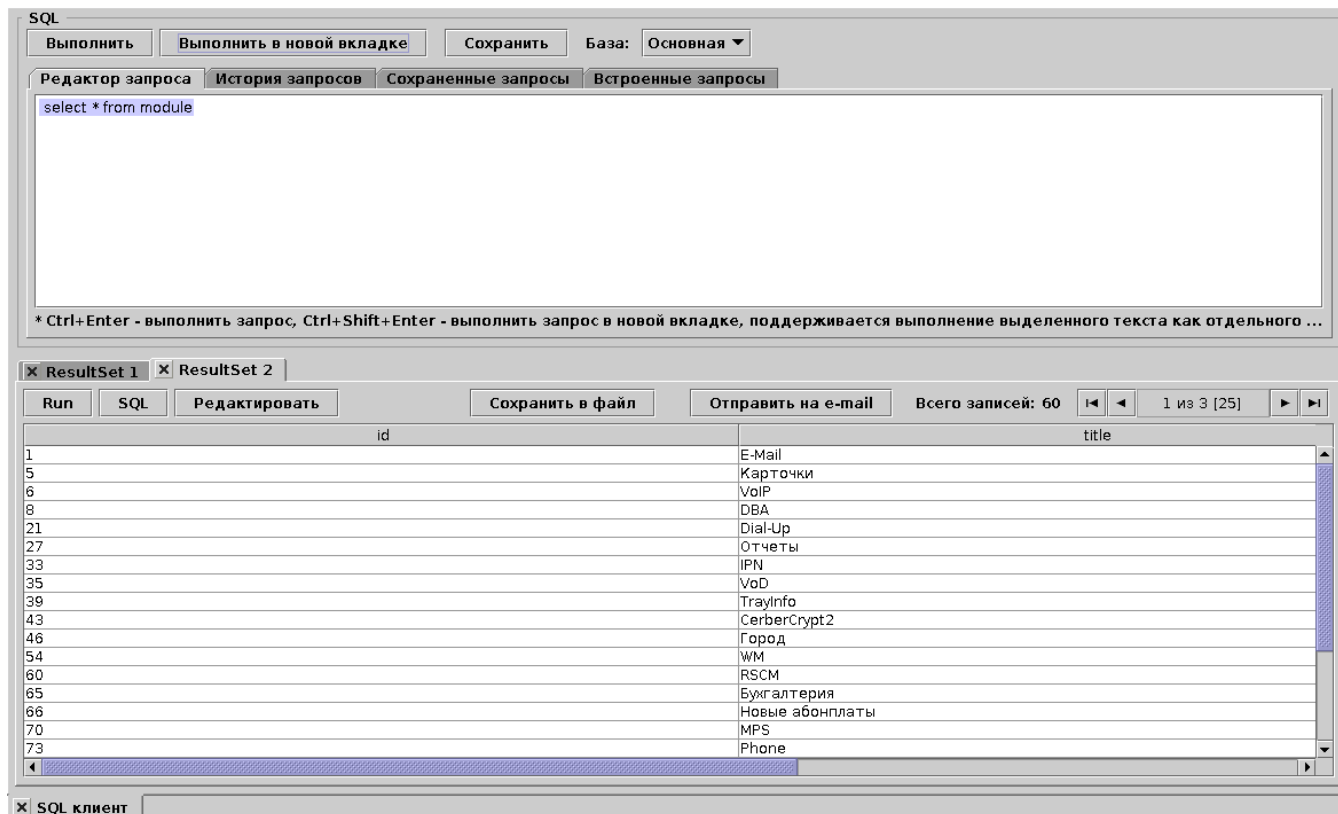
так же из данного интерфейса возможно активировать любую из доступных лицензий с помощью контекстного меню

1.2018	74	сервер	CCAC93B2E905C7053B67FB936...	buymoney
3.2018	73	сервер	D2A023C5EB6FF78B05A735B6D...	bvcom
2.2018	72	сервер	E9732024ED1594463924289C2...	card
1.2018	71	база	7F7C732CR366AD1DD68BE19E7...	cerbercrypt
6.2017	70	база	624762EC92AE36F3B9781978A...	dba
6.2017	68	сервер	15D1A3F8D070043C1E74DB5FA...	dialup
9.2016	69	сервер	AD96A5ED8075F3A9DE4DF35BB...	dialup.wifi
8.2016	68	база	C4FE8904B1AB74CB80CCC34DB...	drweb
8.2016	67	сервер	2A62683C15DE705CBD699EA8E...	drwebn
8.2016	66	база	B40696BEB35EF61ED0832C4CF...	email
6.2016	66	база	62A6340D0BFFB7AE8CE5E4AC7...	enaza
6.2016	65	сервер	BEEE381CFF2EE7908244B9249...	forpost
6.2016	64	база	72B94145A7D267FEB78EB9B71...	gazprombank
4.2016	63	база	BCB8AD7EAF14F46E5DC22F40...	gorod
4.2016	64	база	FEC4CEC3BE20CADC209CEF5C9...	hd

Установить и активировать лицензию
 Просмотр состава лицензии
 Загрузить/перезагрузить лицензию
 Удалить лицензию из базы

SQL Редактор

SQL редактор (**Утилиты=>SQL-Редактор**) представляет собой средство низкоуровневого доступа к БД биллинга. С помощью него можно выполнять произвольные SQL-запросы, получать результаты, сохранять их в файл и отправлять на указанный e-mail.



Каждый запрос можно сохранить для последующего использования, правки и повторного выполнения. Результаты выполнения запроса отображаются в таблице (с возможностью постраничного просмотра). Колонки соответствуют названиям столбцов результата запроса, для этого важно что бы в конце запроса не было знака точка с запятой.

Ведётся локальная история выполненных запросов - они доступны для просмотра и повторения во вкладке "Сохраненные запросы".

На закладках с результатами доступны кнопки "**Отправить на e-mail**" и "**Сохранить в файл**" с помощью которых результаты запросов можно соответственно отправить на электронный адрес или сохранить в файл на локальном компьютере.

Поля запроса и результата разделены подвижным Split-ом.

Редактор SQL - мощное и гибкое средство для манипулирования данными, их обработки и анализа. Но, однако, даже администратору следует применять его с большой осторожностью для выполнения запросов вида UPDATE, DELETE и подобных.

Администрирование и оптимизация

Конфигурация базы данных, память

Каждое приложение биллинга инициализирует пул соединений к базе данных. Необходимость пула вызвана многопоточностью приложений, т. к. в один момент времени соединение с базой данных может потребоваться разным потокам. Пул предотвращает необходимость постоянного создания TCP соединений с MySQL, экономя ресурсы как клиентского приложения, так и MySQL-сервера.

В конфигурации приложения должна быть определена как минимум одна главная (мастер) база. Эта база постоянно актуальна и может выполнять как запросы обновления, так и выборки. В *.properties конфигурации приложения биллинга определены хост, порт MySQL, имя базы данных и учётные данные для MySQL, например так:

```
db.driver=com.mysql.jdbc.Driver
db.url=jdbc:mysql://192.168.184.245/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&elideSetAutoCommit=true&useCursorFetch=true&queryTimeoutKillsConnection=true
db.user=bill
db.pswd=bgbilling
db.maxIdle=10
db.maxActive=50
```

Параметр **db.maxIdle** определяет максимально число простаивающих в данный момент соединений с базой данных; простаивающие соединения, выходящие за данное количество, закрываются. При большом одновременном количестве запросов к приложению биллинга число активных соединений растёт, по мере надобности устанавливаются новые соединения с MySQL, добавляясь в пул. При достижении активного количества соединений **db.maxActive** создание соединений прекращается, потоки ожидают освобождения уже занятых соединений.

Каждое приложение биллинга способно возвращать свой статус, вызовом ***_status.*** скрипта (server_status.sh, radius_status.sh). Сообщение о статусе приложения обязательно содержит время старта приложения и время работы, информацию по соединениям с базой данных и используемой памяти. Рассмотрим как пример вывод server_status.sh:

```
BGBillingServer v 4.6 build 333 from 01.03.2009
Started: 11.03.2009 16:29:49 Uptime: 0 d 00:00:05
Memory total: 6 045 696; max: 66 650 112; free: 2 975 800
Connections pool to Master status Idle: 1; Active: 0; maxActive: 4; maxIdle: 10
Connections pool to Slave "1" status Idle: 0; Active: 0; maxActive: 4; maxIdle: 10
```

В верхней строке отображается версия, номер и дата билда приложения (ядра для сервера). Далее - время старта и время работы.

Затем - использование ОЗУ. Параметр **max** определяет максимальный объем памяти, которую Java-машина может получить у операционной системы; **total** - сколько реально отобрано Java-машиной памяти в настоящий момент; **free** - сколько из этой реально выделенной памяти свободно.

По мере работы сборщик мусора освобождает неиспользуемую память. Если использование памяти все равно растёт - увеличивается параметр total. Однако он никогда не может превысить порога, заданного max. Размер памяти max можно увеличивать, изменяя число после параметра - Xmx в скрипте запуска приложения. На 32х разрядных ОС невозможна установка параметра max более 1.5 Гб. При нормальной работе приложения total не должен достигать max.

Последние строки (одна или более) отображают статусы пулов соединений к мастер и слейв базам данных.

Поддержка репликации

Приложения биллинга могут использовать MySQL-реплику для произведения выборки, приводящих к значительной нагрузке на сервер БД и не требующих при этом немедленной оперативности выполнения. Например, выполнение отчётов модулем отчётов, получение статистики клиентом через Web интерфейс.

Для добавления реплики в *.properties файле приложения (data.properties, radius.properties) указывается после конфигурации основной (Мастер) базы:

```
db.slave.<slave_id>.url=jdbc:mysql://<host>:<port>/<db_name>?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&elideSetAutoCommit=true&useCursorFetch=true&queryTimeoutKillsConnection=true
db.slave.<slave_id>.user=<user>
db.slave.<slave_id>.pswd=<pswd>
db.slave.<slave_id>.maxIdle=<max_idle>
db.slave.<slave_id>.maxActive=<max_active>
```

Где:

- **<slave_id>** - идентификатор Slave-базы;
- **<db_name>** - имя базы данных;
- **<host>** - хост с Slave-базой;
- **<port>** - MySQL порт;
- **<user>** - логин MySQL;
- **<pswd>** - пароль MySQL;
- **<max_idle>** - максимальное число простаивающих соединений в пуле, лишние будут закрыты;
- **<max_active>** - максимальное число активных соединений в пуле.

Для каждого приложения биллинга реплики указываются отдельно в *.properties файле, что позволяет регулировать использование реплик различными приложениями. Необходимо проконтролировать, чтобы пользователь <user> имел права только на SELECT и CREATE TEMPORARY TABLES в реплике. Также хорошим вариантом является выдача полного набора прав с установкой опции **--read_only=1** при старте сервера MySQL (либо установка этой же опции в my.cnf файле).

Например, конфигурация Slave-базы может выглядеть следующим образом:

```
db.slave.1.url=jdbc:mysql://repl:3306/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&elideSetAutoCommit=true&useCursorFetch=true&queryTimeoutKillsConnection=true
db.slave.1.user=bill
db.slave.1.pswd=
db.slave.1.maxIdle=10
db.slave.1.db.maxActive=20
```

Возможно определение нескольких Slave-серверов с разными <slave_id>. В этом случае запрос будет адресован к серверу, имеющему наименьшее отношение активных соединений в настоящий момент к максимальному числу активных соединений. В случае, если количество соединений ко всем подключённым Slave-базам исчерпано, будет возвращено подключение к Мастер-базе.

Можно отключить возможность предотвращения перерасхода slave-соединений, тогда попытки взять соединение из Master-БД делаться не будет, а будет ожидание освобождения slave-пула. Для этого в data.properties надо добавить параметр

```
# slave- (
# slave-, , master-).
db.disable.prevention.slave.overrun=1
```

Для отслеживания актуальности реплик установите в конфигурации сервера биллинга параметр **slave.alarm.second.behind.master=<seconds>**, где <seconds> - количество секунд. При отставании реплики от основной базы на <seconds> и более секунд высылается аларм. Опция, установленная в конфигурации сервера биллинга, применяется ко всем приложениям биллинга. В качестве теста возможна установка опции в 0, что должно спровоцировать высылку сообщения. Для работы опции пользователь MySQL приложения биллинга должен иметь в Slave-базах право **REPLICATION CLIENT**.

Также возможно включить возможность автоматического отключения сильно отстающих реплик с последующим их автоматическим же включением в случае ликвидации критического отставания. Для этого необходимо в конфигурации сервера биллинга установить параметр **slave.disable.second.behind.master=<seconds>**, где <seconds> - количество секунд. При отставании реплики от основной базы на <seconds> и более секунд высылается аларм, а сервер биллинга, в свою очередь, помечает данную реплику отстающей и прекращает обращения к ней. Через некоторое время, если отставание реплики стало менее <seconds>, то обращения к данной реплики биллингом возобновятся автоматически.

"Мусорные" базы данных

"Мусорная" база - это отдельная база данных на ненадёжных быстрых носителях, предназначенная для хранения слабо связанных не критичных для основного сервиса данных. Для неё не используется репликация. Выборки по такой базе производятся редко. Обрыв соединения с "мусорной" базой не критичен для основного сервиса.

В данный момент поддержано хранение в "Мусорной" базе таблиц:

- **log_function_process_yyyyMM** - логи работы VGBS-скриптов на договорах;
- **log_server_<mid>_yyyyMM** - логи RADIUS-запросов модулей DialUp/VoiceIp;
- **reject_to_accept_<mid>** - подмена reject'a accept'ом при ошибке авторизации.

"Мусорная" база определяется в конфигурации сервера биллинга (**Сервис=>Настройки**) следующим образом:

```
db.trash.<trash_id>.url=jdbc:mysql://<host>:<port>/<db_name>?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&elideSetAutoCommit=true&useCursorFetch=true&queryTimeoutKillsConnection=true
db.trash.<trash_id>.user=<user>
db.trash.<trash_id>.pswd=<pswd>
db.trash.<trash_id>.maxIdle=<max_idle>
db.trash.<trash_id>.maxActive=<max_active>
```

Где:

- **<trash_id>** - идентификатор "Мусорной" базы;
- **<db_name>** - имя базы данных;
- **<host>** - хост с Мусорной базой;
- **<port>** - MySQL порт;
- **<user>** - логин MySQL;
- **<pswd>** - пароль MySQL;
- **<max_idle>** - максимальное число простаивающих соединений в пуле, лишние будут закрыты;
- **<max_active>** - максимальное число активных соединений в пуле.

После определения "мусорной" базы необходимо указать префиксы таблиц, которые хранятся в этой базе:

```
trash.table.map.<pos>.<table_prefix>=<trash_id>
```

Где:

- **<trash_id>** - идентификатор "Мусорной" базы;
- **<pos>** - порядковый номер правила для просмотра;
- **<table_prefix>** - префикс таблицы.

Например, так можно определить хранение таблиц log_function_process в отдельной мусорной базе:

```
db.trash.trash_1.url=jdbc:mysql://192.168.184.2:3306/trash_1?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&elideSetAutoCommit=true&useCursorFetch=true&queryTimeoutKillsConnection=true
db.trash.trash_1.user=root
db.trash.trash_1.pswd=
db.slave.trash_1.maxIdle=10
db.slave.trash_1.maxActive=4
#
trash.table.map.1.log_function_process=trash_1
```

Функционал "мусорных" хранилищ экспериментальный, рекомендуется начать использование с таблиц, не используемых критичными сервисами (log_function_process).

Настройка типа хранения помесячных и подневных таблиц в MySQL

Для периодических таблиц (помесячных вида **xxx_mid_yyyyMM** и подневных вида **xxx_mid_yyyyMMdd**) таблиц можно задавать отдельно "движок" и папку для хранения данных и индексов (эта опция поддерживается только для MyIsam). Необходимо указывать в конфигурации сервера биллинга (меню **Сервис=>Настройка=>Конфигурация**).

```
table.create.<table_name>.data.directory=<data_dir>
table.create.<table_name>.index.directory=<index_dir>
table.create.<table_name>.engine=<engine>
```

Где:

- **<table_name>** - это часть имени периодической таблицы, в которую не входит дата;
- **<data_dir>** - каталог для хранения данных таблицы (работает только для MyIsam-таблиц);
- **<index_dir>** - каталог для хранения индексов таблицы (работает только для MyIsam-таблиц);
- **<engine>** - "движок" хранения таблиц (InnoDB, MyIsam).

Например:

```
table.create.log_session_12.data.directory=/home/mysql
table.create.log_session_12.index.directory=/home/mysql
table.create.log_session_12.engine=INNODB
```

Указанные значения преобразуются в атрибуты DATA DIRECTORY, INDEX DIRECTORY и ENGINE команды CREATE TABLE при создании новой периодической таблицы. Более подробно читайте об этом в документации по MySQL.

Рекомендуется использование движка InnoDB для критических таблиц с большим количеством конкурирующих операций чтения-записи и MyIsam-движка для редко считываемых таблиц (различные логи).

Для оптимизации вставки в таблицы логов часто используется опция DELAYED INSERT, которая поддерживается только для "движка" MyIsam.

Параметры запуска клиента

В файле запуска клиента можно указывать следующие параметры:

```
-Dbgbilling.transfer.debug=true - debug
-Dbgbilling.client.os=linux - .
-Dlocal.setting.file.name=config_v.4.5 - .
-Dsun.net.client.defaultConnectTimeout=1000 - URL , FOP, , .
-Duser.home - ( local.setting.file.name)
```

Дополнительные дефолтные настройки локализации, tz и т. Д.

Для приложений биллинга можно указать в конфигурации некоторые настройки, перетирающие default-значения локали, timezone и некоторых других дефолтных значений, связанных с локалью.

Настройки делаются в `data.properties` сервера и в `client.properties` клиента. Для клиента это имеет меньше смысла, а некоторые не имеют никакого значения, как `tz`, например, которая всегда синхронизируется с текущим сервером после подключения.

```
# TZ, , ( , .. TZ )
locale.default.tz=UTC
# , ,
locale.default.locale=de
# DecimalFormatSymbols
locale.default.decimalformatsymbols.decimalseparator=,
```

Импорт договоров

Программа поддерживает ограниченный импорт договоров из csv файла

Интерфейс импорта доступен из клиента **СЕРВИС АВТОМАТИЗАЦИЯ ИМПОРТ ДОГОВОРОВ**

Список утилит

- Импорт договоров
- Импорт баланса
- Изменение текст. параметра договора
- Слияние текст. параметров договора
- Конвертирование параметров договора
- Удаление параметра договора

Импорт договоров

очистить базу перед импортом

Файл с данными

Шаблон договора

шаблон из файла (столбец contractPatterId)

применить для всех шаблон -> 1111111111111111

Договор

Номер договора: из шаблона из колонки ->

Дата создания договора: текущая из колонки ->

Мэпнинг параметров

Поле в файле	Значение в файле
--------------	------------------

Слева список доступных инструментов, справа настройки выбранного инструмента.

Импорт договоров

Данный инструмент позволяет загрузить договора из csv файла по заданному шаблону и заполнить параметры договоров.

Пример формата csv файла

```
contractPatternId,contractTitle,contractPassword,parameter_1,parameter_2,parameter_3,parameter_phone,  
parameter_5,parameter_email,parameter_7,parameter_8  
0,B00094,password,__1,__2,__3,,__4,"", "",
```

название параметров может быть любым, но есть несколько зарезервированных имен, при использовании которых программа автоматически заполнит необходимые настройки инструмента

contractPatternId - код шаблона договора

contractTitle - название договора

contractPassword - пароль договора

parameter_<ID> - параметр договора с соответствующим ID

Импорт договоров

очистить базу перед импортом

Файл с данными

Шаблон договора
 шаблон из файла (столбец contractPatterId)
 применить для всех шаблон ->

Договор
 Номер договора: из шаблона из колонки ->
 Дата создания договора: текущая из колонки ->

Мэпнинг параметров

Поле в файле	Значение в файле	
parameter_1	erater	Ф.И.О. руководителя
parameter_2	reter ertert	Игнорировать
parameter_3	erg aegaERga arrga	Лиц. счёт
parameter_phone		Параметер не установлен
parameter_5	пфцуаыв	Параметер не установлен
parameter_email	ва ываы	Emails
parameter_7	ы аыаы, "аывпаапы	Параметер не установлен
parameter_8	ывывыва	Параметер не установлен

При выборе "очистить базу перед импортом" будут удалены договора у которых установлены метки договоров заданные в конфигурации сервера в параметре loader.import.contract.clean.label.ids=<ID1>,<ID2>,...

Расширение функциональности BGBilling

BGBilling поддерживает возможность гибкого расширения функциональности системы путём написания пользовательских скриптов, которые могут: обрабатывать различные события системы, выполняться обособленно, управлять устройствами и т.д. Имеется возможность разработки расширений функциональности на следующих языках:

- язык Java - обработчики событий могут быть реализованы в виде Java-классов, которые компилируются и загружаются динамически; использование Java в качестве пользовательских скриптов предпочтительнее как с точки зрения удобства и возможностей разработки, так и с точки зрения производительности самих скриптов (быстрее до 50 раз).
- язык BGBS - это интерпретируемый Java-подобный язык программирования, который хорошо подходит для написания небольших скриптов, время выполнения которых не критично. BGBS представляет собой интегрированный в биллинг интерпретатор языка BeanShell (<http://www.beanshell.org/>) + API для управления данными биллинга и набор событий биллинга, которые можно обрабатывать. Разработка на BGBS требует знание языка Java, т.к. большинство вызовов BeanShell прозрачно переадресуются Java API. Синтаксис BeanShell также практически полностью идентичен Java.;

Язык BeanShell уже морально устарел и оставлен для совместимости, чтобы работали старые скрипты. Все новые скрипты нужно писать на java.

Пользовательские скрипты Java/BGBS могут использоваться как:

- **Скрипты поведения** - привязанные к договорам и обрабатывающие определенные события, происходящие с ними;
- **Функции глобальных событий** - обработчики событий системы, не связанных с конкретными договорами;
- **Глобальные скрипты поведения** - выполняемые периодически или единоразово действия.

Только пользовательские скрипты в виде динамических Java классов могут использоваться как:

- **Скрипты пред- и постобработки запросов, скрипты управления сервисом** в модуле Inet.

Только пользовательские BGBS скрипты могут использоваться как:

- **Скрипты предобработки RADIUS запросов**, привязанных к NASам модулей DialUp/VoiceIp и производящих предобработку RADIUS запросов;
- **Скриптовые шлюзы модуля IPN** - скрипт производит управление шлюзом.

Примеры использования скриптов доступны в базе знаний [Wiki](#). Вы также можете публиковать там свои разработки.

При разработке скриптов будет полезен наш API, ссылка на который есть на сайте <http://bgbilling.ru>. (например, [API 7.1](#))

Управление динамическим кодом

Для более полного понимания данной секции необходимо владеть базовыми знаниями Java (прочитать об этом можно, например, на официальном сайте <http://download.oracle.com/javase/tutorial/>).

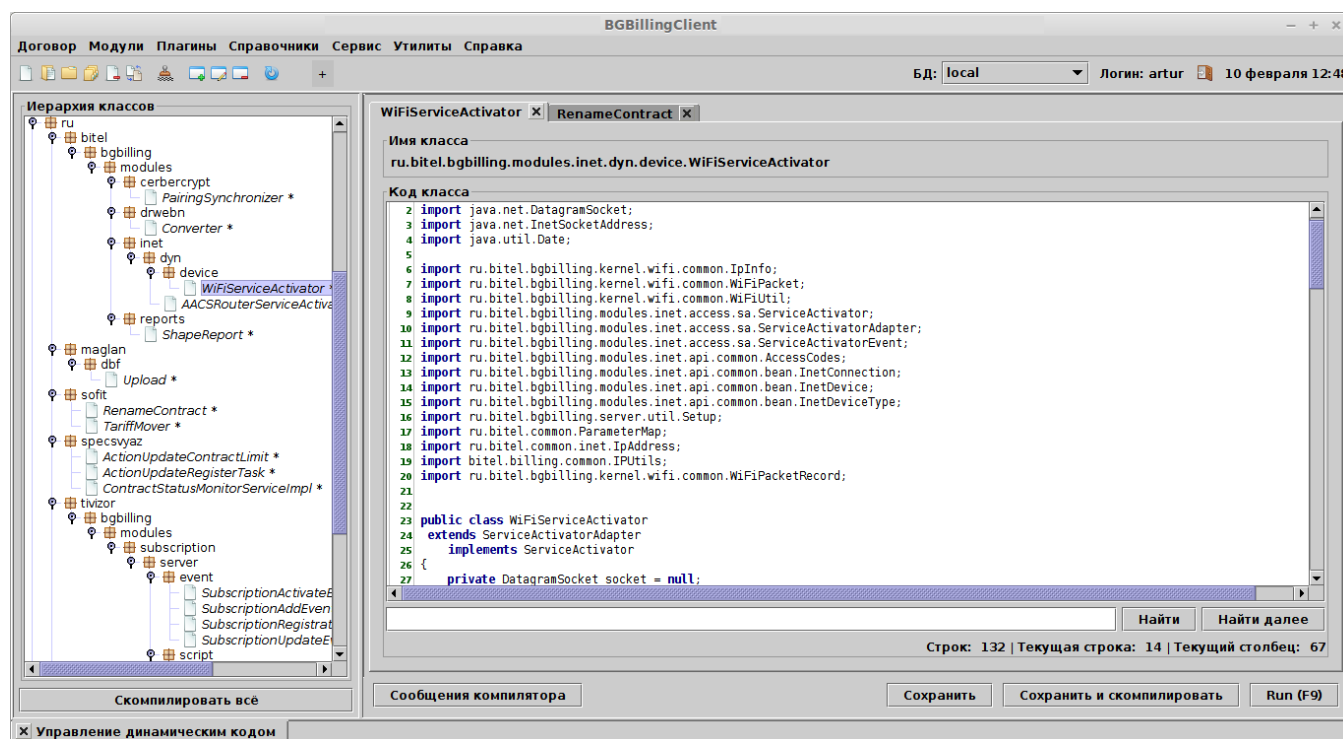
Параметры подсистемы динамического кода настраиваются в [конфигурации сервера](#).

Для встраивания дополнительной функциональности посредством написания Java-классов, необходимо предварительно скомпилировать и загрузить их в базу данных биллинга. Уникальным идентификатором каждого динамического класса является его полное имя - это имя, включающее как имя пакета (спеобразного пространства имён в Java - оно совпадает со структурой каталогов, в которых лежит класс) класса, так и самого имени класса. Например: **ru.bitel.bgbilling.Test** - это полное имя класса **Test**, который лежит в пакете **ru.bitel.bgbilling**.

Исходные коды динамического Java-кода хранятся по умолчанию в каталоге **BGBillingServer/dyn**.

Каждый класс для реализации конкретной функциональности (например, скрипта поведения) должен реализовывать определённый интерфейс (в каждом случае свой). После этого его необходимо скомпилировать и он станет доступен для выбора в соответствующем меню привязки динамического класса.

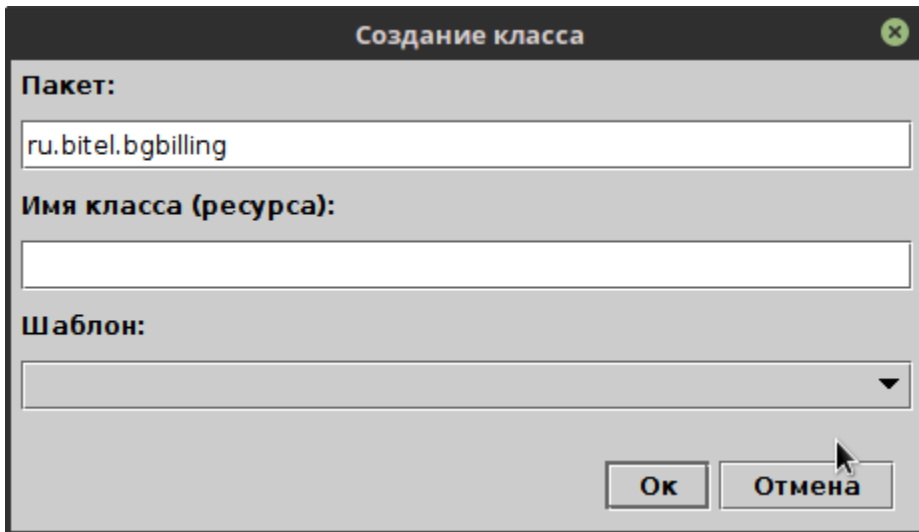
Для работы с динамически загружаемыми классами из клиента биллинга необходимо воспользоваться пунктом меню **Сервис=>Автоматизация=>Управление динамическим кодом**.



Дерево динамических классов можно видеть на панели слева. Изменённые, но не перекомпилированные классы отображаются в дереве со значком * (это будет работать ТОЛЬКО в случае, если файловая система, в которой хранятся скрипты, поддерживает хранение времени последнего изменения файла, в противном случае все исходные коды всегда будут помечены как изменённые).

Для открытия существующего класса на редактирование необходимо выбрать нужный класс в дереве классов слева и щелкнуть на кнопку **Редактировать** на стандартной панели инструментов, либо открыть класс двойным щелчком мыши. Каждый класс открывается в отдельной вкладке.

Для создания нового класса необходимо нажать кнопку **Новый элемент** на стандартной панели инструментов. Откроется диалоговое окно, в котором будет предложено ввести полное имя класса. Полное имя класса включает в себя пакет, в котором располагается класс. Пакет по своей сути представляет собой папку в файловой системе. Пакеты разделяются между собой символом "." (точка). Обратите внимание, что если в дереве классов выбрать предварительно какой-либо пакет и нажать кнопку создания нового класса, то этот пакет автоматически подставится в поле диалогового окна.



В выпадающем списке можно выбрать один из следующих пунктов:

- **Глобальный скрипт** - будет создан экземпляр класса, расширяющий абстрактный класс `ru.bitel.bgbilling.kernel.script.server.dev.GlobalScriptBase`. Данный класс можно запускать как глобальный скрипт, доступный через меню **Сервис -> Автоматизация -> Глобальные скрипты поведения**.
- **Отчеты: класс табличного отчета** - будет создан экземпляр класса, реализующего интерфейс `bitel.billing.server.reports.BGCSVReport.CSVFillerDataFields`, который используется для формирования данных в табличном отчете модуля Reports.
- **Отчеты: xml описатель отчета** - будет создан xml-файл с описанием фильтров отчета модуля Reports.

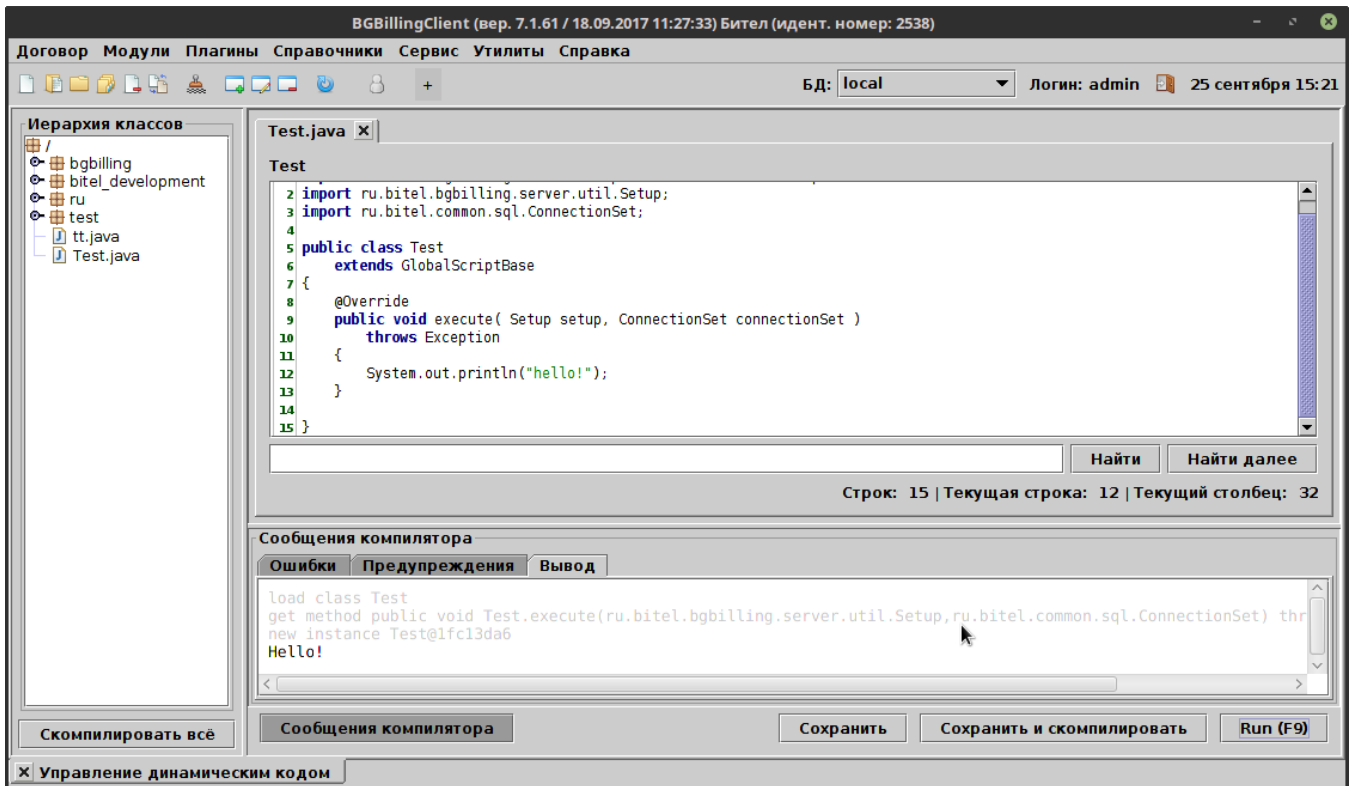
Для полной перекомпиляции всех исходных кодов необходимо нажать кнопку **Скомпилировать всё**. Для компиляции отдельных классов необходимо открыть их в дереве по двойному щелчку и нажать кнопку **Сохранить и скомпилировать**. Ошибки и предупреждения компилятора можно увидеть в открывающейся панели **Сообщения компилятора**. В случае ошибки в процессе компиляции некоторого класса никаких изменений в базе данных скомпилированных классов не произойдет. Это означает, что будет работать последняя работоспособная версия динамического класса.

Для удаления класса (или группы классов) необходимо выбрать их в дереве классов и нажать кнопку **Удалить** в панели инструментов. Перед удалением выбранных классов производится полная перекомпиляция динамических классов без участия удаляемых классов для проверки на наличие зависимостей оставшегося кода от них. В случае, если компиляция пройдет с ошибками, то удаление классов не будет произведено для сохранения работоспособности оставшегося кода!

Обратите внимание, что привязка удаляемых классов в различных подсистемах (скрипты поведения, глобальные скрипты и т.п.) не проверяется! Удаление классов производится на свой страх и риск. В случае оставшейся привязки удаленного класса к какой-либо подсистеме, корректное поведение этой подсистемы не гарантируется!

Одним из преимуществ такого подхода к разработке расширений биллинга состоит в том, что возможна организация полноценной работы в IDE (автодополнение, проверка кода на ошибки и т.п.). Пример, как это сделать, вы можете найти в [Wiki](#).

Для возможностей отладки и удобного запуска можно запустить код прямо из редактора.



Сначала пытается запускаться метод `execute` как у глобальных скриптов (можно просто добавить этот метод с такой сигнатурой). Если такой метод есть, то создаётся экземпляр класса и в него передаются объекты типа `ru.bitel.bgbilling.server.util.Setup` и `ru.bitel.common.sql.ConnectionSet`, как в глобальные скрипты. Первый содержит в себе конфигурацию приложения, второй - набор соединений к БД (основной, slave и "мусорной" и при наличии). Если такого метода нет, то ищется стандартный метод `public static void main(String[] args)`.

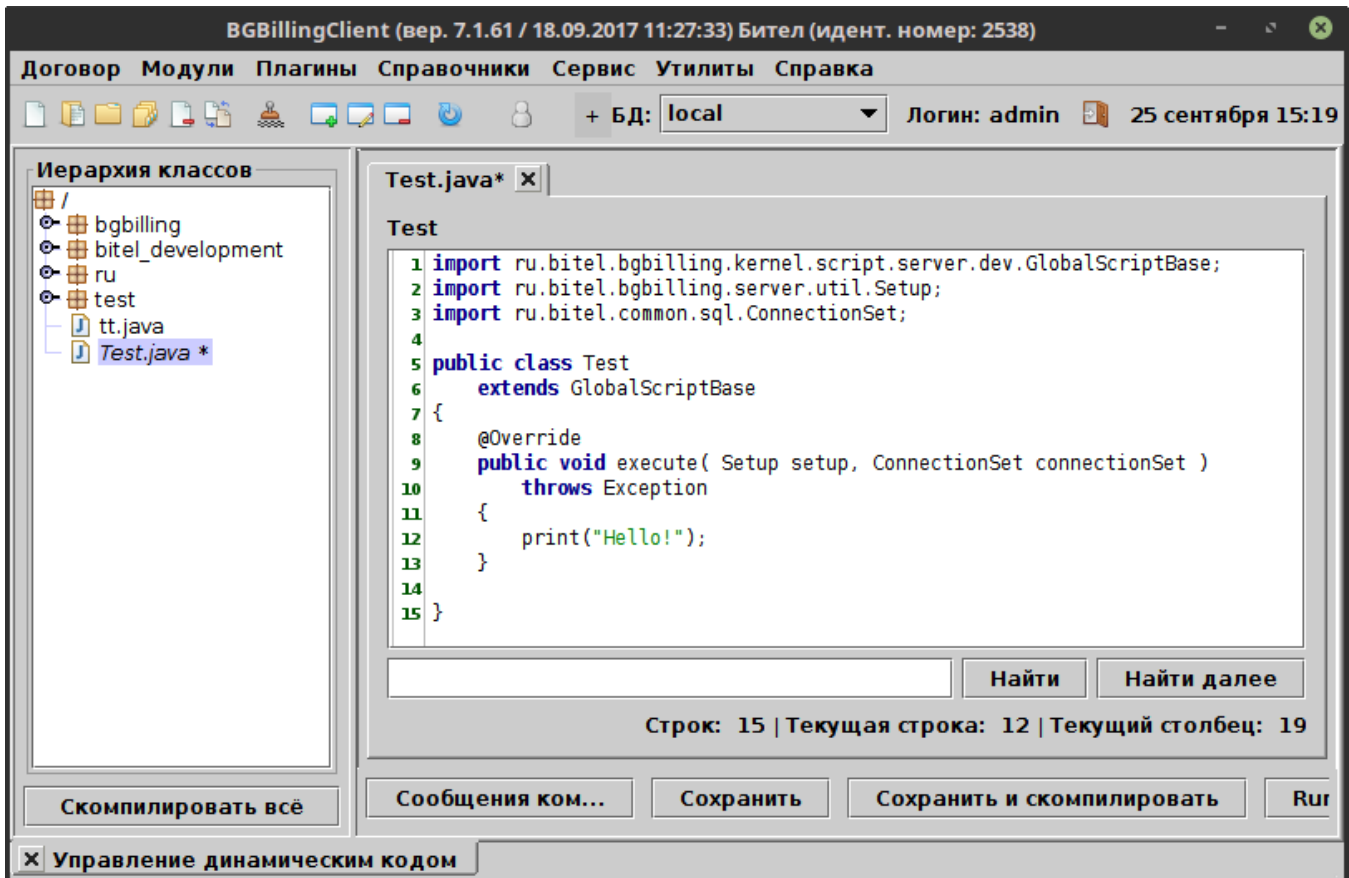
Работа с встроенным редактором

Каждый редактируемый файл открывается в своей собственной вкладке. Ошибки, предупреждения, результат выполнения у каждой вкладки свои.

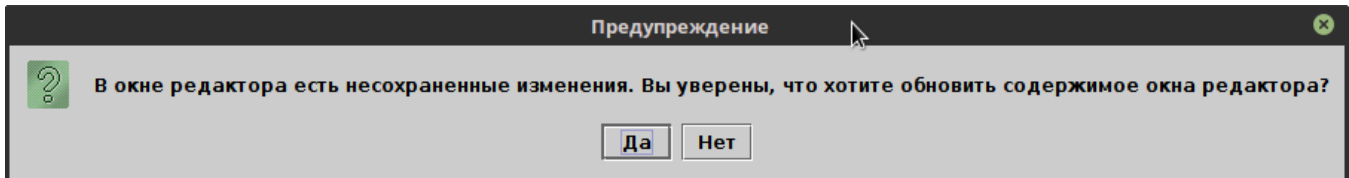
Щелчок правой кнопки мыши на заголовке вкладки откроет контекстное меню вкладки, в котором доступны несколько пунктов:

- **Закреть** - закрыть текущую вкладку
- **Закреть остальные** - закрыть все открытые вкладки кроме текущей
- **Закреть все** - закрыть все открытые в редакторе вкладки.

Как только в окне редактора вы начинаете изменять текст, данная вкладка помечается как измененная (знак "*" в заголовке вкладки рядом с названием файла).

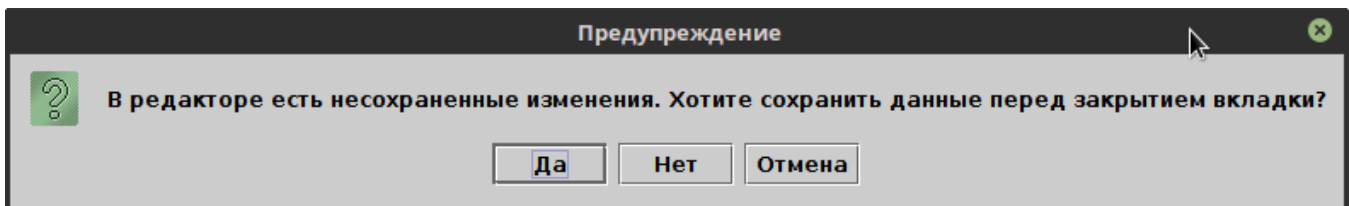


При попытке обновить содержимое вкладки, содержащей несохраненные данные, с помощью кнопки **Обновить** на панели инструментов редактор выдаст предупреждение



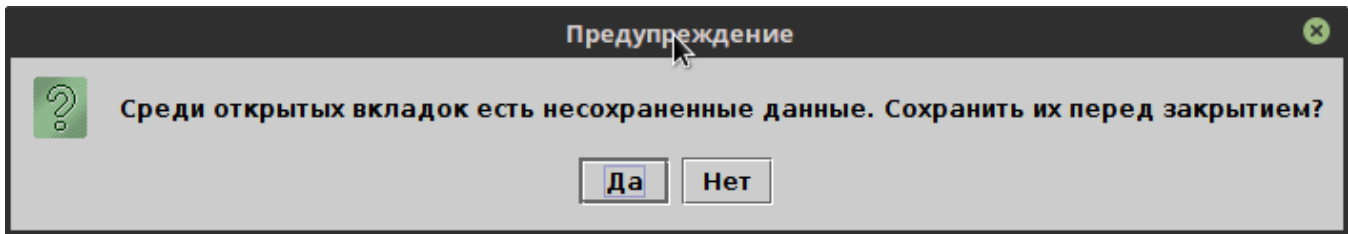
При нажатии на кнопку **Да** из редактора удалятся все ваши внесенные изменения в файл. Нажатие кнопки **Нет** оставит ваши изменения нетронутыми.

При попытке закрыть вкладку, имеющую несохраненные данные, редактор выдаст предупреждение



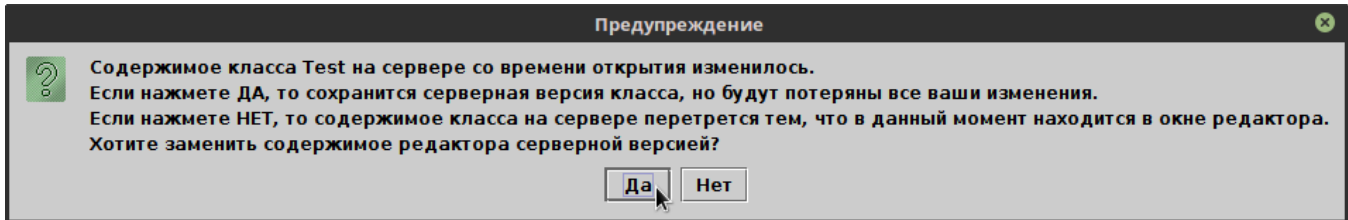
Кнопка **Да** сохранит внесенные изменения и закроет вкладку, кнопка **Нет** закроет вкладку без сохранения изменений, кнопка **Отмена** предотвратит закрытие вкладки, оставив внесенные изменения.

Если в редакторе открыто сразу несколько вкладок и среди них есть как минимум одна с несохраненными изменениями, то предупреждение будет выглядеть следующим образом:



Кнопка **Да** сохранит данные на всех вкладках, имеющих значок измененности, кнопка **Нет** просто закроет указанные вкладки.

Бывает ситуация, когда один и тот же файл редактируется одновременно из двух разных мест. Например, один пользователь правит файл в IDE, а другой в редакторе динамического кода. Если в момент сохранения в редакторе файл уже был сохранен кем то другим, то редактор выдаст предупреждение



При нажатии кнопки **Да** будут применены изменения, которые были сделаны ВНЕ редактора динамического кода. При этом изменения, внесенные в редакторе, будут потеряны. При нажатии кнопки **Нет**, наоборот, будет использовано содержимое вкладки редактора, которое перетрет изменения, внесенные извне.

Горячие комбинации клавиш редактора

В левой стороне указаны комбинация для Linux и Windows, в правой для Mac OS

F9 - запуск

Ctrl + - - переместить выделенный блок текста вверх - (+)

Ctrl + _ - переместить выделенный блок текста вниз - (+)

Ctrl + L - перейти на заданную строку - (+ L)

Ctrl + F - перевести фокус на панель поиска - (+ F)

Shift + C - закомментировать/раскомментировать строку (-и) - (+ /)

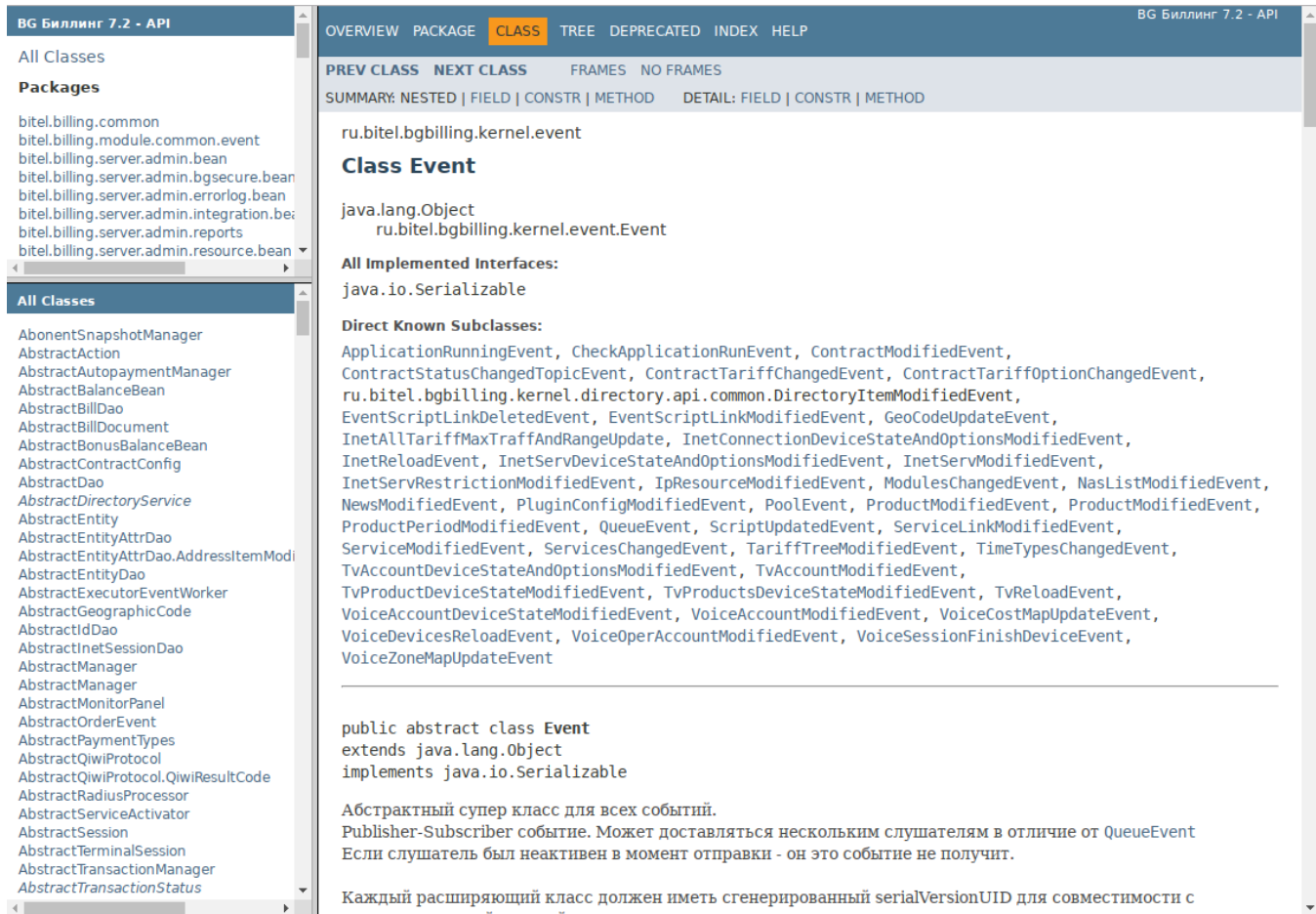
Скрипты поведения

Скрипты поведения предоставляют возможность пользователю произвольным образом обрабатывать события договоров биллинга. События делятся на два типа: синхронные (запросы) и асинхронные (сообщения).

При обработке асинхронного события (сообщения) программа не ждёт ответа обработки события, продолжая работать дальше. Само событие передается через ActiveMQ сервер данных центральному интерпретатору, работающему в процессе сервера биллинга.

Синхронное событие (запрос) обрабатывается в процессе программы, породившем его и результат обработки используется в дальнейшей работе программы. Примером запроса является событие **Запрос учётного периода** в модуле Inet. Обработка синхронного запроса, порожденного сервером биллинга так же производится центральным интерпретатором и не отличается по возможностям от асинхронного.

Все классы-события расширяют базовый класс **ru.bitel.bgbilling.kernel.event.Event**.



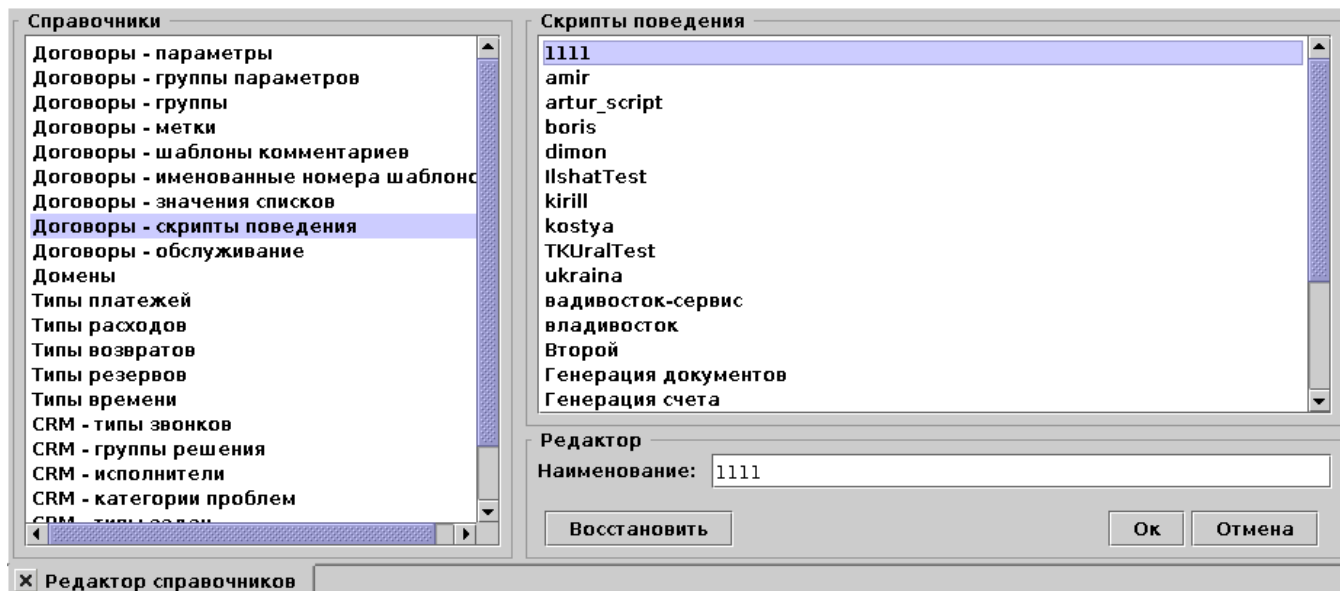
The screenshot shows the JavaDoc API documentation for the `ru.bitel.bgbilling.kernel.event.Event` class. The interface includes a navigation menu on the left with 'All Classes' and 'Packages' sections. The main content area shows the class hierarchy, implemented interfaces, and subclasses. The class `Event` is defined as an abstract class extending `java.lang.Object` and implementing `java.io.Serializable`. The description states: 'Абстрактный супер класс для всех событий. Publisher-Subscriber событие. Может доставляться нескольким слушателям в отличие от `QueueEvent`. Если слушатель был неактивен в момент отправки - он это событие не получит.' A note at the bottom indicates that extending classes should have a `serialVersionUID` for compatibility.

Для получения имени класса события воспользуйтесь горячей клавишей **Ctrl + i**, располагаясь на списке событий в редакторе привязки событий к скрипту VGBS, либо Java-классу (см. далее). В JavaDoc по классу события представлена полная информация по событию.

Для модификации данных в БД можно использовать как напрямую Java SQL API так и предлагаемые с BGBilling API, описанное в разделе документации **API документация для разработки скриптов**.

Создание скрипта поведения

По умолчанию существует скрипт поведения "Общий" - его функции применяются ко всем договорам, независимо от того, закрыты они или нет. В **Справочники=>Другие** Вы можете завести дополнительные скрипты поведения, которые можно будет привязать к договору или указать в шаблоне договора:

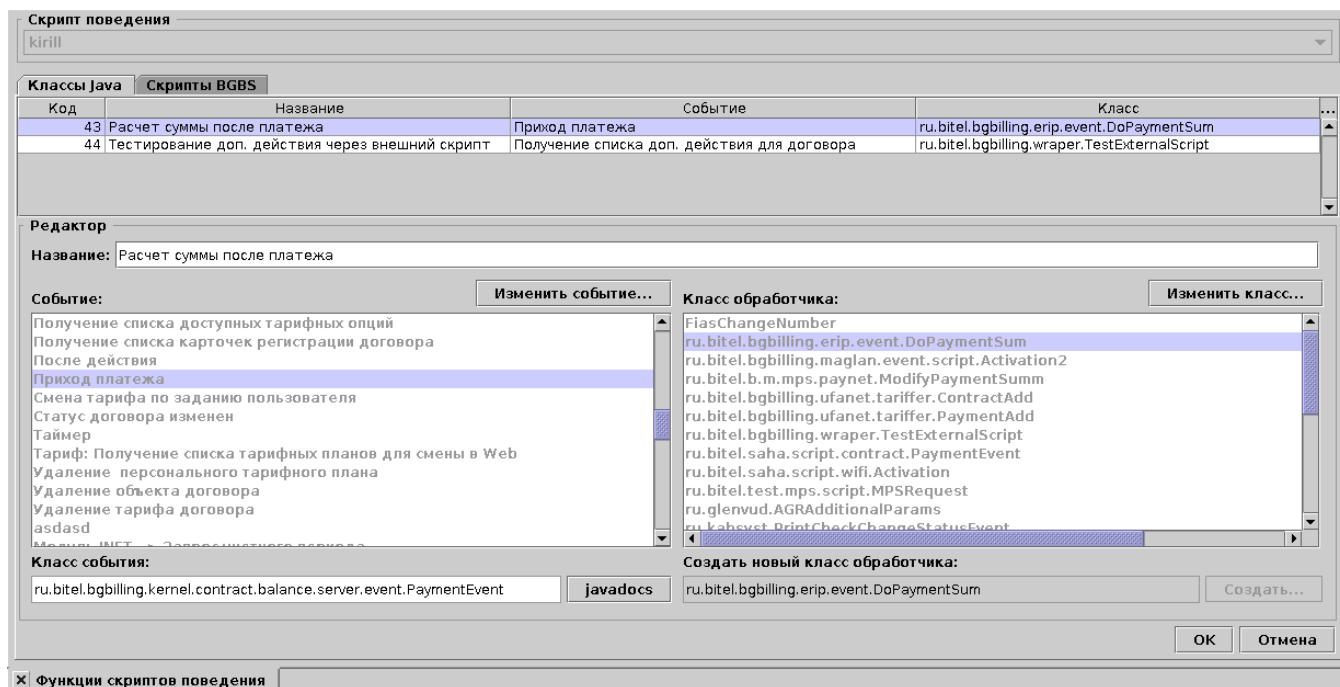


В меню **Сервис=>Автоматизация=>Договоры - скрипты поведения** производится привязка к скрипту поведения его функций. Оно содержит две вкладки: **Скрипты VBScript** и **Классы Java**.

Привязка динамически загружаемых Java-классов к скриптам поведения

Вкладка **Классы Java** содержит интерфейс управления привязкой динамических классов к скриптам поведения. Добавление привязки динамически загружаемого класса Java к скрипту поведения в качестве реакции на определённое событие в целом аналогично такому же действию при добавлении функции скрипта поведения на BGBS. Однако в качестве прямого редактирования кода предлагается выбрать лишь один из динамических классов, реализующих интерфейс `ru.bitel.bgbilling.kernel.script.server.dev.EventScript`, можно использовать наследование от класса `ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase`.

В базовом классе реализованы методы **print** и **error**, позволяющие выводить отладочную информацию и сообщения об ошибках простым способом, обеспечивая их попадание в логи выполнения скриптов.

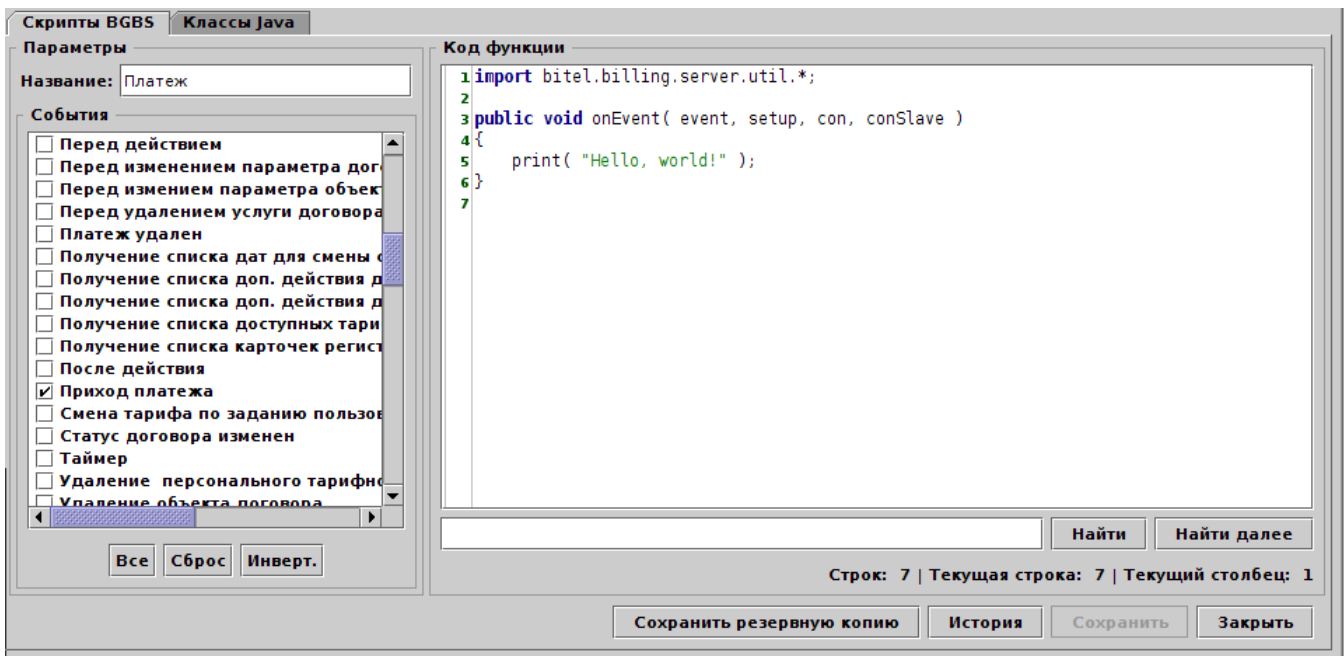
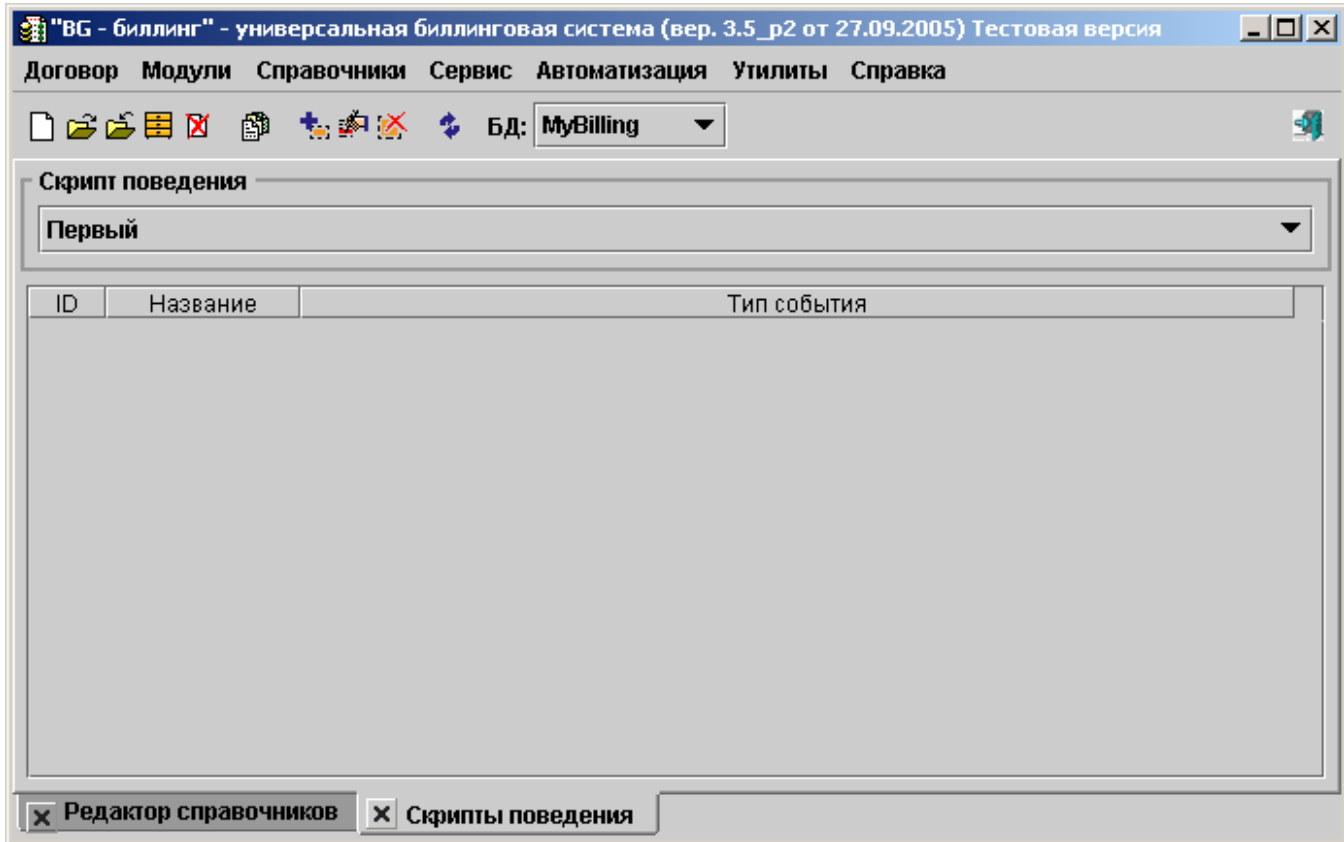


Элемент управления для выбора динамического класса универсален для всех подсистем биллинга, использующих такой вариант расширения функциональности. Он содержит список динамических классов, реализующих необходимый интерфейс, элемент создания нового класса этого интерфейса, а также по двойному клику можно перейти в редактор выбранного класса. По клику на событие в окне под списком выводится класс реализующий его и по кнопке **javadocs** можно перейти на страницу с документацией по этому классу.

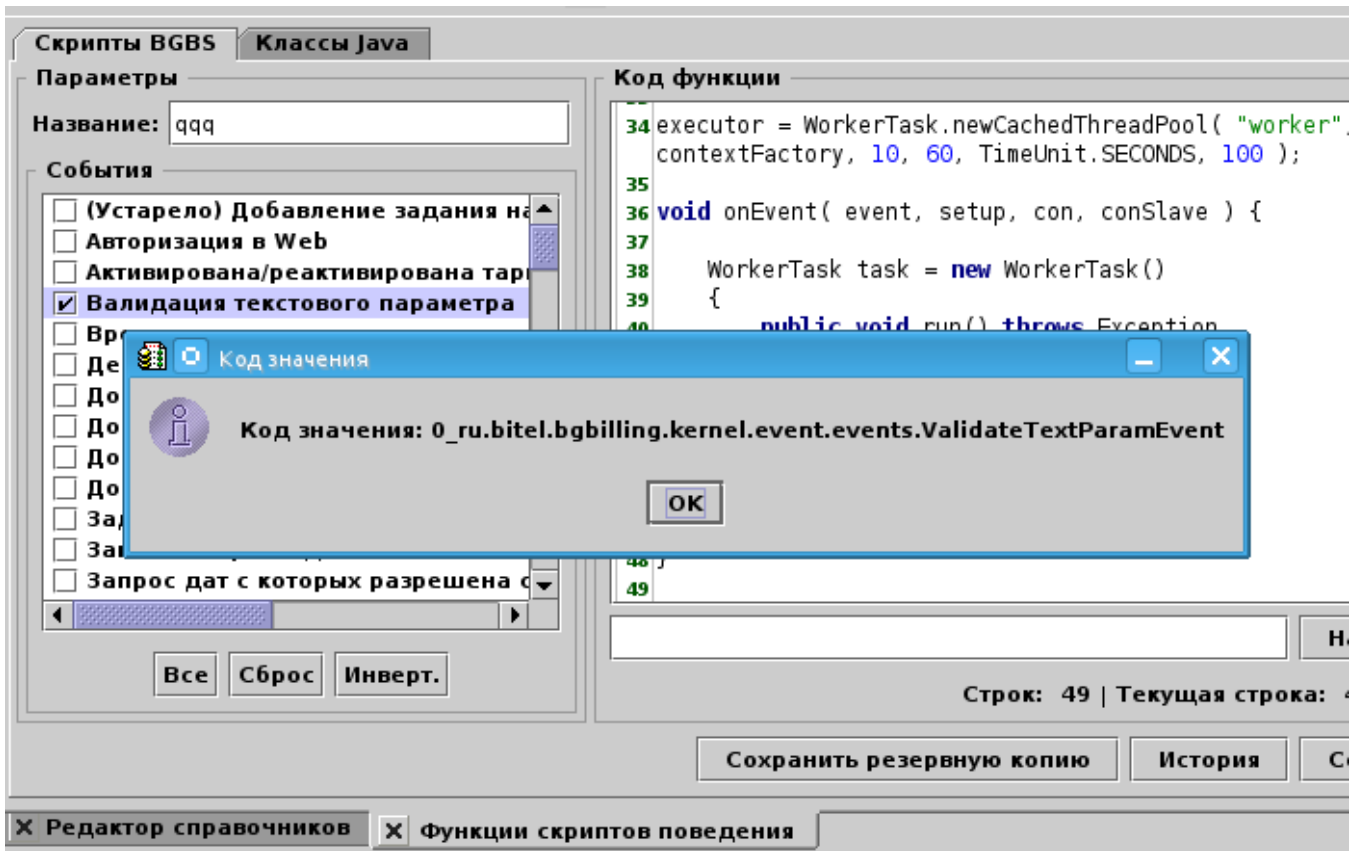
Написание функций скрипта поведения на языке BGS

Первая вкладка содержит возможность управления функциями скрипта поведения, написанными на BGS.

Данная подсистема оставлена для совместимости с ранними версиями биллинга. Предпочтительно использовать динамический Java код.



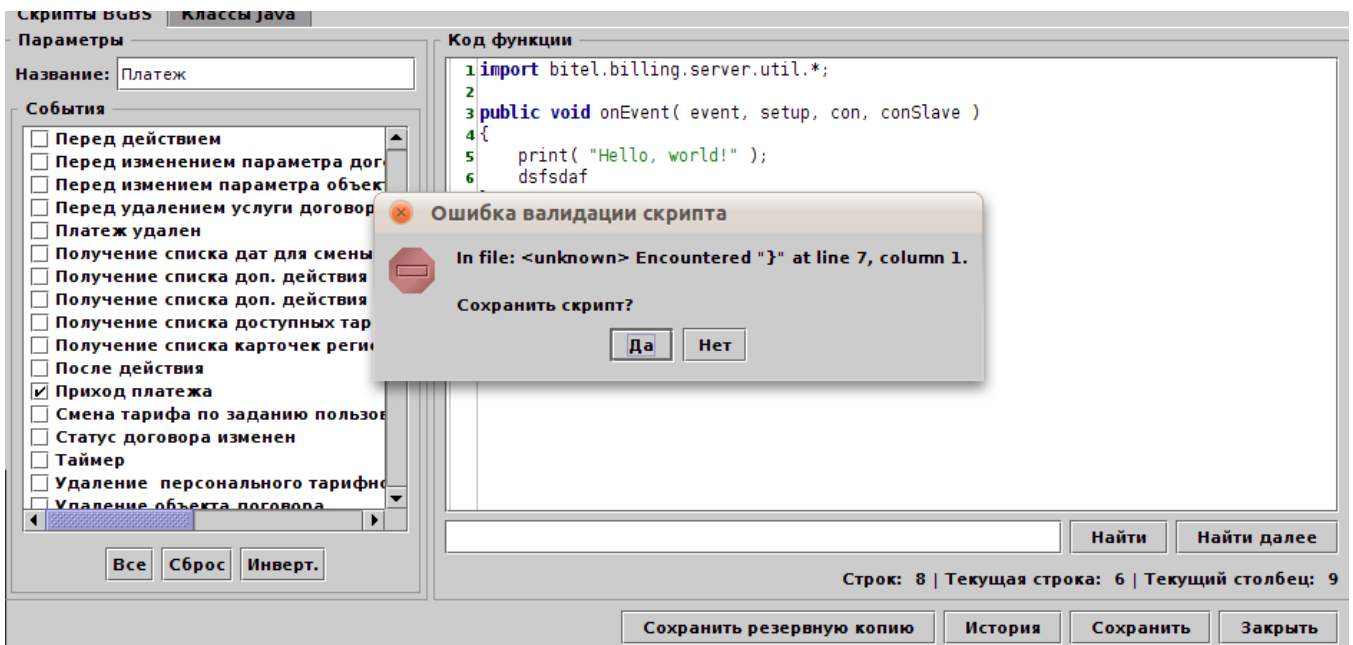
Получение имени класса события по **Ctrl + i**.



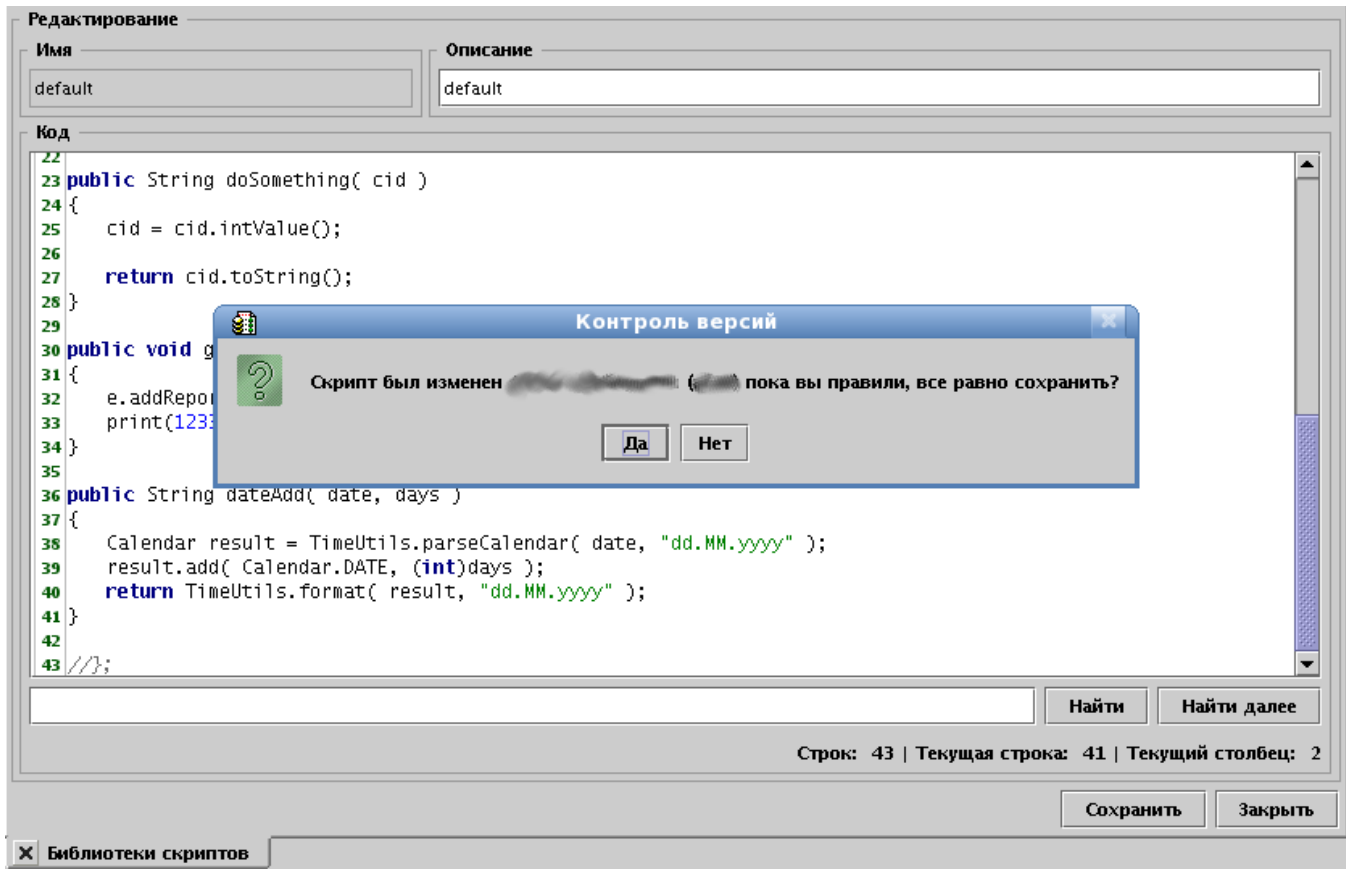
Редактор скриптов обладает подсветкой синтаксиса, индикацией строки и позиции и следующими горячими клавишами:

- **Ctrl+X** - вырезать
- **Ctrl+C** - копировать
- **Ctrl+V** - вставить
- **Ctrl+Z** - отменить
- **Ctrl+R** - повторить
- **Ctrl+L** - переход к строке

При редактировании скриптов нет необходимости перезагрузки запускающих их приложений, достаточно поправить и сохранить нужный скрипт. Перед сохранением скрипта производится его синтаксический анализ с выявлением ошибок, если они есть.



Если во время редактирования скрипт был изменен другим пользователем, то при сохранении выводится сообщение об этом с указанием имени и логина пользователя, внесшего изменения.



Программирование в BeanShell в целом идентично Java, но есть некоторые исключения:

- `print("test")` - вывод строки (будет видна в логе обработки) вместо `System.out.println("")`.
- `error("error")` - вывод ошибки вместо `System.err.println("")`.

При обработке события в скрипт передаются следующие переменные:

- `con` - объект типа `java.sql.Connection` - соединение с базой биллинга;
- `conSlave` - объект типа `java.sql.Connection` - соединение с Slave базой биллинга, либо Master, если ее нет;
- `setup` - объект класса `ru.bitel.bgbilling.server.util.DefaultServerSetup` - конфигурация сервера биллинга;
- `event` - объект, расширяющий `bitel.billing.server.script.bean.event.Event` - содержит класс-описание события.

Тело скрипта может выглядеть, например, следующим образом. Приведен пример обработки абстрактного события. Это оптимальная по производительности схема скрипта, когда главная функция `onEvent` интерпретируется один раз и далее запускается многократно.

```

import bitel.billing.server.contract.bean.*;
import bitel.billing.server.util.*;
import java.sql.*;
import java.util.*;
import bitel.billing.server.contract.bean.*;

includeBGBS( "bgbs://ru.bitel.bgbilling.kernel.script.common.bean.ScriptLibrary/default" );

public void onEvent( event, setup, con, conSlave )
{
    if( event.getActionId() != 3333 )
    {
        return;
    }

    gets(event);

    event.addReport(doSomething("vvv"));
}

```

Вот эта же функция, переписанная по-старому (работает медленнее в 8-10 раз):

```

import bitel.billing.server.contract.bean.*;
import bitel.billing.server.util.*;
import java.sql.*;
import java.util.*;
import bitel.billing.server.contract.bean.*;

includeBGBS( "bgbs://ru.bitel.bgbilling.kernel.script.common.bean.ScriptLibrary/default" );

if( event.getActionId() != 3333 )
{
    return;
}

gets(event);

event.addReport(doSomething("vvv"));

```

Обратите внимание на инструкцию включения библиотеки:

```

includeBGBS( "bgbs://ru.bitel.bgbilling.kernel.script.common.bean.ScriptLibrary/default" );

```

Сами библиотеки скриптов определяются в меню **Сервис=>Автоматизация=>Библиотеки скриптов**. Для каждой библиотеки должно быть определено уникальное имя. Библиотека представляет из себя функции, которые становятся доступными после включения в скрипте инструкции **includeBGBS**. На снимке ниже представлен код библиотеки **default**, использованной в скрипте выше.

Редактирование

Имя: default Описание: default

Код

```
1 import java.util.Calendar;
2 import bitel.billing.common.TimeUtils;
3
4 //includeBGBS( "bgbs://bitel.billing.server.script.bean.Function/56" );
5
6 //common = new Object(){
7
8 public String doSomething()
9 {
10 // return "tada!";
11 rs = getContext().con.createStatement().executeQuery("SELECT 4+5");
12 rs.next();
13 return rs.getString(1);
14 }
15
16 public String doSomething( String a )
17 {
18 print( "111" + a);
19 //throw new Exception();
20 }
```

Найти Найти далее

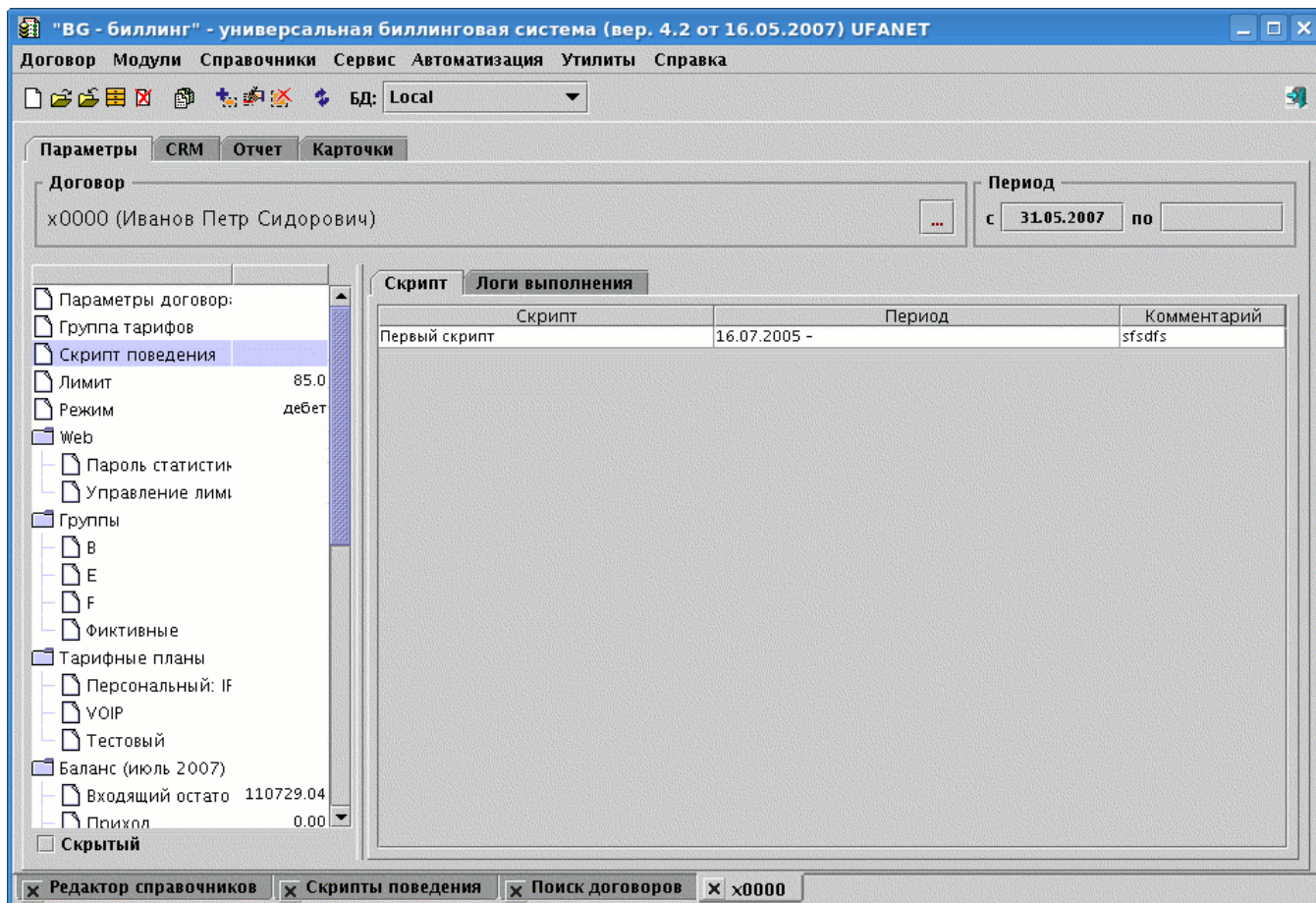
Строк: 43 | Текущая строка: 14 | Текущий столбец: 2

Сохранить резервную копию История Сохранить Закрыть

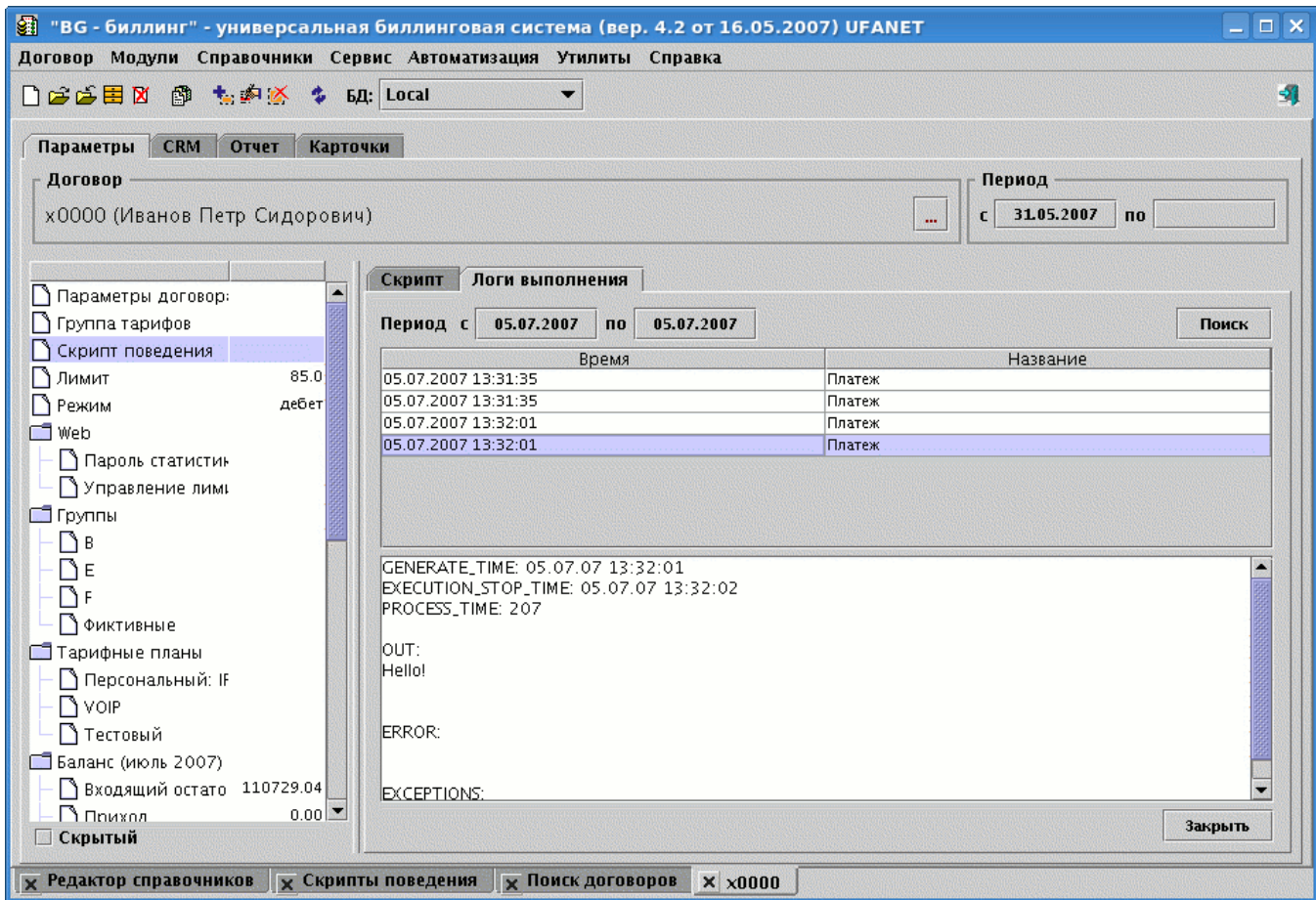
Библиотеку можно включить из другой библиотеки, но данный подход не рекомендуется, т.к. по неосторожности можно создать закливание включения библиотек.

Привязка скриптов поведения к договору

Скрипт поведения "Общий" уже привязан ко всем договорам. Привязка дополнительных же скриптов поведения к договору производится выбором узла дерева карточки договора **Скрипт**. При добавлении нескольких скриптов в договор, событие будет передаваться в каждый из них.



На вкладке **Логи выполнения** отображаются логи выполнения события скриптами. Например, при описанном выше скрипте после прихода платежа в логах будет следующая запись:



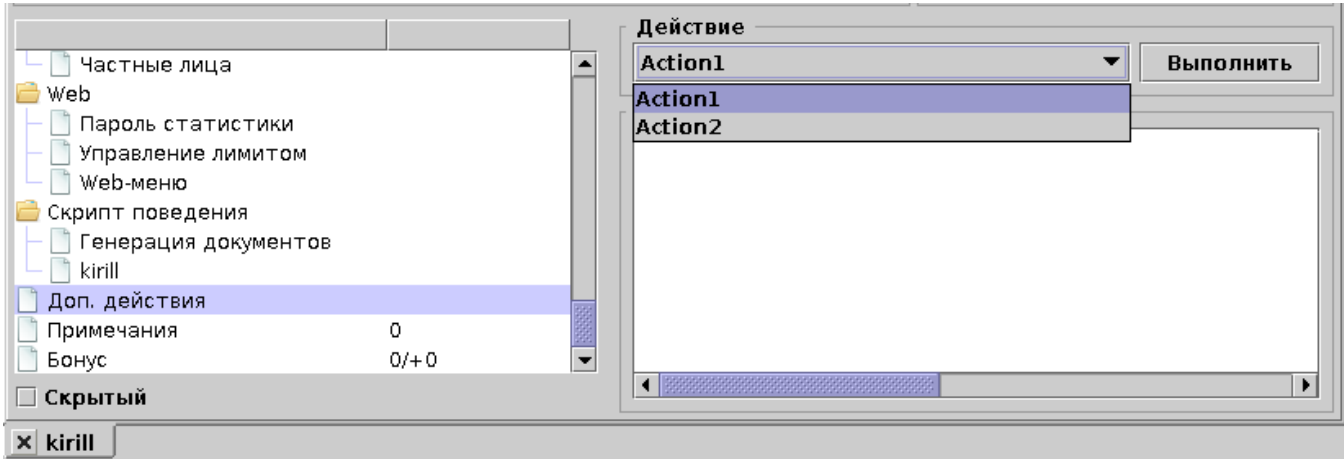
Сообщения об ошибках выполнения скриптов отправляются на E-Mail посредством системы оповещения биллинга.

Обработка событий внешними обработчиками

При необходимости возможно обрабатывать события внешними обработчиками написанными на любых языках (Php, Python, Perl, Bash и т.д.)

Внимание данный механизм обработки событий более ресурсоемкий и медленее, чем обработчик написанный на Java.

Для вызова внешнего обработчика необходимо обернуть его вызов в небольшой Java класс. К примеру для события `GetAdditionalActionListEvent` выводящего список дополнительных действий в договоре



скрипт будет выглядеть следующим образом

```
package ru.bitel.bgbilling.wrapper;

import java.util.Arrays;

import ru.bitel.bgbilling.kernel.event.events.GetAdditionalActionListEvent;
import ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.sql.ConnectionSet;

public class TestExternalScript
    extends EventScriptBase<GetAdditionalActionListEvent>
{
    @Override
    public void onEvent( GetAdditionalActionListEvent event, Setup setup, ConnectionSet set )
        throws Exception
    {
        String result = doExternalScript( event, Arrays.asList( "/usr/bin/echo", "R:addAction\t1\tAction1\nR:
addAction\t2\tAction2" ) );
    }
}
```

Здесь `/usr/bin/echo` внешний обработчик, `"R:addAction\t1\tAction1\nR:addAction\t2\tAction2"` параметр вызова обработчика (их может быть несколько)

Во входной поток обработчика передается событие `event` сериализованное в `json` формат. Выходной поток возвращается в переменную `result`.

Дополнительно выходной поток обработчика обрабатывается на поиск строк определенного формата

R:<МЕТОД КЛАССА EVENT><TAB><ARG_1>[...<TAB><ARG_N>]<ENTER>

например для примера выше, в выходном потоке обработчика будут строки

```
R:addAction\t1\tAction1
R:addAction\t2\tAction2
```

обработка которых приведет к вызову метода `event.addAction(1, "Action1")` и `event.addAction(2, "Action2")`

Скрипты поведения глобальных событий

Скрипты поведения глобальных событий отличаются от скриптов поведения тем, что связанные с ними события не прикреплены к договорам. Для вызова редактора выполните **Сервис=>Автоматизация=>Функции глобальных событий**. Редактор аналогичен редактору скриптов поведения. Помимо этого имеется вкладка **Логи** для просмотра истории запуска событий.

Глобальные скрипты

Глобальные скрипты предоставляют пользователю биллинга возможность выполнения произвольного кода для взаимодействия с API BGBilling без привязки к событийной модели. Глобальные скрипты могут быть использованы как для однократного выполнения некоторого сложного взаимодействия с данными (что сложно было бы реализовать возможностями SQL-редактора), так и для периодического (об этом см. ниже).

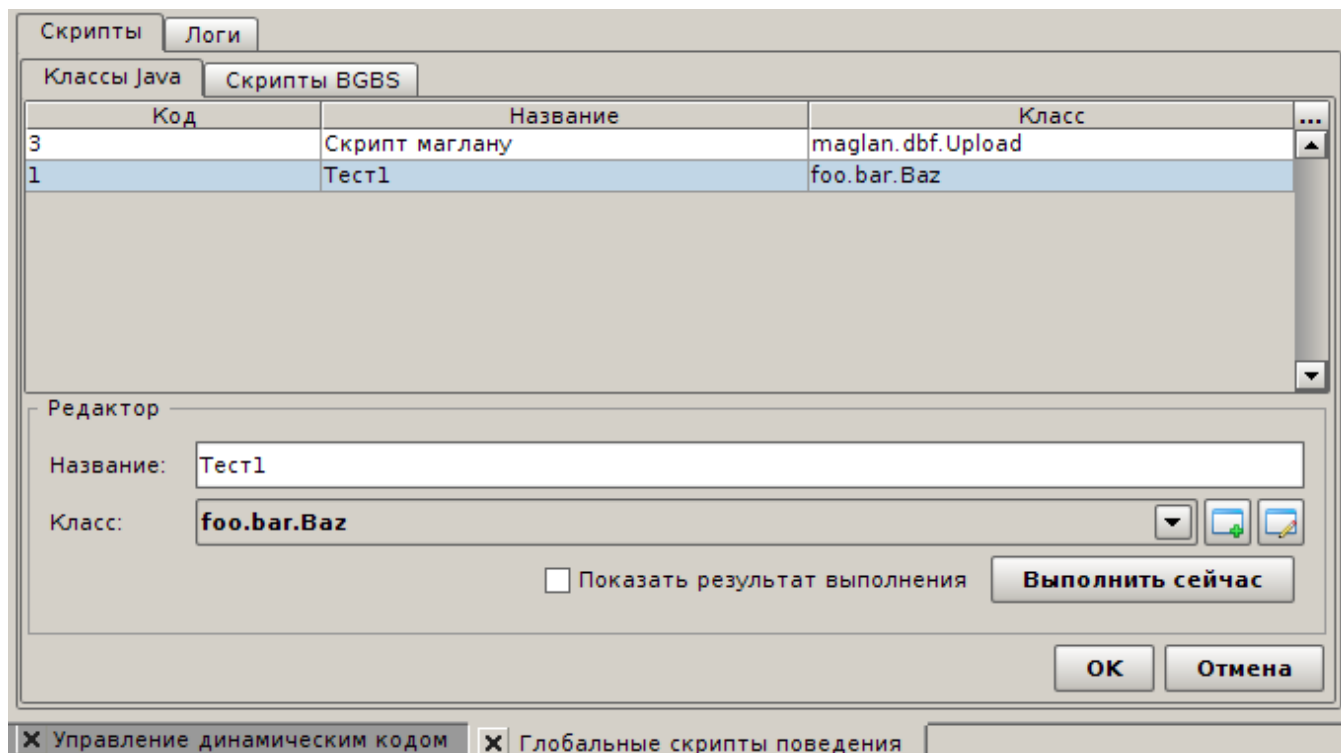
Для доступа к глобальным скриптам необходимо открыть меню **Сервис=>Автоматизация=>Глобальные скрипты поведения**. В открывшейся вкладке представлено две вкладки - **Скрипты** и **Логи**. Во вкладке **Скрипты** представлено две вкладки: **Классы Java** и **Скрипты BGBS**.

Вкладка **Логи** отображает информацию о выполнении скрипта. Абсолютно идентична логам событийных скриптов в карточке договора (см. [здесь](#)).

Глобальные скрипты с использованием динамических классов Java

Во вкладке **Классы Java** расположен список всех доступных глобальных скриптов, привязанных к динамически загружаемым классам. Для создания, удаления и редактирования скриптов необходимо воспользоваться соответствующими кнопками панели инструментов. Двойной щелчок по уже созданному скрипту также открывает редактор.

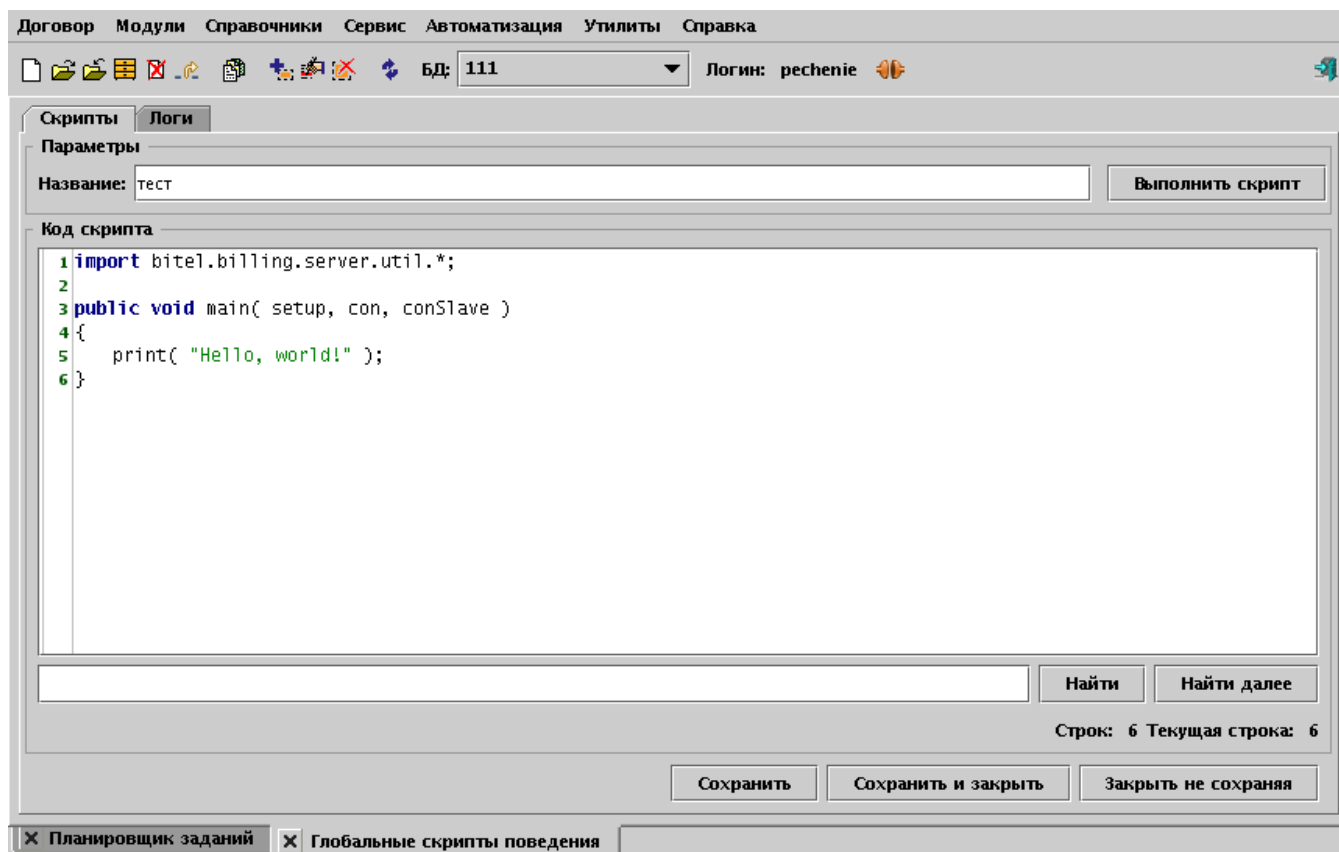
Как и в случае со скриптами поведения, вместо редактора кода скрипта представлена панель привязки динамического класса к глобальному скрипту.



Класс, привязанный к глобальному скрипту, должен расширять базовый класс `ru.bitel.bgbilling.kernel.script.server.dev.GlobalScriptBase`. Для однократного выполнения глобального скрипта необходимо нажать на кнопку **Выполнить скрипт**. Выполнение скрипта происходит асинхронно. Результат выполнения скрипта логируется (см. вкладку **Логи**).

Глобальные скрипты на языке BGBS

Во вкладке **Скрипты BGBS** отображен список всех доступных скриптов. Для создания, удаления и редактирования скриптов необходимо воспользоваться соответствующими кнопками панели инструментов. Двойной щелчок по уже созданному скрипту открывает редактор.



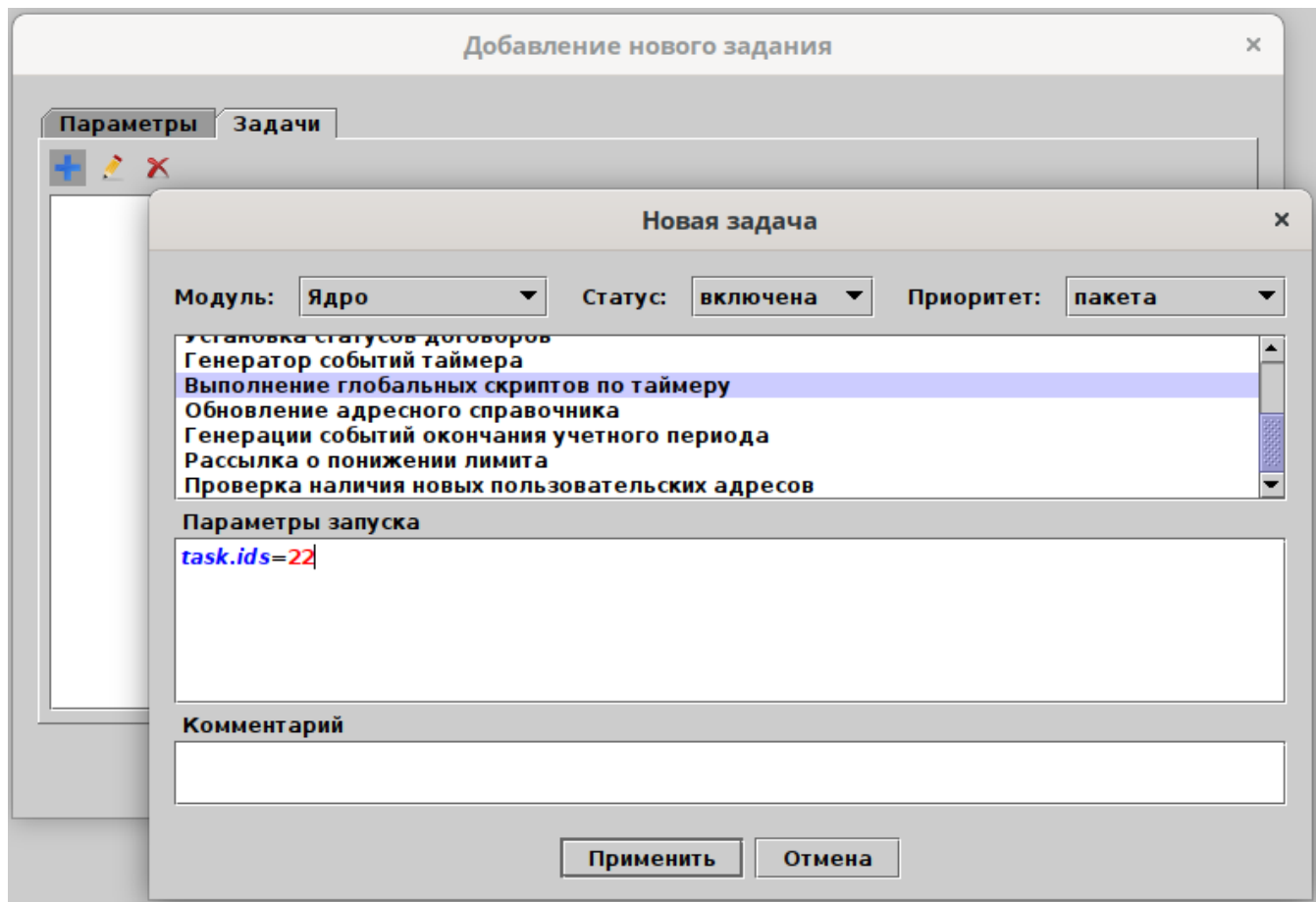
При создании/редактировании скрипта должны быть соблюдены следующие условия. Во-первых, обязательно должно быть задано имя скрипта. Во-вторых, тело скрипта обязательно должно содержать функцию **main**, как представлено на скриншоте выше. В функцию **main** передаются следующие аргументы:

- **con** - объект типа `java.sql.Connection` - соединение с базой биллинга;
- **conSlave** - объект типа `java.sql.Connection` - соединение с Slave базой биллинга либо Master, если ее нет;
- **setup** - объект класса `ru.bitel.bgbilling.server.util.DefaultServerSetup` - конфигурация сервера биллинга.

Для однократного выполнения глобального скрипта необходимо (предварительно его сохранив) нажать на кнопку **Выполнить скрипт**. Выполнение скрипта асинхронно. Результат выполнения скрипта логируется (см. вкладку Логи).

Периодическое выполнение глобальных скриптов

Для периодического выполнения глобальных скриптов необходимо в **Сервис=>Администрирование=>Планировщик заданий** создать задачу планировщика "Выполнение глобальных скриптов по таймеру".



В качестве параметра для задачи необходимо указать **task.ids=X,Y,Z**, где **X,Y,Z** - коды скриптов, перечисленные через запятую.

По причине появления глобальных скриптов двух типов (BGBS и Java) был добавлен опциональный флаг **task.type**, обозначающий желаемый тип выполняемого глобального скрипта. При **task.type=0** в качестве скриптов выполняются соответствующие указанным кодам скрипты на BGBS, в случае же **task.type=1** - на Java. С 9.2210+ по умолчанию **task.type=1**

Возможны случаи, когда внутри выполняемых глобальных скриптов используется commit для соединений с БД, либо commit вызывается внутри api биллинга, при установленной опции autocommit=true у соединения с БД. Для предотвращения ошибок по умолчанию для данной задачи установлен autocommit=false в соединении с БД. Чтобы изменить поведение по умолчанию используйте опцию задачи **autocommit=**, которая может принимать 2 значения: **true** или **false**.

Резервные копии

Для скриптов поведения и библиотек есть возможность сохранять резервные копии. Для этого в всех редакторах доступна кнопка "Сохранить резервную копию". Так же доступна возможность просмотреть список резервных копий с помощью кнопки "История".

ID	Время	Пользователь	Название функции	Комментарий
35	21.05.2012 19:04:05	stark	common	1234
36	22.05.2012 13:08:02	stark	common	yyy

Двойным кликом можно просмотреть конкретную версию скрипта.

Скрипты предобработки RADIUS запросов

Ещё одним применением скриптов являются скрипты предобработки RADIUS-запросов, позволяющие модифицировать запрос, приводя его к требованиям биллинга, определяя при этом вид услуги запроса и т.п. Скрипт предобработки вводится в редакторе NASов модулей DialUp и VoiceIp. В скрипт передаются следующие переменные:

- **con** - объект типа java.sql.Connection, соединение с базой биллинга
- **conSlave** - объект типа java.sql.Connection, соединение с Slave базой биллинга, либо основной, если Slave нет
- **setup** - объект класса bitel.billing.server.radius.RadiusSetup, конфигурация сервера RADIUS
- **request** - объект класса bitel.billing.server.radius.RadiusPacket, RADIUS запрос
- **response** - объект класса bitel.billing.server.radius.RadiusPacket, RADIUS ответ

Образец скрипта. Как и в случае со скриптами предобработки можно не объявлять функцию, сразу размещая логику скрипта. Но скорость работы скрипта при этом будет меньше.

```
import ru.bitel.bgbilling.kernel.network.radius.*;

void processRequest( request, response, setup, con, conSlave )
{
    //
}
```

При модификации скрипта предобработки требуется перезапуск RADIUS-сервера. Вывод работы скриптов и ошибки доступны в логе **script.log** RADIUS-сервера. Скрипты предобработки RADIUS-запроса не привязаны к договору, а привязаны к NASy, т.к. выполняются сразу после получения запроса, когда еще не известно, с каким клиентом будет соотнесен запрос.

Интеграция с внешними системами

Из [структуры](#) биллинга можно определить два способа взаимодействия с ним извне:

- обращение к базе данных - допустимо для выборок данных либо для быстрого внесения изменений, не предполагающих реакции сервера;
- обращение к серверу биллинга - выполнение действий аналогично оператору из АРМ биллинга посредством обращений сторонней системы.

При выполнении обращений к серверу биллинга помимо записи данных в БД могут производиться различные дополнительные обработки самим сервером. Например: разблокировка модулей по платежу.

К серверу биллинга возможно обращаться из сторонних приложений посредством HTTP запросов. Пример запросов можно изучить вызывая различные действия в клиентском приложении, запущенном в через `client_debug.bat` (.sh). Запросы и ответы распечатываются в log файл. Биллинг поддерживает несколько вариантов протоколов обращений для различных действий, их можно определить по внешнему виду запросов.

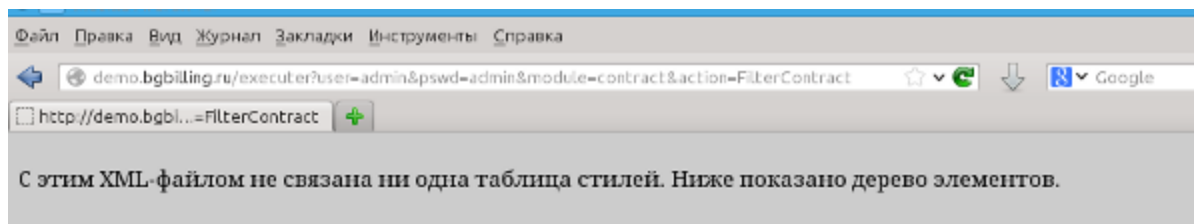
HTTP Action

Более ранний протокол. Подразумевает передачу параметров в HTTP запросе с получением XML ответа. Примерный вид запроса и ответа в логе клиента:

```
http://billing:8081/executer?module=npay&action=ServiceSetList&mid=6&BGBillingSecret=ONM13dhRxs8VDD9wFCzpP0rU&
[ length = 159 ] xml = <?xml version="1.0" encoding="UTF-8"?><data secret="B802E452DF43140146D0EC4219C847D1"
status="ok"><list><item id="0" title=" " /></list></data>
```

Дополнительно необходимо добавить в запрос параметры **user** и **pswd** с логином и паролем пользователя биллинга, от лица которого отправляется запрос. Рекомендуется добавить к запросу параметр **authToSession=0**, это предотвратит создание HTTP сессий, уменьшит потребление памяти на сервере.

В целях отладки запросы можно отправлять прямо в браузере.



```
- <data status="ok">
- <contracts allRecord="4" pageCount="1" pageIndex="1" pageSize="25" recordCount="4">
  <item id="2" title="FW00001-13 [ Иванов Иван Иванович ]"/>
  <item id="4" title="FW00002-13 [ Иванов Иван Иванович ]"/>
  <item id="1" title="test [ Тестовый договор ]"/>
  <item id="3" title="UW00001-13 [ ООО "Рога и копыта" ]"/>
</contracts>
</data>
```

Web-сервис

Более новые вызовы реализованы как Web-сервисы, пример обращения к ним вы можете посмотреть в [Wiki](#). Используется Basic авторизация.

Примерный вид запроса и ответа:

```

http://billing:8081/executer/ru.bitel.bgbilling.kernel.module/ServiceService?wsdl -> {http://service.common.module.kernel.bgbilling.bitel.ru/}ServiceService:serviceList
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><ns5:serviceList xmlns:ns5="http://service.common.module.kernel.bgbilling.bitel.ru/" xmlns:common="http://common.bitel.ru" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><moduleId>9</moduleId></ns5:serviceList></S:Body></S:Envelope>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Header/><S:Body><ns5:serviceListResponse xmlns:ns5="http://service.common.module.kernel.bgbilling.bitel.ru/" xmlns:common="http://common.bitel.ru" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xml="http://www.w3.org/XML/1998/namespace"><return id="15" moduleId="9" title=" Board.ufanet.ru ()" unit="0" using="true"/><return id="7" moduleId="9" title=" Board.ufanet.ru &quot;&quot; ()" unit="0" using="true"/><return id="12" moduleId="9" title=" Board.ufanet.ru &quot;; , &quot; ()" unit="0" using="true"/><return id="6" moduleId="9" title=" Board.ufanet.ru &quot;&quot; ()" unit="0" using="true"/>

```

Web-сервис через JSON RPC

Для более простого обращения к Web сервисам возможно использование протокола JSON RPC.

Запрос выглядит следующим образом, пример:

```

http://127.0.0.1:8080/bgbilling/executer/json/ru.bitel.bgbilling.kernel.contract.api/ContractService

{ "method" : "contractList",
  "user" : { "user" : "shamil", "pswd" : "xxxx" },
  "params" : {
    "title" : "0",
    "fc" : -1,
    "groupMask" : 0,
    "subContracts" : false,
    "closed" : true,
    "hidden" : false,
    "page" : { "pageIndex" : 2, "pageSize" : 2 }
  } }

```

URL запроса определяет Web-сервис, JSON запрос передаётся в теле запроса, кодировка UTF-8. Обязательные параметры:

- **method** - вызываемая функция сервиса;
- **user** - объект с логином и паролем пользователя, от лица которого выполняется действие;
- **params** - объект со всем остальными параметрами сервиса.

Обязательно указание значений всех параметров с примитивными типами: int, boolean.

Примерный вид корректного ответа:

```

{ "status": "ok", "message": "",
  "data":
  {
    "page": { "pageSize": 2, "pageIndex": 2, "pageCount": 49, "recordCount": 97, "pageFirstRecordNumber": 2 },
    "return":
    [ { "id": 353023, "title": "0022010", "groups": 0, "password": "bg2rFZ2PEX", "dateFrom": "2010-01-02", "dateTo": null, "balanceMode": 0, "paramGroupId": 14, "personType": 0, "comment": "", "hidden": false, "superCid": 0, "dependSubList": "", "status": 0, "statusTimeChange": "2010-01-13", "titlePatternId": 0, "balanceSubMode": 0, "sub": false, "independSub": false, "balanceLimit": 0.00, "super": false, "dependSub": false },
      { "id": 353209, "title": "06-10-10/-/0", "groups": 0, "password": "9351220759", "dateFrom": "2010-10-06", "dateTo": null, "balanceMode": 1, "paramGroupId": 14, "personType": 0, "comment": "", "hidden": false, "superCid": 0, "dependSubList": "", "status": 0, "statusTimeChange": "2010-10-06", "titlePatternId": 0, "balanceSubMode": 0, "sub": false, "independSub": false, "balanceLimit": 0.00, "super": false, "dependSub": false } ] } }

```

В **status** возвращается **ok** в случае корректного ответа, либо **error** - ошибка. При этом в **message** выводится текст ошибки.

Там выглядит сообщение об ошибке:

```

{ "status": "error", "message": " .", "data": {} }

```

В случае корректного ответа в **data** возвращаются результаты выполнения. Возвращаемый методом параметр под ключом **return**.

Также возможен возврат результатов из параметров сервиса посредством объекта типа **javax.xml.ws.Holder**.

Для работы с сервисом необходимо найти его класс в JavaDoc, например: [ContractService](#). Далее определить имена и типы передаваемых параметров, переходя к JavaDoc описаниям других классов, если понадобится.

внимание



JavaDoc является первичным при определении имени параметров. Например Если в [InetServ](#) есть метод **getPassword()**, значит нужно использовать **password** как имя атрибута в JSON. Если посмотреть лог работы клиента, там передается другой атрибут **passw**, в данном случае javadoc является первичным.

Некоторую сложность представляет передача параметров, в роли которых могут выступать классы-потомки указанного в JavaDoc класса. Для примера рассмотрим параметр **entityFilter** указанного ранее сервиса, метод **contractList**. В качестве параметров передаются наследники объекта [FilterEntityAttr](#). В JavaDoc данного класса находим:

```
@JsonTypeInfo(use=NAME,
              include=PROPERTY,
              property="type")
@JsonSubTypes(value={
    @JsonSubTypes.Type(name="Address", value=FilterEntityAttrAddress.class),
    @JsonSubTypes.Type(name="Date", value=FilterEntityAttrDate.class),
    @JsonSubTypes.Type(name="House", value=FilterEntityAttrHouse.class),
    @JsonSubTypes.Type(name="Int", value=FilterEntityAttrInt.class),
    @JsonSubTypes.Type(name="List", value=FilterEntityAttrList.class),
    @JsonSubTypes.Type(name="Text", value=FilterEntityAttrText.class),
    @JsonSubTypes.Type(name="Email", value=FilterEntityAttrEmail.class)})
```

Данные строки означают, что объекты маркируются дополнительным полем **type**, в зависимости от которого определяется класс переданного объекта. Вот так, например, выглядит запрос фильтрации договоров по текстовому параметру:

```
{ "method" : "contractList",
  "user" : { "user" : "shamil", "pswd" : "xxxx" },
  "params" :
  {
    "fc" : -1,
    "groupMask" : 0,
    "subContracts" : false,
    "closed" : true,
    "hidden" : false,
    "page" : { "pageIndex" : 1, "pageSize" : 2 },
    "entityFilter" : [ { "type" : "Text", "entitySpecAttrIds" : [2], "value" : "1" } ]
  } }
```

Для тестирования запросов удобно использовать плагин браузера позволяющий отправку запросов (типа `HttpRequester` для FF), либо штатную возможность браузера.

HttpRequester

Request

URL: http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService

POST Save GET POST PUT Raw request Paste Request

Content to Send: Headers Parameters

Content Type: text/json

Content Options: Base64 Parameter Body

Content:

```
{
  "method": "contractList",
  "user": {
    "user": "shami",
    "pswd": "xxxx"
  },
  "params": {
    "to": -1,
    "groupMask": 0,
    "subContracts": false,
    "closed": true,
    "hidden": false,
    "page": {
      "pageIndex": 1,
      "pageSize": 2,
      "entityFilter": {
        "type": "Text",
        "entitySpecAttrId": 2,
        "value": "1"
      }
    }
  }
}
```

Response

POST on http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService

Status: 200 OK [View raw transaction](#)

```
{
  "status": "ok",
  "message": "",
  "data": {
    "page": {
      "pageSize": 2,
      "pageIndex": 1,
      "pageCount": 1,
      "recordCount": 1,
      "pageFirstRecordNumber": 0
    },
    "return": [
      {
        "id": 3266,
        "bel": "000001",
        "groups": 33554452,
        "password": "545454",
        "dateFrom": "2008-01-31",
        "dateTo": null,
        "balanceMode": 1,
        "paramDropList": 3,
        "personType": 1,
        "comment": "Jyuuuââ verreepet naa onââ yakuu",
        "hidden": false,
        "superCid": 0,
        "dependSubList": "",
        "status": 3,
        "statusTimeChange": "2010-03-14",
        "BelPatternId": 0,
        "balanceSubMode": 3,
        "balanceLimit": 0.01,
        "sub": false,
        "independSub": false,
        "dependSub": false,
        "super": false
      }
    ]
  }
}
```

Headers:

Server: Apache-Coyote/1.1

Content-Length: 574

Date: Fri, 29 Aug 2014 07:56:05 GMT

History

Request	Response	Date	ContentLength	ElapsedTime	
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 29 2014 - 1:55:34 PM	574	2057 ms	Clear history
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 29 2014 - 12:28:49 PM	79	2533 ms	Copy to clipboard
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 29 2014 - 12:28:37 PM	929	6338 ms	Delete request
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 29 2014 - 12:28:37 PM	79	2482 ms	Edit raw request
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 29 2014 - 12:27:25 PM	79	19677 ms	Save request
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 27 2014 - 12:54:33 PM	9788	83 ms	Load request
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 27 2014 - 12:54:22 PM	910	124 ms	
POST http://127.0.0.1:8080/bgilling/execute/sonru/btel/bgilling/kernel/contract/api/ContractService	200 OK	Aug 27 2014 - 12:54:14 PM	79	87 ms	

Web API

Для интеграции BGBilling с внешними сайтами или сторонним Личным кабинетом, возможно использование механизм Web API (доступен с версии 6.2)

Для активации данного API в конфигурации ядра необходимо прописать

```
web.api.enable=true
```

Запрос к Web API не проверяется на аутентификацию и права пользователя, поэтому при его включении нужно изолировать доступ к данному URL!

После активации, будет доступны два режима получения данных из BGBilling:

- в XML формате ([http://host\[:port\]/bgbilling/xmlWebApi/<operation>](http://host[:port]/bgbilling/xmlWebApi/<operation>))
- в JSON формате ([http://host\[:port\]/bgbilling/jsonWebApi/<operation>](http://host[:port]/bgbilling/jsonWebApi/<operation>))

где **<operation>** это:

для Ядра

- login
- contractParameters
- contractTarifPlans
- contractBalance
- contractPayments
- lastContractPayments
- contractTariffOptions
- contractModules
- newsList

для других модулей - в разработке, дополнительная информация по запросу на e-mail (<http://bgbilling.ru>) или на форуме

Операция login

пример запроса - [/xmlWebApi/login?login=kirill&password=*****\[&midAuth=0\]](#) или [/jsonWebApi/login?login=kirill&password=*****\[&midAuth=0\]](#)

ответы:

```
<response><status>Ok</status><contractId>386010</contractId><contractTitle>kirill</contractTitle></response>
```

или

```
{ "contractTitle": "kirill", "contractId": 386010, "status": "Ok" }
```

Операция contractParameters

пример запроса - [/xmlWebApi/contractParameters?contractId=386010](#) или [/jsonWebApi/contractParameters?contractId=386010](#)

ответы:

```
<response><status>Ok</status><contractParameters><contractParameter><id>4</id><typeId>1</typeId><title>... </title><read>true</read><write>true</write><value> </value></contractParameter><contractParameter><id>7</id><typeId>1</typeId><title> </title><read>false</read><write>true</write><value/></contractParameter><contractParameter><id>1</id><typeId>1</typeId><title> </title><read>true</read><write>true</write><value></value></contractParameter><contractParameter><id>20</id><typeId>3</typeId><title>e-mail</title><read>true</read><write>true</write><value>&lt;info@company.ru&gt;</value></contractParameter><contractParameter><id>40</id><typeId>9</typeId><title> (</title><read>true</read><write>true</write><value>7 (347) 292-48-23</value></contractParameter></contractParameters></response>
```

или

```
{ "contractParameters": [ { "read": "true", "typeId": "1", "id": "4", "title": "... ", "write": "true", "value": "  " },
{ "read": "false", "typeId": "1", "id": "7", "title": " ", "write": "true" }, { "read": "true", "typeId": "1", "id": "1", "title": "
", "write": "true", "value": "" }, { "read": "true", "typeId": "3", "id": "20", "title": "e-mail", "write": "true", "value": "
&lt;info@company.ru&gt;" }, { "read": "true", "typeId": "9", "id": "40", "title": " ( )", "write": "true", "value": "7 (347)
292-48-23" } ], "status": "Ok" }
```

Операция contractTarifPlans

пример запроса - `/xmlWebApi/contractTarifPlans?contractId=386010[&entityMid=1][&entityId=1]` или `/jsonWebApi/contractTarifPlans?contractId=386010[&entityMid=1][&entityId=1]`

Ответы:

```
<response><status>Ok</status><contractTarifPlans><contractTarifPlan><id>273691</id><tariffPlanId>52<
/tariffPlanId><title> (web)</title><dateFrom>06.01.2015</dateFrom><dateTo/><period>06.01.2015-...<
/period><comment> Web</comment><pos>2</pos></contractTarifPlan></contractTarifPlans></response>
```

или

```
{
  "contractTarifPlans": [
    {
      "period": "06.01.2015-...",
      "pos": "2",
      "dateTo": "",
      "comment": " Web",
      "id": "273691",
      "tariffPlanId": "52",
      "title": " (web)",
      "dateFrom": "06.01.2015"
    }
  ],
  "status": "Ok"
}
```

Операция contractBalance

пример запроса - `/xmlWebApi/contractBalance?contractId=386010` или `/jsonWebApi/contractBalance?contractId=386010`

Поддерживаются параметры в конфиге

```
web.api.contract.balance.format.pattern=###,###,###,###.00
client.gui.currency.title.medium=.
```

Ответы:

```
<response><status>Ok</status><sum>99 878 732,10</sum><currencyTitleMedium>.</currencyTitleMedium></response>
```

или

```
{
  "currencyTitleMedium": ".",
  "sum": "99 878 732,10",
  "status": "Ok"
}
```

Операция contractPayments

пример запроса - `/xmlWebApi/contractPayments?contractId=386010[&members=1][&date1=2016-01-01][&date2=2019-01-01]` или `/jsonWebApi/contractPayments?contractId=386010[&members=1][&date1=2016-01-01][&date2=2019-01-01]`

где **members** - если == 1, то только данный договор, если = 2 то вернет еще и его зависимые субдоговора, если = 3 то только его зависимые субдоговора.

date1 и **date2** - задают период за который выводятся платежи

```
<response><status>Ok</status><contractPayments><contractPayment><id>251748</id><typeId>54</typeId><typeTitle>NetPay</typeTitle><date>01.09.2016</date><comment>    NetPay [ #7]</comment><sum>107,00</sum></contractPayment></contractPayments><totalSum>107,00</totalSum></response>
```

Ответы:

```
{
  "totalSum": "107,00",
  "contractPayments": [
    {
      "date": "01.09.2016",
      "typeId": "54",
      "comment": "    NetPay [ #7]",
      "sum": "107,00",
      "id": "251748",
      "typeTitle": "NetPay"
    }
  ],
  "status": "Ok"
}
```

Операция lastContractPayments

пример запроса - `/xmlWebApi/lastContractPayments?contractId=386010[&members=1][&lastPayments=3]` или `/jsonWebApi/lastContractPayments?contractId=386010[&members=1][&lastPayments=3]`

где **members** - если == 1, то только данный договор, если = 2 то вернет еще и его зависимые субдоговора, если = 3 то только его зависимые субдоговора.

lastPayments - количество платежей

Ответы:

```
<response><status>Ok</status><contractPayments><contractPayment><id>251748</id><typeId>54</typeId><typeTitle>NetPay</typeTitle><date>01.09.2016</date><comment>    NetPay [ #7]</comment><sum>107,00</sum></contractPayment><contractPayment><id>251735</id><typeId>50</typeId><typeTitle>Moneta</typeTitle><date>21.07.2016</date><comment>    Moneta.ru [ #16]</comment><sum>116,00</sum></contractPayment></contractPayments><totalSum>223,00</totalSum></response>
```

или


```

{
  "totalSum": "223,00",
  "contractPayments": [
    {
      "date": "01.09.2016",
      "typeId": "54",
      "comment": "NetPay [ #7]",
      "sum": "107,00",
      "id": "251748",
      "typeTitle": "NetPay"
    },
    {
      "date": "21.07.2016",
      "typeId": "50",
      "comment": "Moneta.ru [ #16]",
      "sum": "116,00",
      "id": "251735",
      "typeTitle": "Moneta"
    }
  ],
  "status": "Ok"
}

```

Операция contractTariffOptions

пример запроса - `/xmlWebApi/contractParameters?contractId=386010` или `/jsonWebApi/contractParameters?contractId=386010`

Форматирование суммы можно задать через параметр конфига

```
web.api.contract.tariff.option.sum.format.pattern=###,###,###,###.00
```

Ответы:

```

<response><status>Ok</status><contractTariffOptions><contractTariffOption><id>81</id><timeFrom>24.07.2015<
/timeFrom><timeTo/><title>
</title><optionId>5</optionId><sum>,00</sum></contractTariffOption></contractTariffOptions></response>

```

или

```

{
  "contractTariffOptions": [
    {
      "timeFrom": "24.07.2015",
      "timeTo": "",
      "optionId": "5",
      "sum": ",00",
      "id": "81",
      "title": " "
    }
  ],
  "status": "Ok"
}

```

Операция contractModules

пример запроса - `/xmlWebApi/contractModules?contractId=386010` или `/jsonWebApi/contractModules?contractId=386010`

Ответы:

```
<response><status>Ok</status><contractModules><contractModule><id>145</id><title>BVCom</title></contractModule><contractModule><id>43</id><title>CerberCrypt2</title></contractModule><contractModule><id>179</id><title>INET</title></contractModule><contractModule><id>213</id><title>Mobi.</title></contractModule><contractModule><id>239</id><title>Moneta</title></contractModule><contractModule><id>70</id><title>MPS</title></contractModule></contractModules></response>
```

или

```
{
  "contractModules": [
    {
      "id": "145",
      "title": "BVCom"
    },
    {
      "id": "43",
      "title": "CerberCrypt2"
    },
    {
      "id": "179",
      "title": "INET"
    },
    {
      "id": "213",
      "title": "Mobi."
    },
    {
      "id": "70",
      "title": "MPS"
    }
  ],
  "status": "Ok"
}
```

Операция newsList

пример запроса - </xmlWebApi/newsList?contractId=386010> или </jsonWebApi/newsList?contractId=386010>

Ответы:

```
<response><status>Ok</status><newsList><item><id>12</id><date>01.12.2010</date><title>!!! !!! 5 !!!</title><body><h1> H1</h1></body></item><item><id>11</id><date>13.07.2010</date><title>123</title><body>12312312312</body></item><item><id>9</id><date>14.12.2009</date><title>vzxlck;glsdfh</title><body>LKNBLNCVXKC;BHJXC;VBJH;XKFGHVBGHDFG</body></item><item><id>10</id><date>07.12.2009</date><title> </title><body> </body></item></newsList></response>
```

или

```
{
  "newsList": [
    {
      "date": "01.12.2010",
      "id": "12",
      "title": "!!! !!! 5 !!!",
      "body": "<h1> H1</h1>\n<h2> H2</h2>"
    },
    {
      "date": "13.07.2010",
      "id": "11",
      "title": "123",
      "body": "12312312312"
    },
    {
      "date": "14.12.2009",
      "id": "9",
      "title": "vzxlck;glsdfh",
      "body": "LKNBLNCVXKC;BHJXC;VBJH;XKFGHBGHDFG"
    },
    {
      "date": "07.12.2009",
      "id": "10",
      "title": " ",
      "body": " "
    }
  ],
  "status": "Ok"
}
```

Динамические Web-Сервисы

Помимо вызова стандартных Web-сервисов можно создавать еще свои Web-сервисы в [динамическом коде](#). Динамические Web-сервисы работают точно так же как обычные, описанные в основной главе [Интеграция с внешними системами](#). Для аутентификации используется HTTP Basic-аутентификация, вызовы логируются в журнале запросов и их можно ограничивать правами пользователя. Пример обращения к ним вы можете посмотреть в [WiKi](#).

Для создания своего Web-сервиса в конфигурации сервера необходимо прописать параметр:

```
dynservice:ru.company.CustomService=ru.company.impl.CustomServiceImpl
```

Где:

- **ru.company.CustomService** - некий интерфейс .
- **ru.company.impl.CustomServiceImpl** - его реализация.

Пример интерфейса:

```
package ru.company;

import javax.ws.WebService;

@WebService
public interface CustomService
{
    public String doit();
}
```

И его реализация:

```
package ru.company.impl;

import ru.company.CustomService;
import javax.ws.WebService;

@WebService(endpointInterface = "ru.company.CustomService")
public class CustomServiceImpl implements CustomService
{
    @Override
    public String doit()
    {
        return "Hello World";
    }
}
```

В данном примере wsdl-файл доступен по ссылке: <http://host:port/bgbilling/executer/ru.company/CustomService?wsdl>

Общие рекомендации по разработке скриптов

При разработке скриптов будет полезен [наш API](#), ссылка на который есть на нашем сайте. Примеры использования скриптов доступны в базе знаний [Wiki](#).

При сохранении/удалении/просмотре сущностей более приоритетным способом является использование Web-сервисов. В этом случае гарантируется целостность данных, все необходимые события будут сгенерированы, действия будут произведены.

Пример добавления платежа с помощью Web-сервиса:

```
//
ServerContext context = ServerContext.get();
// Web-
PaymentService paymentService = context.getService( PaymentService.class, 0 );
//с
Payment payment = new Payment();
payment.setSum( sum );
...
// ,
paymentService.updatePayment( payment, null);
```

PaymentService - Web-сервис для работы с платежами. Второй параметр метода **getService** - это код модуля, в данном случае 0, так как это Web-сервис ядра. Web-сервис для любого другого модуля можно получить аналогично, указав верный код модуля.

Альтернативным способом сохранения является прямое использование менеджеров работы с базой данных. Например, для платежа это PaymentDAO. Но данный подход не рекомендуется, так как в этом случае всю дополнительную логику (события для смены статуса и открытия доступа при приходе платежа, обновление баланса договора и т.п.) придется делать самостоятельно.

Модули абонплат и разовых услуг

Модуль NPay (абонплаты)

Назначение модуля

Модуль предназначен для начисления наработки за периодические услуги (абонплаты).

Настройка модуля

Установите модуль на сервер. Затем создайте экземпляр модуля, назвав его произвольным образом (например, **Мои абонплаты**).

Определите в редакторе модулей и услуг услуги, связанные с этим модулем. Каждая абонентская плата - отдельная услуга. Для начала можете определить несколько тестовых услуг, скорректировав список впоследствии.

с версии 8.2101 вместо настройки режимов начисления абонплат в разных статусах с помощью параметров `contract.status.suspend.codes=` и `service.no.suspend.<status_list>=<service_codes>` рекомендуется использовать матрицу статусов (см. скриншот ниже). По умолчанию используется матрица, если хотите использовать параметры вместо матрицы, укажите параметр `npay.service.status.mode=param`

ID	Услуга	Активен
120	qqq (Abonplata)	не начислять
198	Wut?	не начислять
79	Абонплата	не начислять
75	Абонплата 60 в месяц	не начислять
76	Абонплата 70 дневная	не начислять
230	Абонплата за пользование сервисом	начислять
103	Абонплата за телефоны	не начислять
114	Абонплата	
78	Абонплата	
77	Абонплата	
99	Абонплата	
115	Аренда ка	
305	Базовый п	
197	Дополнит	
117	Дополнит	
116	Предоста	
316	тест 2	
128	Тестирова	
129	Тестирова	
147	Тестирую	

Статус	Начислять	Не начислять	Текущее значение
Активен	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
В отключении	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Отключен	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Закрыт	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Приостановлен	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
В подключении	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Создайте конфигурацию модуля:

Конфигурация

```
# ,
contract.status.suspend.codes=3,4
#
#
# 0 - , 1 - , 2 - ,
recalculate.on.service.change=1
# E-Mail , -
#auto.recalculate.email=
#
#auto.recalculate.email.service.set=
#
max.periodic.errors=30
#
#-----
#
#
#closed.date.disabled.ActionMovePay=1
#
#closed.date.disabled.ActionRecalculatePay=1
#
#closed.date.disabled.ActionServiceObjectDelete=1
#
#closed.date.disabled.ActionServiceObjectUpdate=1
#
#closed.date.disabled.ActionWrapPay=1
#-----
```

Для исключения приостановки некоторых абонентских плат по статусу договора укажите в конфигурации:

```
# ,
service.no.suspend.<status_list>=<service_codes>
```

где:

- **<status_list>** - код статусов договора через запятую;
- **<service_codes>** - коды услуг договора через запятую.

Например, не приостанавливать в статусе 3 абонплаты 9 и 36, по всем другим начисления не будет. В статусе 4 не приостанавливать только абонплату 9.

```
# ,
service.no.suspend.3=9,39
service.no.suspend.4=9
```

Если в статусе 3 и 4 не нужно приостанавливать только абонплату 9, можно было бы записать так:

```
# ,
service.no.suspend.3,4=9
```

На один статус договора не должно быть несколько записей.

Первоначально модуль не требует дополнительной настройки и способен производить начисления простых видов абонентских плат. Дополнительные настройки устанавливаются в конфигурации модуля и указаны в контексте решаемой задачи далее.

Привязка абонплат к клиентам

Подключение модуля к договору осуществляется выбором узла **Модули** дерева договора с последующим выбором модуля **Нрау** из списка установленных модулей в системе. Выбрав экземпляр модуля в дереве договора, можно просмотреть список всех подключенных к договору абонплат с возможностью их фильтрации на выбранную дату.

The screenshot shows the BGBillingClient application window. The title bar reads "BGBillingClient". The menu bar includes "Договор", "Модули", "Плагины", "Справочники", "Сервис", "Утилиты", and "Справка". The status bar shows "БД: local", "Логин: artur", and "12 Февраль 15:43". The main window has tabs for "Параметры", "Отчет", "Карточки", "Бонусы", "HelpDesk", "Рассылки", "CRM", and "Документы". The "Договор" section shows "artur_test [Гареев Артур Маратович]" with ID "352984" and a period from "24.11.2009" to "...". A checkbox "выводить только элементы, актуальные на дату" is checked with "12.02.2014". A tree view on the left shows "Модули" expanded to "MPS". A table displays the following data:

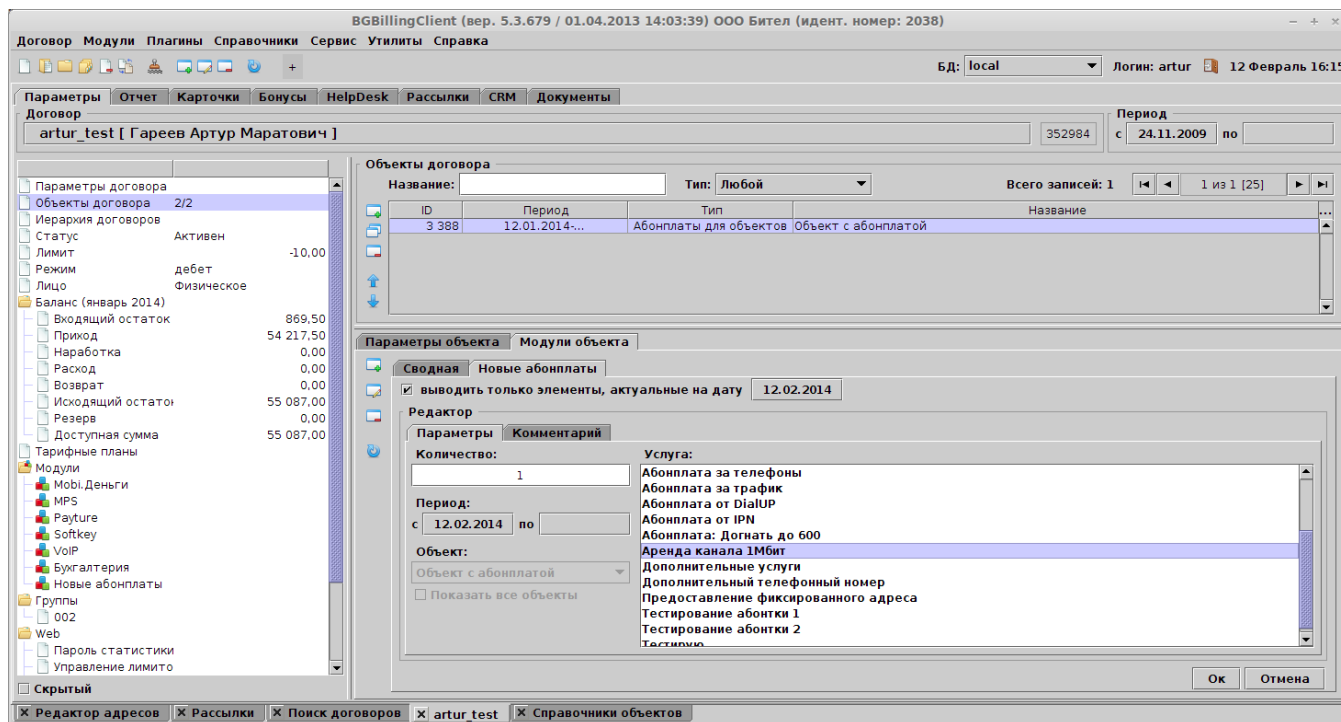
Услуга	Кол-во	Период
Абонплата за трафик	1	12.01.2014-11.02.2014

Для подключения абонплаты к договору необходимо нажать кнопку **Новый элемент** на стандартной панели инструментов. При этом откроется окно редактора абонплат, в котором необходимо выбрать необходимую абонплату из списка услуг модуля, а также период действия данной абонплаты.

The screenshot shows the BGBillingClient application window with the "Редактор" window open. The "Редактор" window has tabs for "Параметры" and "Комментарий". The "Параметры" tab is active, showing "Количество: 1" and "Период: с 12.02.2014 по ...". The "Услуга:" list includes "Абонплата 60 в месяц", "Абонплата 70 дневная", "Абонплата за пользование сервисом" (highlighted), "Абонплата за телефоны", "Абонплата за трафик", "Абонплата от DialUP", "Абонплата от IPN", "Абонплата: Догнать до 600", "Аренда канала 1Мбит", "Дополнительные услуги", "Дополнительный телефонный номер", and "Предоставление фиксированного адреса". Buttons "Ок" and "Отмена" are at the bottom right of the editor.

При этом сразу можно указывать объекты, к которым относятся абонплаты. Если установлена галочка **Показать все объекты**, то в списке объектов будут выведены все объекты, в противном случае - объекты с активным периодом. **Количество абонплаты** указывает множитель цены - во сколько раз больше вычисленной по тарифу стоимости будет взята абонплата.

Для договоров с несколькими точками подключения, телефонами и т.п., для каждого из которых устанавливаются свои абонплаты, возможна установка абонлат непосредственно для **объектов**.



Возможно перенесение абонплаты со всей наработкой с текущего договора на другой открытый. Для этого выбирается строка с услугой, правой кнопкой мыши вызывается меню и активируется пункт **Перенести на другой договор**. После переноса услуги система активирует переназначение по выбранным договорам за месяцы, когда была активна данная услуга. Задачу переначисления выполняет планировщик биллинга, он должен быть запущен.

Алгоритм начисления, примеры тарифов

Модуль определяет потребление договором абонплаты по наличию услуги из модуля в договоре. Алгоритм работы следующий. Начисление производится за определённый месяц.

1) производится выборка потребителей услуг данного модуля, получая набор сочетаний: **договоры - услуга - период - количество**;

2) для каждого сочетания производится разбивка по действующим на период тарифные планы, получая тарифицируемые сочетания: **договор - услуга - тариф - период - количество**;

В договоре определены:

- Абонплата 1 - с 1 по 10 число;
- Абонплата 2 - с 9 по 31 число;
- Тариф 1 - с 2 по 31 число.

Получаем два тарифицируемых сочетания:

- Абонплата 1 - Тариф 1 - с 2 по 10 число;
- Абонплата 2 - Тариф 1 - с 9 по 31 число;

Каждое сочетание тарифицируется по тарифу.

1) При **помесячном режиме** начисления цена в тарифе запрашивается один раз, используя дату окончания периода для передачи в тарифный запрос. При стоимости **пропорционально периоду** цена определяется пропорционально отношению количество дней, когда абонплата в месяце была открыта и статус активен к общему числу дней месяца. **Количества** для помесечного режима также всегда рассчитываются на дату окончания периода. Помесячный режим всегда рассчитывается до конца расчётного месяца.

2) При **подневном режиме** передаётся запрос в тариф на каждый день, входящий в период сочетания. Для каждого дня оценивается стоимость и количества, анализируется статус договора. Снятие может быть произведено как до текущего дня, так и до конца месяца, в зависимости от настройки узла.

Исходя из этого, помесечный режим начислений оптимальнее в плане тарификации, однако цена и количество сущностей, которым пропорциональна абонплата в нем может изменяться только один раз в месяц. В этом её отличие от подневного режима с начислением до конца месяца.

3) При **погодном режиме** начисление производится только в том месяце, в который была добавлена абонплата. Стоимость абонплаты определяется на конец месяца.

4) При **авансовом режиме** начисления:

Если период абонплаты закрыт, то цена запрашивается на день начала периода и абонплата снимается **только за месяц начала** согласно стоимости дня или месяца. При указании цены за месяц дневная цена получается делением месячной на количество дней в месяце начала абонплаты. Соответственно при начислении за месяц авансовые абонплаты с закрытым периодом, открытые не в месяце начисления игнорируются.

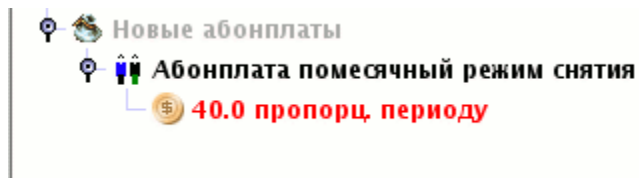
Если период абонплаты открыт, то цена запрашивается на ДАТУ_НАЧАЛА=МАКСИМУМ(День начала периода; День начала месяца). Стоимость дня умножается на количество дней от ДАТЫ_НАЧАЛА до конца месяца. При указании цены за месяц дневная цена получается делением месячной на количество дней в месяце начисления.

Рассмотрим несколько примеров. Для начала, простейшие абонплаты, не зависящие от наработок в других модулях.

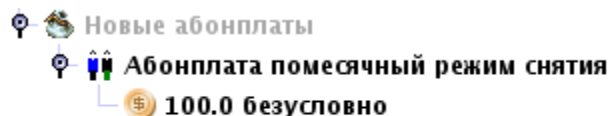
Абонплаты, не зависящие от других модулей

Пропорциональная абонплата 40 рублей за месяц.

При этом, если человек подключён в середине месяца, то за первый месяц должно сняться только 20 рублей (половина). Название услуги - **Абонплата**.



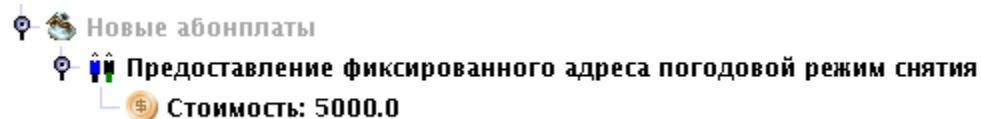
Простая абонплата - 100 рублей в месяц независимо от даты подключения услуги.



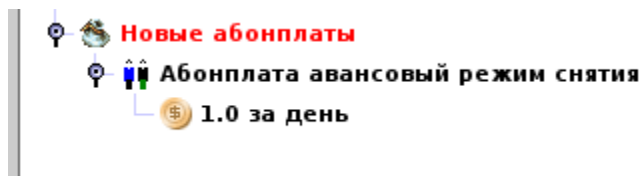
Начисление 1 го рубля за каждый день до текущей даты



Погодовой режим снятия - начисление только в месяц добавления абонплаты на договор (раз в год).



Авансовый режим снятия - начисление ежемесячно (открытый период), либо единовременно в месяц открытия (закрытый период).



Абонплаты, зависящие от наработки по объёму в других модулях

Модуль абонлат может производить начисление в зависимости от объёмов наработок в других модулях.

Зависимости можно выставить только для подневного и помесячного режима начислений.

Все используемые при тарификации объёмы должны быть описаны в конфигурации экземпляра модуля. Объём наработки описывается в конфигурации экземпляра модуля абонлат следующим образом.

```
module.amount.<id>.title=<title>
module.amount.<id>.mid=<mid>
module.amount.<id>.class=<class_name>
module.amount.<id>.sids=<sids>
```

Где:

- **<id>** - уникальный числовой идентификатор объёма, нумерация должна быть последовательной и непрерывной, число не должно меняться впоследствии, при изменении нумерации необходима правка всех тарифов, где используются данные объёмы;
- **<title>** - название объёма;
- **<mid>** - код экземпляра модуля, в котором рассчитывается объём;
- **<class_name>** - класс, рассчитывающий объём;
- **<sids>** - коды услуг через запятую.

В данный момент поддерживаются следующие классы, которые могут быть указаны в **<class_name>** для расчёта объёма услуги (байт, либо секунд):

- **ru.bitel.bgbilling.modules.npay.server.bean.amount.DialUpModuleAmount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.DialUpModuleAmount**) - количество байт или секунд в экземпляре модуля DialUp;
- **ru.bitel.bgbilling.modules.npay.server.bean.amount.IPNModuleAmount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.IPNModuleAmount**) - количество байт в экземпляре модуля IPN;
- **ru.bitel.bgbilling.modules.npay.server.bean.amount.InetModuleAmount** (до версии 8.XXXX - **ru.bitel.bgbilling.modules.inet.npay.InetModuleAmount**) - количество байт в экземпляре модуля Inet;
- **ru.bitel.bgbilling.modules.npay.server.bean.amount.PhoneModuleAmount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.PhoneModuleAmount**) - количество секунд округлённой длительности в экземпляре модуля Phone.
- **ru.bitel.bgbilling.modules.npay.server.bean.amount.VoipModuleAmount**
- **ru.bitel.bgbilling.modules.npay.server.bean.amount.VoiceModuleAmount**

Указанный в конфигурации объём можно использовать в узле тарифного плана типа **Условие по объёму услуги**. При размещении данного узла в узле типа **Дневной режим снятия** объём будет вычисляться за каждый день для использования в условиях по объёму. При размещении в узле типа **Месячный режим снятия** объём будет вычисляться за рассчитываемый период (период сочетания, см. [алгоритм](#)).

Рассмотрим пример конфигурации экземпляра модуля для начисления несколько типов абонлат и использующую при начислении значения входящих трафиков модуля DialUP, IPN и голосового трафика модулей Телефонии (Phone, VoiceIp, Voice). Если необходимо учитывать суммарную наработку по нескольким услугам, они перечисляются через запятую.

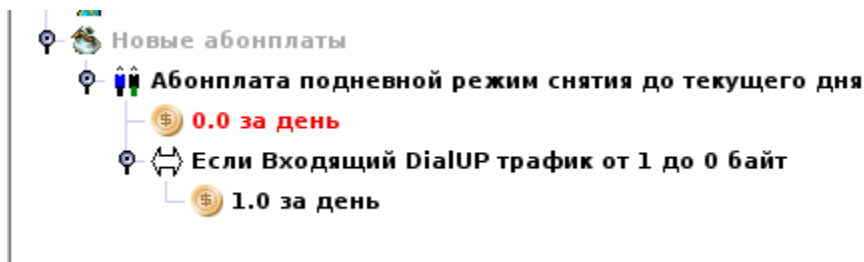
```

module.amount.1.title= DialUP
module.amount.1.mid=21
module.amount.1.class=ru.bitel.bgbilling.modules.npay.server.bean.amount.DialUpModuleAmount
module.amount.1.sids=23
#
module.amount.2.title= IPN
module.amount.2.mid=33
module.amount.2.class=ru.bitel.bgbilling.modules.npay.server.bean.amount.IPNModuleAmount
module.amount.2.sids=40
#
module.amount.3.title=
module.amount.3.mid=73
module.amount.3.class=ru.bitel.bgbilling.modules.npay.server.bean.amount.PhoneModuleAmount
module.amount.3.sids=83,84,85,86
#
module.amount.4.title= INET
module.amount.4.mid=179
module.amount.4.class=ru.bitel.bgbilling.modules.npay.server.bean.amount.InetModuleAmount
module.amount.4.sids=203
#
module.amount.5.title= Voiceip
module.amount.5.mid=79
module.amount.5.class=ru.bitel.bgbilling.modules.npay.server.bean.amount.VoipModuleAmount
module.amount.5.sids=122
#
module.amount.6.title= Voice
module.amount.6.mid=217
module.amount.6.class=ru.bitel.bgbilling.modules.npay.server.bean.amount.VoiceModuleAmount
module.amount.6.sids=150

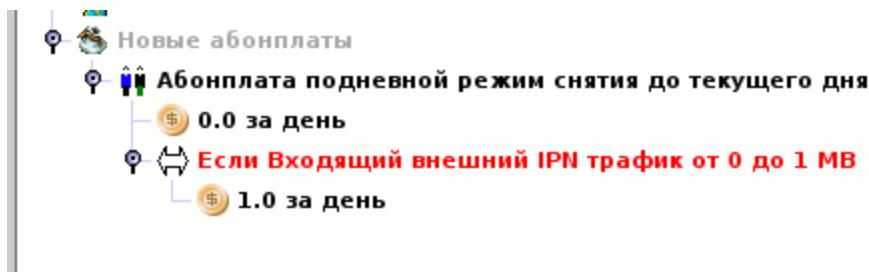
```

Приведённая выше конфигурация используется в следующих примерах тарифов:

Начисление 1 го рубля только за те дни, когда была наработка в модуле DialUP



Начисление за дни, когда у клиента не было достаточной наработки по IPN, "штрафа" в 10 рублей, стимулирующий тариф.



Все приведённые выше примеры могут быть совмещены с узлами типа: **Фильтр по времени** и **Период**.

Тариф, снимающий абонплату за prepaid 100 MB в месяц, если клиент наработал меньше, то снятие происходит пропорционально реально потреблённому объёму.

Новые абонплаты

Абонплата помесячный режим снятия

- Если Входящий внешний IPN трафик от 0 до 100 MB
50.0 пропорц. объему
- Если Входящий внешний IPN трафик от 100 до 0 MB
50.0 безусловно

Тот же тариф, но снимающий в случае, если объем менее 100, как выгоднее: т.е. анализируется доля потреблённых дней в месяце и доля потреблённого объёма, большая доля умножается на цену за месяц.

Новые абонплаты

Абонплата помесячный режим снятия

- Если Входящий внешний IPN трафик от 0 до 100 MB
50.0 как выгоднее
- Если Входящий внешний IPN трафик от 100 до 0 MB
50.0 безусловно

Необходимо учитывать, что в узле типа "Условие" нижняя граница включена в диапазон, а верхняя - нет. 0 в верхней границе диапазона означает неограниченный объем.

Абонплаты, зависящие от денежной наработки в других модулях

Модуль абонлат может производить начисление в зависимости от денежной наработки за услуги других модулей. Все используемые при тарификации денежные наработки должны быть описаны в конфигурации экземпляра модуля абонлат следующим образом.

```
module.account.<id>.title=<title>
module.account.<id>.mid=<mid>
module.account.<id>.class=<class_name>
#module.account.<id>.sids=<sids>
```

Где:

- **<id>** - уникальный числовой идентификатор денежной наработки, нумерация должна быть последовательной и непрерывной, число не должно меняться впоследствии, при изменении нумерации необходима правка всех тарифов, где используются данные денежные наработки;
- **<title>** - название денежной наработки;
- **<mid>** - код экземпляра модуля, в котором рассчитывается объём, 0 для класса KernelAccount;
- **<class_name>** - класс, рассчитывающий объём;
- **<sids>** - коды услуг через запятую (необязательный параметр).

В данный момент поддерживаются следующие классы, которые могут быть указаны в **<class_name>** для расчёта денежной наработки:

- **ru.bitel.bgbilling.modules.npay.server.bean.account.DialUpModuleAccount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.DialUpModuleAccount**) - денежная наработка в экземпляре модуля DialUp;
- **ru.bitel.bgbilling.modules.npay.server.bean.account.PhoneModuleAccount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.PhoneModuleAccount**) - денежная наработка в экземпляре модуля Phone;
- **ru.bitel.bgbilling.modules.npay.server.bean.account.InetModuleAccount** (до версии 8.XXXX - **ru.bitel.bgbilling.modules.inet.npay.InetModuleAccount**) - денежная наработка в экземпляре модуля Inet;
- **ru.bitel.bgbilling.modules.npay.server.bean.account.KernelAccount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.KernelAccount**) - денежная наработка в ядре, возможен расчёт только целиком за месяц;
- **ru.bitel.bgbilling.modules.npay.server.bean.account.VoipModuleAccount** (до версии 8.XXXX - **bitel.billing.server.npay.bean.VoipModuleAccount**) - денежная наработка в экземпляре модуля Voicеip;

Указанную в конфигурации денежную наработку использовать в узле тарифного плана типа **Условие денежной наработке**. Данный узел может быть размещён только в узле типа **Месячный режим снятия**, денежная наработка будет вычисляться за рассчитываемый период (период сочетания, см. [алгоритм](#)) за исключением случая использования класса KernelAccount, который может вычислить только сумму за весь месяц, зато подходит для вычисления наработки любого модуля.

Рассмотрим пример конфигурации экземпляра модуля абонлат для начисления несколько типов абонлат и использующую при начислении значения денежной наработки экземпляра модуля DialUp с кодом 21.

```
module.account.1.title= DialUp
module.account.1.mid=21
module.account.1.class=ru.bitel.bgbilling.modules.npay.server.bean.account.DialUpModuleAccount
module.account.1.sids=< , >
#
module.account.1.title= Inet
module.account.1.mid=179
module.account.1.class=ru.bitel.bgbilling.modules.npay.server.bean.account.InetModuleAccount
```

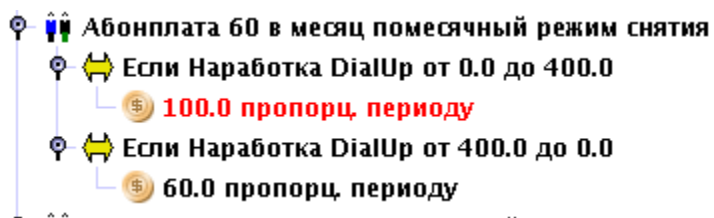
Приведённая выше конфигурация используется в следующих примерах тарифов:

Тариф начисляет дополнительную наработку в случае, если клиент недостаточно наработал по услугам DialUP модуля.



В данном случае **Догнать до 600** - это название услуги модуля абонлаты. Пропорциональный периоду режим обозначает, что 600 будет преобразовано в 300, если клиент подключён в середине месяца и период абонлаты в услугах договора также будет открыт серединой месяца. При этом наработка DialUP модуля также анализируется за этот период. Возможен и безусловный режим снятия, когда "доводимая" сумма не изменяется в зависимости от периода действия абонлаты. Режим **как выгоднее** оценивает сумму снятия при безусловном и пропорциональном режимах и выбирает максимальную из них.

Тариф начисляет клиенту абонплату в 100 рублей в том случае, если его наработка DialUP была менее 400 рублей и 60 рублей в противном случае. Абонплата называется **60 в месяц**.



Абонплаты, пропорциональные количеству телефонов (+линий), логинов и сервисов

В модуле возможно создание абонплат, начисляемых пропорционально количеству поинтов (телефонов) в модуле Phone, либо логинов в модулях VoiceIP, DialUP, либо количеству сервисов в модуле Inet, либо количеству аккаунтов модуля Voice. Предположим, существует абонплата с кодом 103, в тарифном плане для него установлена обычная месячная стоимость, 73 - код модуля телефонии.

Зависимость поддерживается только для помесечного и подневного режимов начисления абонплат.

Предположим что абонплата с кодом 103 - это абонплата за аренду телефонного модуля, в тарифном плане для него установлена обычная месячная стоимость, 73 - код модуля телефонии.

Добавление подобной конструкции в конфигурацию модуля умножит сумму абонплаты на количество телефонов в договоре.

```
module.quantity.1.mid=73
# 8.XXXX - bitel.billing.server.npay.bean.PhoneModuleQuantity
module.quantity.1.class=ru.bitel.bgbilling.modules.npay.server.bean.quantity.PhoneModuleQuantity
module.quantity.1.sids=103
```

Аналогично для модулей VoiceIP, DialUP возможно использования класса (вместо <mid> - код модуля DialUP, VoiceIP):

```
module.quantity.1.mid=<mid>
# 8.XXXX - bitel.billing.server.npay.bean.CallModuleQuantity
module.quantity.1.class=ru.bitel.bgbilling.modules.npay.server.bean.quantity.CallModuleQuantity
module.quantity.1.sids=103
```

Для модуля Inet используйте (<mid> - код модуля Inet)

```
module.quantity.1.mid=<mid>
# 8.XXXX - ru.bitel.bgbilling.modules.inet.npay.InetModuleQuantity
module.quantity.1.class=ru.bitel.bgbilling.modules.npay.server.bean.quantity.InetModuleQuantity
module.quantity.1.sids=103
```

Для модуля Voice используйте (<mid> - код модуля Voice)

```
module.quantity.1.mid=<mid>
# 8.XXXX - bitel.billing.server.npay.bean.VoiceModuleQuantity
#
module.quantity.1.class=ru.bitel.bgbilling.modules.npay.server.bean.quantity.VoiceModuleQuantity
#
module.quantity.1.class=ru.bitel.bgbilling.modules.npay.server.bean.quantity.VoiceLineModuleQuantity
module.quantity.1.sids=103
```

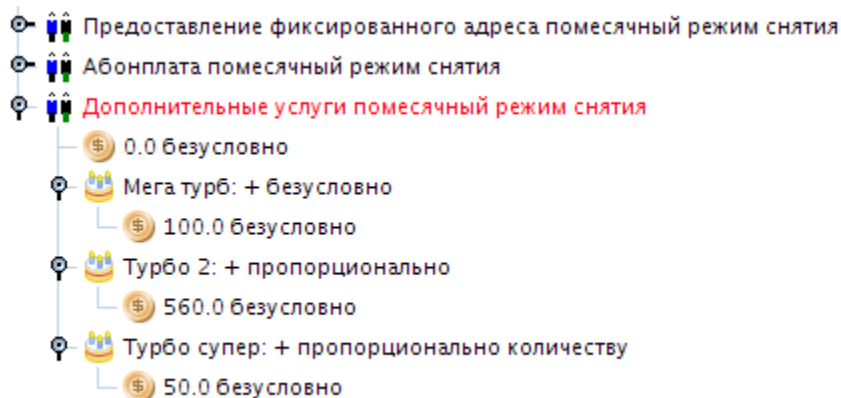
В параметре **sids** могут быть указаны несколько кодов услуг через запятую.

При помесечном режиме начисления количество высчитывается для даты окончания периода сочетания абонплата-тариф (см. [алгоритм](#)). При подневном количество высчитывается для каждых суток.

Данный коэффициент используется совместно с количеством услуги, указываемым в свойствах абонплаты. При указании в свойствах абонплаты количества А и вычисленном количестве В по конфигурации, абонплата составит СУММА * А * В.

Абонплаты, зависящие от тарифных опций

На активированную тарифную опцию можно делать начисление. Начисление по тарифной опции работает как для подневного режима снятия, так и для месячного, при этом учитывается период (день/месяц).



Для этого в услуге добавьте ветку Опция и выберите нужную опцию в редакторе ветки. "+" или "=" означает соответственно добавить цену к текущей или установить цену принудительно.

Параметры начисления для опции:

- **безусловно** - если опция была активирована один или более раз за месяц (для месячного режима снятия) или день (для подневного режима снятия), то добавить (установить) цену, рассчитанную из потомков ветки.
- **пропорционально** - пропорционально периоду активации. Для месячного режима снятия период - месяц, подневного - день. Если опция была активирована несколько раз за период, то периоды складываются. Рассчитывается с точностью до секунды.
- **пропорционально количеству** - пропорционально количеству активаций. Цена умножается на количество активаций опции за период.

Цена, в зависимости от параметров начисления, рассчитывается после прохождения подветок.

Подветки Опции работают также, как описано выше для тарифов без использования ветки Опции, т.е. установка в подветке Стоимость режима пропорц. периоду будет означать, что цена зависит от периода услуги.

Для большей наглядности наработки для отдельных опций или для групп опций можно создавать отдельные услуги.

Абонплаты, зависящие от других абонплат

С версии 9.2204+ доступен новый узел "Фильтр по услуге для помесечных абонплат" с несколькими режимами работы, этот узел позволяет реализовать несколько различных способов начисления абонплат

Фильтр по услуге: **Абонплата ТВ**

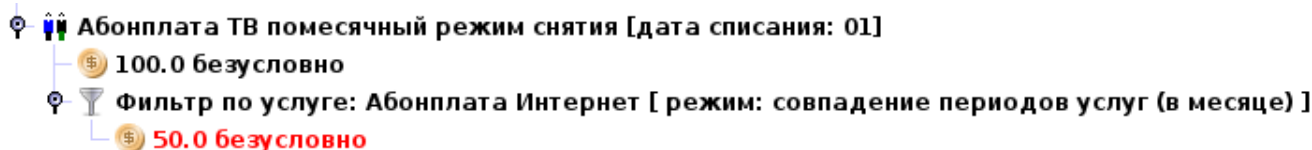
Режим фильтрации:

- весь месяц
- совпадение периодов услуг (в месяце)
- дней услуги в месяце: больше или равно ->
- месяц с начала услуги: меньше или равно ->
- месяц с начала услуги: больше или равно ->
- последний месяц услуги

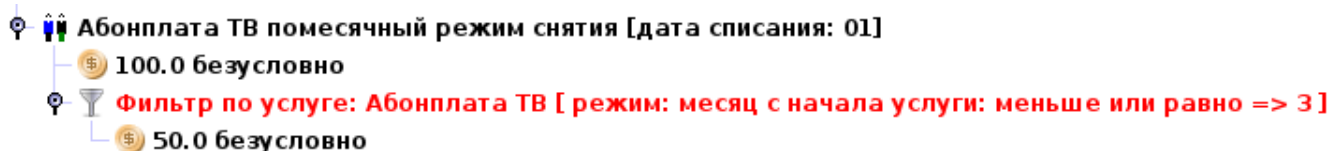
✓ ✗

Рассмотрим несколько возможных вариантов

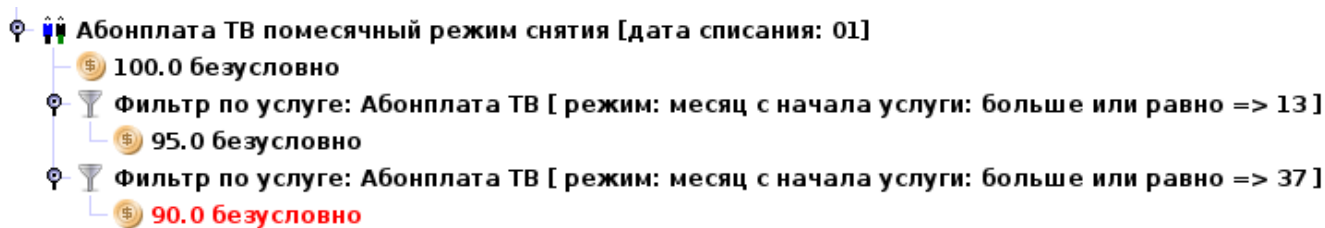
1) Скидка на одну из абонплат при наличии на договоре другой абонплаты, например у нас есть две абонплаты ТВ и Интернет, и мы хотим предоставить на абонплату ТВ скидку 50% при наличие абонплаты Интернет



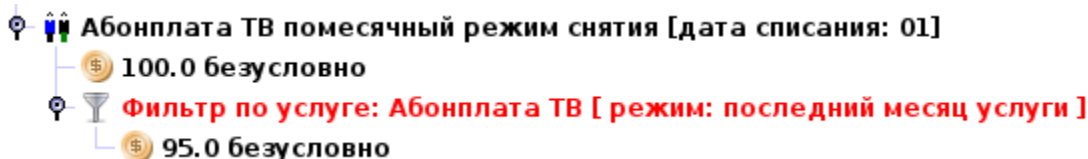
2) Скидка на абонплату на первый месяц (или на несколько месяцев), например у нас абонплата ТВ и первые 3 месяца скидка 50%



3) Скидка для лояльных клиентом, например если с момента подключение клиента прошел 1 год скидка - 5%, больше 3 лет - 10%



4) Скидка на последний месяц абонплаты

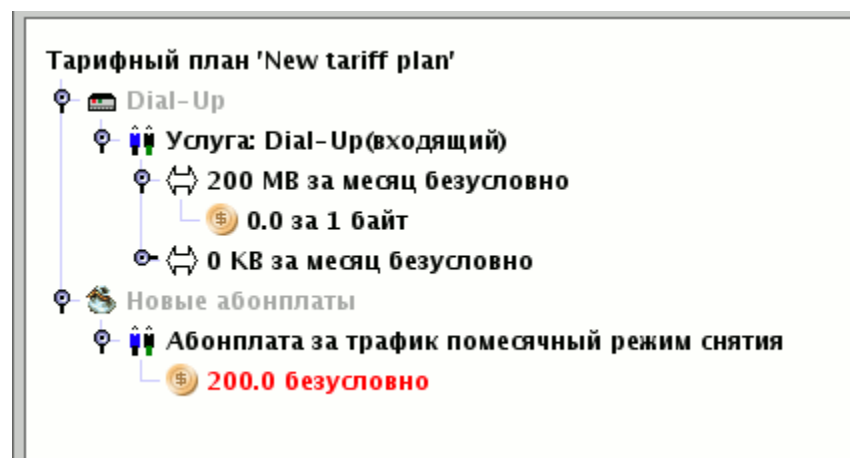


Методики построения тарифных планов

Особенностью модуля абонплат является то, что он почти всегда используется в связке с другими модулями. При этом можно выделить абонплаты, логически связанные с услугами в каком-либо из модулей и логически отдельные.

Например, если в модуле DialUP есть тариф с включённым трафиком за абонплату, то абонплата логически связана с тарифом DialUP. Примером логически не связанной абонплатой может являться абонплата за предоставление фиксированного IP-адреса.

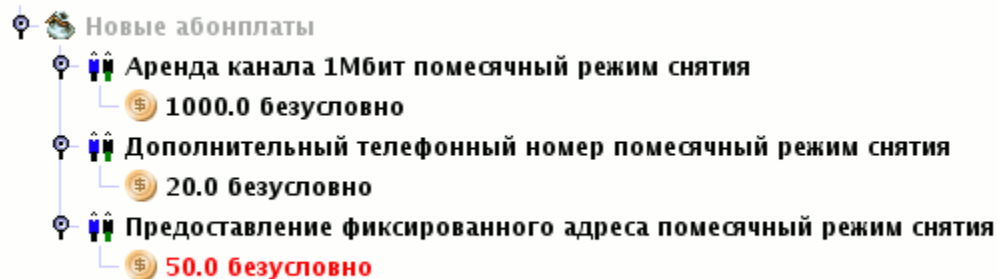
Тарифы логически связанных абонплат удобнее всего объединять в одно дерево с тарифами за услуги. Например, есть линейка тарифов DialUP с предоплаченным трафиком: 100 МБ - 100 руб, 200 МБ - 200 руб. Указаны через тире включённый трафик и абонплата за него. Создаётся услуга **А** **бонплата за трафик**, одна должна присутствовать во всех договорах, использующих данный тариф. Вид тарифных планов будет следующим:



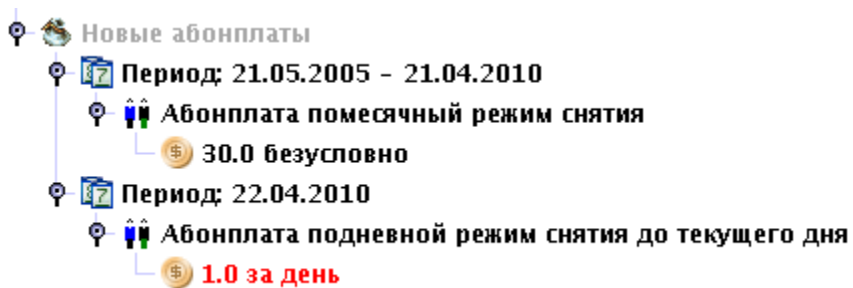
Применение такой схемы гарантирует изменение размера абонплаты при переходе абонента на другой DialUP тариф.

Логически не связанные абонплаты можно вынести в один тариф и указывать этот тариф дополнительно всем договорам. Пример такого тарифа:

Тарифный план 'New tariff plan'



Если необходимо в тарифном плане организовать перевод абонплаты с одного режима снятия на другой, то необходимо построить тарифный план следующим образом: на верхнем уровне добавить узлы типа Период для первого и второго режимов снятия, а затем внутри этих узлов создать дочерние узлы для абонплат первого и второго типа соответственно.



Начисление

Использование наборов услуг позволяет группировать абонплаты по времени снятия. Например, если необходимо чтобы фиксированные абонплаты с кодами 3,4 и 5 снимались в начале месяца, а **Доводящая** абонплата с кодом 18 снималась в конце месяца, то создаются 2 набора услуг в конфигурации модуля.

```
service.set.1.title=  
service.set.1.sids=18  
service.set.2.title=  
service.set.2.sids=3,4,5
```

Набор услуг может быть также задан как полный перечень услуг модуля за исключением некоторых наборов услуг. Например:

```
service.set.3.title=,  
service.set.3.sids=-{1,2}
```

В модуле всегда определён набор услуг **Полный набор услуг**, содержащий в себе все услуги модуля. Он используется, если все абонплаты снимаются одновременно.

Начисление может быть произведено в автоматическом и ручном режиме. В ручном режиме необходимо на вкладке **Начисление** модуля выбрать обсчитываемый месяц и запустить расчёт. При этом возможно выбрать набор услуг модуля, на которые производятся начисления. В случае, если начисление было произведено ошибочно, возможно очистить наработку, при этом также используется набор услуг. Если в конфигурации сервера прописать опцию

```
client.gui.npay.recalc.month.range=1
```

то период можно будет задавать в виде диапазона месяцев.

Задачу начисления выполняет процесс планировщика, он должен быть запущен.

В автоматическом - в планировщике заданий необходимо добавить задачу **Начисление Npay абонплат**.

Периодичность запуска задачи определяется требуемой частотой обновления объёма абонплаты. При необходимости в конфигурации может быть явно указан набор услуг:

```
service.set=< >
```

Если он не указан, используется **Полный набор услуг**. С использованием наборов услуг возможна настройка снятия различных абонплат в разное время. Необходимо учитывать, что при отработке задачи начисления берётся час, предшествующий текущему. Это даёт возможность снимать абонплату в конце месяца, установив запуск задачи на 0 часов последующего месяца. Данная особенность может мешать произвести съём абонплат при подневном режиме снятия ранее, чем первый час новых суток. При запуске задачи в 0 часов абонплаты будут начислены лишь по предыдущие сутки. Для отключения перевода часа назад добавьте в конфигурации задачи опцию:

```
hour.minus=0
```

Дебетовые абонплаты

Режим дебетовых абонплат позволяет изменять статус дебетовых договоров в случае, если начисление им абонентской платы может привести к опусканию остатка меньше лимита. Режим применяется только к договорам с режимом **Дебет**. Режим включается в конфигурации экземпляра модуля следующим образом.

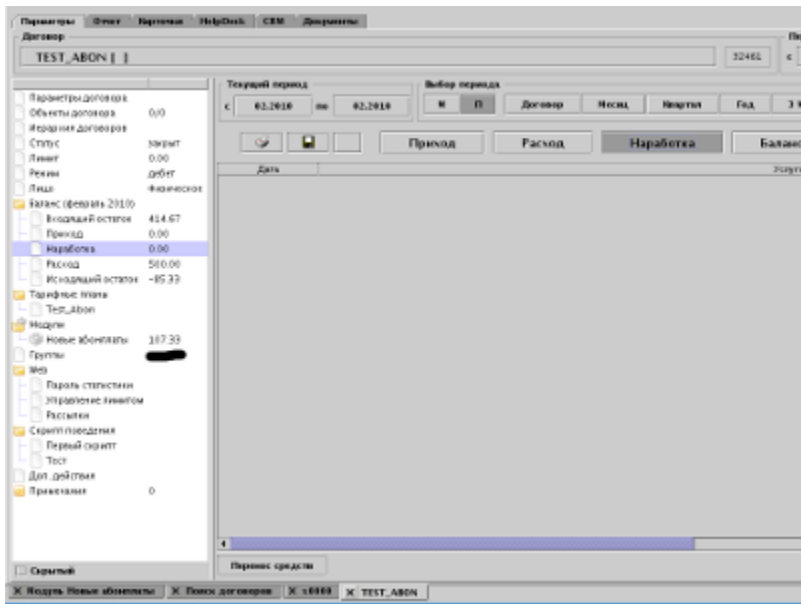
```
debet.npay.status.manage=1
#
debet.npay.active.status=0
#
debet.npay.locked.status=3
# , , .
#debet.npay.status.rule=0:3,3:0,4:1,1:4
# , ;
#debet.npay.status.manage.groups=
# , ;
#debet.npay.status.manage.label.ids=
# ; .
# ,
#debet.npay.status.manage.tariff.ids=
# ( , , )
#debet.npay.service.set=
# ,
#debet.npay.unlock.balance.limit=0
# , 30
#debet.npay.unlock.balance.for.days=0
# (1-, 0 - ).
debet.npay.processDependentSub=0
# , (0 - )
check.account.changed=1
```

Закрытие статусов договоров производится задачей планировщика **Закрытие статуса NPay договоров по балансу**, запуск которой должен осуществляться в начале суток **до** переобсчёта абонентских плат.

Для каждого активного договора оценивается сумма начислений абонентских плат при тарификации до текущих суток. Оценивается уже начисленная договору наработка за абонентские платы. В случае, если уже начисленная наработка более или равна планируемой к начислению, не выполняется никаких действий. В случае, если планируемая к начислению наработка больше уже начисленной и её начисление приведёт к понижению баланса договора ниже лимита, то статус договора меняется на определённый в переменной конфигурации **debet.npay.locked.status**.

При подневном начислении абонентских плат до текущего дня блокировка будет производиться с начала любых суток. При подневном начислении до конца месяца, либо помесечном режиме начисления блокировка возможна только в начале месяца перед первым начислением. Для супердоговоров оценивается их совместная наработка со всеми зависимыми субдоговорами.

Перевод договора в активный статус, указанный в переменной **debet.npay.active.status**, происходит при обновлении баланса(платеж, расход, наработка) или изменении лимита, когда остаток баланса позволяет открыть договор от текущей даты, начислить ему абонентскую плату и баланс при этом не должен опуститься ниже лимита. Минимально необходимая для открытия сумма платежа должна отображаться в дереве карточки договора напротив экземпляра модуля NPay.



При приходе платежа и открытии договора производится переназначение абонентских плат до текущей даты с учётом нового статуса.

Групповые операции

На данный момент доступна только одна групповая операция: "Добавление/прерывание абонплат".

Групповая операция "Добавление/прерывание абонплат"

Данная групповая операция позволяет добавить или прервать (в частном случае - закрыть) абонплату у договоров.

Новые абонплаты: Добавление/прерывание абонплат

Изменяется таблица
pray_service_object_66

Действие

Добавить Прервать

Период

с 01.07.2010 по

Количество

1

Абонплаты

- Абонплата за трафик
- Абонплата от DialUP
- Абонплата от IPN
- Абонплата: Догнать до 600
- Аренда канала 1Мбит
- Дополнительные услуги
- Дополнительный телефонный номер
- Предоставление фиксированного адреса

Все Сброс Инверт.

При установленной опции **Добавить** выбранные абонплаты будут добавлены всем указанным договорам с заданным периодом и количеством.
При установленной опции **Прервать** выбранные абонплаты будут прерваны на данный период. Если вторая граница периода будет открытой, то выбранные абонплаты будут закрыты датой, указанной в левой границе периода.

Модуль RSCM

Назначение модуля

Модуль RSCM (Random Service Calculate Module - Обсчёт произвольных услуг) предназначен для начисления на счёт пользователя наработки за потребление различных типов услуг, единовременных начислений.

В отличие от простого занесения расхода, в модуле используется справочник возможных услуг, а стоимость задается через тарифный план; возможен переобсчет.

Установка и настройка модуля

Модуль устанавливается с помощью **bg_installer**. После создания его экземпляра в редакторе модулей и услуг заносится перечень услуг данного модуля. Например: **Вызов мастера**, **Получение пакета обновлений**. После этого необходимо закрыть BGBillingClient и подключиться им снова к серверу для того чтобы модуль появился в меню.

Для каждого типа услуги устанавливается единица измерения. Например: "Вызов мастера" в часах, "Получение пакета обновлений" в КБ. Привязка производится на вкладке **Параметры услуг** модуля.

ID	Название	Базовый модуль
233	Robokassa	Модуль Robokassa
60	RSCM	RSCM - обсчет произвольных услуг
247	RunTel	RunTel
207	RuRuPay	RuRuPay
218	Sberbank	Sberbank
234	SimplePay	Модуль SimplePay
255	Smarthouse	Smarthouse
187	Softkey	Softkey
253	SoftLine	SoftLine

ID	Услуга	Рассрочка (платежи по месяцам в %)	Ед. измерения
36	Вызов мастера		шт
39	Новая услуга	[20,8,8,8,8,8,8,8,8,8]	единица
33	Получение пакета обновлений		КБ
40	Пользование почтовым ящиком		шт

При добавлении новой услуги на вкладке **Параметры услуг** в **Редакторе** из выпадающего списка можно выбрать лишь те услуги, которые еще не были добавлены ранее. Это сделано для того, чтобы нельзя было добавить несколько одинаковых услуг с разными единицами измерения.

Добавление услуги (на ts03.local)

Услуга: **Регистрация домена в зоне .RU**

Единица измерения:

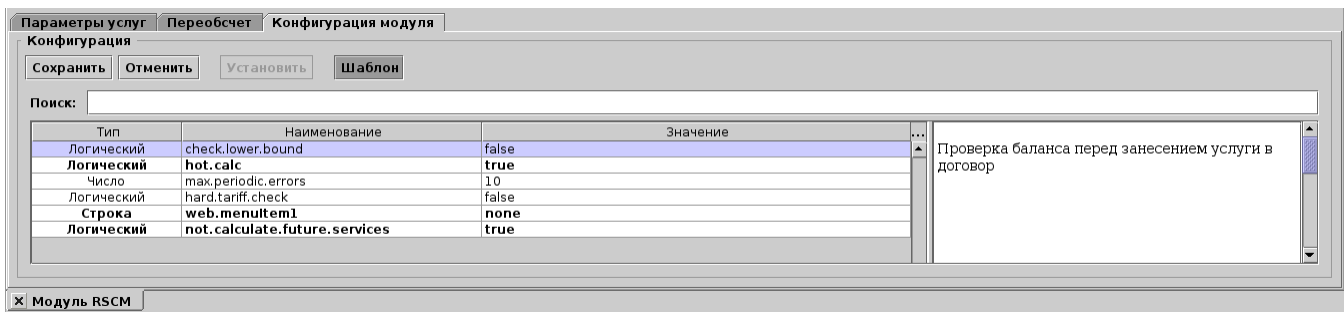
Оплата в рассрочку:

Взносы по месяцам в %:

Примеры задания взносов в % (сумма % по месяцам должна=100%):
*/10 - равными долями 10 месяцев
50;35;15 - три месяца: 50% в первый, далее 35% и 15%
0;*/10 - первый месяц 0%, далее 10 месяцев равными долями

Применить **Отмена**

Установите конфигурацию модуля, заполните параметры, сохраните конфигурацию и сделайте ее активной:



Для произведения начисления наработки необходимо в планировщике добавить задачу **Начисление RSCM** с параметром в конфигурации

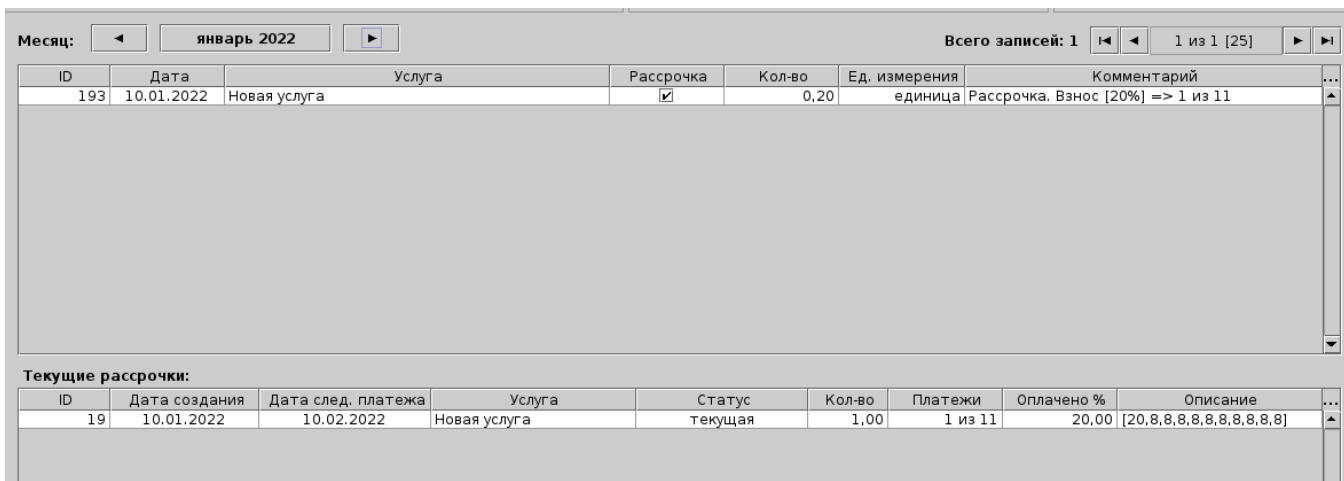
```
mid=< >
```

Также начисление за определённый месяц можно вызвать вручную на вкладке **Переобсчёт**.

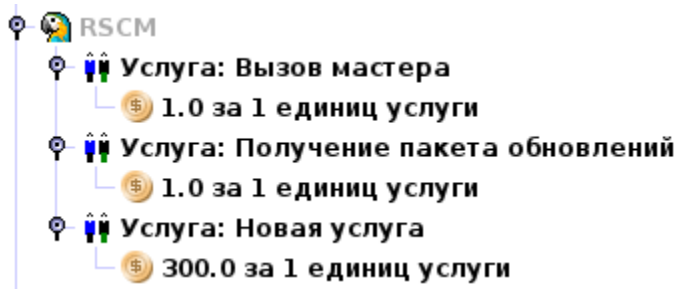
При использовании рассрочек также необходимо настроить задачу планировщика "**Рассрочка, начисление ежемесячных платежей**" (частота запуска раз в сутки).

Использование модуля

Подключение модуля к договору осуществляется через выбор узла **Модули** дерева договора, нажатием на кнопку **Новый элемент** стандартной панели инструментов. После занесения в договор наработки по услуге и выполнения переобсчёта в баланс договора будет занесена наработка в соответствии с текущим тарифным планом.



В тарифном плане должны быть определены цены для каждого вида услуги. Логика поиска тарифа соответствует [Алгоритму 1](#).



Узел с ценой услуги может быть заключён в контейнеры **Период**, либо **Временной фильтр**.

События модуля

Для данного модуля определено одно событие [BCBS](#) :

"Изменение/добавление начисления услуги RSCM в договор" - вызывается при изменении/добавлении начисления услуги RSCM в договор.

Модуль Subscription

Назначение модуля

Модуль предназначен для начисления наработки за периодические услуги. Модуль похож по функционалу на модуль NPay, за следующими исключениями: начисление возможно за произвольные периоды от 1 секунды до бесконечности, соответственно период не привязан к календарным дням, месяцам и т.д.; начисление происходит сразу в момент активации нового учетного периода при условии наличия на момент активации необходимой суммы на счете, в случае нехватки средств новый период не активируется; продление подписки возможно на новый учетный период как в автоматическом режиме так и вручную.

Настройка модуля

Установите модуль на сервер, используя утилиту **bg_installer**, обновите клиент биллинга. Затем создайте экземпляр модуля, назвав его произвольным образом (например, **Подписки**).

Определите в редакторе модулей и услуг услуги, связанные с этим модулем. Рекомендуем для каждой подписки создать отдельную услугу. Для начала можете определить несколько тестовых услуг, скорректировав список впоследствии.

Для работы модуля необходимо создать "Типы подписок" в соответствующем справочнике **Меню = Модули = Подписки**, закладка "Типы подписок". Откройте редактор щелкнув по иконке "Новый элемент" на панели инструментов.

ID	Название	ID услуги	Период	Длительность	Задержка	Округление	Автопродление	Комментарий
1	Интернет (1Мб, 7 дней)	228	12.11.2012 - ...	7 DAY	1 HOUR	1 HOUR	<input checked="" type="checkbox"/>	

Редактор

Название: Интернет (1Мб, 7 дней)

Услуга: Интернет (1Мб, 7 дней)

Период с 12.11.2012 по

Продолжительность: 7 DAY

Задержка: 1 HOUR

Округление: 1 HOUR

Разрешить автопродление подписки

Комментарий

Конфигурация

Сохранить Отмена

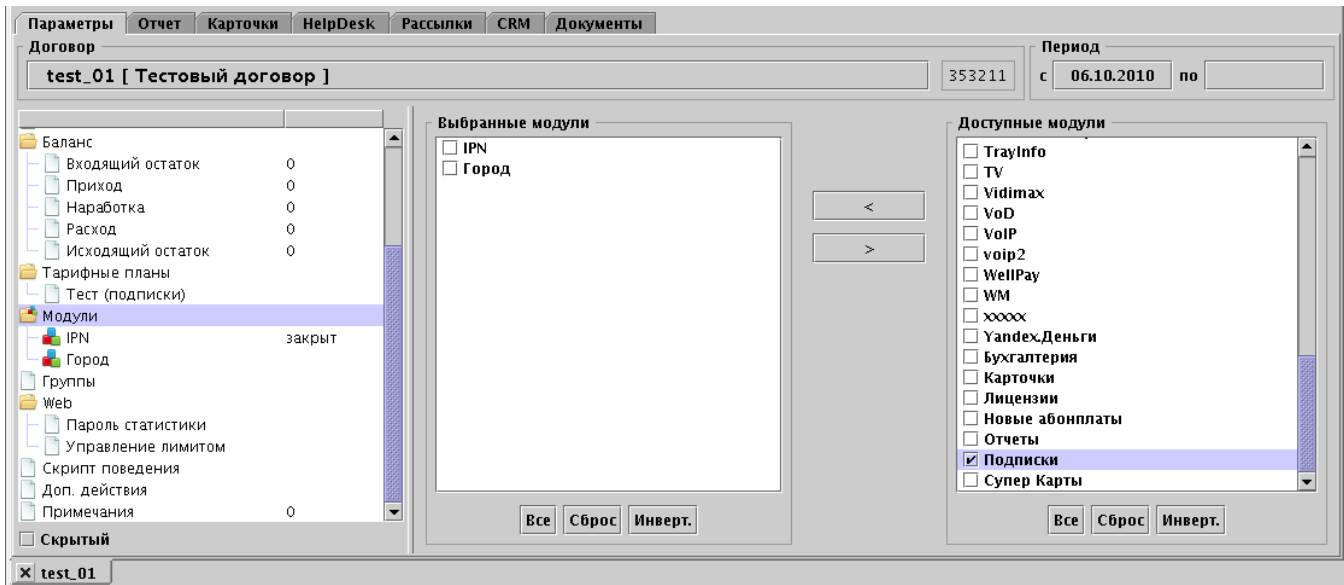
Модуль Подписки

В редакторе необходимо указать название подписки, выбрать связанную с подпиской услугу, указать период действия, продолжительность, время начала подписки (задается с помощью параметров Задержка и Округление, подробнее ниже), флаг разрешения автопродления подписки, комментарий и конфигурацию.

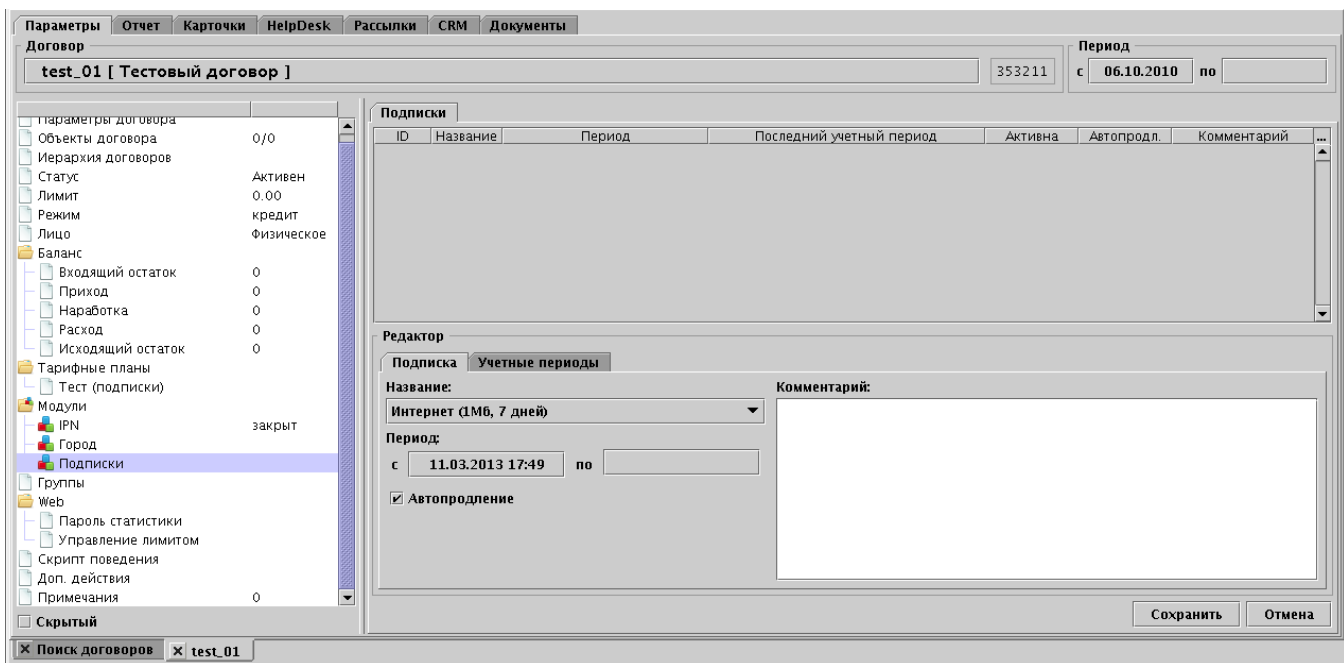
Время начала подписки задается двумя параметрами Задержка и Округление, которые используются следующим образом: берем время активации очередного учетного периода, к нему прибавляем время заданное параметром Задержка и округляем до значения заданного параметром Округление. Например, если параметр Задержка и Округление равны 1 HOUR, то время начала подписки будет установлено в 00 минут следующего часа (XX/XX/XXXX 15:34 + 1 HOUR = XX/XX/XXXX 16:34 окр. до 1 HOUR = XX/XX/XXXX 16:00). Если необходимо, что бы подписка была активирована сразу необходимо ЗАДЕРЖКУ установить в 0 и ОКРУГЛЕНИЕ в 1.

Привязка подписок к клиентам

Подключение модуля к договору осуществляется выбором узла **Модули** дерева договора, выбором экземпляра модуля в правом списке и нажатием кнопки "<".



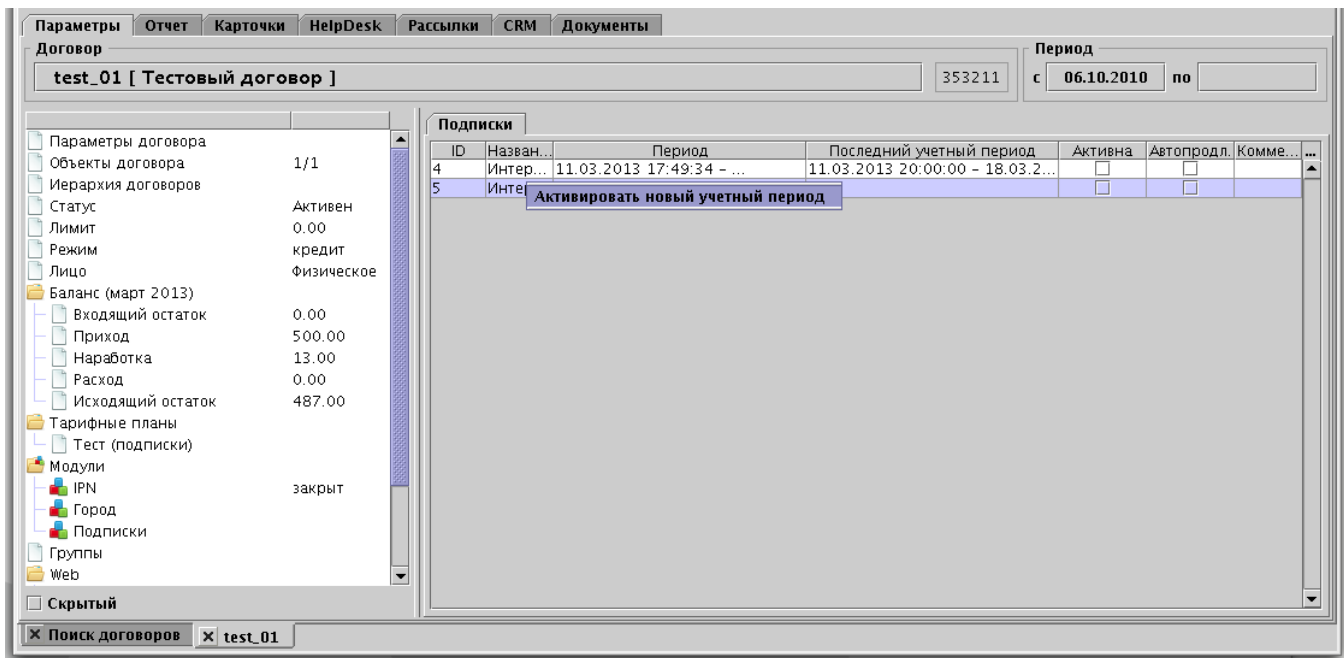
После добавления модуля к договору, необходимо добавить нужные подписки на договор. Для этого выбрать в дереве договора элемент Подписки и открыть редактор кликнув по иконке "Новый элемент" на Панели инструментов. В редакторе нужно выбрать Тип подписки, ввести период, при необходимости выставить флаг автопродления и ввести комментарий если нужно.



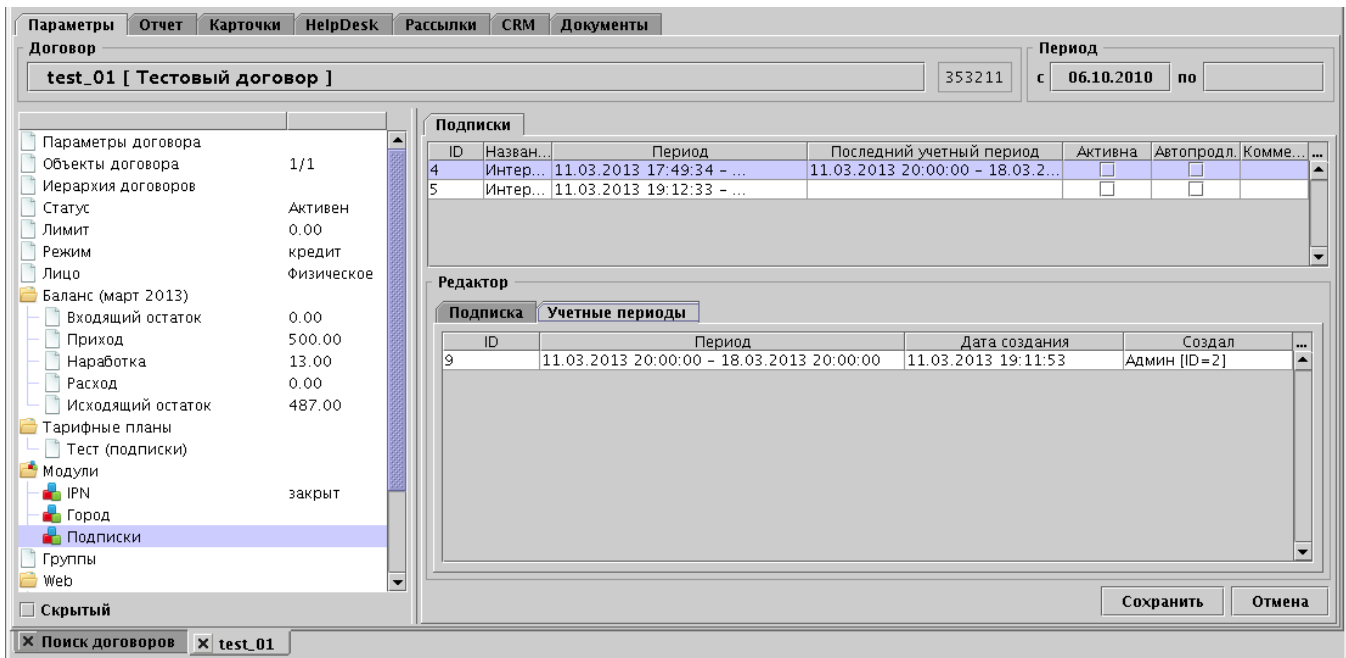
После добавления подписки на договор, подписку необходимо активировать, для этого выбираем подписку из списка в договоре, правой клавишей мыши вызываем контекстное меню и из выпадающего меню выбираем "Активировать новый учетный период".

Модуль поддерживает два режима активации/продления (режим задается в конфигурации модуля в параметре **subscription.mode**):

- 1) По балансу, активация происходит только при условии, что на договоре есть необходимая для активации сумма и услуга прописана в тарифном плане.
- 2) По статусу договора, активации происходит только при условии, что при текущем статусе договора разрешено производить начисления (настройка в каком статусе разрешено производить начисления задается в редакторе **Матрица статусов**)



При удачной активации в списке подписок в столбце "Последний учетный период" отображается период на который подписка активирована. По окончании этого периода система попытается повторно активировать новый учетный период, при условии, что на договоре достаточно средств и стоит флаг автопродления. Если флаг автопродления включен, а средств не достаточно, активация будет приостановлена до прихода на договор средств достаточных для активации подписки. Новый период не будет активирован если период действия "Типа подписки" или период подписки на договоре заканчивается раньше, чем период подписки.



Примеры тарифных планов модуля

📍 🌐 Подписки
📅 📅 Период: 01.03.2013
👤 📶 Услуга: Интернет (1Мб, 7дней)
💰 📄 13.0 за 1 единиц услуги

📍 🌐 Подписки
👤 📶 Услуга: Интернет (1Мб, 7дней)
📅 📅 Период: 01.03.2013
💰 📄 13.0 за 1 единиц услуги

Групповые операции модуля Subscription

На данный момент для модуля нет доступных групповых операций.

Модули Интернета

Модуль Inet предназначен для организации доступа, тарификации времени и трафика клиентов, потребляющих услуги передачи данных по RADIUS, Netflow v1,v5,v7,v9, IPFIX, sFlow, SNMP. Поддерживаются различные механизмы доступа: RADIUS-авторизация, управление ПО и оборудованием по ssh, telnet, snmp и т.д.; Cisco ISG, Redback CLIPS, staticCLIPS, dot1Q; DHCP.82-авторизация и др.

Устаревшие модули:

Модуль IPN предназначен для учёта постоянных подключений по протоколу IP путём сбора и анализа первичных логов. Первичные логи собираются либо напрямую коллектором VGIPNetFlowCollector, либо сторонней программой, которая "подсовывает" бинарные логи в коллектор, оставляя ему лишь функцию обработки.

Модуль DialUp предназначен для подсчёта времени и трафика клиентов, работающих по выделенным телефонным линиям, либо через VPN-туннели. Учёт трафика возможен на основании данных из RADIUS пакетов (суммарный трафик), либо NetFlow потока (трафик может быть разбит по категориям).

Модуль Inet

Назначение модуля

Модуль **Inet** предназначен для организации доступа, тарификации времени и трафика клиентов, потребляющих услуги передачи данных по RADIUS, Netflow v1,v5,v7,v9, IPFIX, sFlow, SNMP. Поддерживаются различные механизмы доступа: RADIUS-авторизация, управление ПЮ и оборудованием по ssh, telnet, snmp и т.д.; Cisco ISG, Redback CLIPS, staticCLIPS, dot1Q; DHCP.82-авторизация и др.

Базовые сведения о модуле


Базовые понятия модуля:

- **Тип трафика** - категория для разделения потребляемого трафика;
- **Устройство** - в дереве устройств определяется иерархия устройств разного типа, имеющих значение для модуля;
- **Тип устройства** - определяет поведение устройства, механизм управления сервисов на устройствах данного типа;
- **Сервис** - предоставляемый клиенту какой-то способ доступа к передаче данных, например логин или порт коммутатора;
- **Тип сервиса** - определяет параметры, которые должны быть определены у сервиса;
- **Сессия сервиса** - сущность в модуле для учёта потребляемых сервисом услуг;
- **Опция** - набор опций определяет параметры сессии сервиса в конкретный момент времени, опция абстрактна, конкретное её воплощение реализует тип устройства.

Приложения модуля:

- **BGInetAccess** - управляет доступом сервисов к сети и параметрами этого доступа;
- **BGInetAccounting** - тарификация сессий сервисов.

Связь между приложениями осуществляется посредством базы данных и MQ-сообщений.

После очередного обновления модуля необходимо в **Автоматизация->Управление динамическим кодом** скомпилировать все классы, т.к.  перекомпиляция после обновления автоматически не происходит, а классы, входящие в сборку, могли обновиться.

Настройка модуля

Для корректной работы модуля необходима установка модуля Card.

Установите модуль на сервер, создайте экземпляр. Определите в **Редакторе модулей и услуг** услуги, обчисляемые этим модулем. Например: "Входящий трафик", "Исходящий трафик", "Время". Услуги используются для разделения наработки по типам в балансе договора.

В конфигурации модуля укажите:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# ID , ( ) InetAccounting-.
# ID , ID Cisco/SmartEdge/...
# Access+Accounting, ProcessGroup. !
accounting.deviceTypeIds=

# : 0 - , 1 - , ,
# , 2 - , ( - 1)
#accounting.tariffication.checkPrice=1

# , ,
# 0 ( ) - ( ), 1 - .
# , " "
#accounting.period.activation.mode=0

# " ",
#serv.disableOnTariffError=0

# Web -
web.menuItem1= Inet
```

```

web.menuItem2= Inet
web.menuItem3=none
#web.menuItem3= Inet

# , " , ,
#
# ru.bitel.bgbilling.modules.inet.api.server.bean.InetAccountGenerate
# ru.bitel.bgbilling.modules.inet.server.bean.InetAccountGenerate ( 8.2101+)
#account.login.class=

# .
#
#serv.login.min=1
# (.. 1,2,3 10000000,
# 4, 10000001)
# 2147483647 -
#serv.login.max=9999999

#
#
serv.login.chars=1234567890ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyz
# ,
serv.login.chars.description=

# . , ,
# ( ):
# , " , ,
#
# ru.bitel.bgbilling.modules.inet.api.server.bean.InetAccountGenerate
#
#account.password.class=
#
serv.password.length.min=5
#
serv.password.length.max=16
# ( )
serv.password.chars=1234567890ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyz
# ,
serv.password.chars.description= .
#
serv.password.length.auto=6
# ( serv.password.chars)
#serv.password.chars.auto=

# card InetRadiusProcessor,
# , .
# card
#card.moduleId=
# id
#card.activate.serviceIds=
# , , ;
# 0, .
#card.login.min=0
# , , ;
# 0, .
#card.login.max=0

#
#/ .
# : 0 - , 1 - . , 1.
#client.gui.expand.device.tree=1

# ( nas.radius)
#nas.radius.legacy.fix=1

```

На вкладке **Опции** экземпляра модуля укажите хотя бы одну опцию.

При использовании старых версий настройки с **nas.radius** , необходимо добавить флаг **nas.radius.legacy.fix=1**

Общий алгоритм модуля, установка и принципы настройки серверов

BGInetAccess и BGInetAccounting обновляются как обычные [серверные приложения биллинга](#). Необходимо обновить каждое из приложений ~~перед~~ первым запуском.

В документации представлены общие настройки. Примеры для конкретных схем вы можете найти на [BGBilling Wiki](#).

В модуле представлены два типа серверов: Access и Accounting. Первый тип отвечает за управление доступом сервисов и параметрами оказания услуг: скорость, перенаправления, различные ограничения и т.п. Второй тип производит тарификацию сессий. Оба типа серверов по сути своей являются контейнерами, в которых могут быть запущены **Слушатели**, поддерживающие протоколы (RADIUS, DHCP, NetFlow-сервера). Обработку пакетов слушателей выполняют **Обработчики**, которые также определяются в конфигурации.

Установка серверов Inet

Установка Access и Accounting-серверов происходит одинаково. Разница только в названиях папок, служб и системных переменных.

BGInetAccess и BGInetAccounting обновляются как обычные [серверные приложения биллинга](#). Необходимо обновить каждое из приложений перед первым запуском. Обращаем внимание, что обновление приложений необходимо произвести **ПЕРЕД** первым запуском, поскольку при первичном обновлении приложения скачивают все необходимые для своей работы библиотеки с сервера биллинга. Для корректной работы системы обновлений, убедитесь, что в конфигурационных файлах приложений (application.xml) должен быть прописан доступный url биллинга в параметре **server.update.url**. Путь должен в себя включать host, port (опционально), context из data.properties.

Установка Access-сервера

Установка на платформу Linux

- 1) Извлеките **BGInetAccess** из архива и скопируйте в каталог **/opt/bgbilling**;
- 2) Перейдите в каталог **/opt/bgbilling/BGInetAccess**;
- 3) Удалите все .ini, .bat и .exe файлы:

```
rm -f /*.bat & rm -f /*.exe & rm -f /*.ini
```

- 4) Откройте для редактирования файл **setenv.sh** и пропишите в нем путь к Java-машине, например так:

```
...
    cd ${0%${0##*/}}.

    JAVA_HOME=/opt/java/jdk8

    if [ -z "$JAVA_HOME" ]; then
        echo "The JAVA_HOME environment variable is not defined"
        echo "This environment variable is needed to run this program"
        exit 1
    fi
    ...
```

- 5) Проверьте .sh файлы на наличие символов ^M, если символы присутствуют их можно удалить вручную, либо воспользоваться утилитой:

```
dos2unix *.sh
```

- 6) Установите права запуска для всех *.sh файлов:

```
chmod 744 *.sh
```

- 7) При необходимости настройте запуск данных приложений как служб ОС. Примеры скриптов доступны в архиве с программой.
- 8) Произведите настройку **application.xml**;
- 9) Обновите как обычные [серверные приложения биллинга](#);
- 10) Для ручного запуска и останова сервера **BGInetAccess** используйте скрипты **start.sh** и **stop.sh**.

При необходимости установки нескольких BGInetAccess-серверов на одной машине конечный каталог может быть переименован, например, в **BGInetAccessVPN**. Также требуется переименование и корректировка скрипта запуска, разнесение портов в **application.xml**.

Установка на платформу Windows

Для установки BGInetAccess на платформу Windows на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить BGInetAccess стоит Java-машина. Если её нет, установите версию не меньше 1.6.20. Загрузить можете с нашего сайта;

- 2) Загрузите с сервера BGInetAccess;
- 3) Распакуйте архив на диск C.;
- 4) Установите переменную окружения **BGINET_ACCESS_HOME =C:\BGInetAccess**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;
- 5) Установите службу BGInetAccess, для чего запустите файл **access_install.bat**;
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **access_uninstall.bat**;
- 7) Обновите как обычные [серверные приложения биллинга](#) ;
- 8) Для запуска и останова сервера BGInetAccess используйте консоль запуска и управления службами, служба BGInetAccess.

Установка Accounting-сервера

Установка на платформу Linux

- 1) Извлеките **BGInetAccounting** из архива и скопируйте в каталог **/opt/bgbilling**;
- 2) Перейдите в каталог **/opt/bgbilling/BGInetAccounting**;
- 3) Удалите все .ini, .bat и .exe файлы:

```
rm -f *.*.bat & rm -f *.*.exe & rm -f *.*.ini
```

- 4) Откройте для редактирования файл **setenv.sh** и пропишите в нем путь к Java-машине, например так:

```
...
    cd ${0%${0##*/}}.

    JAVA_HOME=/opt/java/jdk8

    if [ -z "$JAVA_HOME" ]; then
        echo "The JAVA_HOME environment variable is not defined"
        echo "This environment variable is needed to run this program"
        exit 1
    fi
...

```

- 5) Проверьте .sh файлы на наличие символов ^M, если символы присутствуют их можно удалить вручную, либо воспользоваться утилитой:

```
dos2unix *.sh
```

- 6) Установите права запуска для всех *.sh файлов:

```
chmod 744 *.sh
```

- 7) Возьмите из каталога **BGInetAccounting/script** скрипт запуска **bginet_accounting** и скопируйте его в каталог **/etc/init.d**, установите права на исполнение (см. выше). Если вы изменили каталог установки или переименовывали BGInetAccounting, скорректируйте скрипт.

- 8) Выясните текущий уровень запуска системы командой:

```
[root@gate init.d]# runlevel
N 3
```

- 9) Создайте линк для автоматического запуска Accounting-сервера:

```
ln -s /etc/init.d/bginet_accounting /etc/rcN.d/S99bginet_accounting
```

где N - требуемый уровень запуска.

10) Произведите настройку **inet-accounting.xml** (в версиях 8.2101+ заменен на **application.xml**);

11) Обновите как обычные [серверные приложения биллинга](#).

11) Для запуска и останова сервера **BGInetAccounting** используйте скрипты **start.sh** и **stop.sh**.

При необходимости установки нескольких BGInetAccounting-серверов на одной машине конечный каталог может быть переименован, например, в **BGInetAccountingVPN**. Также требуется переименование и корректировка скрипта запуска, разнесение портов в **inet-accounting.xml**.

Установка на платформу Windows

Для установки BGInetAccounting на платформу Windows на диск C:.

1) Убедитесь, что на машине, где вы собрались ставить BGInetAccounting стоит Java-машина. Если её нет, установите версию не меньше 1.6.20. Загрузить можете с нашего сайта;

2) Загрузите с сервера BGInetAccounting;

3) Распакуйте архив на диск C:;

4) Установите переменную окружения **BGINET_ACCOUNTING_HOME =C:\BGInetAccounting**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;

5) Установите службу BGInetAccounting, для чего запустите файл **accounting_install.bat**;

6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **accounting_uninstall.bat**;

7) Обновите как обычные [серверные приложения биллинга](#) . ;

8) Для запуска и останова сервера BGInetAccounting используйте консоль запуска и управления службами, служба BGInetAccounting.

Общая часть конфигурации

Слушатели, процессоры и другие сущности контейнера определяются в конфигурационном файле **application.xml** (старых версиях **inet-access.xml**, либо **inet-accounting.xml**).

Рассмотрим общую часть XML-конфигурации обоих серверов.

Код

```
<?xml version="1.0" encoding="UTF-8"?>
<application context="access">
  <!-- -->
  <param name="app.name" value="BGInetAccess"/>
  <!-- id -->
  <param name="app.id" value="" />

  <!-- -->
  <param name="db.driver" value="com.mysql.jdbc.Driver"/>
  <param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&
allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&
queryTimeoutKillsConnection=true"/>
  <param name="db.user" value="bill"/>
  <param name="db.pswd" value="bgbilling"/>

  <!-- MQ -->
  <param name="mq.url" value="failover:(tcp://localhost:61616)"/>
  <param name="mq.user" value="bill"/>
  <param name="mq.pswd" value="bgbilling"/>

  <!-- -->
  <param name="server.apps.type" value="inet"/>
  <param name="server.update.url" value="http://localhost:8080/bgbilling"/>
  <param name="server.update.check.ssl" value="no"/>

  <!-- id -->
  <param name="moduleId" value="" />
  <!-- id -->
  <param name="rootDeviceId" value="" />

  <param name="datalog.radius.dir" value="data/radius"/>
  <param name="datalog.dhcp.dir" value="data/dhcp" />
  ....
```

Параметры:

- **app.name** определяет имя приложения, оно используется, например в системе алармов;
- **app.id** - уникальный числовой идентификатор приложения среди всех приложений биллинга с данным параметром в XML-конфигурации, значение его не должно меняться всё время жизни системы;
- **moduleId** - код экземпляра модуля Inet, к которому относится сервер;
- **server.update.url** - url, по которому доступен сервер биллинга для получения обновлений модуля inet и всех необходимых библиотек.

Далее следуют стандартные параметры с настройкой доступа к серверу БД и к MQ-серверу (серверам).

Каждый сервис привязан к своему устройству. В конфигурации каждого из серверов Access и Accounting указывается корневое устройство, от которого, включительно, начинается загрузка в память устройств и сервисов. Код этого устройства указывается в параметре **rootDeviceId**.

Параметры вида **datalog.*** определяют каталоги для хранения бинарных логов RADIUS, DHCP, NetFlow-пакетов. Хранение данных Access-сервером необходимо для возможности по запросу предоставления пакетов авторизации сессии в интерфейсе клиента. Хранение данных Accounting-сервером позволяет выполнять переобработку логов. Access-сервер хранит RADIUS (Access-запросы) и DHCP-пакеты. Accounting - RADIUS (Accounting-запросы) и NetFlow-пакеты. Хранение в бинарном виде в файловой системе позволяет разгрузить БД от большого объёма информации. Логи сохраняются с разбивкой по каталогам устройств-источников.

Каждому **устройству** может быть сопоставлен свой объект класса-активатора сервисов и объект класса-процессора протокола. Классы объектов указываются в настройках типа устройства, классы должны расширять абстрактные классы **ru.bitel.bgbilling.modules.inet.access.sa.ServiceActivatorAdapter** и **ru.bitel.bgbilling.modules.inet.access.sa.ProtocolHandlerAdapter** соответственно. В момент старта сервера для каждого устройства, в типе которого указаны классы, создаётся отдельный объект и вызывается метод **init**, в котором могут быть считаны параметры конфигурации. Все методы объекта класса-обработчика активации вызываются последовательно от устройства далее у всех его предков до корневого узла сервера, что позволяет производить настройки в иерархии устройств.

При создании сервиса на устройстве, к которому оно привязано, вызывается метод **serviceCreate** объекта класса-активатора. В этом методе возможно указание настроек, которые должны быть добавлены на устройство только при создании сервиса. Этот же метод вызывается, когда сервис был добавлен будущим числом и это число наступило. При удалении сервиса или истечении даты его закрытия на устройстве вызывается метод **serviceCancel**.

Опции могут быть определены на сервисе и сессии. Помимо опции у сервиса определяется его состояние. С помощью опций сервиса возможно управление параметрами статического доступа. При этом нет необходимости в потреблении сервисом трафика, нет необходимости в сессиях. Опции указываются непосредственно в свойствах сервиса. При изменении статуса сервиса или опций сервиса вызывается метод объекта класса-активатора устройства **serviceModify**.

Сервис потребляет услуги модуля в ходе сессий, при этом для тарификации используются тарифные планы. Каждая сессия привязана к устройству, не обязательно к тому, к которому привязан сервис. Логика привязки сессии к устройству определяется слушателем (RADIUS, DHCP, NetFlow), об этом будет описано позднее. В конфигурации того устройства, к которому привязана сессия, может быть определено устройство, с которого получается информация по трафику данной сессии. Устройства и интерфейсы определяется переменной: **flow.agent.link=<device_id>:<iface_id>**

Где:

- **<device_id>** - код устройства, с которого снимается трафик;
- **<iface_id>** - код интерфейса устройства, через который выходит трафик сессии.

Например:

```
flow.agent.link=1:-1
```

Указание другого устройства, отличного от устройства сессии имеет смысл при съёме трафика по протоколу NetFlow с вышестоящего роутера. Код интерфейса при этом должен совпадать с кодом интерфейса, идущим во Flow-записях и обозначающий интерфейс роутера, смотрящий на устройство сессии. Более подробно об это переменной можно прочитать [тут](#).

Сессия сервиса также обладает набором опций, который состоит из опций сервиса и опций тарифного плана. Второй набор опций может меняться в ходе тарификации. Первый - в результате правки сервиса. При изменении параметров сессии в объекте класса-активатора устройства, к которому привязана сессия вызываются метод **connectionModify**. При завершении - **connectionClose**. Для старта сессии необходимо наличие NetFlow-трафика по IP-адресу сервиса, либо наличие сигнала (RADIUS, DHCP).,

Как уже указывалось ранее, в конфигурации серверов могут быть указаны слушатели, рассмотрим их.

Слушатель InetRadiusListener

Слушатель предназначен для обработки RADIUS-пакетов. Общие сведения о протоколе RADIUS доступны [здесь](#).

Добавляется в контейнер следующим образом.

Код

```
<context name="radius">
    <!-- C RADIUS- -->
    <bean name="radiusProcessor" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor"/>

    <!-- ScheduledExecutorService, dataLogger -->
    <scheduledExecutorService name="hrlydtlggr" corePoolSize="1" />

    <!-- C dataLogger, RADIUS- ( ) -->
    <bean name="radiusDataLogger" class="ru.bitel.bgbilling.modules.inet.radius.RadiusHourlyDataLogger">
        <param name="scheduledExecutor">hrlydtlggr</param>
    </bean>

    <!-- C RADIUS- dataLogger -->
    <bean name="radiusListener" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusListener">
        <constructor>
            <!-- (), . - -->
            <param name="host" value="" />
            <!-- , -->
            <param name="port" value="1812" />
            <!-- -->
            <param name="recvBufferSize">512 * 1024</param>
            <!-- SO_RCVBUF- -->
            <param name="soRCVBUF"></param>
            <!-- - -->
            <param name="threadCount">10</param>
            <!-- -->
            <param name="maxQueueSize">200</param>
            <!-- -->
            <param name="processor">radiusProcessor</param>
            <!-- , RadiusListener.Mode.authentication -->
            <param name="mode">RadiusListener.Mode.authentication</param>
            <!-- dataLogger -->
            <param name="dataLogger">radiusDataLogger</param>
        </constructor>
    </bean>
</context>
```

Параметры:

- **radiusProcessor** - класс процессора, реализующий логику обработки пакетов;
- **radiusDataLogger** - объект, осуществляющий запись бинарных логов, по умолчанию это объект типа **ru.bitel.bgbilling.modules.inet.radius.RadiusHourlyDataLogger** (сохранение бинарных логов с разбивкой по источникам и часам);
- **host** - IP-адрес, на котором слушатель, пустое значение - прослушивать любой IP-адрес;
- **port** - прослушиваемый UDP-порт;
- **soRCVBUF** - рекомендуемый SO_RCVBUF (socket receive buffer) для сокета. На FreeBSD большие значения могут вызвать ошибку и сервер не запустится;
- **byteBufferCapacity** - размер буфера для приёма пакетов;
- **threadCount** - число потоков-обработчиков пакетов, рекомендуемые значения 10-30, не рекомендуется указывать более 100;
- **maxQueueSize** - максимальный размер очереди пакетов, при превышении размера очереди пакеты начинают отбрасываться и высылается аларм, рекомендуемые значения 200-1000. Следует учитывать, что если по какой-то причине сервер не успевает обрабатывать пакеты, очередь растёт, то какие-то пакеты из очереди могут быть обработаны с опозданием и NAS уже выслал повторный запрос, который опять попадет в очередь и опять может быть обработан с опозданием. Поэтому большое значение вместо распределения нагрузки может вызвать ее увеличение;
- **mode** - режим работы, может принимать значения **RadiusListener.Mode.authentication** для обработки Access-Request-пакетов в InetAccess и **RadiusListener.Mode.accounting** для обработки Accounting-пакетов в InetAccounting.

Процессор InetRadiusProcessor

Предназначен для авторизации и аккаунтинга по протоколу RADIUS большинства типов коммутируемых соединений.

Загружает список NAS'ов из дочерних узлов корневого узла сервера, начиная с самого корневого узла. Если нужно фильтровать по типам устройств (т.е. какие-то дочерние узлы не учитывать как NAS'ы), то в конфигурации корневого устройства в параметре **radius.deviceTypeIds** необходимо указать id типов устройств-NAS'ов через запятую. **У устройства-NAS'a поле Идентификатор должно быть заполнено.**

Поиск NAS'a для пришедшего пакета производится сначала по атрибуту NAS-Identifier с поиском по идентификатору устройства, затем, если устройство не найдено, по NAS-IP-Address с поиском по хостам устройств. Сессия сервиса привязывается к устройству, представляющему NAS.

Поиск сервиса/логина

При авторизации поиск сервиса Inet может осуществляться как по логину, так и по интерфейсу или VLAN'у. Для указания режима поиска необходимо прописать в конфигурации устройства-NAS'a или в конфиге любого его устройства-предка параметр:

```
# : 0 ( ) - , 1 - (
# AGENT_REMOTE_ID AGENT_CIRCUIT_ID INTERFACE_ID), 2 - VLAN (
# AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID), 4 - VLAN
# ( AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID),
# 5 - MAC- ( MAC_ADDRESS), 6 - MAC-
# ( MAC_ADDRESS).
radius.servSearchMode=0
```

radius.servSearchMode может принимать значения:

- **0** - поиск по логину из атрибута User-Name;
- **1** - поиск по интерфейсу на (найденном) устройстве;
- **2** - поиск по VLAN'у на устройстве (в предобработке должны быть проставлены опции AGENT_REMOTE_ID и AGENT_CIRCUIT_ID или VLAN_ID);
- **4** - поиск по VLAN'у на устройстве и его дочерних устройствах (в предобработке должны быть проставлены опции AGENT_REMOTE_ID и AGENT_CIRCUIT_ID или VLAN_ID);
- **5** - поиск по MAC-адресу на устройстве (в предобработке должна быть проставлена опция MAC_ADDRESS);
- **6** - поиск по MAC-адресу на устройстве и дочерних устройствах (в предобработке должна быть проставлена опция MAC_ADDRESS);
- **7** - поиск по адресу, указанному в User-Name, из диапазона адресов сервиса (в типе сервиса должно быть указано serv.search.address=1);
- **9** - поиск по адресу, указанному в User-Name, из диапазона адресов сервиса (в типе сервиса должно быть указано serv.search.address=1) на агентских (дочерних) устройствах;
- **10** - поиск по MAC-адресу на всех устройствах;
- **16** - поиск по VLAN + IP (атрибута User-Name) (8.2201+);

Также возможен дополнительный поиск дочернего сервиса:

```
radius.servSearchMode=<servSearchMode>-<subServSearchMode>
```

<subServSearchMode> может принимать значения:

- 0 или отсутствует - нет поиска дочернего сервиса
- 1 - поиск дочернего сервиса по MAC-адресу, если такого дочернего сервиса нет - ошибка авторизации;
- 2 - поиск дочернего сервиса по MAC-адресу, если такого дочернего сервиса нет - сессия будет привязана к родительскому сервису;
- 5 - поиск дочернего сервиса по IP-адресу из поля User-Name, если такого дочернего сервиса нет - ошибка авторизации;
- 6 - поиск дочернего сервиса по IP-адресу из поля User-Name, если такого дочернего сервиса нет - сессия будет привязана к родительскому сервису;
- 7 - проверка IP-адреса из поля User-Name на совпадение с IP-адресом сервиса или поиск дочернего сервиса по IP-адресу, если не совпадает и такого дочернего сервиса нет - ошибка авторизации;
- 8 - проверка IP-адреса из поля User-Name на совпадение с IP-адресом сервиса или поиск дочернего сервиса по IP-адресу, если не совпадает и такого дочернего сервиса нет - сессия будет привязана к родительскому сервису.

Например, radius.servSearchMode=1 - поиск по порту, radius.servSearchMode=2-1 - поиск по VLAN с проверкой MAC-адреса, radius.servSearchMode=2-7 - поиск по VLAN с проверкой IP-адреса.

Возможен последовательный поиск с разными режимами (если по первому режиму сервис не был найден - применяется следующий), для этого режимы нужно указать через запятую:

```
radius.servSearchMode=7,2
```

При поиске по логину поиск сервиса происходит по значению атрибута **User-Name**, по его совпадению с логином сервиса. После логина в атрибуте User-Name может быть указан **реалм** с разделителем **@**, например "vasya@local". Если реалм не указан, то предполагается, что пользователь использует реалм default. При этом принудительное указание реалма default (например, "vasya@default") не допускается.

По умолчанию перед поиском сервиса по логину из User-Name удаляется значение домена и не удаляются пробелы в начале и в конце. Также поиск идет с учетом регистра. Изменить поведение можно в конфигурации устройства (см. Пример конфигурации устройства-NAS'a).

Список допустимых реалмов указывается в параметре устройства/типа сервиса radius.realm через запятую, например:

```
#radius.realm=default  
radius.realm=default, local
```

Параметр может быть указан как в конфигурации устройства (или типа устройства), так и в конфигурации типа сервиса - в последнем случае значение будет главнее, чем тот же параметр в конфигурации устройства. По умолчанию допускается только реалм default.

При поиске по интерфейсу или VLAN'у, сервис может быть привязан к дочернему для NAS'a устройству - например, в случае Cisco ISG или Redback CLIPS - сервис привязан к коммутатору, NAS'ом выступает маршрутизатор. В этом случае необходимо в предобработке RADIUS-запроса установить опцию **AGENT_REMOTE_ID** со значением идентификатора дочернего коммутатора. Если опция будет присутствовать и такое дочернее устройство существует, то поиск сервиса будет идти относительно этого агентского устройства, иначе, по умолчанию, поиск идет относительно устройства-NAS'a.

```
request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, "1CBDB9E64878" );
```

Данную опцию необходимо устанавливать как в preprocessAccessRequest, так и в preprocessAccountingRequest. Значение обычно извлекается из RADIUS-атрибутов запроса, содержащих, например, значение субопций DHCP option 82. Значение можно установить как строкой, так и массивом байт (byte[]). В случае, если опция установлена в предобработке запроса, будет произведен поиск агентского устройства на совпадение с полем Идентификатор.

Также возможна схема с несколькими NAS'ами, работающими параллельно, в этом случае ветку с устройствами коммутаторами нужно завести отдельно (не делать ее дочерней по отношению к одному из устройств-NAS'ов), а в конфигурации устройств-NAS'ов указать параметр **device.agentDeviceRootId=<x>**, где <x> - ID устройства-папки, в котором находится подветка с коммутаторами:

Для определения интерфейса или VLAN возможны два способа: установка опции **AGENT_CIRCUIT_ID** или установка напрямую опции **INTERFACE_ID/VLAN_ID**. В первом случае значение номера интерфейса или VLAN из опции будет извлечено по параметрам конфигурации агентского устройства **dhcp.option82.interfaceId.position**, **dhcp.82.interfaceId.length**, **dhcp.option82.vlanId.position**, **dhcp.option82.vlanId.length**. Во втором - значение должно быть равно интерфейсу или порту.

Пример (установка опций из атрибутов-значений субопций DHCP option 82, данный код есть в обработчиках, которые входят в дистрибутив модуля):

```
RadiusAttribute<?> agentRemoteId = request.getAttribute( 2352, 96 );  
if( agentRemoteId != null )  
{  
    ByteBuffer data = agentRemoteId.getData();  
    data.position( 2 );  
    data = data.slice();  
  
    request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, data );  
}  
  
RadiusAttribute<?> agentCircuitId = request.getAttribute( 2352, 97 );  
if( agentCircuitId != null )  
{  
    ByteBuffer data = agentCircuitId.getData();  
    data.position( 2 );  
    data = data.slice();  
  
    //    VLAN'  
    request.setOption( InetRadiusProcessor.AGENT_CIRCUIT_ID, data );  
}
```

Пример (vlan per user - поиск по vlan):

```

/*
//      - , 2;
//      - 4 ( vlan )
RadiusAttribute<?> agentRemoteId = request.getAttribute( 2352, 96 );
if( agentRemoteId != null )
{
    ByteBuffer data = agentRemoteId.getData();
    data.position( 2 );
    data = data.slice();

    request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, data );
}*/

Pattern vlanPattern = Pattern.compile( "(.+) vlan-id (\\d+)" );

String nasPortId = request.getStringAttribute( -1, 87, null );
if( nasPortId != null )
{
    Matcher m = vlanPattern.matcher( nasPortId );
    if( m.find() )
    {
        String vlanId = m.group( 2 );
        request.setOption( InetRadiusProcessor.VLAN_ID, vlanId );
    }
}
}

```

Таким образом для поиск сервиса возможен:

- - по атрибуту **User-Name** (по умолчанию);
- - по опциям **AGENT_REMOTE_ID** (идентификатор агентского устройства - устройства, на котором, собственно, интерфейс) и **INTERFACE_ID** (или **AGENT_CIRCUIT_ID**, содержащий номер интерфейса);
- - по опциям **AGENT_REMOTE_ID** (не обязательна, если в пределах NAS'a VLAN уникальны) и **VLAN_ID** (или **AGENT_CIRCUIT_ID**, содержащий номер **VLAN**'а).

При аутентификации во всех случаях происходит проверка пароля, поэтому в некоторых случаях пароль нужно устанавливать на NAS'е, в коде `preprocessAccessRequest` или, при необходимости, отключить проверку пароля для NAS'a:

```

# (1, ) (0) NAS'
radius.password.verification=0

```

Проверку пароля также можно отключить для определенного типа сервиса, указав в его конфигурации:

```

# (1, ) (0) NAS'
serv.radius.password.verification=0

```

При отключенной проверке пароля и MS-CHAPv2 аутентификации биллинг не будет отдавать в ответе Access-Асcept подпись, т.к. для генерации правильной подписи требуется, чтобы пароль у обеих сторон совпадал.

RADIUS-атрибуты

Когда сервис определен, определяется набор атрибутов сессии, последовательным добавлением:

- атрибутов, определенных для реалма;
- атрибутов, определенных для сервиса(логина);
- атрибутов, определенных для опций.

Атрибуты задаются перечислением <имя атрибута>=<значение> через точку с запятой, если в значении атрибута должна присутствовать точка с запятой (т.е. это не разделение двух разных атрибутов), то её нужно указать два раза: ";";".

Атрибуты реалма

Атрибуты реалма определяются в конфигурации устройства-NASa следующим образом:


```
radius.realm.<realm>.attributes=<attributes>
```

Где:

- **<realm>** - реалм;
- **<attributes>** - атрибуты.

Пример:

Пример

```
radius.realm.default.attributes=Acct-Interim-Interval=300;Session-Timeout=604800
```

Есть возможность добавлять динамические атрибуты, с помощью макросов:

```
radius.realm.<realm>.attributes.macros=<attributes>

radius.realm.default.attributes.macros=Interface-Name=$ipParam(ifaceName)
```

Запись атрибутов производится в таком же виде, как и в параметрах `radius.realm.<realm>.attributes`, однако здесь поддерживаются макросы вида `$имя` или `$имя(параметр1,параметр2)`, например:

```
radius.realm.default.attributes.macros=Interface-Name=$ipParam(ifaceName)
```

Поддерживаются следующие макросы:

- **\$ipGate** - значение поля "Роутер" IP-ресурса, из которого выдан IP-адрес;
- **\$ipDns** - значение поля "DNS" IP-ресурса, из которого выдан IP-адрес;
- **\$ipSubnetMask** - значение поля "Маска подсети" IP-ресурса, из которого выдан IP-адрес;
- **\$ipParam(name)** - значение параметра конфигурации IP-ресурса, из которого выдан IP-адрес, аргумент макроса - имя параметра, например `$ipParam(ifaceName)` вернет значение параметра `ifaceName=` конфигурации IP-ресурса;
- **\$param(\$device,name,default)** - значение параметра конфигурации устройства-NAS'a или агентского устройства (коммутатора), указывается так: `$param($device,myParam,myDefault)` или `$param($agentDevice,myParam,myDefault)`;
- **\$requestAttribute(vendor,type)** или **\$requestAttribute(name)** - значение атрибута из запроса, например `$requestAttribute(User-Name)`;
- **\$login** - логин сервиса (поле Логин);
- **\$identifier** - идентификатор сервиса (поле Идентификатор);
- **\$inetServId** - ID сервиса;
- **\$contractId** - ID договора;
- **\$contractTitle** - название договора.

Последние макросы также можно использовать для подмены `User-Name` в аккаунтинге для схем IPoE или для добавления информационного атрибута `Class`, который затем будет передаваться в RADIUS-аккаунтинге:

```
radius.realm.default.attributes.macros=User-Name=$login;Class=IPoE/$contractId/$inetServId
```

Или, если `User-Name` для поиска сервиса мы совсем не используем, можем указать там номер договора для информации или для СОРМ (см. также [Пересылка RADIUS-запросов](#)).

```
radius.realm.default.attributes.macros=User-Name=$contractTitle
```

Атрибуты опций

Полный набор опций сессии определяется объединением опций, указанных в самом сервисе и опций из тарифного плана. Соответствие кодов опций атрибутам определяется в конфигурации устройства-NASa следующим образом:

```
radius.inetOption.<option_id>.attributes=<attributes>
```

Где:

- **<option_id>** - числовой код опции;
- **<attributes>** - RADIUS атрибуты.

Пример конфигурации, где определены атрибуты для реалмов и опций:

```
radius.realm.default.attributes=Service-Name:1=RSE-SVC-EXT;Service-Options:1=1  
  
radius.inetOption.1.attributes=Service-Parameter:1=Rate=100000 Burst=12500000  
radius.inetOption.2.attributes=Service-Parameter:1=Rate=100000 Burst=12500000
```

Для удобства указания однотипных наборов атрибутов, например, для указания атрибутов скорости можно использовать шаблоны. Для этого опции скорости должны находиться в одной родительской опции (которую можно назвать, например, "Скорость"). В конфигурации устройства нужно будет указать:

```
radius.inetOption.<parent_option_id>.template=Service-Parameter:1=Rate=$rate Burst=$burst
```

где **<parent_option_id>** - ID родительской опции ("Скорость").

В конфигурации опций-скоростей необходимо указать необходимые значения `rate=` и `burst=`. Таким образом отпадает необходимость указания практически одинакового набора атрибутов для каждой опции-скорости в конфигурации устройства, достаточно лишь прописать один раз шаблон. Также значения параметров можно прописать в конфигурации родительской опции, для того чтобы использовать как значения по умолчанию (если значение не переопределено в дочерней опции) или просто как самостоятельную переменную шаблона.

Атрибуты сервиса/логина

Атрибуты специфичные для **типа сервиса** можно прописать в конфигурации типа сервиса:

```
serv.radius.realm.default.attributes=
```

Для режима Reject-To-Accept (см. далее) пропишите атрибуты в параметре **serv.radius.disable.attributes**:

```
serv.radius.disable.attributes=
```

При необходимости выдавать специфичные атрибуты (значения) индивидуально для сервиса, в конфигурации типа сервиса укажите:

```
param.myParam.title=
```

Теперь на вкладке "Дополнительно" в редакторе сервиса на договоре можно будет добавить данный параметр, указав его значение. Теперь этот параметр можно использовать при выдаче RADIUS-атрибутов, прописав в конфигурации типа сервиса, в том же параметре **serv.radius.<realm>.default.attributes** атрибут с помощью макроса:

```
serv.radius.realm.default.attributes=My-Attribute=$myParam
```

Выдача IP-адреса

Выдача IP-адреса производится через атрибут Framed-IP-Address из диапазона (адреса), указанного в самом сервисе на договоре (т.е. выдача статического IP-адреса), либо, если адрес в сервисе не указан или занят другой сессией этого же сервиса - из пула, определённого в конфигурации устройства параметром **radius.realm.<realm>.ipCategories=<cat_codes>**, где:

- **<realm>** - реалм;
- **<cat_codes>** - id коды [категорий ресурсов IP-адресов](#) через запятую.

Например, выдача динамических адресов из категории IP-ресурсов с кодом 4:

```
radius.realm.default.ipCategories=4
```

Вы также можете выдавать адреса из разных пулов в зависимости от активной [опции Inet](#), указав опцию в тарифе или статически в сервисе на договоре. Таким образом, можно в зависимости от различных условий выдавать адреса из разных подсетей (для ситуации, когда баланс меньше лимита смотрите ниже [Reject-To-Accept](#)). Благодаря гибкости тарифа можно выдавать адреса из разных пулов в зависимости от баланса, активных тарифных опций, групп договоров, времени, периода и других условий.

Для этого в конфигурации устройства-NAS'а или корневого устройства укажите именованный пул адресов. Пул состоит из имени и перечисления id [категорий ресурсов IP-адресов](#) через запятую: **resource.ip.pool.<name>.ipCategories=<cat_codes>**

```
resource.ip.pool.whitePool.ipCategories=5
```

Далее создайте [опции Inet](#), например, создав опцию-ветку "Пулы IP-адресов" и внутри нее опцию "Белый пул IP-адресов". В конфигурации устройства или типа устройства-NAS'а или же корневого устройства укажите привязку данной опции к пулу IP-адресов, например, если у созданной опции "Белый пул IP-адресов" ID=72:

```
radius.inetOption.72.ipPool=whitePool
```

Теперь, если в сервисе договора будет активна (с помощью тарифа или указана статически в сервисе) опция "Белый пул IP-адресов", то адрес будет выдаваться из пула whitePool, который состоит из IP-адресов категорий, указанных в параметре **resource.ip.pool.whitePool=**, т.е. в данном примере - из категории IP-ресурсов с ID=5. Если же опция будет не активна - то адрес будет выдан из параметра **radius.realm.<realm>.ipCategories=<cat_codes>**, если он указан.

Также вместо указания списка IP-категорий для реалма можно воспользоваться указанием имени пула, т.е. вместо **radius.realm.default.ipCategories=4** указать, например:

```
resource.ip.pool.greyPool.ipCategories=4
radius.realm.default.ipPool=greyPool
```

При использовании схем IPoE возможна ситуация, когда с Cisco/SE приходит подряд несколько Access-Request'ов. Например, абонентский DHCP-клиент шлет сразу несколько DHCP-Discover-пакетов, или в Cisco инициация сессии срабатывает не только на DHCP-Discover, но и на DHCP-Request. По умолчанию для динамической выдачи IP-адресов InetAccess на каждый Access-Request выдает Access-Accept с новым IP-адресом, что может быть проблемой для связки DHCP-клиент+Cisco/SE. В этом случае можно использовать параметр **radius.address.authReserveTimeout**. При выдаче адреса он будет запоминаться по ключу сервис+устройство-NAS+(агентское_устройство)+User-Name+(MAC-address)+(CallingStationId)+(identifier) и в течении указанного времени (в миллисекундах) на очередной Access-Request с таким же ключом будет предлагать тот же адрес.

```
# IP- ( IPoE), - 0 ()
radius.address.authReserveTimeout=3000
```

Выдача статической подсети

Для того, чтобы в Access-Accept биллинг выдавал подсеть в виде атрибутов **Framed-IP-Address** и **Framed-IP-Netmask**, то в типе сервиса нужно указать "Тип адреса": "статическая сеть", а также добавить в конфигурации типа сервиса:

```
# Access-Accept Framed-IP-Address=<net> Framed-IP-Netmask=<mask>
serv.radius.staticSubnet=1
```

Не выдавать адрес из диапазона сервиса

Возможен случай, когда в сервисе или дочернем сервисе нужно указать IP-адрес, диапазон или подсеть, которые будут нужны только для управления. Т.е. задача в том, чтобы из данного сервиса IP-адрес не выдавать. Например, в родительском сервисе мы указываем подсеть, а в дочерних сервисах - разрешенные абоненту IP-адреса из этой подсети (т.е. абонент должен получать адрес из дочерних сервисов, а не из родительского). Для того, чтобы адрес из сервиса не выдавался, в конфигурации типа сервиса нужно указать:

```
# IP-
serv.radius.skipStaticAddress=1
```

Не добавлять Framed-IP-Address в Access-Accept

В некоторых схемах, например, IP Subscriber, не требуется выдача IP-адреса в **Access-Accept**, т.к. IP-адрес у абонента уже есть, и адрес в сервисе указан только для того, чтобы найти абонента по нему при запросе **Access-Request**. В этом случае в конфигурации типа сервиса нужно указать:

```
# Framed-IP-Address Access-Accept
serv.radius.noAddress=1
```

Выдача Framed-Route

Для того, чтобы выдавать в **Access-Accept** абоненту атрибут **Framed-Route** нужно создать тип сервиса, который будет дочерним, в поле "Тип адреса" выбрать значение "статическая сеть", указать в конфигурации типа сервиса:

```
# Framed-Route
serv.radius.framedRoute=1
```

Выдача адреса из Access-Request

Для того, чтобы InetAccess в Access-Accept просто выдавал Framed-IP-Address из запроса (если он там присутствует), в конфигурации устройства нужно указать:

```
# Framed-IP-Address, Access-Request
radius.address.fromRequest=1
```

Если при этом необходимо проверять, подходит ли IP-адрес из Access-Request для диапазона адресов или сети договора (например, IPoE в accel-ppp), то в конфигурации устройства необходимо указать:

```
# Framed-IP-Address, Access-Request,
radius.address.fromRequest.check=1
```

Reject-To-Accept

В случае ошибки авторизации высылается пакет AUTHENTICATION_REJECT с отображением ошибки и её кода в мониторе модуля. [Допустимые коды ошибок данного процессора](#).

Вместо AUTHENTICATION_REJECT может быть выслан AUTHENTICATION_ACCEPT-пакет с дополнительными атрибутами. Перечень кодов ошибок, для которых производится подмена Reject-To-Accept определяется переменной конфигурации NAS'a **radius.disable.accessCodes=<codes>**, где **<codes>** - перечень кодов ошибок авторизации через запятую. Например:

```
radius.disable.accessCodes=1,2,3,4,10,11,12
```

Соответственно, для Reject-To-Accept могут быть собственные диапазоны IP-адресов, например:

```
radius.disable.ipCategories=3
```

Если же нужно выдавать статический IP-адрес такой же, как при обычном доступе (также, как когда не возникает ошибок авторизации), нужно указать режим:

```
# :
# 0 ( ) - radius.disable.ipCategories,
# 1 - , - radius.disable.ipCategories
# 2 - ( )
radius.disable.mode=1
```

Если нужно выдавать динамический IP-адрес такой же, как при обычном доступе - то можно прописать в **radius.disable.ipCategories** те же самые пулы.

При ошибке авторизации, атрибуты выдаются из параметра radius.disable.attributes:

```
# ,
radius.disable.attributes=
# , 1 ( / ), 2 3
#radius.disable.1,2,3.attributes=
#
#radius.disable.attributes.macros=
```

Атрибуты из опций Inet в данном режиме не выдаются.

Для поддержки Reject-To-Accept при ошибке с кодом 1 (Логин не найден), т.е. для предоставления гостевого доступа, необходимо создать договор с балансом меньше лимита и сервисом модуля Inet со статусом закрыт, а id сервиса прописать в конфигурации устройства:

```
radius.disable.servId=<inetServId>
```

Таким образом, сессии с ненайденным сервисом будут привязываться к указанному сервису.

При превышении числа сессий над ограничением, установленным для сервиса, генерируется ошибка авторизации. Данная ошибка также может быть обработана механизмом Reject-To-Accept с выдачей адреса из пула фиктивных адресов.

MAC-адрес и идентификатор

При RADIUS Access-Request можно также использовать MAC-адрес, чтобы авторизовывать запросы только с определенным MAC-адресом. MAC-адрес из Access-Request пакета устанавливается в Обработчике процессора протокола, в методе preprocessAccessRequest: request.setOption(InetRadiusProcessor.MAC_ADDRESS, macAddress). Стандартные обработчики процессора протокола, поставляемые с модулем уже реализуют этот функционал, нужно только прописать в конфигурации устройства или типа устройства:

```
# , MAC-
radius.macAddress.vendor=-1
# , MAC-
radius.macAddress.type=31
# ( ), MAC-. , cisco avpair
#radius.macAddress.prefix=client-mac-address=
```

Или:

```
# , MAC-
radius.macAddress.vendor=9
# , MAC-
radius.macAddress.type=1
# ( ), MAC-. , cisco avpair
radius.macAddress.prefix=client-mac-address=
```

Таким образом, при извлечении MAC-адреса из RADIUS-пакета, он будет сравнен с MAC-адресом из аутентифицированного сервиса при условии, что в этом сервисе заведен MAC-адрес.

Для автоматического привязывания MAC-адреса к сервису в конфигурации модуля/типа сервиса/устройства/типа устройства можно прописать:

```
# MAC-, .
# , , .
# 0 - , 1 - , 2 - , 3 -
# ( MAC- )
serv.macAddress.auto=1
```

Таким образом при первой удачной авторизации MAC-адрес будет привязан к сервису и с другим MAC-адресом клиент уже не сможет авторизоваться.

Аналогично MAC-адресу можно использовать поле Идентификатор сервиса. Например, в качестве идентификатора использовать значение атрибута Calling-Station-Id, устанавливая в Обработчике процессора протокола, в методе preprocessAccessRequest: request.setOption(InetRadiusProcessor.IDENTIFIER, callingStationId).

Для автоматического привязывания идентификатора к сервису в конфигурации модуля/типа сервиса/устройства/типа устройства можно прописать:

```
# , .
# , , .
# 0 - , 1 - , 2 - , 3 -
# ( )
serv.identifier.auto=1
```

Calling-Station-Id

Атрибут **Calling-Station-Id** используется для подстановки в поле сессии "С номера". При отсутствии данного атрибута можно настроить, чтобы обработчик процессора протокола (CoAProtocolHandler, SmartEdgeProtocolHandler, ISGProtocolHandler и другие) при предобработке RADIUS-запроса устанавливал значение в данный атрибут из другого, например, из атрибута Mac-Address Redback (ID вендора 2352, ID типа атрибута 145):

```
# Calling-Station-Id.
# ID RADIUS-
radius.callingStationId.vendor=2352
# RADIUS-
radius.callingStationId.type=145
# RADIUS-, ( )
#radius.callingStationId.prefix=
```

Привязка сервиса к NAS'у или ветке NAS'ов

Для привязки сервиса к устройству - т.е. для того, чтобы клиент мог авторизоваться только на устройстве (NAS'e), привязанному к сервису, или являющимся дочерним по отношению к нему (устройству), необходимо указать в конфигурации устройства/типа устройства/типа сервиса:

```
# ,
# 0 - (NAS'), 1 - ,
# 2 -
serv.device.link=2
```

Действие при получении Access-Request при наличии активной сессии

При подключении абонента может быть ситуация, когда в биллинге сессия еще активна, а клиент на самом деле уже отключился и пытается подключиться заново. Это может произойти при потере связи с NAS'ом (т.е. STOP-пакет не пришел, но connection.close.timeout еще не произошел), или, например, при использовании IPoE с Cisco/Redback (когда абонент подключил другое устройство, а Cisco/Redback по таймауту DHCP-lease еще не поняли, что старое соединение можно закрывать). Если возникает ошибка о превышении количества активных сессий, то по умолчанию биллинг выдает Access-Reject или же срабатывает правило Reject-To-Accept (см. выше). Для изменения обработки такой ситуации можно использовать параметр конфигурации **radius.connection.checkDuplicate**. Он работает в связке с Calling-Station-Id - если происходит попытка авторизации, а количество активных соединений превышено и среди активных соединений есть соединение с **таким же Calling-Station-Id**, то при указании radius.connection.checkDuplicate:

- 0 - не пускаем абонента (по умолчанию);
- 1 - происходит попытка сброса старого соединения (например, отправка PoD-пакета), абонента на этой авторизации не пускаем;
- 2 - происходит попытка сброса старого соединения, его завершение в биллинге, абонента на этой авторизации не пускаем, должно пустить при следующей попытке, т.к. старой сессии уже не будет;
- 3 - происходит завершение старого соединения в биллинге, абонента на этой авторизации не пускаем, должно пустить при следующей попытке, т.к. старой сессии уже не будет;
- 4 - происходит попытка сброса старого соединения, затем через 5 секунд завершение в биллинге, абонента на этой авторизации не пускаем;
- 5 - попытка сброса старого соединения в биллинге, завершение и пускаем немедленно (т.е. игнорируется ошибка);
- 6 - завершение старого соединения в биллинге, пускаем абонента немедленно;
- 7 - попытка в течении 5 секунд сбросить соединение, затем закрытие сессии в биллинге с ожиданием полного выполнения закрытия (т.е. IP-адрес станет свободным), пускаем абонента
- 8 - попытка сброса и сразу закрытие в биллинге с ожиданием полного выполнения закрытия (т.е. IP-адрес станет свободным), пускаем абонента;
- 9 - закрытие старого соединения в биллинге с ожиданием полного выполнения закрытия (т.е. IP-адрес станет свободным), пускаем абонента.

Для более быстрого подключения абонента при использовании и динамическим адресом рекомендуем использовать 4, 5, 6. Для статических адресов (чтобы абонент мог получить свой адрес) - 8,9.

Для обработки ситуации, когда количество активных соединений превышено, но соединения с таким Calling-Station-Id не найдено, и нужно сбросить просто самое старое соединение, нужно указать цифру во втором разряде, например: 88.

Для сброса соединения **с таким же Calling-Station-Id**, даже если количество соединений не превышено, нужно указать цифру в третьем разряде, например, 808, 909, 888 или 818.

Этот параметр можно указать для типа сервиса, в конфигурации типа сервиса: **serv.radius.connection.checkDuplicate**.

В некоторых IPoE-схемах для обработки этого события требуется использовать не Calling-Station-Id, а User-Name. Для этого нужно указать параметр **radius.connection.checkDuplicate.type=1**. В этом режиме будет проверяться User-Name из Access-Request с полем username сессии. Этот параметр можно указать для типа сервиса, в конфигурации типа сервиса: **serv.radius.connection.checkDuplicate.type**.

Т.к. параметр 8 и 9 (закрытие в биллинге с ожиданием полного выполнения закрытия, т.е. IP-адрес станет свободным) ожидает полного завершения соединения в биллинге, а необходимо такое ожидание только для некоторых схем, например, статических адресов, то рекомендуется задавать этот параметр в типе сервиса. Например, в конфигурации NAS'a **radius.connection.checkDuplicate=5**, в конфигурации типа сервиса для статических адресов **radius.connection.checkDuplicate=8**.

Без атрибута Calling-Station-Id в Access-Request-пакете данная функция работать не будет. При невозможности присылать Calling-Station-Id в Access-Request можно настроить подмену в обработчике процессора протокола (в предобработке RADIUS-запросов).

Блокировка частых Access-Request -> Access-Reject

Спам-запросы на авторизацию порождаются, обычно, забытым оборудованием, самостоятельно предпринимающем попытки авторизации на RADIUS. Постоянные обращения при таких запросах могут существенно повышать загрузку. Признаком спам-запроса являются несколько Reject-ответов на одинаковые запросы авторизации в течение определённого интервала времени. В этом случае подобные запросы попадают в спам-кэш на определённое время и по ним автоматически выдаётся Reject-ответ без полных проверок, производимых при авторизации. Для включения антиспам-системы в **конфигурации модуля** добавьте (требуется перезапуск InetAccess):

```
antispam.key.attributes=<key_attributes>
antispam.reject.count=<reject_count>
antispam.reject.per.time=<per_time>
antispam.ban.time=<ban_time>
```

Где:

- **<key_attributes>** - RADIUS-атрибуты, идентифицирующие запрос, через запятую;
- **<reject_count>** - количество Reject, возвращённых на запрос с одинаковыми идентифицирующими атрибутами;
- **<per_time>** - за какое количество секунд выдано указанное в **<reject_count>** количество Reject-ответов;
- **<ban_time>** - время в секундах, на которое данный запрос попадает в спам-базу.

Например, запрос идентифицируется атрибутами User-Name и Calling-Station-Id. При десяти Reject-ответах в течении минуты запрос попадает в спам-базу на 15 минут:

```
antispam.key.attributes=User-Name,Calling-Station-Id
antispam.reject.count=10
antispam.reject.per.time=60
antispam.ban.time=900
```

Шифрование и атрибуты MS-CHAPv2

```
# 1 ( ) - Access-Accept MPPE-128 MS_mppe_encryption_types
# ( 128 ) MS_mppe_encryption_policy=1 ( ) RFC2548
add.mppe.enc.types.and.policy=1
```

Посервисный аккаунтинг

Посервисный аккаунтинг

Для поддержки посервисного аккаунтинга (например, при использовании Cisco ISG или Redback CLIPS) в обработчике процессора протокола необходимо указывать AcctSessionId родительского соединения (аккаунтинга) и имя сервиса (ServiceName), по которому пришел аккаунтинг-пакет: `request.setOption(InetRadiusProcessor.PARENT_ACCT_SESSION_ID, parentAcctSessionId);` и `request.setOption(InetRadiusProcessor.SERVICE_NAME, serviceName);`

Пример для Redback SmartEdge 100:

Код

```
@Override
public void preprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
    throws Exception
{
    int acctStatusType = request.getIntAttribute( -1, RadiusDictionary.Acct_Status_Type, -1 );

    switch( acctStatusType )
    {
        //
        case 101:
        case 102:
        case 103:
        {
            // id
            final String parentAcctSessionId = request.getStringAttribute( -1,
parentAcctSessionIdType, null );
            // ,
            final String serviceName = request.getStringAttribute( radiusVendor,
serviceNameType, null );

            // Acct-Status-Type,
            request.setIntAttribute( -1, RadiusDictionary.Acct_Status_Type, acctStatusType
- 100 );

            // id
            request.setOption( InetRadiusProcessor.PARENT_ACCT_SESSION_ID,
parentAcctSessionId );

            //
            request.setOption( InetRadiusProcessor.SERVICE_NAME, serviceName );
        }
        break;

        default:
        {
        }
        break;
    }
}
```

Тогда Accounting считает данное соединение как дочернее родительскому, а в поле username соединения (в базе) устанавливается ServiceName.

При обработке трафиков используется ServiceName - он может выступать в качестве фильтра правила привязки трафиков: если указано ServiceName, то данное правило будет обрабатывать только для сервисной сессии с таким ServiceName, и наоборот, если поле ServiceName в правиле пусто, то данное правило не будет обрабатывать ни для какой сервисной сессии.

В поле ServiceName можно указать несколько сервисов через запятую (RSE-SVC-EXT, RSE-SVC-EXT1024) или же указать REGEXP внутри двух слешей (^RSE-SVC-EXT.*\$).

Например, если у вас внешний трафик идет посервисным аккаунтингом с ServiceName=RSE-SVC-EXT, то нужно назначить такую привязку:

Типы трафика Привязка типов трафика

Редактор

Название:

radius



Тип	Период	Тип трафика	Комментарий
RADIUS	...-...	Входящий	
RADIUS	...-...	Исходящий	

Редактор правила

Тип

RADIUS с по

RADIUS

ServiceName:

RSE-SVC-EXT

Код вендора: Код атрибута: Префикс:

-2 1

Коллектор

Направление: Диапазон адресов:

любое -

DiffServ/ToS: Диапазон портов:

-

Источник:

любой

Интерфейс:

любой

Тип трафика

Входящий

Комментарий:

Пример конфигурации устройства-NAS'a

Конфигурация

```
# secret RADIUS- ( - Community/secret )
#radius.secret=<secret >

# PoD CoA- ( - , /)
#radius.port=< >

# Access-Accept .
# H, Reject-To-Accept Start-
connection.start.fromAccept=1
# , Start- Accounting-. ,
# 1 ( ) - ,
# 2 - Accounting , Update/Stop , connection.close.timeout , ,
# connection.close.timeout, ,
# 0 - Start- .
#connection.start.fromUpdate=1
# Update- IP- (Framed-Ip-Address).
# Update- (0).
#connection.start.fromUpdate.ignoreFramedIpLack=0

# suspended
connection.suspend.timeout=900
# ( connection.suspend.timeout)
connection.close.timeout=900
# , 0 ( ) - ,
# 1 - NAS' ( connectionClose )
#connection.close.timeout.forceClose=1

# , default (default - )
radius.realm.default.attributes=

# , Access-Accept disable (rejectToAccept)
radius.disable.accessCodes=
# Access-Accept disable:
# 0 ( ) - radius.disable.ipCategories, 1 - , ( )
#radius.disable.mode=0

# , Access-Accept disable
radius.disable.attributes=

# Id , , ( : 1, ).
# , radius.disable.accessCodes 1
#radius.disable.servId=

# , DISABLE (.. )
#radius.disable.pattern.attributes=

# , MAC-
#radius.macAddress.vendor=9
# , MAC-
#radius.macAddress.type=1
# ( ), MAC-. , cisco avpair
#radius.macAddress.prefix=client-mac-address=

# : 0 ( ) - , 1 - (
# AGENT_REMOTE_ID AGENT_CIRCUIT_ID INTERFACE_ID), 2 - VLAN (
# AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID), 4 - VLAN
# ( AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID),
# 5 - MAC- ( MAC_ADDRESS), 6 - MAC-
# ( MAC_ADDRESS).
#radius.servSearchMode=0
# : 0 - , 1 ( ) - .
#radius.password.verification=1
```

```

#      Access-Request.      , NAS ( ) ,
# Stop-
# callingStationId      : 1 -      (connectionClose),
# 2 -      (connectionClose)      ,      , 3 -      .
#radius.connection.checkDuplicate=0

#      User-Name.      - (1).
#      , CoA PoD NAS' User-Name.
radius.username.removeDomain=1
#      User-Name      .      - (0).
#      , CoA PoD NAS' User-Name.
#radius.username.removeWhitespace=0
#      .      - (0).
#radius.username.ignoreCase=0

# (1, ) (0) NAS'
#radius.password.verification=1

#      RADIUS-
#radius.accessError.infoPattern=LOGIN:$User-Name

#
#      AccessAccept
radius.inetOption.1.attributes=
radius.inetOption.2.attributes=
radius.inetOption.3.attributes=

#
#      ,
#sa.error.pause=60
#
#sa.batch.size=20
# ( ) ( )
#sa.batch.wait=5
# ( )
#sa.batch.pause=0
# ( ) disconnect.
#sa.batch.waitNext=5

#
#      CoA ( - , )
#sa.radius.option.attributesPrefix=nas.radius.inetOption.
#sa.radius.connection.attributes=NAS-Port, Acct-Session-ID, User-Name, Framed-IP-Address, NAS-IP-Address, NAS-
Identifier
# CoA- ( sa.radius.connection.withoutBreak=1)
#sa.radius.disable.attributes={@radius.disable.attributes}
#      , CoA
#sa.radius.coa.attributes=
#      CoA- ( default - radius.realm.default.attributes)
#sa.radius.realm.addAttributes=0
#      , PoD
#sa.radius.pod.attributes=

```

Пересылка RADIUS-запросов

В BGBilling есть возможность осуществлять пересылку Radius-пакетов (как запросов, так и ответов) во внешние системы. Это бывает необходимо для реализации зеркалирования radius-трафика на сервера СОРМ. Для каждого типа пакета можно задать отдельные настройки пересылки. В конфигурации устройства или типа устройства можете указать следующие опции (также их можно указать в конфигурации корневого устройства):

```
#,      accounting- (start, stop, update)
#      radius.forward.host=<host>
radius.forward.accounting.request.host=<host>
#,      accounting- (start, stop, update). - 1813
#      radius.forward.port=<port>
radius.forward.accounting.request.port=<port>
# ,      accounting- (start, stop, update). ,
radius.forward.accounting.request.source.port=<source_port>

#,      accounting-
radius.forward.accounting.response.host=<host>
#,      accounting-
radius.forward.accounting.response.port=<port>
# ,      accounting-. ,
radius.forward.accounting.response.source.port=<source_port>

#,      access-
radius.forward.access.request.host=<host>
#,      access-
radius.forward.access.request.port=<port>
# ,      access-. ,
radius.forward.access.request.source.port=<source_port>

#,      access-
#      radius.answer.forward.host=<host>
radius.forward.access.response.host=<host>
#,      access-
#      radius.answer.forward.port=<port>
radius.forward.access.response.port=<port>
# ,      access-. ,
radius.forward.access.response.source.port=<source_port>

# RADIUS-, , NAS
radius.forward.secret=<secret>
#      User-Name accounting-, -
radius.forward.user.name.prefix=<prefix>
#      User-Name accounting-, -
radius.forward.user.name.suffix=<suffix>
#      User-Name. :
# 1) $contractTitle - User-Name
# 2) $contractId - User-Name id contract
# 3) $servId - id Inet.
# 4) $contractId_$servId - id id ( , )
#      radius.forward.after.process=1, ..
#radius.forward.user.name.pattern=$contractTitle
#      radius- . 500. .
#radius.forward.maxQueueSize=500
```

Обратите внимание на следующий момент. Все запросы и ответы будут иметь один и тот же **srcIp:srcPort** и **dstIp:dstPort**. И получается, что биллинг как бы является автором этих запросов и ответов, что может привести к невозможности сопоставления друг другу запросов и ответов на стороне СОРМа. Возможно, что в данном случае придется осуществлять подмену адресов и портов в ином месте (с помощью iptables или на коммутаторе или иным способом).

Данные параметры (если необходимы) нужно указывать в конфигурации модуля:

```
# , ( - )
#radius.forward.after.preprocess=0
# ,
#radius.forward.after.process=0
# , access- .
#radius.forward.access.request=0
```

Где:

- **<host>** - хост, на который будут ретранслироваться пакеты, единственный обязательный параметр для включения функционала;
- **<port>** - порт, на который будут ретранслироваться пакеты, по умолчанию равен 1813;
- **<secret>** - RADIUS-секрет, которым будет подписан пакет, по умолчанию берётся секрет NASA;
- **<prefix>** - префикс к значению атрибута User-Name из аккаунтинг пакета, если не указан - префикс не добавляется;
- **<suffix>** - суффикс к значению атрибута User-Name из аккаунтинг пакет, если не указан - суффикс не добавляется.

Суффикс и префикс могут быть полезны при необходимости ретранслирования пакетов с нескольких InetAccounting с пересекающимися логинами на единый сервер.

Например:

```
radius.forward.host=bitel.ru
radius.forward.port=4444
radius.forward.secret=4343
radius.forward.user.name.prefix=test_
radius.forward.user.name.suffix=_forv
```

Пример с подстановкой номера договора в атрибут User-Name, в конфигурации модуля:

```
radius.forward.after.process=1
```

В конфигурации типа устройства-NAS'a:

```
radius.forward.host=bitel.ru
radius.forward.port=4444
radius.forward.secret=4343
radius.forward.user.name.prefix=test_
radius.forward.user.name.suffix=_forv
radius.forward.user.name.pattern=$contractTitle
```

Слушатель InetDhcpListener

Предназначен для обработки DHCP-пакетов для схем DHCP.82, ISG/CLIPS+DHCP.82. Используется в InetAccess, добавляется в контейнер следующим образом:

Код

```
<context name="dhcp">
  <!-- C dhcp- -->
  <bean name="dhcpProcessor" class="ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor"/>

  <scheduledExecutorService name="hrlydtlggr" corePoolSize="1" />

  <!-- C dataLogger, dhcp- -->
  <bean name="dhcpDataLogger" class="ru.bitel.bgbilling.modules.inet.dhcp.DhcpHourlyDataLogger">
    <param name="scheduledExecutor">hrlydtlggr</param>
  </bean>

  <!-- C dhcp- dataLogger -->
  <bean name="dhcpListener" class="ru.bitel.bgbilling.kernel.network.dhcp.DhcpListener">
    <constructor>
      <!-- (), . - -->
      <param name="host" value="" />
      <!-- , -->
      <param name="port" value="67" />
      <!-- -->
      <param name="byteBufferCapacity">512 * 1024</param>
      <!-- - -->
      <param name="threadCount">10</param>
      <!-- -->
      <param name="maxQueueSize">200</param>
      <!-- -->
      <param name="processor">dhcpProcessor</param>
      <!-- dataLogger -->
      <param name="dataLogger"></param>
    </constructor>
  </bean>
</context>
```

Параметры:

- **host** - IP-адрес, на котором слушатель, пустое значение - прослушивать любой IP-адрес;
- **port** - прослушиваемый UDP порт;
- **byteBufferCapacity** - размер буфера для приёма пакетов;
- **threadCount** - число потоков-обработчиков пакетов, рекомендуемые значения 10-50, не рекомендуется указывать более 100;
- **maxQueueSize** - максимальный размер очереди пакетов, при превышении размера очереди пакеты начинают отбрасываться и высылается аларм. Рекомендуемые значения 200-1000. Следует учитывать, что если по какой-то причине сервер не успевает обрабатывать пакеты, очередь растёт, то какие-то пакеты из очереди могут быть обработаны с опозданием и DHCP-клиент уже выслал повторный запрос, который опять попадет в очередь и опять может быть обработан с опозданием. Поэтому большое значение вместо распределения нагрузки может вызвать ее увеличение;
- **dataLogger** - объект, осуществляющий запись бинарных логов, по умолчанию это объект типа **ru.bitel.bgbilling.modules.inet.dhcp.DhcpHourlyDataLogger** (сохранение бинарных логов с разбивкой по источникам и часам);
- **processor** - процессор, реализующий логику обработки пакетов.

Процессор InetDhcpProcessor

Предназначен для выдачи IP-адресов по протоколу DHCP с опцией 82. Опцию в запрос должен подставить коммутатор, пропускающий запрос, далее запрос обязательно должен переслать DHCP-Relay. Идентификация сервисов осуществляется на основании полей **circuitId** и **remoteId**, но также могут быть настроены любые другие поля, определяющие порт коммутатора клиента.

Загружает список устройств-коммутаторов, начиная с корневого узла. Типы устройств-коммутаторов определяются в переменной конфигурации корневого узла **dhcp.relay.deviceTypeIds** через запятую. Также загружаются привязанные к коммутаторам сервисы договоров.

Когда приходит DHCP-запрос, из него извлекается поле **giaddr (Relay-IP)**. Осуществляется поиск устройства-релея сначала по совпадению этого поля с адресом устройства в биллинге. Затем, если поиск был отрицательным - осуществляется поиск по совпадению IP-адреса, с которого пришёл DHCP-запрос с IP-адресом устройства.

Для обработки DHCP-RENEW-запросов без опции DHCP Option 82 в конфигурации корневого устройства необходимо указать:

```
# RENEW- Option 82 ( InetAccess)
# 0 - ( ), 1 - ( )
dhcp.renew=1
```

Если по каким-то причинам клиентские устройства (например, NetGear JWNR2000) в DHCP-REQUEST посылают xid, отличный от DHCP-DISCOVER, можно убрать привязку к xid-запросам, прописав в конфигурации корневого устройства:

```
# xid DHCP- ( InetAccess)
# 0 - ., 1 ( ) - .
dhcp.xid=0
```

В случае, если необходимо отвечать на INFO-запросы, то в конфигурации нужно указать:

```
# DHCP-INFO-
# 0 - . ( ), 1 - .
dhcp.offer.infoDiscover=1
```

Опция 82

Смотри также [Настройка разбора опции DHCP Option 82](#).

Далее алгоритм работы определяется переменными конфигурации найденного устройства-релея. Следующие параметры определяют, какие опции извлекаются для идентификации устройства-коммутатора клиента и непосредственно клиента по порту или VLAN.

```
# , , 0 - , 2 - 2 (+) DHCP-.
# position agentRemoteId, vlanId, interfaceId -
#dhcp.option82.removeHeader=0
dhcp.option82.removeHeader=2

# agentRemoteId
# agentRemoteId: 0 ( ) - , 1 -
#dhcp.option82.agentRemoteId.type=0
# 82, ,
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.position=0
dhcp.option82.agentRemoteId.length=6

# 82, VLAN,
dhcp.option82.vlanId.code=1
dhcp.option82.vlanId.position=0
dhcp.option82.vlanId.length=2
# 82, ,
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.position=3
dhcp.option82.interfaceId.length=1
```

Параметры, описанные выше подойдут для обработки такого пакета:

```

Message type: BOOT_REQUEST
Dhcp message type: DHCP Discover{1}
htype: 1, hlen: 6, hops: 1
xid: 1067065417, secs: 0, flags: 0
Client IP: 0.0.0.0
Your IP: 0.0.0.0
Server IP: 0.0.0.0
Relay IP: 109.233.170.1
Client MAC: {00804840A46F}
Host name{12}={support-desktop}
Parameter request list{55}={1, 28, 2, 3, 15, 6, 119, 12, 44, 47, 26, 121, 42}
Agent information{82}=
  sub{1}={000403420101}
  sub{2}={00060012CF539F5E}

```

В данном случае будет проигнорирован заголовок: sub{1}={03420101}, sub{2}={0012CF539F5E} и извлечены значения agentRemoteId=0012CF539F5E (MAC-адрес в виде набора байтов), VLAN=0x0342, interfaceId=0x01.

От некоторых коммутаторов значения могут приходить в строковом виде, например: sub{2}={3137322E31362E39392E33363A313031} (на самом деле, это обычная строка в шестнадцатичном представлении), для обработки этой опции нужен такой конфиг:

```

# , , 0 - , 2 - 2 (+) DHCP-.
# position agentRemoteId, vlanId, interfaceId -
dhcp.option82.removeHeader=0

# agentRemoteId
# agentRemoteId: 0 ( ) - , 1 -
#dhcp.option82.agentRemoteId.type=1
# 82, ,
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.position=0
dhcp.option82.agentRemoteId.length=0

# vlanId interfaceId ,
# , .
# 82, VLAN,
#dhcp.option82.vlanId.code=1
#dhcp.option82.vlanId.position=0
#dhcp.option82.vlanId.length=2
# 82, ,
#dhcp.option82.interfaceId.code=1
#dhcp.option82.interfaceId.position=3
#dhcp.option82.interfaceId.length=1

```

При необходимости, данные параметры можно установить вручную в обработчике процессора протокола, в методе **preprocessDhcpRequest** (аналогично [InetRadiusProcessor](#)).

```

final DhcpOption agentRemoteId = request.getSubOption( option82RemoteIdCode );
if( agentRemoteId != null )
{
  byte[] value = new byte[option82RemoteIdLength];
  System.arraycopy( agentRemoteId.value, option82RemoteIdPosition, value, 0, option82RemoteIdLength );
  request.setOption( InetDhcpProcessor.AGENT_REMOTE_ID, value );
}

final DhcpOption vlanId = request.getSubOption( option82VlanIdCode );
if( vlanId != null )
{
  int vlan = InetUtils.parseInt( vlanId.value, option82VlanIdPosition, option82VlanIdLength );
  request.setOption( InetDhcpProcessor.VLAN_ID, vlan );
}

```

Пример обработки для строкового представления (извлечение agentRemoteId не обязательно, если указано dhcp.option82.agentRemoteId.type=1):


```

@Override
public void preprocessDhcpRequest( DhcpPacket request, DhcpPacket response )
    throws Exception
{
    DhcpOption circuitId = request.getSubOption( (byte)1 );
    DhcpOption remoteId = request.getSubOption( (byte)2 );

    request.setOption( InetDhcpProcessor.AGENT_REMOTE_ID, new String( remoteId.value, "UTF-8" ) );
    request.setOption( InetDhcpProcessor.INTERFACE_ID, new String( circuitId.value, "UTF-8" ) );
}

```

При необходимости, для релея можно отключить проверку на наличие DHCP Option 82 в пакете. В этом случае поиск абонента должен осуществляться без их использования, например, по MAC-адресу.

```

# DHCP Option 82 ( RENEW-),
# 0 - , 1 ( ) - .
dhcp.option82.required=0

```

Поиск абонента

По описанным выше значениям AGENT_REMOTE_ID, INTERFACE_ID и VLAN, которые будут извлечены из пакета, происходит поиск устройства и сервиса. Конфигурация поиска устройства и сервиса на устройстве:

```

dhcp.deviceSearchMode=<deviceSearchMode>
dhcp.servSearchMode=<servSearchMode>
#dhcp.servSearchMode=<servSearchMode>-<subServSearchMode>

```

Идентификация коммутатора, расположенного под релеем и сервиса на коммутаторе, может производиться в нескольких режимах. Режимы определяется параметром конфигурации устройства-релея.

<deviceSearchMode> может принимать значения:

- **0 (рекомендуется)** - по giaddr или IP-адресу источника идет поиск устройства, далее у этого устройства вызывается преобработка preprocessDhcpRequest (где можно при необходимости извлечь и установить AGENT_REMOTE_ID, а также INTERFACE_ID или VLAN_ID), далее по установленному AGENT_REMOTE_ID или, если AGENT_REMOTE_ID не установлен - по конфигурации dhcp.option82.agentRemoteId.x agentRemoteId извлекается из пакета и идет поиск агентского устройства по совпадению **идентификатора** устройства, далее у агентского устройства, если таковое найдено вызывается preprocessDhcpRequest (где можно при необходимости извлечь и установить INTERFACE_ID или VLAN_ID). Здесь запоминаются оба устройства, как deviceId и agentDeviceId;
- **1** - по giaddr или IP-адресу источника идет поиск устройства, по его конфигурации идет извлечение agentRemoteId, далее по agentRemoteId идет поиск агентского устройства. Здесь запоминается последнее найденное устройство как deviceId, agentDeviceId для биллинга будет 0;
- **2** - по giaddr или IP-адресу источника идет поиск устройства, найденное устройство будет запомнено как deviceId, agentDeviceId для биллинга будет 0.

Поиск сервиса происходит на агентском устройстве (agentDeviceId), если оно найдено, иначе на устройстве (deviceId).

<servSearchMode> может принимать значения:

- **0** - поиск по логину на всех устройствах (извлечение значения логина должно быть настроено в конфигурации или произведено в преобработке);
- **1** - поиск по интерфейсу на (найденном) устройстве;
- **2** - поиск по VLAN'у на устройстве;
- **3** - поиск на устройстве по интерфейсу и MAC-адресу;
- **4** - поиск по VLAN'у на устройстве и его дочерних устройствах;
- **5** - поиск по MAC-адресу на устройстве;
- **6** - поиск по MAC-адресу на устройстве и дочерних устройствах;
- **10** - поиск по MAC-адресу на всех устройствах;
- **11** - поиск по VLAN и MAC-адресу на устройстве, а также на его потомках и его предках;
- **12** - поиск по VLAN и порту на устройстве;
- **20** - поиск по логину аналогично 0, но только на устройстве и его дочерних устройствах.

После поиска сервиса можно дополнительно использовать поиск дочернего устройства (как элемент дополнительной авторизации).

<subServSearchMode> может принимать значения:

- 0 или отсутствует - нет поиска дочернего сервиса
- 1 - поиск дочернего сервиса по MAC-адресу, если такого дочернего сервиса нет - ошибка авторизации;
- 2 - поиск дочернего сервиса по MAC-адресу, если такого дочернего сервиса нет - сессия будет привязана к родительскому сервису.

Процесс DHCP-авторизации состоит из двух запросов: DISCOVER и REQUEST. В первом запросе клиент запрашивает IP-адреса, какие ему могут предложить DHCP сервера. Во втором просит закрепить за ним конкретный IP-адрес. На DHCP-сервер биллинга попадают запросы с опцией 82, которая позволяет идентифицировать клиента. После идентификации клиента ему выдаётся IP-адрес. Идентификатором сессии при DHCP.82 авторизации выступает MAC-адрес клиента. Допускается одновременная инициализация нескольких сессий за одним портом коммутатора.

Выдача IP-адреса

Адрес сессии выделяется либо из диапазона, указанного в самом сервисе, либо, если он исчерпан или не указан - из пула, определённого в конфигурации устройства. Пул адресов устройства определяется параметром конфигурации `dhcp.ipCategories=<cat_codes>`, где `<cat_codes>` - id коды [категорий ресурсов IP адресов](#) через запятую. Например:

```
dhcp.ipCategories=4
```

При ошибке авторизации DISCOVER-запросы будут игнорироваться, а на все REQUEST-запросы будет высылаться ответ DHCP_NAK. Для предотвращения нагрузки на DHCP-сервер постоянной обработкой запросов возможно определение пула фиктивных адресов, выдаваемых при ошибках авторизации. Пул определяется переменной конфигурации устройства `dhcp.disable.ipCategories=<cat_codes>`, где `<cat_codes>` - id коды [категорий ресурсов IP адресов](#) через запятую. Например:

```
dhcp.disable.ipCategories=3,4
```

При превышении количества сессий сервиса над ограничением в его свойствах при включенном `pool.erro` выдаются несколько адресов из этого пула, после чего DHCP-запросы игнорируются. Это сделано для невозможности исчерпания пула фиктивных адресов отправкой большого количества DHCP-запросов с разными MAC-адресами. Количество сессий сверх ограничения, для которых могут быть выданы фиктивные адреса задаётся переменной конфигурации `dhcp.additionalUnauthorizedSessionCount`.

Также адреса для динамической выдачи можно указать с помощью [именованных пулов IP-адресов](#):

```
dhcp.ipPool=myWhitePool
dhcp.disable.ipPool=myGreyPool
```

Для выдачи адреса, в случае, если сервис не был найден (например, для предоставления гостевого доступа), необходимо создать договор с балансом меньше лимита и сервисом модуля `Inet` со статусом закрыт, а ID сервиса прописать в конфигурации устройства:

```
# ID
dhcp.disable.servId=1
```

Таким образом, сессии с ненайденным сервисом будут привязываться к указанному сервису.

Эту опцию можно использовать в схемах, когда клиента подключают изначально в какой-то новый порт, он получает фиктивный адрес, для него создается сессия на специальном сервисе. Далее он заходит в личный кабинет и через дополнительные действия создает сервис, скрипт находит его сессию на фиктивном договоре (например, по IP), берет с нее информацию (порт, MAC-адрес и т.п.) и создает уже обычный сервис на договоре этого абонента.

Если необходимо, чтобы адрес выдавался независимо от баланса/статуса/состояния, т.е. всегда, как при положительном балансе и открытом статусе договора, нужно указать параметр `dhcp.disable.mode=1` или `2`. При значении `1` `InetAccess` будет выдавать адрес всегда так, как если авторизация прошла успешно. Однако при необходимости переключить сессию из состояния отключена в подключена или наоборот будет выдан NAK, сессия завершится и создастся новая, при этом вызывая в обработчике активации сервисов `onAccountingStop` и `onAccountingStart`. При значении `2` при необходимости переключить состояние сессия не завершается (но `connectionModify` вызывается во всех трех случаях).

```
# . 0 ( ) - dhcp.disable.*,
# 1 - ( NAK, onAccountingStop, ACK, onAccountingStart),
# 2 - ( ).
dhcp.disable.mode=0
```

В случае, если состояние сессии изменяется с помощью обработчика активации сервисов или указан параметр `sa.connection.stateModify=1` (меняющий состояние сессии, даже если нет обработчика активации сервисов), то при последующем получении DHCP-запроса состояние сессии будет уже совпадать с необходимым и будет выдан ACK даже если `dhcp.disable.mode=0`/1. Т.е. в этом случае подразумевается, что доступ абоненту уже был переключен и нет необходимости производить переполучение адреса.

В случае, если в сервисе или дочернем сервисе необходимо указать IP-адрес для управления, но при этом адрес из этого сервиса не должен выдаваться по DHCP, то в конфигурации типа сервиса следует указать:

```
# DHCP
serv.dhcp.noAddress=1
```

Выдача IP-адреса в зависимости от опций Inet

При необходимости выдачи IP-адресов из разных пулов в зависимости от опций Inet, необходимо описать [именованный пул IP-адресов](#). и указать данный пул в конфигурации опции Inet:

```
# IP-
dhcp.ipPool=myWhitePool
```

Или в конфигурации устройства:

```
# с ID 5 IP-
dhcp.inetOption.5.ipPool=myWhitePool
```

Выдача DHCP-опций

Помимо IP-адреса в ответе DHCP-запроса могут передаваться различные опции. Их можно указать как в IP-ресурсе, так и в конфигурации устройства. При указании в IP-ресурсе полей Router, Маска и DNS, значения, при выдаче адреса из этого ресурса соответственно попадут в опции gate, subnetMask и dns. Также в конфигурации IP-ресурса можно указать дополнительные опции с помощью параметров **dhcp.option.<option_name>=<option_value>**.

Задать опции в конфигурации устройства можно с помощью переменных конфигурации **dhcp.option.<option_name>=<option_value>** и **dhcp.net.option.<net>.<mask>.<option_name>=<option_value>**, где:

- **<net>** - сеть, для которой выдаётся параметр;
- **<mask>** - маска сети;
- **<option_name>** - название опции;
- **<option_value>** - значение опции.

Первый параметр устанавливает опции безусловно, второй - в зависимости от выданного IP-адреса.

Возможные значения названий опций и их значений перечислены в таблице.

Название опции	Значение в виде	В DHCP пакете
leaseTime	Число в секундах.	Опция 51, срок аренды IP-адреса
timeOffset	Число в секундах.	Опция 2.
gate	Строка с IP-адресом в виде NNN.NNN.NNN.NNN.	Опция 3, маршрутизатор
serverIdentifier	Строка с IP-адресом в виде NNN.NNN.NNN.NNN.	Опция 54, идентификатор DHCP-сервера
dns	Строка с одним или несколькими адресами вида NNN.NNN.NNN.NNN, разделённых запятой.	Опция 6, DNS-сервера
domainName	Строка.	Опция 15, домен
subnetMask	Строка с IP-адресом в виде NNN.NNN.NNN.NNN.	Опция 1, маска подсети
renewalTime	Число в секундах.	Опция 58, время, после которого DHCP-клиент должен перейти в RENEW
rebindingTime	Число в секундах.	Опция 59, время, после которого DHCP-клиент должен перейти в REBIND

Также можно добавить опцию с помощью ее кода/типа и указания значения в формате HEX, например, dhcp.option.43=68656c6c66.

Для поддержки прямых RENEW запросов (т.е. когда RENEW запрос идет напрямую от абонента, не проходя через relay), в конфигурации нужно указать **dhcp.renew=1** (см. [Получение IP-адреса DHCP-клиентом](#)). При этом для таких запросов можно указать специфичный набор опций, как **dhcp.renew.option.<option_name>**.

Пример конфигурации:

```
dhcp.option.serverIdentifier=0.0.0.0
dhcp.option.leaseTime=60
#
#dhcp.option.renewalTime=
#dhcp.option.rebindingTime=
#dhcp.inetOption.1.leaseTime=120
#
dhcp.net.option.193.106.88.0:255.255.255.0.gate=193.106.88.1
dhcp.net.option.193.106.88.0:255.255.255.0.dns=194.165.18.6
#
dhcp.net.option.172.16.24.0:255.255.255.0.gate=172.16.24.1
dhcp.net.option.172.16.24.0:255.255.255.0.dns=194.165.18.6
```

Блокировка бесконечных DHCP-DISCOVER

Возможна ситуация, когда DHCP-клиент в роутере абонента повисает и начинает слать бесконечные DISCOVER. Чтобы постоянно не резервировать IP-адреса из пула предусмотрена система блокировки. Например, по умолчанию, если за две минуты пришло 10 DISCOVER-пакетов, то блокировать на полчаса. Рекомендуется менять только параметры **dhcp.ban.discoverCount** и **dhcp.ban.timeout**. Конфигурацию можно указывать в устройстве и типе устройства:

```
# OFFER ( 60 )
dhcp.offer.timeout=25
# , - DISCOVER
dhcp.ban.checkTimeout=120
# - DISCOVER,
dhcp.ban.discoverCount=10
# , DISCOVER-
dhcp.ban.timeout=1800
```

При необходимости можно сбросить бан с помощью командной строки:

```
./access.sh dhcpBanRemove
```

Действие при получении DHCP-Discover при наличии активной сессии

При работе по протоколу DHCP нельзя достоверно узнать, когда закончилась DHCP-сессия (доверять DHCP-Release мы не можем), поэтому в биллинге такая сессия по умолчанию завершается по таймауту `connection.close.timeout`, который выставляется больше, чем время выданное lease.

При смене абонентом устройства идет новый запрос DHCP-Discover, но может быть ситуация, когда старая сессия еще не закрыта по таймауту (пример - только что было продление адреса и абонент сменил устройство). В этом случае будет считаться, что у абонента уже есть одна сессия и что IP-адрес этой сессии еще занят. Для того, чтобы на DHCP-Discover происходило закрытие активных сессий на сервисе, нужно указать **dhcp.connection.closeOnNew=1**. Но в этом случае на одном сервисе возможна только одна активная сессия.

С версии 7.0 для обработки такой ситуации (вместо использования **dhcp.connection.closeOnNew=1**) можно использовать параметр конфигурации **dhcp.connection.checkDuplicate**. Он работает аналогично `radius.connection.checkDuplicate`, поэтому нужно указать цифру во втором разряде (для обработки ситуации, когда количество активных соединений превышено, но соединения с таким MAC-адресом не найдено, и нужно отключить просто самое старое соединение):

- 4 - происходит попытка сброса старого соединения, затем через 5 секунд завершение в биллинге, абонента на этой авторизации не пускаем;
- 5 - попытка сброса старого соединения в биллинге, завершение и пускаем немедленно (т.е. игнорируется ошибка);
- 6 - завершение старого соединения в биллинге, пускаем абонента немедленно;
- 7 - попытка в течении 5 секунд сбросить соединение, затем закрытие сессии в биллинге с ожиданием полного выполнения закрытия (т.е. IP-адрес станет свободным), пускаем абонента
- 8 - попытка сброса и сразу закрытие в биллинге с ожиданием полного выполнения закрытия (т.е. IP-адрес станет свободным), пускаем абонента;

- 9 - закрытие старого соединения в биллинге с ожиданием полного выполнения закрытия (т.е. IP-адрес станет свободным), пускаем абонента.

В большинстве случаев подойдет значение **dhcp.connection.checkDuplicate=90**.

Дополнительные параметры

В некоторых случаях возможна ситуация, когда запросы с неизвестных биллингу relay-агентов нужно игнорировать (т.к. в случае RENEW-запроса и dhcp.renew=1 биллинг будет отвечать NAK). Для этого в конфигурации корневого устройства укажите:

```
#          giaddr ( InetAccess)
# 0 - ( ), 1 -
dhcp.skipUnknownDevices=1
```

При поиске relay-агента сначала происходит поиск по giaddr, затем, если устройство еще не найден - поиск по source IP-адресу DHCP-пакета. Если данный дополнительный поиск необходимо отключить, укажите в конфигурации корневого устройства:

```
# IP-          giaddr ( InetAccess)
# 0 - , 1 - ( )
dhcp.findDeviceBySocketAddress=0
```

Если Вы не хотите указывать в дереве устройств все relay-агенты, а производить поиск только по agentRemoteId, то в конфигурации корневого устройства вы можете указать ID устройства, к которому будут привязываться DHCP-пакеты, вместо поиска по giaddr:

```
# DHCP- ,          giaddr ( InetAccess)
dhcp.overrideRelayDeviceId=
```

Процессор InetDhcpHelperProcessor

Предназначен для работы в связке с RADIUS-процессором. При успешной авторизации или старте сессии **InetDhcpHelperProcessor** запоминает сессию по ключу, который создается из шаблона **dhcp.key.pattern**, указанного в конфиге. В шаблоне могут быть указаны следующие параметры:

- **deviceId** - устройство, к которому привязана сессия, т.е. NAS, с которого идет RADIUS-аккаунтинг,
- **remoteId** - агентское устройство, обычно коммутатор с которого пришел DHCP-relay-запрос на NAS,
- **circuitId** - порт или VLAN, в зависимости от указанного типа поиска,
- **mac** - MAC-адрес, для данной схемы он должен быть в RADIUS-атрибуте Calling-Station-Id.

При получении DHCP-запроса по указанному **dhcp.key.pattern** извлекаются данные из DHCP-пакета, т.е. по **giaddr** или IP-адресу, от которого запроса идет поиск устройства **deviceId**, аналогично **InetDhcpProcessor** из пакета извлекаются **agentRemoteId** и **circuitId**, по **agentRemoteId** идет поиск агентского устройства **remoteId**, и по **circuitId** - порта или VLAN, MAC-адрес извлекается из соответствующего поля DHCP-пакета.

По этому ключу идет поиск среди запомненных RADIUS-сессий и выдача соответствующего ответа с IP-адресом из сессии.

Т.е. задача настроить так, чтобы из RADIUS-пакета и DHCP-пакета извлекались одни и те же данные (устройство, агентское устройство, порт /VLAN, MAC-адрес). Для этого в обработчике процессора протокола происходит извлечение из RADIUS-пакета DHCP-опций **remoteId** и **circuitId**.

Работа процессора зависит от параметра **radius.servSearchMode**:

- при поиске по логину (не рекомендуется, устарел, не удобен для связки с DHCP), в логине пользователя должны содержаться разделёнными двоеточием субопции DHCP.82 **remoteId** и **circuitId**, в номере звонящего - строка с HEX MAC-адреса. IP-адрес запоминается по ключу код устройства + опции + MAC-адрес;
- при остальных видах поиска из DHCP-запроса извлекаются идентификатор агентского устройства и номер порта или VLAN, аналогично **InetDhcpProcessor**. IP-адрес запоминается по ключу код устройства + код агентского устройства + номер интерфейса/VLAN + MAC.

Во всех случаях в RADIUS-атрибуте Calling-Station-Id должен быть проставлен MAC-адрес, чтобы при перезагрузке Access-сервера данные загрузились из сессий.

```
# IP- ( ) DHCP
dhcp.key.pattern=${deviceId}:${remoteId}:${circuitId}:${mac}
# - RADIUS- /VLAN, + MAC:
#dhcp.key.pattern=${deviceId}:${mac}
```

Процессор загружает список устройств-коммутаторов, начиная с корневого узла. Типы устройств-коммутаторов можно отфильтровать в переменной конфигурации корневого узла **dhcp.relay.deviceTypeIds** через запятую. Алгоритм поиска устройства при получении запроса идентичен **ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor**.

Пример пакетов для обработчика ISGProtocolHandler:

Код

RADIUS:

```
Packet type: Access-Request
Identifier: 140
Authenticator: {12 6B D7 9F 37 1E A1 39 BA 8C CD 13 0B CD 98 7B}
Attributes:
  User-Name=00060012cf539f5e:000403420101:0080.4840.a46f
  NAS-Identifier=7201-ipoe.nettrans.ru
  NAS-Port-Id=0/0/1/834
  User-Password=123
  Event-Timestamp=1314183142
  NAS-IP-Address=94.125.95.117
  NAS-Port=412
  Service-Type=5
  Acct-Session-Id=720000000000019C
  NAS-Port-Type=33
  cisco-avpair=circuit-id-tag=000403420101
  cisco-avpair=remote-id-tag=00060012cf539f5e
  cisco-NAS-Port=0/0/1/834
```

DHCP:

```
Message type: BOOT_REQUEST
Dhcp message type: DHCP Discover{1}
htype: 1, hlen: 6, hops: 1
xid: 1067065417, secs: 0, flags: 0
Client IP: 0.0.0.0
Your IP: 0.0.0.0
Server IP: 0.0.0.0
Relay IP: 109.233.170.1
Client MAC: {00804840A46F}
Host name{12}={support-desktop}
Parameter request list{55}={1, 28, 2, 3, 15, 6, 119, 12, 44, 47, 26, 121, 42}
Agent information{82}=
  sub{1}={000403420101}
  sub{2}={00060012CF539F5E}
```

Пример конфигурации для таких пакетов:

Конфигурация

```
#      RADIUS-
#      ()
radius.agent.option.removeHeader=2
# RADIUS- cisco-avpair
radius.agent.option.remoteId.type=1
#      cisco-avpair  remoteId
radius.agent.option.remoteId.prefix=remote-id-tag=
# RADIUS- cisco-avpair
radius.agent.option.circuitId.type=1
#      cisco-avpair  circuitId
radius.agent.option.circuitId.prefix=circuit-id-tag=

#      DHCP-
#      ()
dhcp.option82.removeHeader=2
#      agentRemoteId
#      82,      ,
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.position=0
dhcp.option82.agentRemoteId.length=6

#      INTERFACE_ID  VLAN
# (position length      circuitId,  RADIUS-!)
#      82,  VLAN,
dhcp.option82.vlanId.code=1
dhcp.option82.vlanId.position=0
dhcp.option82.vlanId.length=2
#      82,      ,
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.position=3
dhcp.option82.interfaceId.length=1

#      ,      RADIUS-  DHCP-.
# $deviceId - ( NAS relay-,  DHCP- )
# $remoteId - ( ,  ),  agentRemoteId
# $circuitId -      interfaceId,  VLAN
# $mac - MAC-
radius.key.pattern=$deviceId:$remoteId:$circuitId:$mac
```


Настройка разбора опции DHCP Option 82

Перед прочтением данной главы рекомендуется к прочтению глава [Процессор InetDhcpProcessor](#).

- Извлечение значений идентифицирующих абонента из DHCP-пакета
 - Базовые значения
 - Логин
 - Опции в виде HEX-строки
 - Разные форматы
- Извлечение значений идентифицирующих абонента из RADIUS-пакета
- Извлечение значений, идентифицирующих абонента из DHCP-пакета, с указанием позиции и длины
- Разные типы устройств
- Разные типы устройств с разным форматом agentRemoteId
- Cisco ISG и SmartEdge
- Cisco ISG и SmartEdge и разные типы устройств с разным форматом agentRemoteId
- Пример совмещения предобработки с конфигурацией (PON)

Для корректной работы нужно правильно извлекать значения agentRemoteId, circuitId (port/VLAN) из DHCP-пакета. А в случае использования IPoE с Cisco ISG или SmartEdge еще и из RADIUS-пакетов (в этом случае значения субопций option 82 находятся внутри RADIUS-пакетов). Извлекать значения можно с помощью конфигурации с указанием regex, с помощью конфигурации с указанием позиций значений, а также с помощью скрипта предобработки пакета.

Извлечение значений идентифицирующих абонента из DHCP-пакета

Для извлечения значения требуется указать код субопции, а также один или два regex - **hex** и **string**. **hex** указывает, что нужно выбрать в опции в 16-ричном формате (один байт - это два символа), **string** - указывает, что нужно выбрать в опции, преобразованной в строку.

Базовые значения

Предположим, у нас в DHCP-пакете присутствует такая option 82:

```
Agent information{82}=
sub{1}={076D00020201}
sub{2}={D067B3932607}
```

agentRemoteId является вся HEX-строка D067B3932607 в субопции 2 - это MAC-адрес, укажем как его извлечь:

```
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.hex=.*
```

Чтобы получить номер порта, который находится в субопции 1 в третьем байте ({076D00020201}), указываем:

```
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.hex=^\w{6}(\w{2})
```

Таким образом с помощью regex мы указываем, что нам нужны два символа (02) в этой HEX-строке.

Для того чтобы получить VLAN, который находится в субопции 1 в первых двух байтах ({076D00020201}), нужно указать:

```
dhcp.option82.vlanId.code=1
dhcp.option82.vlanId.hex=^\w{2})
```

Подобным образом можно извлечь и SVLAN:

```
dhcp.option82.agentSvlanId.code=1
dhcp.option82.agentSvlanId.hex=^\w{2})
```

Логин

Или логин для поиска по логину (для схемы PON, когда в agentRemoteId приходит MAC-адрес ONU):

```
dhcp.option82.login.code=2
dhcp.option82.login.hex=.*
```

Извлеченный логин можно сохранить в сессии (сохранение сработает, если включен поиск по логину или логин был установлен в предобработке):

```
dhcp.option82.login.save=1
```

Опции в виде HEX-строки

Приходящие значения могут представлять собой строку:

```
Agent information{82}=
sub{1}={7072617664792D3331612D706F7274332D3330352D766C616E}
sub{2}={33362D6C65736F7A61766F642065746820312F313235}
```

Такие значения обычно длиннее, чем бинарный вариант. В данном примере в опции 1 записана строка "pravdy-31a-port3-305-vlan", в опции 2 - "36-lesozavod eth 1/125". Для извлечения значения из такой опции можно использовать regex, прописанный в параметре **string**:

```
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.string=port(\d+)
```

```
dhcp.option82.vlanId.code=1
dhcp.option82.vlanId.string=(\d+)-vlan
```

Если **hex** при этом не указан, то это равнозначно, как если бы он был указан равным ".*":

```
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.hex=.*
dhcp.option82.interfaceId.string=port(\d+)
```

Если в 16-ричном виде присутствует префикс, который нужно исключить перед преобразованием в строку, то можно использовать оба **regex** - сначала с помощью **hex** выберем 16-ричное значение, затем с помощью **string** преобразуем его в строку и выберем подстроку. Например, у нас в значении есть префикс-длина:

```
Agent information{82}=
sub{1}={00197072617664792D3331612D706F7274332D3330352D766C616E}
sub{2}={001633362D6C65736F7A61766F642065746820312F313235}
```

Тогда исключить первые два байта и получить порт можем так:

```
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.hex=^{4}(.)$
dhcp.option82.interfaceId.string=port(\d+)
```

Разные форматы

Regex-ов можно указать несколько - это может быть удобно при наличии устройств разных марок - они могут посылать option 82 в разном формате:

```
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.1.hex=^0006(.{6})$
dhcp.option82.agentRemoteId.2.hex=.*
```

Для таких случаев рекомендуется, чтобы в дереве устройств биллинга было устройство с таким идентификатором (agentRemoteId), у него был свой тип устройства, а в конфигурации типа устройства была задана конфигурация для извлечения порт/VLAN и т.п. (см. главу Разные типы устройств). Но если разницу можно выделить через regex, то извлечение полей circuitId можно также произвести набором regex:

```
dhcp.option82.interfaceId.code=1
dhcp.option82.interfaceId.1.hex=^0006\w{2}(\w{2})
dhcp.option82.interfaceId.2.hex=.*
dhcp.option82.interfaceId.2.string=port(\d+)
```

При осуществлении настройки проверить ваш Regex вы можете на одном из множества онлайн-сервисов, например, <https://regex101.com/>. Для преобразования набора байт в 16-ричном представлении в строку также можно воспользоваться онлайн-сервисом.

Извлечение значений идентифицирующих абонента из RADIUS-пакета

Из RADIUS-пакета необходимо извлечь **Agent-Remote-Id** и **Circuit-Id** - значения DHCP option 82. Значения порта/VLAN/SVLAN будут извлечены из **Circuit-Id** с помощью правил описанных в параметрах **dhcp.option82.*** (см. выше).

Если приходящие значения закодированы в бинарном формате (например, значение Agent-Remote-Id совпадает с MAC-адресом агентского устройства) и нет префикса с указанием длины опции, то достаточно указать RADIUS-vendor и тип атрибута, например, для Redback Agent-Remote-Id и Agent-Circuit-Id:

```
radius.agentRemoteId.vendor=2352
radius.agentRemoteId.type=96
radius.option82.circuitId.vendor=2352
radius.option82.circuitId.type=97
```

Если в значениях опций присутствует префикс-длина размером два байта, то его нужно исключить при получении agentRemoteId:

```
radius.agentRemoteId.vendor=2352
radius.agentRemoteId.type=96
radius.agentRemoteId.hex=^{4}(.*)$
radius.option82.circuitId.vendor=2352
radius.option82.circuitId.type=97
```

Если в значении закодирована строка, а не бинарные значения, например:

```
Agent-Remote-Id={70 72 61 76 64 79 2D 33 36 61 2D 70 6F 72 74 35 2D 38 30 35 2D 76 6C 61 6E}
Agent-Circuit-Id={34 36 2D 6C 65 73 6F 7A 61 76 6F 64 20 65 74 68 20 30 2F 31 32 35}
```

то в конфигурации указываем преобразование в строку:

```
radius.agentRemoteId.vendor=2352
radius.agentRemoteId.type=96
radius.agentRemoteId.string=.*
radius.option82.circuitId.vendor=2352
radius.option82.circuitId.type=97
```

Circuit-Id в двух последних случаях мы извлекаем как есть - подразумевается, что извлеченное значение будет идентично значению DHCP-опции в DHCP-пакете и его обработка уже прописана в параметрах **dhcp.option82.***.

Для более сложных вариантов возможно извлечение с помощью преобработки (см. далее).

Извлечение значений, идентифицирующих абонента из DHCP-пакета, с указанием позиции и длины

Это старый вариант извлечения значений, но при больших нагрузках (от 300 тыс. сессий онлайн) он может быть предпочтительнее, т.к. будет тратить меньше ресурсов процессора.

```
# , , 0. .
# , , circuitId DHCP- RADIUS- .
# , position removeHeader.
dhcp.option82.removeHeader=0
```

```
# agentRemoteId 2
dhcp.option82.agentRemoteId.code=2
#
dhcp.option82.agentRemoteId.position=2
# , -1
dhcp.option82.agentRemoteId.length=6
# 0, remoteId , , MAC-; 1,
dhcp.option82.agentRemoteId.type=0
```

```
# interfaceId 1 (circuitId)
dhcp.option82.interfaceId.code=1
#
dhcp.option82.interfaceId.position=5
#
dhcp.option82.interfaceId.length=1
```

```
# vlanId 1 (circuitId)
dhcp.option82.vlanId.code=1
#
dhcp.option82.vlanId.position=2
#
dhcp.option82.vlanId.length=2
```

Разные типы устройств

Если используются разные типы устройств, у которых разные форматы circuitId, тип поиска DHCP-устройства должен быть 0 (в этом режиме сначала находится устройство по giaddr, у него вызывается rpreprocess, затем находится агентское ус-во, у него тоже вызывается rpreprocess) или 1 (в этом режиме сначала находится устройство по giaddr, затем находится агентское ус-во, у него тоже вызывается rpreprocess).

```
dhcp.deviceSearchMode=0
```

Конфигурация парсинга agentRemoteId должна быть указана в устройстве, с которого приходит запрос на InetAccess (т.е. чей giaddr указан в пакете). А конфигурация извлечения порта/VLAN из circuitId должна быть указана в агентских типах устройствах. Таким образом InetAccess найдет relay agent по giaddr, по его конфигурации извлечет agentRemoteId, по agentRemoteId найдет дочернее агентское устройство и уже по его конфигурации извлечет значения порта/VLAN.

Разные типы устройств с разным форматом agentRemoteId

В этом случае нельзя однозначно указать в конфигурации как извлечь agentRemoteId, в отличие от варианта выше.

Поэтому нужно воспользоваться набором **regex**, описанных выше:

```
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.1.hex=^0006(.{6})$
dhcp.option82.agentRemoteId.2.hex=.{8}(.{6})
```

Или же воспользоваться предобработкой пакетов. Для этого укажите тип поиска DHCP-устройства = 0 (в этом режиме сначала находится устройство по giaddr, у него вызывается rpreprocess, затем находится агентское ус-во, у него тоже вызывается rpreprocess).

```
dhcp.deviceSearchMode=0
```

Расширьте обработчик процессора протокола типа устройства, с которого приходит запрос на InetAccess (т.е. чей giaddr указан в пакете). По giaddr InetAccess однозначно найдет устройство. Затем вызовет у него предобработку, в которой нужно определить как распарсить и в ручную проставить agentRemoteId.

```
import java.util.Arrays;

import ru.bitel.bgbilling.kernel.network.dhcp.DhcpPacket;
import ru.bitel.bgbilling.kernel.network.dhcp.DhcpProtocolHandler;
import ru.bitel.bgbilling.modules.inet.access.sa.ProtocolHandlerAdapter;
import ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor;

public class Dhcp82ProtocolHandler
    extends ProtocolHandlerAdapter
    implements DhcpProtocolHandler
{
    @Override
    public void preprocessDhcpRequest( DhcpPacket request, DhcpPacket response )
        throws Exception
    {
        byte[] agentRemoteId = request.getSubOption( (byte)2 ).value;
        // DLink
        if( agentRemoteId.length == 8 )
        {
            request.setOption( InetDhcpProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 2, 6 ) );
        }
        else
        {
            request.setOption( InetDhcpProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 5, 6 ) );
        }
    }
}
```

Конфигурация извлечения порта/VLAN из circuitId должна быть указана в агентских типах устройствах. Таким образом InetAccess найдет relay agent по giaddr, предобработка извлечет и проставит agentRemoteId, по agentRemoteId InetAccess найдет дочернее агентское устройство и уже по его конфигурации извлечет значения порта/VLAN.

Cisco ISG и SmartEdge

В отличие от схемы DHCP82 без Cisco ISG/SmartEdge, здесь еще нужно извлечь remoteId и circuitId из RADIUS-пакета.

```
# , , 0. .
# , , circuitId DHCP- RADIUS- .
# , position removeHeader.
radius.agent.option.removeHeader=2
```

```
# SmartEdge
#
radius.agent.option.remoteId.type=96
#
radius.agent.option.remoteId.position=0
#
radius.agent.option.remoteId.length=-1
radius.agent.option.circuitId.type=97

#
radius.agent.option.remoteId.type=202
radius.agent.option.remoteId.position=0
radius.agent.option.remoteId.length=-1
radius.agent.option.circuitId.type=202
```

```
# Cisco ISG
radius.agent.option.remoteId.type=1
radius.agent.option.remoteId.prefix=remote-id-tag=
radius.agent.option.circuitId.type=1
radius.agent.option.circuitId.prefix=circuit-id-tag=
```

Cisco ISG и SmartEdge и разные типы устройств с разным форматом agentRemoteId

В этом случае нельзя однозначно указать в конфигурации как извлечь agentRemoteId и circuitId из RADIUS-пакета. Поэтому это нужно сделать в предобработке (извлечение из RADIUS-пакета и сейчас происходит в предобработке, но согласно конфигурации - это можно увидеть в динамических классах ISGProtocolHandler и SmartEdgeClipsProtocolHandler). Расширьте класс предобработки Cisco ISG/SmartEdge:

```
import java.util.Arrays;

import ru.bitel.bgbilling.kernel.network.radius.RadiusAttribute;
import ru.bitel.bgbilling.kernel.network.radius.RadiusPacket;
import ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor;

public class XSmartEdgeClipsProtocolHandler
    extends SmartEdgeClipsProtocolHandler
{
    @Override
    protected void setAgentOptions( RadiusPacket request )
    {
        RadiusAttribute<byte[]> agentRemoteIdAttribute = request.getAttribute( 2352, 96 );
        RadiusAttribute<byte[]> circuitRemoteIdAttribute = request.getAttribute( 2352, 97 );

        byte[] agentRemoteId = agentRemoteIdAttribute.getValue();
        byte[] circuitRemoteId = circuitRemoteIdAttribute.getValue();

        // DLink
        if( agentRemoteId.length == 8 )
        {
            request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 2, 6 ) );
            request.setOption( InetRadiusProcessor.AGENT_CIRCUIT_ID, Arrays.copyOfRange(
circuitRemoteId, 2, circuitRemoteId.length - 2 ) );
        }
        else
        {
            request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 5, 6 ) );
            request.setOption( InetRadiusProcessor.AGENT_CIRCUIT_ID, Arrays.copyOfRange(
circuitRemoteId, 2, circuitRemoteId.length - 2 ) );
        }
    }
}
```

Если в DHCP-пакете указан giaddr Cisco/SmartEdge, то в этот класс нужно добавить предобработку DHCP

```

import java.util.Arrays;

import ru.bitel.bgbilling.kernel.network.dhcp.DhcpPacket;
import ru.bitel.bgbilling.kernel.network.radius.RadiusAttribute;
import ru.bitel.bgbilling.kernel.network.radius.RadiusPacket;
import ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor;
import ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor;

public class XSmartEdgeClipsProtocolHandler
    extends SmartEdgeClipsProtocolHandler
{
    @Override
    protected void setAgentOptions( RadiusPacket request )
    {
        RadiusAttribute<byte[]> agentRemoteIdAttribute = request.getAttribute( 2352, 96 );
        RadiusAttribute<byte[]> circuitRemoteIdAttribute = request.getAttribute( 2352, 97 );

        byte[] agentRemoteId = agentRemoteIdAttribute.getValue();
        byte[] circuitRemoteId = circuitRemoteIdAttribute.getValue();

        // DLink
        if( agentRemoteId.length == 8 )
        {
            request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 2, 6 ) );
            request.setOption( InetRadiusProcessor.AGENT_CIRCUIT_ID, Arrays.copyOfRange(
circuitRemoteId, 2, circuitRemoteId.length - 2 ) );
        }
        else
        {
            request.setOption( InetRadiusProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 5, 6 ) );
            request.setOption( InetRadiusProcessor.AGENT_CIRCUIT_ID, Arrays.copyOfRange(
circuitRemoteId, 2, circuitRemoteId.length - 2 ) );
        }
    }

    @Override
    public void preprocessDhcpRequest( DhcpPacket request, DhcpPacket response )
        throws Exception
    {
        byte[] agentRemoteId = request.getSubOption( (byte)2 ).value;
        // DLink
        if( agentRemoteId.length == 8 )
        {
            request.setOption( InetDhcpProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 2, 6 ) );
        }
        else
        {
            request.setOption( InetDhcpProcessor.AGENT_REMOTE_ID, Arrays.copyOfRange(
agentRemoteId, 5, 6 ) );
        }
    }
}

```

Если же giaddr в DHCP-пакете relay agent'a, от которого получил запрос Cisco/SmartEdge, то для типа устройства relay agent'a нужно указать отдельный обработчик, см. [выше](#).

Пример совмещения предобработки с конфигурацией (PON)

В данном примере из пакета извлекаются нужные значения, устанавливаются в запрос, а также формируется строка, которая далее окажется в поле "На номер" сессии (InetDhcpProcessor.CALLED_STATION_ID). Данную строку можно будет использовать как информационную (она отображается в клиенте биллинга), а также первом подключении (регистрации) абонента для заполнения полей сервиса договора.

Конфигурация:

```

# ""
dhcp.servSearchMode=0

# - MAC- PON
dhcp.option82.login.code=2
dhcp.option82.login.hex=. *
# ""
dhcp.option82.login.save=1

# agentRemoteId ( )
dhcp.option82.agentRemoteId.code=0
# SVLAN
dhcp.option82.agentSvlanId.code=1
dhcp.option82.agentSvlanId.hex=^\w{4})

#
# OLT
dhcp.option82.oltPort.code=1
dhcp.option82.oltPort.hex=^{6}(\w{2})
# ont ID OLT
dhcp.option82.ontId.code=1
dhcp.option82.ontId.hex=^{8}(\w{2})

```

Предобработка:

```

package ru.provider.bgbilling.modules.inet.dyn.device;

import javax.annotation.Resource;

import org.apache.log4j.Logger;

import ru.bitel.bgbilling.kernel.network.dhcp.DhcpOption;
import ru.bitel.bgbilling.kernel.network.dhcp.DhcpPacket;
import ru.bitel.bgbilling.modules.inet.access.sa.ProtocolHandlerAdapter;
import ru.bitel.bgbilling.modules.inet.api.common.bean.InetDevice;
import ru.bitel.bgbilling.modules.inet.api.common.bean.InetDeviceType;
import ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor;
import ru.bitel.bgbilling.modules.inet.runtime.device.PacketValueExtractor;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.ParameterMap;
import ru.bitel.common.io.BinaryValueExtractor;

public class PonDhcpRelayProtocolHandler
    extends ProtocolHandlerAdapter
{
    private static final Logger logger = Logger.getLogger( PonDhcpRelayProtocolHandler.class );

    @Resource
    private PacketValueExtractor packetValueExtractor;

    private byte oltPortCode;
    private BinaryValueExtractor oltPortExtractor;

    private byte ontIdPortCode;
    private BinaryValueExtractor ontIdPortExtractor;

    @Override
    public void init( Setup setup, int moduleId, InetDevice inetDevice, InetDeviceType inetDeviceType,
        ParameterMap deviceConfig )
        throws Exception
    {
        oltPortCode = (byte)deviceConfig.getInt( "dhcp.option82.oltPort.code", 1 );
        oltPortExtractor = packetValueExtractor.getExtractor( deviceConfig, "dhcp.option82.oltPort." );

        ontIdPortCode = (byte)deviceConfig.getInt( "dhcp.option82.ontId.code", 1 );
        ontIdPortExtractor = packetValueExtractor.getExtractor( deviceConfig, "dhcp.option82.ontId." );
    }
}

```



```

@Override
public void preprocessDhcpRequest( final DhcpPacket request, final DhcpPacket response )
    throws Exception
{
    final int svlan = packetValueExtractor.getOption82AgentSvlanId( request, true );
    final int vlan = packetValueExtractor.getOption82VlanId( request, true );

    final int oltPort = extract( request, oltPortCode, oltPortExtractor );
    final int ontId = extract( request, ontIdPortCode, ontIdPortExtractor );

    final String onuIdentifier = (String)packetValueExtractor.getOption82Login( request, true );

    final String calledStationId = svlan + "." + vlan + "/" + oltPort + ":" + ontId + "/" + onuIdentifier;

    logger.info( calledStationId );

    // VLAN,
    request.setOption( InetDhcpProcessor.VLAN_ID, vlan );
    // VLAN circuitId
    request.setOption( InetDhcpProcessor.CIRCUIT_ID, vlan );
    // calledStationId ,
    request.setOption( InetDhcpProcessor.CALLED_STATION_ID, calledStationId );
}

private static int extract( final DhcpPacket request, final byte code, final BinaryValueExtractor extractor
)
{
    final DhcpOption subOption = request.getSubOption( code );
    if( subOption != null )
    {
        return extractor.extractInt( subOption.value, -1 );
    }

    return -1;
}
}

```

Получение IP-адреса DHCP-клиентом

Для корректной работы абонента требуется передавать в OFFER/RESPONSE, кроме выданного IP-адреса, как минимум опции **serverIdentifier** (идентификатор DHCP-сервера), **gate** (шлюз), **subnetMask** (маска подсети), **leaseTime** (время в секундах, на которое выдает IP-адрес), **dns** (DNS-сервера в порядке важности).

```
dhcp.option.serverIdentifier=10.0.6.56
dhcp.option.leaseTime=600
#
dhcp.net.option.193.106.88.0:255.255.255.0.gate=193.106.88.1
dhcp.net.option.193.106.88.0:255.255.255.0.subnetMask=255.255.255.0
dhcp.net.option.193.106.88.0:255.255.255.0.dns=194.165.18.6
```

В опции **serverIdentifier** обычно нужно указать IP-адрес InetAccess; в некоторых случаях - 0.0.0.0 или адрес Cisco ASR/Redback (если он - relay, который пересылает запросы биллингу).

Опции **gate**, **subnetMask**, **dns** можно указать в IP-ресурсе в соответствующих полях, а не через конфигурацию устройства. Также в конфигурации IP-ресурса можно указать опции, привязанные к этому ресурсу, аналогично конфигурации устройства, через префикс **dhcp.option**.

Важно учитывать что при истечении времени T1 (или по другому - renewal time) DHCP-клиент переходит в стадию RENEW и начинает отправлять запросы RENEW на продление lease. Если ответа он не получает, то при истечении времени T2 (или по другому - rebinding time) DHCP-клиент переходит в стадию REBIND и отправляет REBIND-запросы для продления lease выданного ранее IP-адреса. Время T1 по умолчанию - это (leaseTime * 0.5), T2 - это (leaseTime * 0.875).

Существуют опции **renewalTime** и **rebindingTime**, с помощью которых можно указать конкретные значения, когда DHCP-клиент должен перейти в стадию RENEW, а когда в REBIND. Значение **renewalTime** должно быть меньше **rebindingTime**, а значение **rebindingTime** - меньше **leaseTime**.

```
dhcp.option.leaseTime=600
dhcp.option.renewalTime=180
dhcp.option.rebindingTime=420
```

По умолчанию сервер InetAccess не отвечает на RENEW-запросы (чтобы заставить DHCP-клиент перейти в стадию REBIND), но это работает далеко не всегда и далеко не во всех схемах. Чтобы включить обработку RENEW-запросов, в конфигурации корневого устройства (обычно это Access+Accounting) нужно указать (требуется перезапуск InetAccess):

```
dhcp.renew=1
```

Существует множество реализаций DHCP-клиентов и не все они работают согласно RFC.

Например, по RFC DHCP-клиент должен посылать DISCOVER и REQUEST из одной сессии запроса IP-адреса с одинаковым xid (Transaction ID), но некоторые DHCP-клиенты в REQUEST, отправляют другой xid (причем isc-dhcp-server обрабатывает это нормально, т.е. не смотрит на xid). Чтобы InetAccess не связывал DISCOVER и REQUEST по xid и обрабатывал такие запросы нормально, в конфигурации корневого устройства (обычно это Access+Accounting) нужно указать (требуется перезапуск InetAccess):

```
dhcp.xid=0
```

В конфигурации isc-dhcp-server есть параметр **always-broadcast**:

Цитата

always-broadcast flag;

The DHCP and BOOTP protocols both require DHCP and BOOTP clients to set the broadcast bit in the flags field of the BOOTP message header. Unfortunately, some DHCP and BOOTP clients do not do this, and therefore may not receive responses from the DHCP server. The DHCP server can be made to always broadcast its responses to clients by setting this flag to 'on' for the relevant scope; relevant scopes would be inside a conditional statement, as a parameter for a class, or as a parameter for a host declaration. To avoid creating excess broadcast traffic on your network, we recommend that you restrict the use of this option to as few clients as possible. For example, the Microsoft DHCP client is known not to have this problem, as are the OpenTransport and ISC DHCP clients.

Аналогично этому параметру, в конфигурации устройства (не обязательно корневого) можно указать (требуется нажать "Перечитать конфигурацию на серверах"):

```
dhcp.alwaysBroadcast=1
```

Возможные проблемы при получении/продлении адреса

На DISCOVER биллинг не отправляет OFFER

Запрос не доходит до биллинга или биллинг не находит сервис договора.

На OFFER от биллинга клиент не присылает REQUEST

Ответ от биллинга не доходит до клиента или клиенту не нравится содержимое пакета. В последнем случае нужно проверить опции **serverIdentifier, gate, subnetMask, leaseTime**.

На REQUEST от клиента биллинг отвечает NAK, в логах

```
Unknown packet (linked offer not found). Discard packet.
```

В случае, если это получение адреса (SELECTING state), то возможно, что клиент в REQUEST посылает xid, отличный от того, что он посылал в DISCOVER. Для поддержки таких роутеров в конфигурации корневого устройства (Access+Accounting) нужно указать параметр `dhcp.xid=0` (см. выше).

В случае, если это продление адреса (RENEWING или REBINDING), это означает, что биллинг не нашел сессию на данном сервисе с таким MAC-адресом. Возможно, что **connection.close.timeout** меньше чем **leaseTime**.

Через время leaseTime у клиента происходит реполучение адреса с DISCOVER-пакетом (SELECTING state)

Скорее всего клиент не поддерживает (из-за ошибки в реализации DHCP-клиента) нормальный REBINDING, а сразу переходит в SELECTING, и в биллинге отключена поддержка RENEW-запросов (`dhcp.renew=1`) или **serverIdentifier** указан как `0.0.0.0` или RENEW-пакеты не доходят до биллинга. Нужно настроить поддержку RENEW-запросов - скорее всего DHCP-клиент через RENEWING state (никогда не переходя в REBINDING) будет работать нормально.

Слушатель InetFlowListener

Для того, чтобы принимать и обрабатывать flow-пакеты (netflow/netflow v9/sflow) в **inet-accounting.xml** должен быть прописан соответствующий слушатель. Если необходимо приходящие пакеты перед обработкой сохранять на диск, в слушатель нужно передать dataLogger. Объект dataLogger описывается один раз и передается всем flow-слушателям, с которых нужно сохранять данные.

У слушателя необходимо указать параметры:

- type - его тип: netflow, netflow9, sflow;
- host - хост (интерфейс), на котором будет открыт сокет, по умолчанию на всех интерфейсах;
- port - порт, на котором будет открыт сокет;
- recvBufferSize - размер буфера приема слушателя;
- soRCVBUF - рекомендуемый размер буфера приема сокета (SO_RCVBUF);
- threadCount - количество потоков-обработчиков;
- agentDeviceIds - id устройств, источников flow-потоков, если на данном порту нужно обрабатывать данные только у определенных источников;
- dataLogger - dataLogger, с помощью которого приходящие пакеты будут записываться в лог файлы.

Код

```
<!-- ScheduledExecutorService, dataLogger -->
<scheduledExecutorService name="hrlydtlggr" corePoolSize="1"/>

<!-- C dataLogger, flow- ( ) -->
<bean name="flowDataLogger" class="ru.bitel.bgbilling.modules.inet.collector.IPHourlyDataLogger"
>
    <param name="scheduledExecutor">hrlydtlggr</param>
</bean>

<!-- C flow- dataLogger -->
<bean name="flowListener" class="ru.bitel.bgbilling.modules.inet.collector.InetFlowListener">
    <constructor factoryMethod="newInstance">
        <!-- , netflow, netflow9 sflow -->
        <param name="type" value="netflow"/>
        <!-- (), . - -->
        <param name="host" value=""/>
        <!-- , -->
        <param name="port" value="2001"/>
        <!-- -->
        <param name="recvBufferSize">4 * 1024 * 1024</param>
        <!-- SO_RCVBUF -->
        <param name="soRCVBUF">512 * 1024</param>
        <!-- - -->
        <param name="threadCount" value="10"/>
        <!-- id -, c -->
        <param name="agentDeviceIds" value=""/>
        <!-- id -, c -->
        <param name="processAgentDeviceIds" value=""/>
        <!-- 1 - , IP- IP-,
            2 - IP- -->
        <param name="ipResourceFilter" value=""/>
        <!-- dataLogger -->
        <param name="dataLogger">flowDataLogger</param>
    </constructor>
</bean>
```

Каждый слушатель использует буферы памяти в direct memory, которая не находится в обычном heap java, общий размер которых можно рассчитать как $recvBufferSize + recvBufferSize + threadCount * datalog.flow.chunk.size(datalog.chunk.size)$. Direct memory может использоваться также другими слушателями, а также при обработке логов. При превышении доступной памяти произойдет ошибка `java.lang.OutOfMemoryError: Direct buffer memory`. В старых билдах JRE `MaxDirectMemorySize` по умолчанию ограничен 64МБ, в новых - ограничен размером свободной оперативной памяти. Для того, чтобы указать конкретное значение, используется параметр в строке запуска `-XX:MaxDirectMemorySize`.

Обычно достаточно оставить параметр `agentDeviceIds` пустым, что будет означать, что поток будет приниматься со всех устройств-источников, начиная от корневого устройства Accounting-сервера. Однако в ситуации, когда с одного IP-адреса приходит два потока с разных источников, чтобы Accounting корректно разделял данные на два источника, необходимо создать два устройства-источника с одним и тем же IP-адресом и добавить два слушателя на двух разных портах и в `agentDeviceIds` одного прописать первый источник, во втором - второй. Таким образом трафик, идущий с одного и того же адреса на один порт будет считаться как трафик с одного источника, на другой порт - с другого источника.

В конфигурации типа устройства-источника (или в конфигурации устройства) необходимо указать какой поток идет с источника:

```
# , netflow, netflow9, ipfix sflow
# - netflow
flow.agent.type=netflow
```

Для тарификации по Netflow9/IPFIX/Next Gen NetFlow в шаблоне должны присутствовать следующие поля:

i
IN_BYTES, PROTOCOL, SRC_TOS, L4_SRC_PORT, IPV4_SRC_ADDR, INPUT_SNMP, L4_DST_PORT, IPV4_DST_ADDR, OUTPUT_SNMP, IPV4_NEXT_HOP, LAST_SWITCHED, FIRST_SWITCHED, IPV6_SRC_ADDR, IPV6_DST_ADDR

Когда сессия стартует на каком-либо устройстве, по умолчанию считается, что это устройство и есть источник flow потока для данной сессии. Например, когда устройство одновременно является NAS'ом и источником netflow.

В схеме, когда flow-сенсор находится выше, необходимо указать для каждого устройства (например, NAS'a) привязку к источнику. Например, источник - это устройство с кодом 3. В конфигурации устройства-NAS'a прописываем, что трафик с любого интерфейса источника 3 может принадлежать сессии с данного NAS'a:

```
flow.agent.link=3:-1
```

Или, трафик с интерфейсов 1 или 2 может принадлежать сессии с данного NAS'a. Т.е. flow пакеты даже с таким же IP-адресом, как у сессии, но с другим интерфейсом, привязаны к ней не будут.

```
flow.agent.link=3:1,2
```

Для того, чтобы указать, что устройство является источником для всех своих дочерних устройств, достаточно в конфигурации типа устройства-источника прописать:

```
flow.agent.link={@deviceId}:-1
```

o
При использовании InetFlowListener возможна схема, когда будет множество устройств-коммутаторов с интерфейсами, которые никак не связаны с Flow-агентом и поэтому у них не указан в конфигурации flow.agent.link. Исторически сложилось, что если параметр flow.agent.link не указан, то устройство само считается Flow-агентом с указанными интерфейсами. Для экономии памяти рекомендуется указать для таких устройств flow.agent.link={@deviceId}:-1.

Если необходимо исключить из сохранения и обработки пакеты, в которых отсутствуют записи с IP-адресами, принадлежащими одному из заведенных диапазонов IP-ресурсов, в inet-accounting.xml для слушателя нужно указать параметр <param name="ipResourceFilter" value="1"/>, или же глобально (т.е. для всех слушателей) - в конфигурации модуля:

```
# flow :
# 0 ( ) - , 1 - , IP- IP-,
# 2 - , 1, IP-.
flow.ipResourceFilter=1
```

RADIUS/DHCP сессии

IP-адрес RADIUS или DHCP сессии автоматически привязывается к слушателю Netflow, привязка осуществляется по параметру **flow.agent.link**, описанному выше.

В схемах IP Subscriber Sessions возможна ситуация, когда RADIUS-сессии мы выдаем подсеть с помощью **Framed-IP-Netmask**, но в RADIUS-Accounting-пакетах не приходят ни **Framed-IP-Netmask**, ни **Framed-Route**. Для того, чтобы статическая подсеть сервиса всегда привязывалась к слушателю Netflow и трафик попадал на сессию, в конфигурации типа сервиса нужно указать:

```
# RADIUS-
serv.flow.staticSubnet=1
```

Экспорт данных в CSV (Netflow/sFlow, NAT logging)

Для проверки, отладки или других целей существует возможность экспортировать данные по трафику из лог-файлов .bgdl/.data. Для этого необходимо выполнить консольную команду `./accounting.sh flowExport`, например:

```
./accounting.sh flowExport -s 2 -h 2013-10-17-22 -tFmt "dd.MM.yyyy HH:mm:ss" -f flows.csv
```

Возможные параметры:

- `-s` - ID устройства - источника данных (source_ID);
- `-h` - фильтр по одному часу, формат ГГГГ-ММ-ДД-ЧЧ, например, 2013-10-17-22;
- `-tFrom` - фильтр по периоду, начало периода, формат ГГГГ-ММ-ДДТТЧЧ:ММ:СС, символ Т - латинский;
- `-tTo` - фильтр по периоду, конец периода, формат ГГГГ-ММ-ДДТТЧЧ:ММ:СС, символ Т - латинский;
- `-i` - фильтр по интерфейсам, через запятую, например 1,4,7,41;
- `-r` - фильтр по одному IP-адресу или диапазону IP-адресов, например, 192.168.0.1 или 192.168.0.1-192.168.0.24;
- `-f` - имя файла, в который будет сохранен вывод;
- `-tFmt` - формат времени в выводе, например, dd.MM.yyyy HH:mm:ss;
- `-dir` - корневая директория с flow-логами, по умолчанию используется значение параметра `datalog.flow.dir`, т.е. та директория, в которую сохраняет логи InetAccounting;
- `-maxSort` - кол-во выбранных записей для одного лог-файла, когда сортировку по времени нужно прекратить (для экономии памяти);
- `-writeIfaces` - будет ли выводиться значения интерфейсов в CVS (1 или 0);
- `-tZone` - зона (если данные сохранены в одной зоне, а запускаете flowExport с машины в другой зоне).

Минимально необходимые для выполнения параметры - `-s`, `-h` или `-tFrom/-tTo`, `-f`, например:

```
./accounting.sh flowExport -s 2 -tFrom 2014-01-01T00:00:00 -tTo 2014-01-31T23:59:59 -f 2014-01.csv -tFmt "yyyy-MM-dd HH:mm:ss"
```

Копия класса, производящего экспорт также находится в [динамических классах](#) - `ru.bitel.bgbilling.modules.inet.dyn.accounting.detail`.

FlowExport. Его можно модифицировать под свои нужды и запустить командной:

```
./accounting.sh dynClassRun ru.bitel.bgbilling.modules.inet.dyn.accounting.detail.FlowExport -s 1 -h 2013-11-20-13 -f 2013-11-20-13.csv
```

Или лучше создать копию этого класса, модифицировать и запускать уже его, т.к. динамические классы из стандартной поставки перезаписываются при обновлении биллинга.

Команды, указанные выше, запускают экспорт внутри запущенного процесса InetAccounting. При необходимости Вы можете запустить экспорт отдельным процессом:

```
`${JAVA_HOME}/bin/java -cp lib/app/*:lib/ext/* ru.bitel.bgbilling.modules.inet.accounting.detail.FlowExport -dir /dir/to/netflow/logs -s 2 -tFrom 2014-01-01T00:00:00 -tTo 2014-01-31T23:59:59 -f 2014-01.csv -tFmt "yyyy-MM-dd HH:mm:ss"
```

В этом случае также необходимо указать параметр `-dir`, т.к. значение параметра `datalog.flow.dir` при таком запуске недоступно.

Cisco NAT high-speed logging (HSL), Ericsson (Redback) SmartEdge NAT logging и MikroTik NAT log

Модуль Inet поддерживает логирование NAT Cisco NAT HSL, SE NAT logging и MikroTik NAT log, осуществляющееся с помощью Netflow9/IPFIX /Next Gen NetFlow. Для сохранения NAT-логов достаточно настроить прием/сохранение Netflow9.

Для получения информации по NAT-логам необходимо выполнить консольную команду `./accounting.sh natLogExport`, например:

```
./accounting.sh natLogExport -s 1320 -tFrom 2014-08-09T14:02:00 -tTo 2014-08-09T16:03:00 -a 95.94.51.35 -f flow_nat.csv -tFmt "dd.MM.yyyy HH:mm:ss.SSS"
```

Возможные параметры:

- -s - ID устройства - источника данных (source_ID);
- -h - фильтр по одному часу, формат ГГГГ-ММ-ДД-ЧЧ, например, 2013-10-17-22;
- -tFrom - фильтр по периоду, начало периода, формат ГГГГ-ММ-ДДТЧЧ:ММ:СС, символ Т - латинский;
- -tTo - фильтр по периоду, конец периода, формат ГГГГ-ММ-ДДТЧЧ:ММ:СС, символ Т - латинский;
- -a - внешний IP-адрес;
- -p - внешний порт;
- -f - имя файла, в который будет сохранен вывод;
- -tFmt - формат времени в выводе, например, dd.MM.yyyy HH:mm:ss;
- -dir - корневая директория с flow-логами, по умолчанию используется значение параметра datalog.flow.dir, т.е. та директория, в которую сохраняет логи InetAccounting;
- -tZone - зона (если данные сохранены в одной зоне, а запускаете flowExport с машины в другой зоне).

Минимально необходимые для выполнения параметры - **-s**, **-h** или **-tFrom/-tTo**, **-f**, например:

```
./accounting.sh natLogExport -s 1320 -h 2014-08-09-15 -f flow_nat.csv
```

При необходимости Вы можете запустить экспорт отдельным процессом:

```
${JAVA_HOME}/bin/java -cp lib/app/*:lib/ext/* ru.bitel.bgbilling.kernel.network.datalog.netflow.nat.NatLogProcessor -dir /dir/to/netflow/logs -s 1320 -tFrom 2014-08-09T14:02:00 -tTo 2014-08-09T16:03:00 -a 95.94.51.35 -f flow_nat.csv -tFmt "dd.MM.yyyy HH:mm:ss.SSS"
```

В этом случае также необходимо указать параметр **-dir**, т.к. значение параметра datalog.flow.dir при таком запуске недоступно.

Описание формата файлов логов BG Data Log (BGDL)

Название файла - **log_YYYY-MM-DD-НН.ХХХ.bgdl**, где **YYYY** - год, **MM** - месяц, **DD** - день, **НН** - час, **ХХХ** - порядковый номер файла в данном часе, например **000, 001, 002** и т.д.

Файл начинается с 16 байтового префикса

MAGIC	VERSION	TYPE	HEADER LENGTH
42 47 44 4C	00 00 00 04	00 00 00 01	00 00 00 C9
BGDL	4	IP	201

где TYPE: IP=1, RADIUS=2, DHCP=3

Далее идет блок с параметрами заголовка, длина блока берется из префикса (например, 201 - в таблице выше)

4 байта	4 байта	N байт
ID первого параметра	длина первого параметра, N байт	значение первого параметра
...
ID последнего параметра	длина последнего параметра, N байт	значение последнего параметра

Параметры заголовка

ID	Название	Длина, байт	Описание	Пример
3	FINISHED_ID	1	Флаг окончания записи файла-лога, 0 - не закончен, 1 - закончен	TRUE
4	BUFFER_ID	5	Если тип CHUNKED(1), данные в файле разбиты на блоки размером Size байт. Если тип NONE(0), то данные хранятся одним блоком размером Size байт	Type: CHUNKED; Size: 524288
5	COMPRESSION_ID	3	Сжатие блока с данными: NONE(0), ZLIB(1), GZIP(2), XZ(3)	Type: ZLIB; Level: 5; Strategy: 1
6	DISTRIBUTED_ID	1	если 0 - данные хранятся в том же файле что и заголовок, если 1 - данные в отдельном файле log_YYYY-MM-DD-НН.ХХХ.data	TRUE
7	STREAMING_ID	1	Флаг, значения 0 и 1.	TRUE
100	IP_DATALOG_TYPE_ID	3	формат логов: RAW(0), NETFLOW(1), SFLOW(2), SNMP(3), NETFLOW9(4), IPFIX(5)	Type: NETFLOW9
110	NETFLOW9_TEMPLATES_ID	Y	описание шаблонов с данными: 2 байта ID шаблона, 2 байта количество полей в шаблоне, далее для каждого поля 2 байта ID поля + 2 байта длина поля, далее следующий шаблон пока не переберем Y байт	Template IDs: 256, 257, 258, 259

После параметров заголовка в зависимости от значения параметра **DISTRIBUTED_ID** идут пакеты с данными если **DISTRIBUTED_ID=0** или файл **bgdl** заканчивается, а данные находятся в отдельном файле **log_YYYY-MM-DD-НН.ХХХ.data**.

В зависимости от значения параметра **COMPRESSION_ID** данные могут быть сжаты или нет.

Формат файла с данными. В зависимости от значения параметра **BUFFER_ID** данные могут располагаться одним блоком или могут быть разбиты на равные куски (**CHUNK**). Если **BUFFER_ID=1**, данные разбиты на куски размером **size** байт, если **BUFFER_ID=0** данные хранятся одним блоком размером **size** байт.

Если данные хранятся кусками, то в начале каждого куска идет заголовок 16 байт

4 байта	4 байта	4 байта	4 байта
реальная длина блока с данными, может быть меньше длины куска	номер куска в файле	?	?

Далее идут пакеты flow в том формате в котором они приходят от оборудования.

Сбор статистики с помощью InetSnmpWorker

Для того, чтобы считывать статистику по счетчикам snmp в **inet-accounting.xml** должен быть прописан соответствующий следующий обработчик:

```
<bean name="snmpWorker" class="ru.bitel.bgbilling.modules.inet.accounting.InetSnmpWorker">
  <constructor>
    <param name="agentDeviceIds" value="" />
    <param name="period" value="30" />
  </constructor>
</bean>
```

Тут доступны такие параметры

agentDeviceIds - коды устройств, с которых нужно собирать статистику. Если не указан то собирает со всех. Собирает snmp только с тех устройств, в конфигурации которых указан параметр flow.agent.type=snmp.

period - интервал в секундах, через который опрашиваются snmp-счетчики устройств.

Настройка BGInetAccess сервера

В конфигурации модуля присутствует параметр **accounting.deviceTypeIds**, определяющий **коды типов устройств**, которые являются Accounting-серверами. Поиск устройств начинается с корневого устройства Access-сервера, включительно. Т.к. одно и то же устройство может быть и Access и Accounting сервером. Таким образом, "под" одним устройством Access-сервера может располагаться несколько устройств Accounting-серверов.

Пример типовой настройки inet-access.xml, которая идет в дистрибутиве:

Код

```
<application context="access">
  <!-- -->
  <param name="app.name" value="BGInetAccess"/>
  <!-- id -->
  <param name="app.id" value="" />

  <!-- -->
  <param name="db.driver" value="com.mysql.jdbc.Driver"/>
  <param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&queryTimeoutKillsConnection=true&connectTimeout=1000"/>
  <param name="db.user" value="bill"/>
  <param name="db.pswd" value="bgbilling"/>
  <param name="db.validationTimeout" value="10"/>

  <!-- MQ -->
  <param name="mq.url" value="failover:(tcp://localhost:61616)"/>
  <param name="mq.user" value="bill"/>
  <param name="mq.pswd" value="bgbilling"/>

  <!-- -->
  <param name="server.apps.type" value="inet"/>
  <param name="server.update.url" value="http://localhost:8080/bgbilling"/>
  <param name="server.update.check.ssl" value="no"/>

  <!-- id -->
  <param name="moduleId" value="" />
  <!-- id -->
  <param name="rootDeviceId" value="" />

  <!-- , -->
  <param name="commonIdentifierName" value="rootDeviceId"/>

  <!-- -->
  <!-- , radius -->
  <param name="datalog.radius.dir" value="data/radius" />
  <!-- , -->
  <param name="datalog.radius.chunk.size" value="262144" />
  <!-- radius : 0 - , 1 - zlib -->
  <param name="datalog.radius.compression.type" value="1" />
  <!-- , flow -->
  <param name="datalog.dhcp.dir" value="data/dhcp" />
  <!-- , -->
  <param name="datalog.dhcp.chunk.size" value="131072" />
  <!-- flow : 0 - , 1 - zlib -->
  <param name="datalog.dhcp.compression.type" value="1" />

  <!-- Access -->
  <bean name="access" class="ru.bitel.bgbilling.modules.inet.access.Access" />

  <context name="radius">
    <!-- C RADIUS- -->
    <bean name="radiusProcessor" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor"
  />

  <!-- ScheduledExecutorService, dataLogger -->
  <scheduledExecutorService name="hrlydtlggr" corePoolSize="1" />
```

```

        <!-- C dataLogger, radius- ( ) -->
        <bean name="radiusDataLogger" class="ru.bitel.bgbilling.modules.inet.radius.
RadiusHourlyDataLogger">
            <param name="scheduledExecutor">hrlydtlggr</param>
        </bean>

        <!-- C RADIUS-          dataLogger -->
        <bean name="radiusListener" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusListener">
            <constructor>
                <!-- (), . - -->
                <param name="host" value=""/>
                <!-- , -->
                <param name="port" value="1812"/>
                <!-- -->
                <param name="recvBufferSize">512 * 1024</param>
                <!-- SO_RCVBUF -->
                <param name="soRCVBUF"></param>
                <!-- - -->
                <param name="threadCount">10</param>
                <!-- -->
                <param name="maxQueueSize">200</param>
                <!-- -->
                <param name="processor">radiusProcessor</param>
                <!-- , RadiusListener.Mode.authentication -->
                <param name="mode">RadiusListener.Mode.authentication</param>
                <!-- dataLogger -->
                <param name="dataLogger">radiusDataLogger</param>
            </constructor>
        </bean>
    </context>

    <context name="dhcp">
        <!-- C dhcp- -->
        <bean name="dhcpProcessor" class="ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor"/>

        <scheduledExecutorService name="hrlydtlggr" corePoolSize="1" />

        <!-- C dataLogger, DHCP- -->
        <bean name="dhcpDataLogger" class="ru.bitel.bgbilling.modules.inet.dhcp.DhcpHourlyDataLogger">
            <param name="scheduledExecutor">hrlydtlggr</param>
        </bean>

        <!-- C dhcp-          dataLogger -->
        <bean name="dhcpListener" class="ru.bitel.bgbilling.kernel.network.dhcp.DhcpListener">
            <constructor>
                <!-- (), . - -->
                <param name="host" value=""/>
                <!-- , -->
                <param name="port" value="10067"/>
                <!-- -->
                <param name="recvBufferSize">512 * 1024</param>
                <!-- - -->
                <param name="threadCount">10</param>
                <!-- -->
                <param name="maxQueueSize">200</param>
                <!-- -->
                <param name="processor">dhcpProcessor</param>
                <!-- dataLogger -->
                <param name="dataLogger">dhcpDataLogger</param>
            </constructor>
        </bean>
    </context>
</application>

```

Настройка BGInetAccounting сервера

Пример типовой настройки, которая идет в дистрибутиве:

Код

```
<application context="accounting">
  <!-- -->
  <param name="app.name" value="BGInetAccounting"/>
  <!-- id -->
  <param name="app.id" value="" />

  <!-- -->
  <param name="db.driver" value="com.mysql.jdbc.Driver"/>
  <param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&queryTimeoutKillsConnection=true&connectTimeout=1000"/>
  <param name="db.user" value="bill"/>
  <param name="db.pswd" value="bgbilling"/>
  <param name="db.validationTimeout" value="10"/>

  <!-- MQ -->
  <param name="mq.url" value="failover:(tcp://localhost:61616)"/>
  <param name="mq.user" value="bill"/>
  <param name="mq.pswd" value="bgbilling"/>

  <!-- -->
  <param name="server.apps.type" value="inet"/>
  <param name="server.update.url" value="http://localhost:8080/bgbilling"/>
  <param name="server.update.check.ssl" value="no"/>

  <!-- id -->
  <param name="moduleId" value="" />
  <!-- id -->
  <param name="rootDeviceId" value="" />

  <!-- , -->
  <param name="commonIdentifierName" value="rootDeviceId"/>

  <!-- radius- -->
  <!-- , radius -->
  <param name="datalog.radius.dir" value="data/radius" />
  <!-- , -->
  <param name="datalog.radius.chunk.size" value="524288" />
  <!-- radius : 0 - , 1 - zlib -->
  <param name="datalog.radius.compression.type" value="1" />
  <!-- flow- -->
  <!-- , flow -->
  <param name="datalog.flow.dir" value="data/flow" />
  <!-- , -->
  <param name="datalog.flow.chunk.size" value="524288" />
  <!-- flow : 0 - , 1 - zlib -->
  <param name="datalog.flow.compression.type" value="1" />

  <!-- Accounting -->
  <bean name="accounting" class="ru.bitel.bgbilling.modules.inet.accounting.Accounting"/>

  <context name="radius">
    <!-- C RADIUS- -->
    <bean name="radiusProcessor" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor"
  />

  <!-- ScheduledExecutorService, dataLogger -->
  <scheduledExecutorService name="hrlydtlgr" corePoolSize="1"/>

  <!-- C dataLogger, RADIUS- ( ) -->
  <bean name="radiusDataLogger" class="ru.bitel.bgbilling.modules.inet.radius.
RadiusHourlyDataLogger">
```

```

        <param name="scheduledExecutor">hrlydtlggr</param>
    </bean>

    <!-- C radius-          dataLogger -->
    <bean name="radiusListener" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusListener">
        <constructor>
            <!-- (), . - -->
            <param name="host" value=""/>
            <!-- , -->
            <param name="port" value="1813"/>
            <!-- -->
            <param name="recvBufferSize">1 * 1024 * 1024</param>
            <!-- SO_RCVBUF -->
            <param name="soRCVBUF"></param>
            <!-- - -->
            <param name="threadCount">10</param>
            <!-- -->
            <param name="maxQueueSize">200</param>
            <!-- -->
            <param name="processor">radiusProcessor</param>
            <!-- , RadiusListener.Mode.accounting -->
            <param name="mode">RadiusListener.Mode.accounting</param>
            <!-- setup -->
            <param name="setup">setup</param>
            <!-- dataLogger -->
            <param name="dataLogger">radiusDataLogger</param>
        </constructor>
    </bean>
</context>

    <!-- C flow- -->
    <context name="collector">
        <!-- ScheduledExecutorService, dataLogger -->
        <scheduledExecutorService name="hrlydtlggr" corePoolSize="1"/>

        <!-- C dataLogger, flow- ( ) -->
        <bean name="flowDataLogger" class="ru.bitel.bgbilling.modules.inet.collector.IPHourlyDataLogger"

            <param name="scheduledExecutor">hrlydtlggr</param>
        </bean>

        <!-- C flow-          dataLogger -->
        <bean name="flowListener" class="ru.bitel.bgbilling.modules.inet.collector.InetFlowListener">
            <constructor factoryMethod="newInstance">
                <!-- , netflow, netflow9, ipfix sflow -->
                <param name="type" value="netflow"/>
                <!-- (), . - -->
                <param name="host" value=""/>
                <!-- , -->
                <param name="port" value="2001"/>
                <!-- -->
                <param name="recvBufferSize">4 * 1024 * 1024</param>
                <!-- SO_RCVBUF -->
                <param name="soRCVBUF">512 * 1024</param>
                <!-- - -->
                <param name="threadCount" value="10"/>
                <!-- id -, -->
                <param name="agentDeviceIds" value=""/>
                <!-- dataLogger -->
                <param name="dataLogger">flowDataLogger</param>
            </constructor>
        </bean>

        <!--
        <bean name="flowListener" class="ru.bitel.bgbilling.modules.inet.collector.InetFlowListener">
            <constructor factoryMethod="newInstance">
                <param name="type" value="netflow9"/>
                <param name="host" value=""/>
                <param name="port" value="9367"/>
                <param name="recvBufferSize">4 * 1024 * 1024</param>
                <param name="soRCVBUF">512 * 1024</param>

```

```
        <param name="threadCount" value="8"/>
        <param name="agentDeviceIds" value="4"/>
        <param name="dataLogger">flowDataLogger</param>
    </constructor>
</bean>

<bean name="flowListener" class="ru.bitel.bgbilling.modules.inet.collector.InetFlowListener">
    <constructor factoryMethod="newInstance">
        <param name="type" value="netflow"/>
        <param name="host" value=""/>
        <param name="port" value="9368"/>
        <param name="recvBufferSize">4 * 1024 * 1024</param>
        <param name="soRCVBUF">512 * 1024</param>
        <param name="threadCount" value="8"/>
        <param name="agentDeviceIds" value="20"/>
        <param name="dataLogger">flowDataLogger</param>
    </constructor>
</bean>
-->

<context name="detail">
    <!-- C flow- -->
    <bean name="detailWorker" class="ru.bitel.bgbilling.modules.inet.accounting.detail.
InetDetailWorker"/>
    </context>
</context>

</application>
```

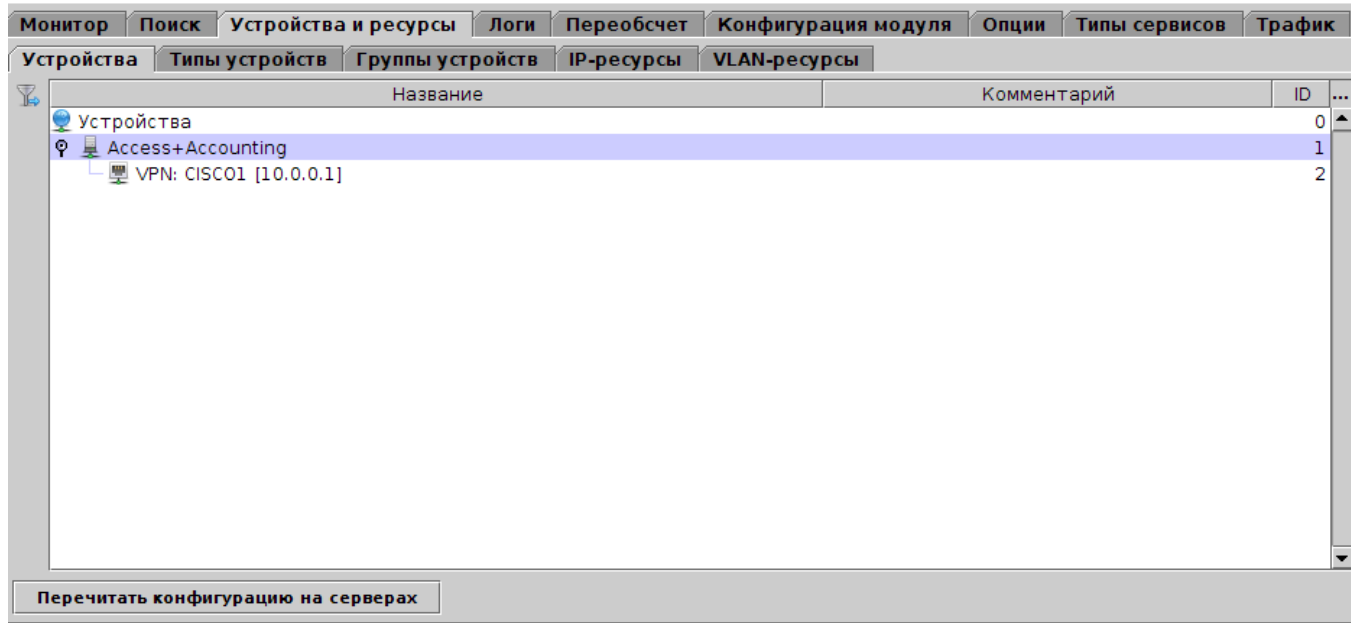
Общий алгоритм настройки

Это общая обзорная статья, описывающая общий алгоритм настройки модуля Inet, рекомендуется к прочтению. Далее можно перейти к примерам настроек [VPN](#) и [IPoE](#), в них некоторая информация дублируется, а некоторая представлена в большем объёме. Примеры более сложных схем, таких как Cisco ISG, RedBack Clips и описаны в нашей [wiki](#), но вначале рекомендуем изучить, понять и попробовать простые схемы, а потом уже приступить к более сложным.

Настройка начинается с того, что заводится корневое [устройство](#) с отдельным [типом](#) (обычно тип и устройство называют Access+Accounting). Это устройство является общим, с него не собирают трафик, не дают через него доступ и т.п. Это все лишь корень дерева, который может содержать некоторые общие настройки для приложений InetAccess и InetAccounting.

И это тот корень, который должен быть прописан в [rootDeviceId](#) настроек InetAccess/InetAccounting серверов.

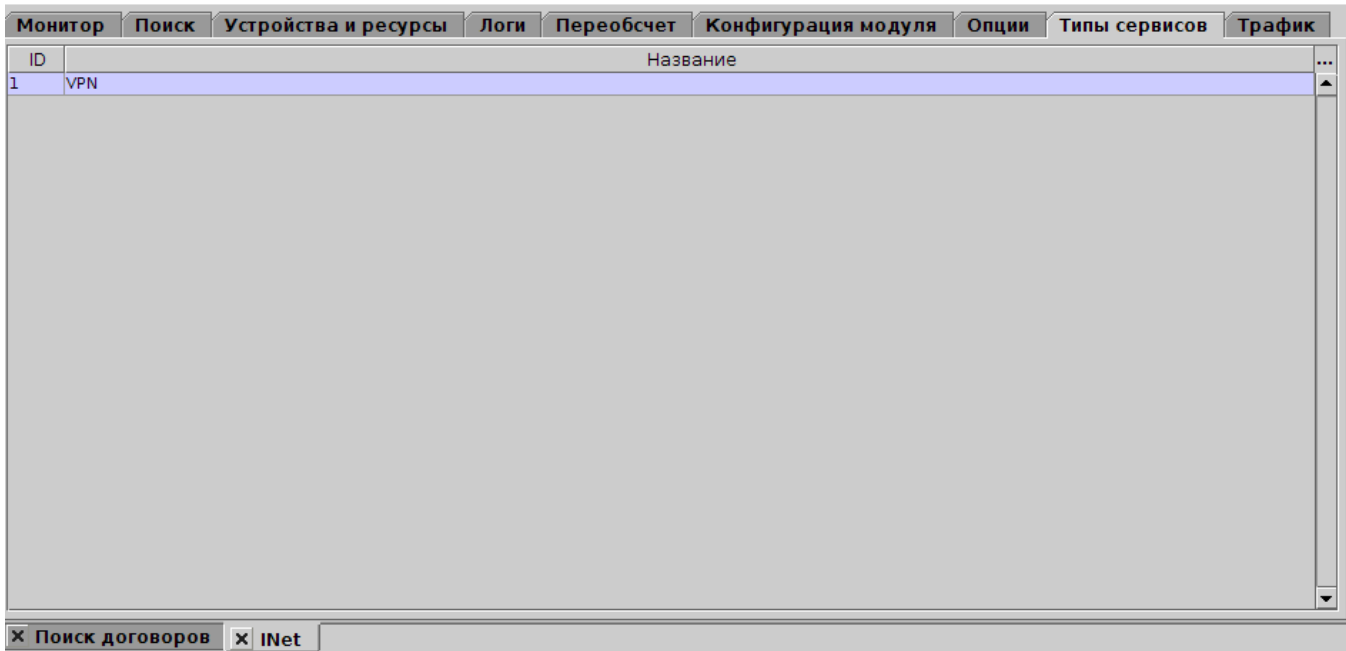
[Access](#) и [Accounting сервера](#) должны быть обязательно установлены и настроены (даже если в них нет ни одного слушателя, и Вы не используете RADIUS/Netflow/DHCP).



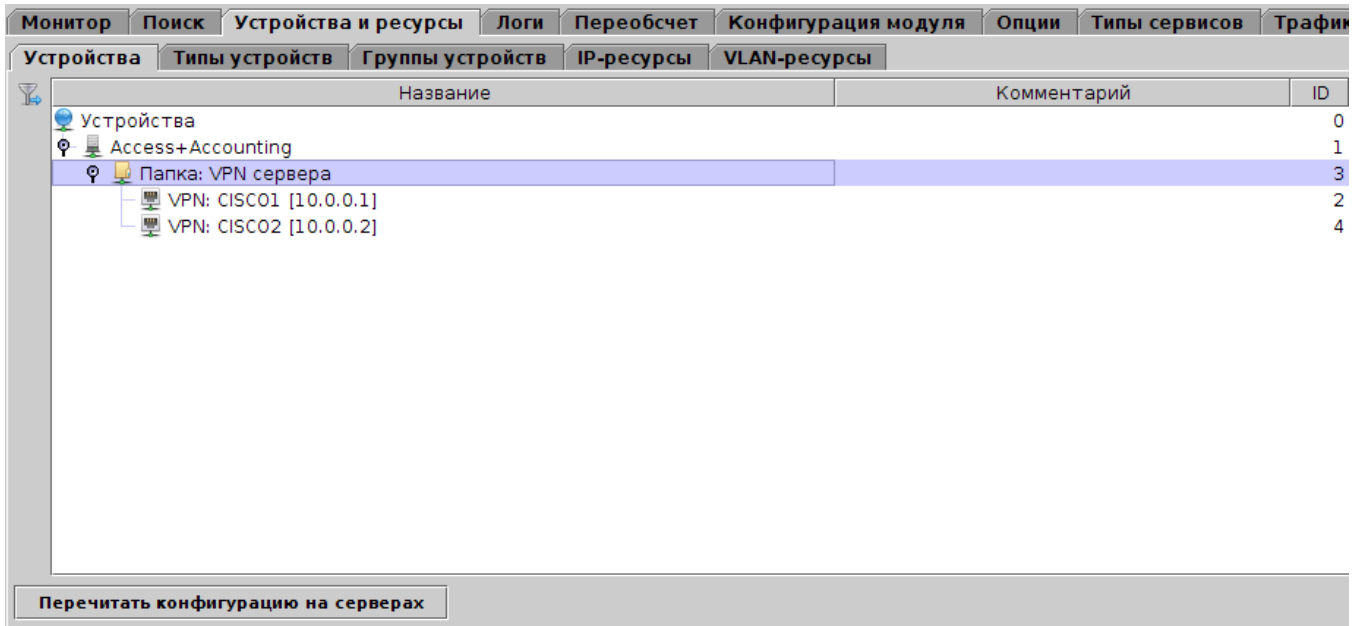
The screenshot shows a web-based configuration interface for a network device. At the top, there are several tabs: "Монитор", "Поиск", "Устройства и ресурсы", "Логи", "Переобсчет", "Конфигурация модуля", "Опции", "Типы сервисов", and "Трафик". Below these, there are sub-tabs: "Устройства", "Типы устройств", "Группы устройств", "IP-ресурсы", and "VLAN-ресурсы". The main area displays a tree view of the configuration hierarchy. The root node is "Устройства" (ID 0). Under it is a child node "Access+Accounting" (ID 1), which is highlighted in blue. Under "Access+Accounting" is a further child node "VPN: CISCO1 [10.0.0.1]" (ID 2). At the bottom of the interface, there is a button labeled "Перечитать конфигурацию на серверах".

Далее заводится дочернее устройство с отдельным [типом](#). В общем случае в иерархии может быть сколько угодно устройств. Но это минимальная конфигурация.

Далее заводится [тип сервиса](#).



Важно чтобы сервис на договоре был привязан к какому-либо устройству. Для этого в типе сервиса либо указывается галочка Устройство (тогда каждый раз придётся указывать устройство при добавлении сервиса на договор), либо в конфигурации типа сервиса указана переменная `const.device.id` (в этом случае устройство с данным типом будет привязано автоматически). В случае VPN, например, часто договора привязаны к корневому устройству Access+Accounting или к одному NAS-у, поэтому имеет смысл указывать `const.device.id`. Также можно объединить NAS'ы в одну общую папку и прописать в `const.device.id` ID этой папки.



Далее добавляем сервис на договор.

Параметры Отчет Карточки

Договор

test [Иванов Иван Иванович] 1

Период с 26.11.2011 по

Лицо физическое

Баланс (ноябрь): 100,00

Входящий остаток: 100,00

Приход: 0,00

Наработка: 0,00

Расход: 0,00

Возврат: 0,00

Исходящий остаток: 100,00

Резерв: 0,00

Доступная сумма: 100,00

Тарифные планы

Модули

- Bill
- Dialup
- INet**
- Nray
- Web

Скрытый

Сервисы Учётные периоды

Сервис	Тип	Устройство	Период	Статус	Состоян...	Коммент...	ID
Сервисы							
ivanov VPN	VPN	Папка: VPN сервера	03.12.2013-...	открыт	удален		1

Поиск договоров x INet x test

Вид сервиса зависит от переменной `title.pattern` в типе сервиса(после исправления переменной нужно пересохранить сервис). В самом сервисе может быть в общем случае что угодно (логин, устройство и т.п.), зависит все это от настроек типа сервиса. Более сложные схемы позволяют заводить дочерние сервисы.

Если нужно указать статический IP-адрес на сервисе, то не забываем про `IP ресурсы` и переменную `ip.resource.categoryId`.

Для выдачи динамических IP-адресов не забываем про параметр `radius.realm.<realm>.ipCategories` в конфигурации устройства.

Тут мы должны создать (если нам это нужно) **типы трафика**. И **привязку** этих типов трафика. Привязка трафика может быть либо по Netflow /sFlow или RADIUS. Про настройку привязки читайте [тут](#). Пример привязки для Radius есть [тут](#), для netflow - [тут](#). Новую привязку нужно указать в **типе сервиса**. Естественно до настройки привязки не забываем добавить сами **типы трафиков**.

Если вы не хотите учитывать трафик вообще(не по netflow не по radius), то можете этот момент пропустить, либо вернуться в нему позже.

Далее заводим простой тариф ([пример тарифа](#)) . В тарифе должны быть цены и услуги для всех типов трафика(в том числе и для типа трафика Время). И добавляем тариф на договор .

После изменения тарифа не забываем выбрать в контекстном меню **"Оповещение об изменениях"**, чтобы Access и Accounting сервера об этом узнали.

В этом месте, если ещё не настроили, то можно настроить и запустить **Access и Accounting сервера**. Если они уже запущены, то стоит зайти в **деповое устройство** и нажать там кнопку **"Перечитать конфигурацию на серверах"** или перезапустить Access и Accounting чтобы они получили новые данные.

После изменений в дереве устройств, в типе устройств, в типе сервиса нужно нажимать кнопку **"Перечитать конфигурацию на серверах"** в дереве устройств.

При нормальной работе access сервера состояние сервиса на договоре (не путать со статусом) должно смениться с **удален** на **подключен**(если баланс больше лимита). В случае VPN можно проверить установку сессий.

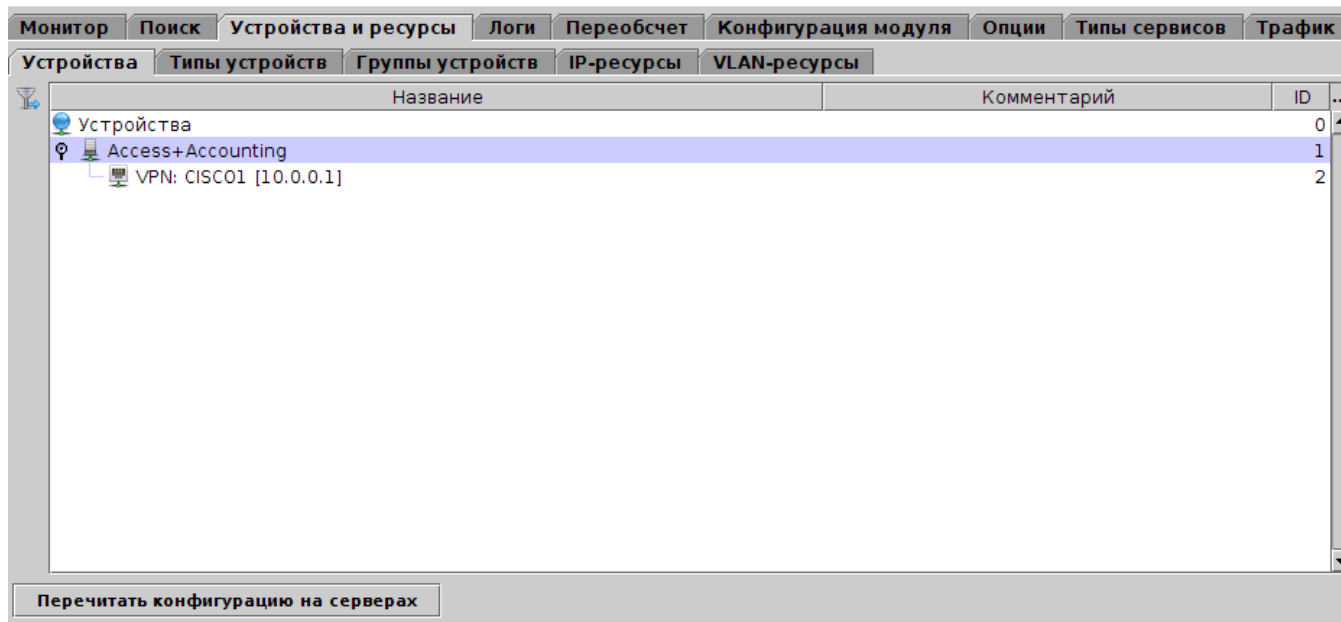
В случае если нужно управлять доступом в зависимости от состояния баланса и статуса договором(что фактически отражено в состоянии сервиса), то нужно чтобы на **типе сервиса** стоял **обработчик активации сервисов**. Есть несколько стандартных обработчиков, которые есть сразу в выпадающем списке в типе сервиса, если они не подходят, то можно создать свой. Есть примеры на нашей [wiki](#). Пример настройки подобной схеме есть в главе про **IPoE**.

Если нужно в нужно менять какие-то скорости или другие параметры, то это делается с помощью **опции** (не путать с тарифными опциями) в тарифе. [Пример тарифа с опциями](#). Привязка **опции к конкретным атрибутам радиуса** указывается в конфигурации устройства (параметр `radius.inetOption.x`). Если опция меняет параметры доступа, то **обработчик активации сервисов** должен обрабатывать её смену в соответствующем методе.

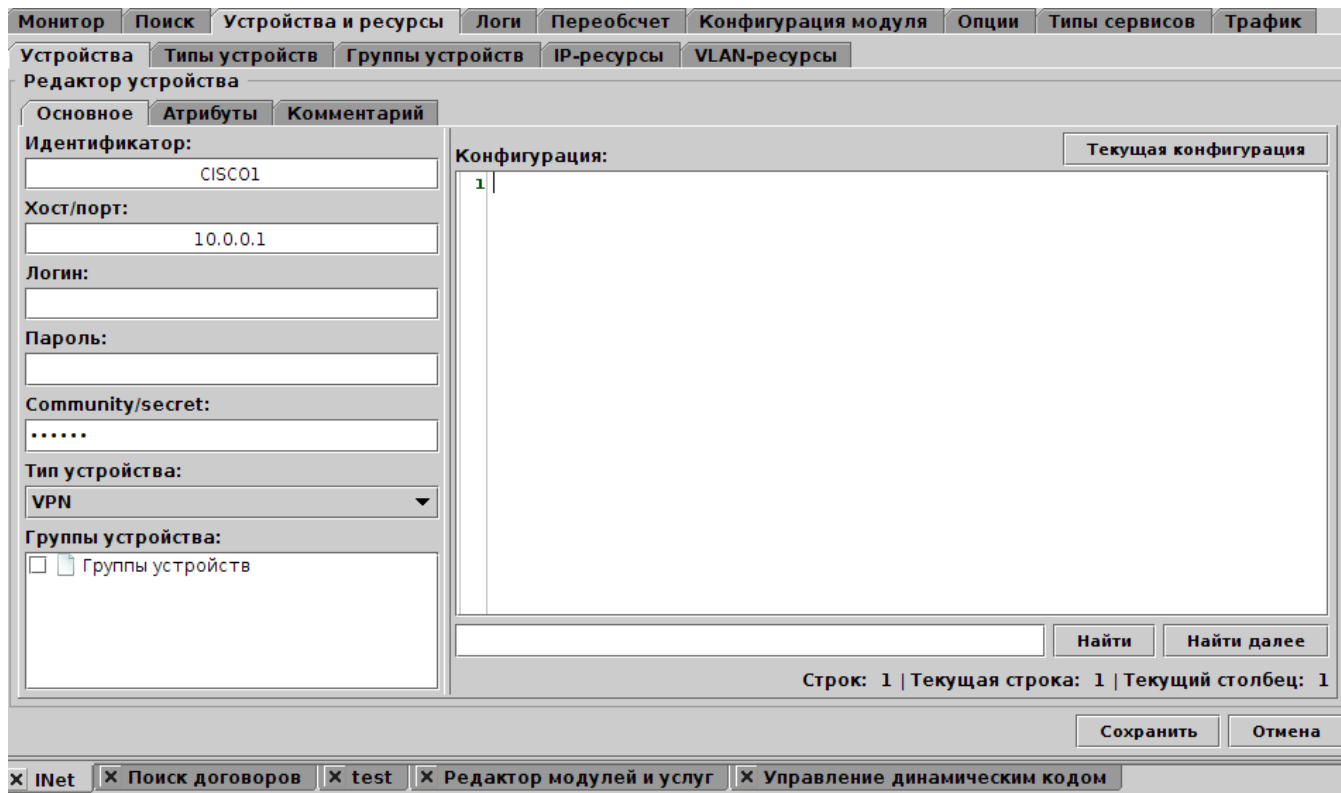
В случае VPN если нужно менять опции на лету, на уже установленных соединениях с помощью Radius Coa запросов и сбрасывать сессий из биллинга с помощью Radius Pod запросов, то нужно использовать обработчик активации сервиса `ru.bitel.bgbilling.modules.inet.dyn.device.radius.CoAServiceActivator`(он есть среди доступных).

Пример настройки VPN доступа (PPPoE/PPTP)

Заводится [тип устройства](#) VPN. И [Устройство](#) этого типа как дочернее к Access+Accounting.

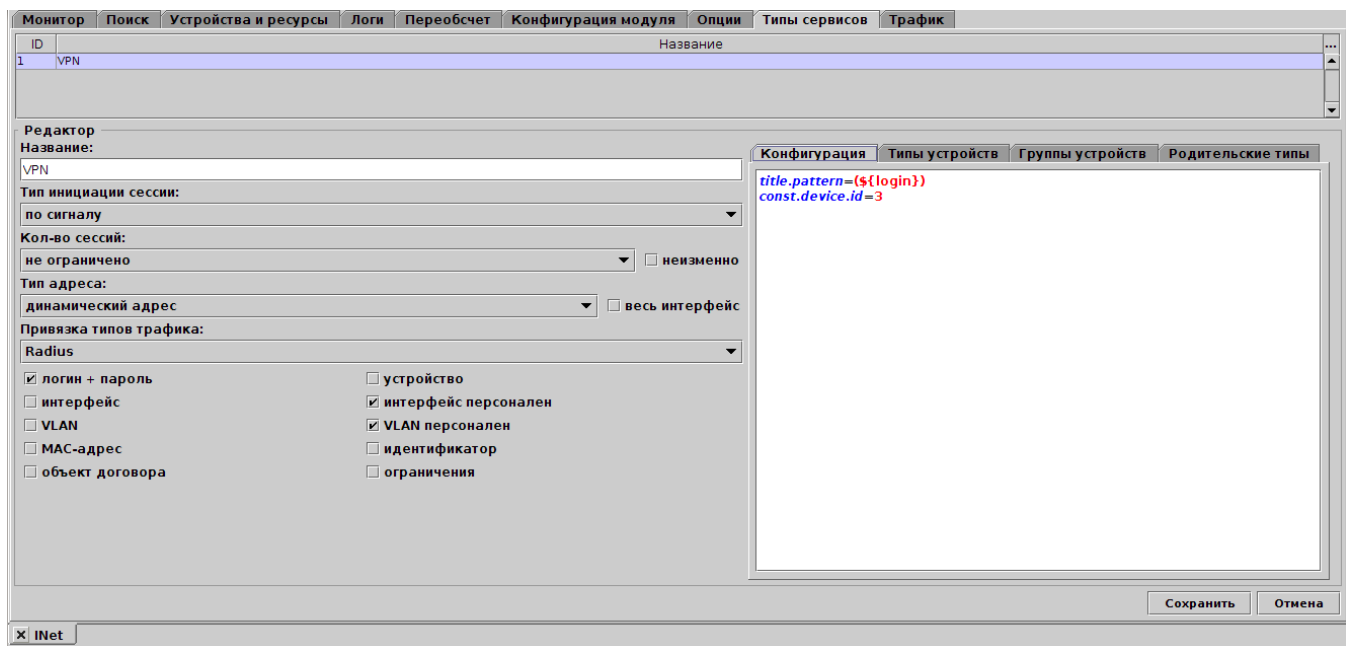


Тут желательно ещё раз перечитать главу об [InetRadiusProcessor](#). В самом устройстве настроить



Тут в качестве идентификатора нужно указать значение из атрибута NAS-Identifier radius-запросов, в качестве Хост/порт указать значение из атрибута NAS-IP-Address radius-запросов. Должно быть обязательно заполнено одно из этих полей (при наличии соответствующего атрибута), иначе система не сможет определить к какому устройству привязать radius-пакет. Подробнее об этом описано [здесь](#).

Далее заводится [тип сервиса](#).

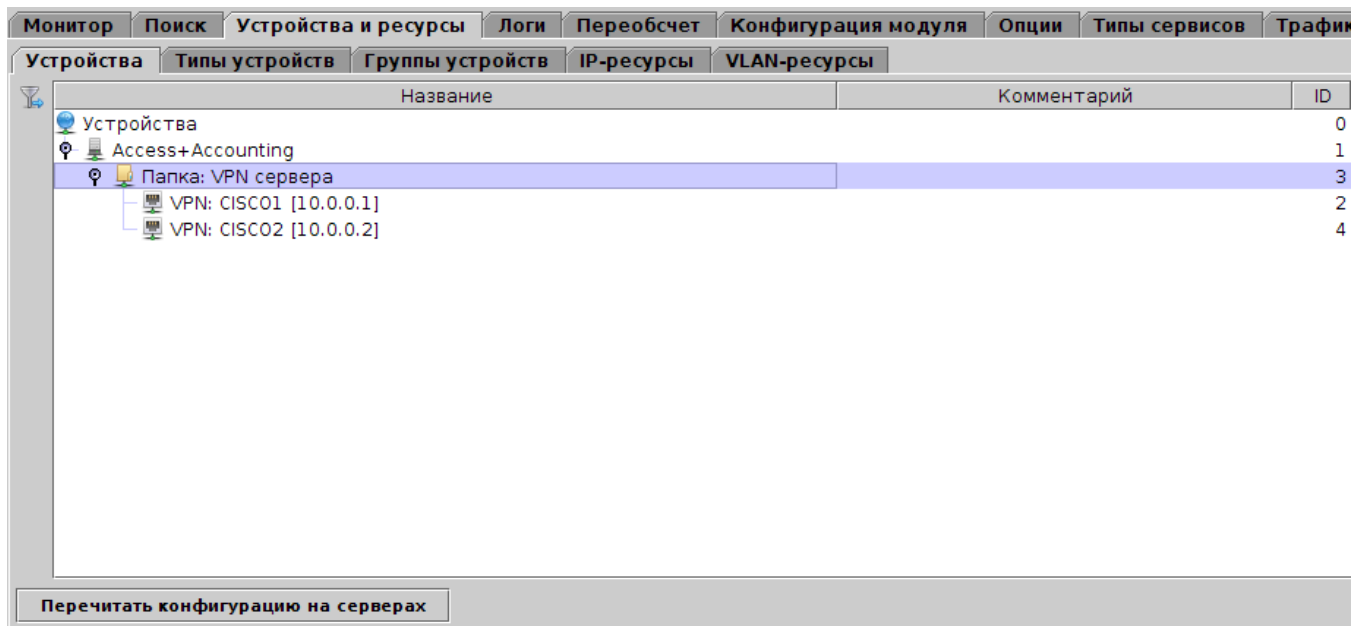


Тут тип инициации по сигналу, так как сессия стартует по radius access запросу. Тип адреса - **динамический адрес**. Про статический описано ниже. Как настраивать привязки типов трафика описано [тут](#).

Привязку Radius нужно указывать, если учёт трафика идёт по Radius протоколу. Как настроить конкретный пример для привязки Radius описано [ниже](#). В случае, если подсчёт трафика идёт по Netflow, указываете привязку **Netflow**. О том, как её настроить, описано [тут](#). На первом этапе пока можете не указывать привязку и вернуться в ней позже.

Ставите галочку возле поля **Логин + Пароль**. Если вы хотите при добавлении сервиса на договор каждый раз указывать устройство явно, к которому оно привязано, то ставите галочку "устройство". Но в случае VPN обычно чаще бывают такие варианты:

- 1) Есть всего один сервер NAS и к нему привязаны все сервисы. В этом случае проще указать параметр **const.device.id** в типе сервиса и галочку **устройство** не указывать. Тогда все сервисы этого типа будут автоматически привязаны к этому устройству.
- 2) Если несколько NAS-ов и сессия привязывается по факту к тому NAS-у, с которого она вышла, то все NAS объединяются в устройствах в одну общую папку и в **const.device.id** прописывается эта папка. Вот так это выглядит в устройствах.



Так же указываем **title.pattern** в типе сервиса.

Далее добавляем **сервис** на договор.

Сервисы **Учётные периоды**

Настройки **Опции** Дополнительно

Тип: VPN

Период: с 03.12.2013 по

Статус: Открыт

Кол-во сессий: не ограничено

Логин: ivanov авто

Пароль: авто

Комментарий

Состояние: Удален, активные опции: -

Сохранить Отмена

Параметры **Отчет** Карточки

Договор: test [Иванов Иван Иванович]

Период: с 26.11.2011 по

Сервисы **Учётные периоды**

Сервис	Тип	Устройство	Период	Статус	Состоян...	Коммент...	ID
Сервисы							
ivanov	VPN	Папка: VPN сервера	03.12.2013-...	открыт	удален		1

Лицо: физическое

- Баланс (ноябрь):
 - Входящий ос: 100,00
 - Приход: 0,00
 - Наработка: 0,00
 - Расход: 0,00
 - Возврат: 0,00
 - Исходящий о: 100,00
 - Резерв: 0,00
 - Доступная су: 100,00
- Тарифные плань
- Модули:
 - Bill
 - Dialup
 - INet**
 - Nray
 - Web

Скрытый

Поиск договоров INet test

Отображение сервиса(в данном примере показывает логин) зависит от переменной `title.pattern` в типе сервиса(после исправления переменной нужно пересохранить сервис).

Если нужно указать статический ip адрес на сервисе, то не забываем про IP ресурсы и переменную `ip.resource.categoryId`.

Далее заводим простой тариф ([примеры](#)). В тарифе должны быть цены и услуги для всех типов трафика. И добавляем тариф на договор .

Тут уже можно настроить `RadiusListener`-ы в Access и Accounting серверах(если еще не настроили). Перезапустить их, и попробовать авторизовать клиента.

Если нужно учитывать трафики по Radius, то вначале добавляем трафики

Монитор Поиск Устройства и ресурсы Логи Переобсчет Конфигурация модуля Опции Типы сервисов Трафик

Привязка типов трафика Типы трафика

ID	Название	Ед. измерения	...
0	Время	секунда	▲
1	Входящий	байт	
2	Исходящий	байт	

× INet × Поиск договоров × test × Редактор модулей и услуг × Управление динамическим кодом

Потом добавляем привязку для Radius

Монитор Поиск Устройства и ресурсы Логи Переобсчет Конфигурация модуля Опции Типы сервисов Трафик

Привязка типов трафика Типы трафика

Редактор

Название: radius

Тип	Период	Тип трафика	Комментарий	...
RADIUS	...	Входящий		▲
RADIUS	...	Исходящий		

Редактор правила

Тип: RADIUS с [] по []

Комментарий:

RADIUS

Реалм: [] ServiceName: []

Код вендора: Код атрибута: Префикс:

[-2] [1] []

Коллектор

Направление: Диапазон адресов:

любое [] - []

DiffServ/ToS: Диапазон портов:

[] - []

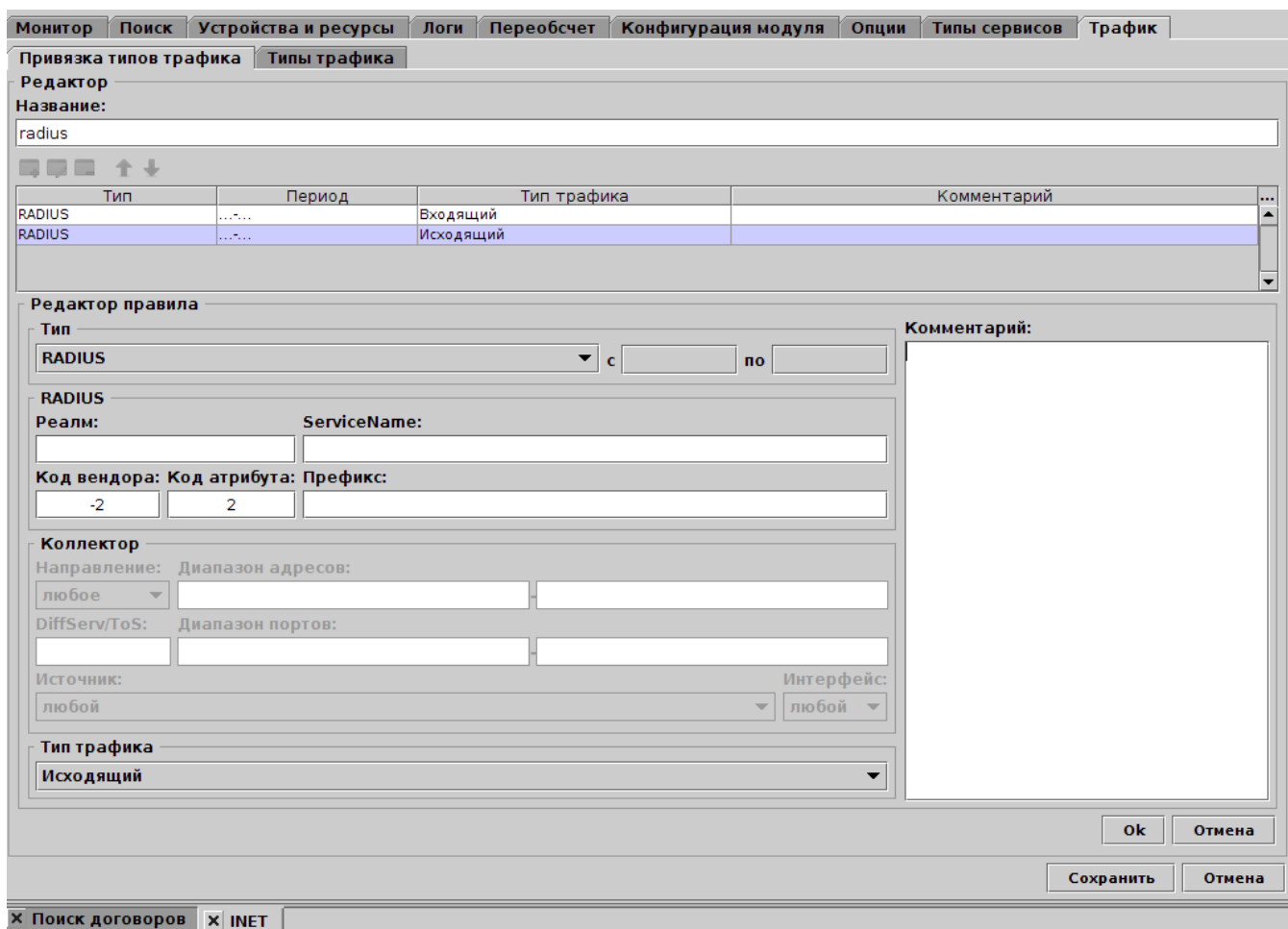
Источник: [любой] Интерфейс: [любой]

Тип трафика: Входящий

Ok Отмена

Сохранить Отмена

× Поиск договоров × INET



Если нужно настроить сбор трафика по netflow, то читаем вот эту [статью](#).

После настройки привязки, не забываем её указать в типе сервиса.

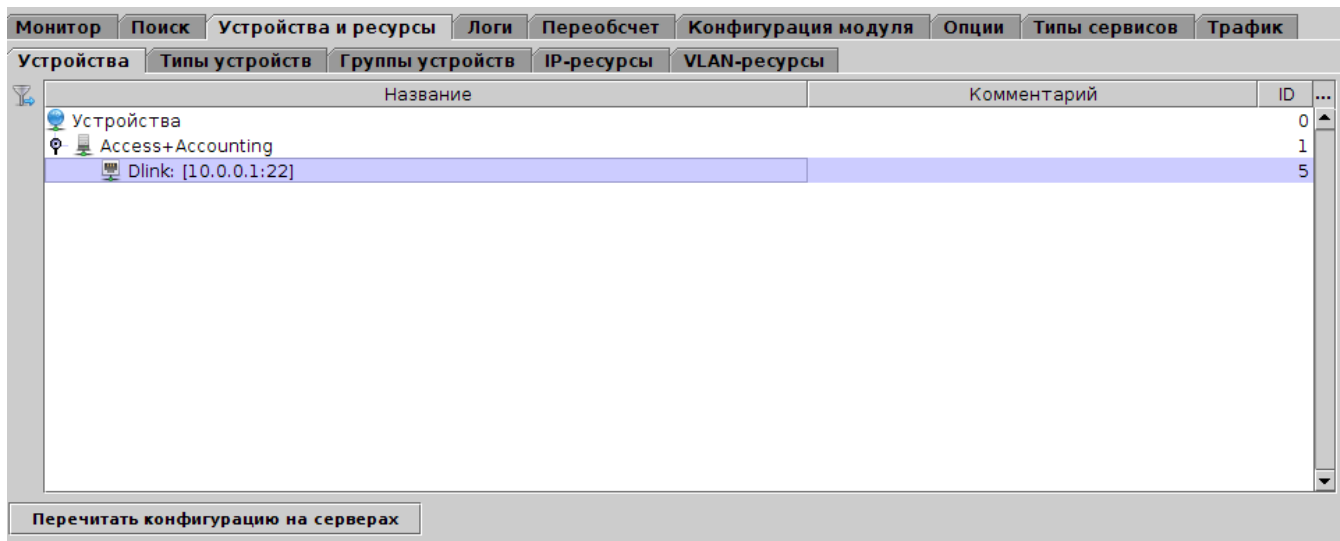
Если нужно менять какие-то скорости или другие параметры, то это делается с помощью [опции Inet](#) (не путать с тарифными опциями) в тарифе. Привязка [опции к конкретным атрибутам радиуса](#) указывается в конфигурации устройства (параметр **inet.option**). Для настройки корректного сброса VPN сессий (с помощью Radius Pod запросов) и для смены параметров(скорости) на лету, без установки новой сессии(Radius CoA-запрос) нужно в [типе устройства](#) поставить обработчик активации сервиса **ru.bitel.bgbilling.modules.inet.dyn.device.radius.CoAServiceActivator**(он есть среди доступных).

Пример настройки VPN соединения на основе MPD есть в нашей [wiki](#).

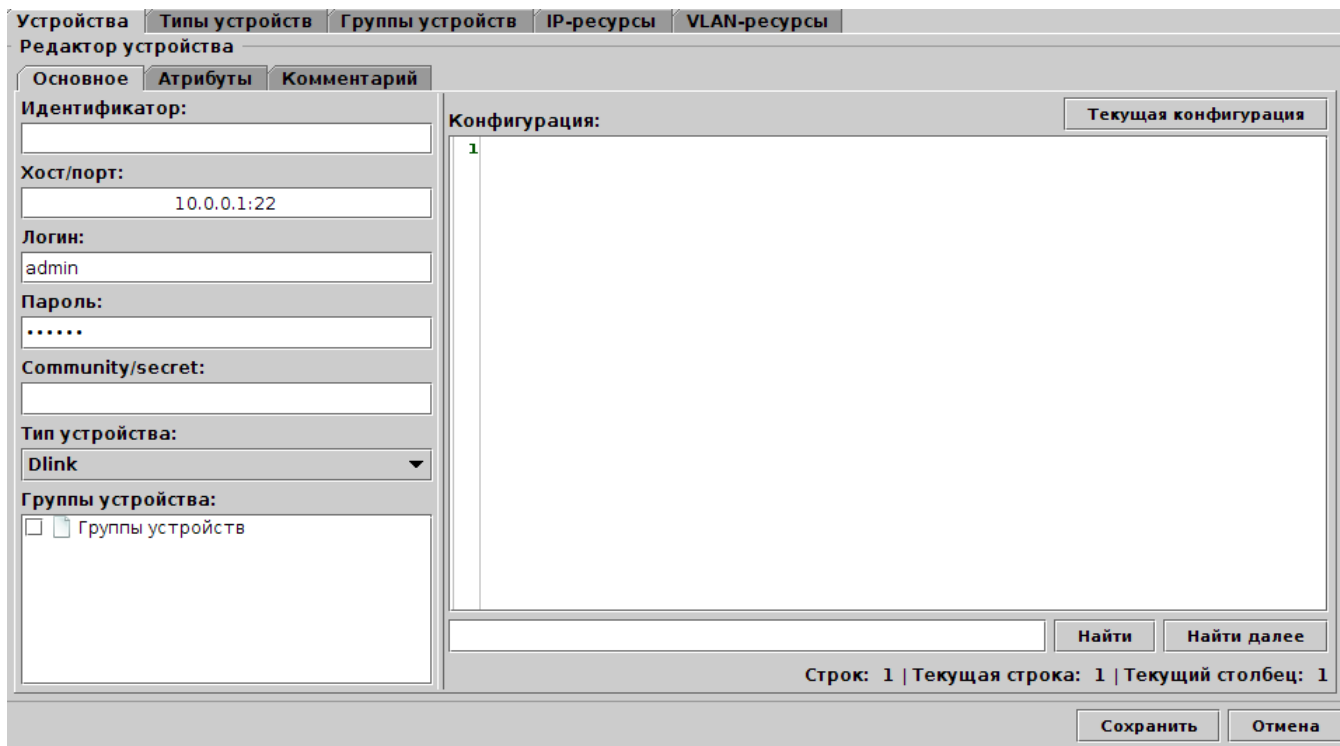
Пример настройки IPoE, управление доступом

Заводим новый [тип устройства](#), назовём его Dlink.

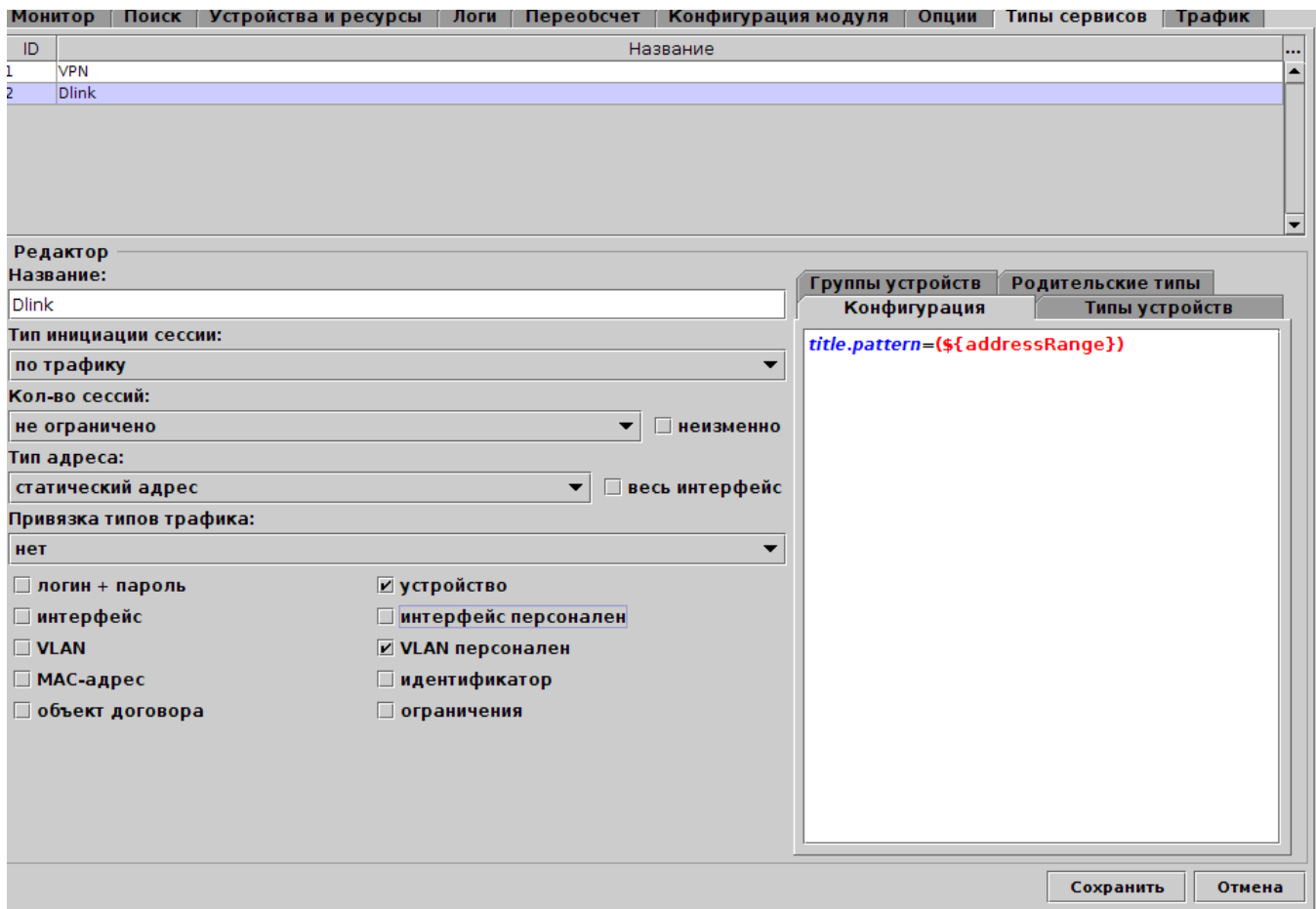
Добавляем [устройство](#) этого типа как дочернее к устройству Access+Accounting.



В самом устройстве прописываем хост:порт(порт ставим 22 для ssh), и логин/пароль пользователя для управления.



Заводим [тип сервиса](#):



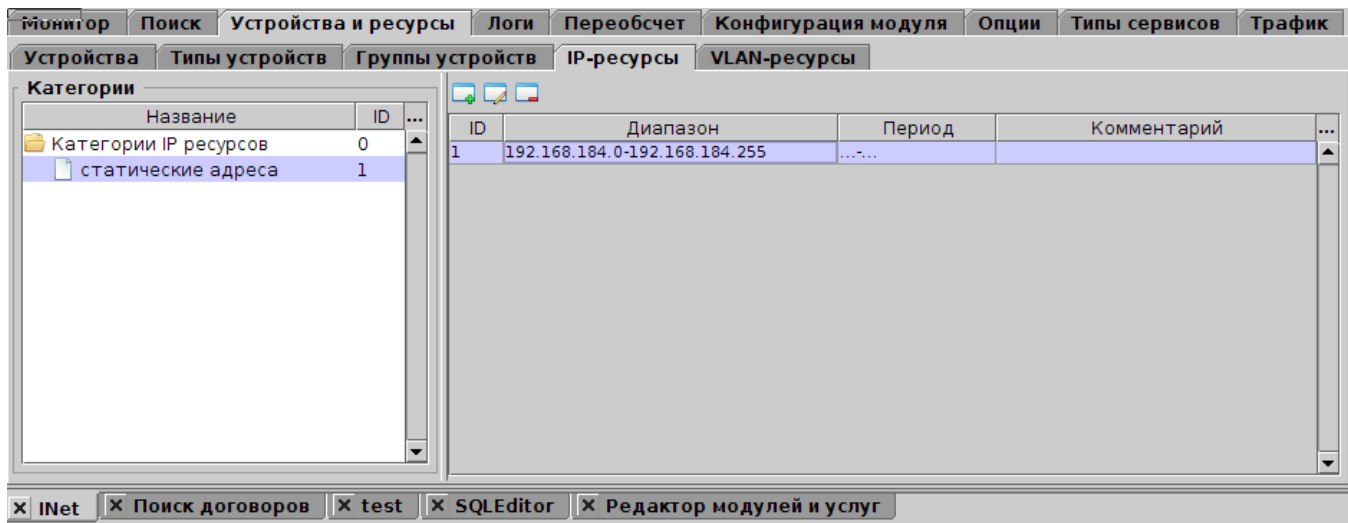
Тут тип инициации сессий:

- 1) по трафику(это если вы собираете трафик по netflow и хотите его учитывать).
- 2) По сигналу(Если вы хотите чтобы сессия стартовала по dhcp запросу).

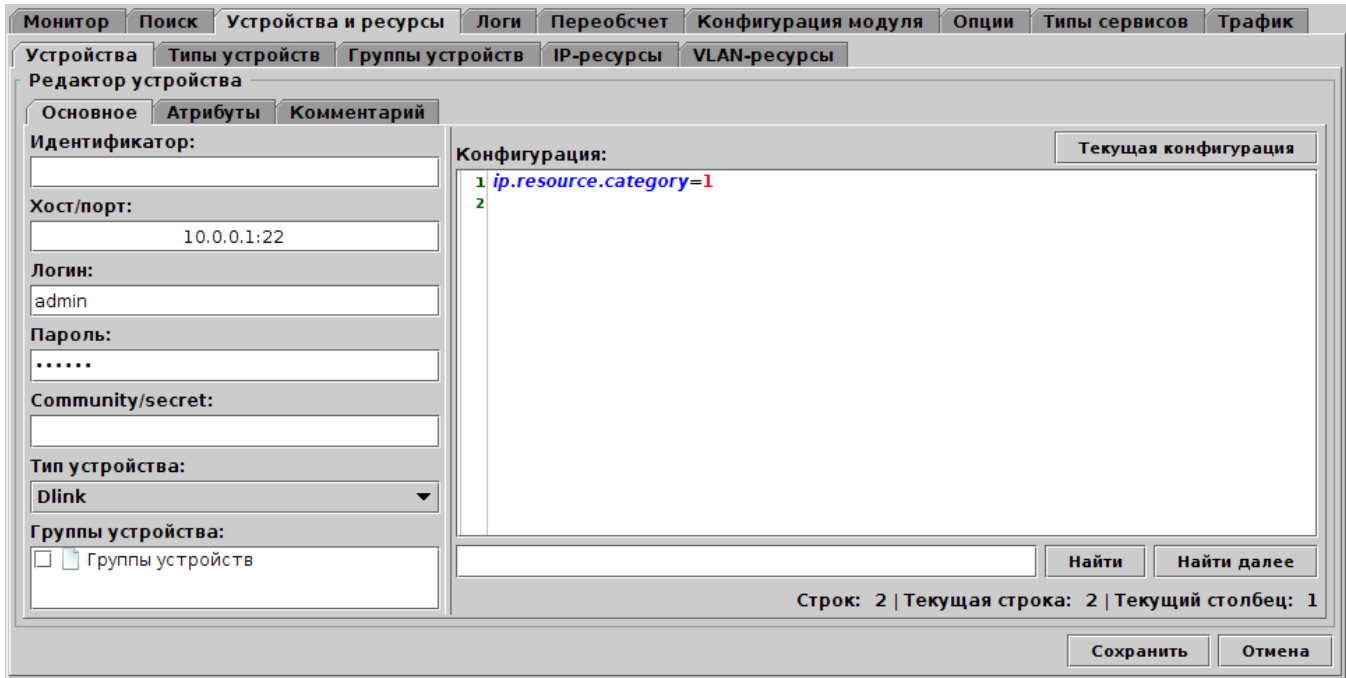
Так как режим какой-то должен быть выбран, то поставьте пока по трафику, даже если сессии вообще не будет(не будет не сигналов, ни трафика). Потом можно будет поменять по мере настройки.

Тип адреса - статический адрес(про динамический будет описано позже). Ставим галочку возле устройства. В `title.pattern` настроили так, чтобы отображался ip адрес.

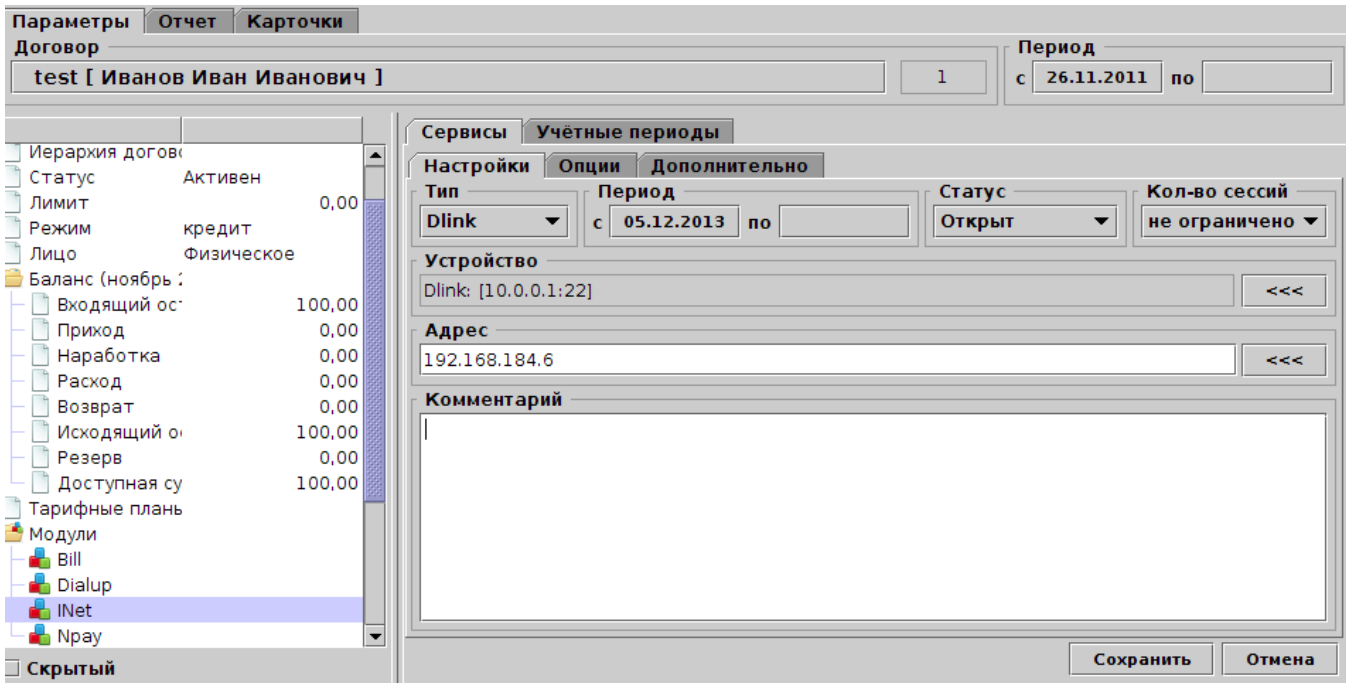
Так как мы выбрали статический ip, то нам надо создать для него категорию в [ресурсах ip-адресов](#).

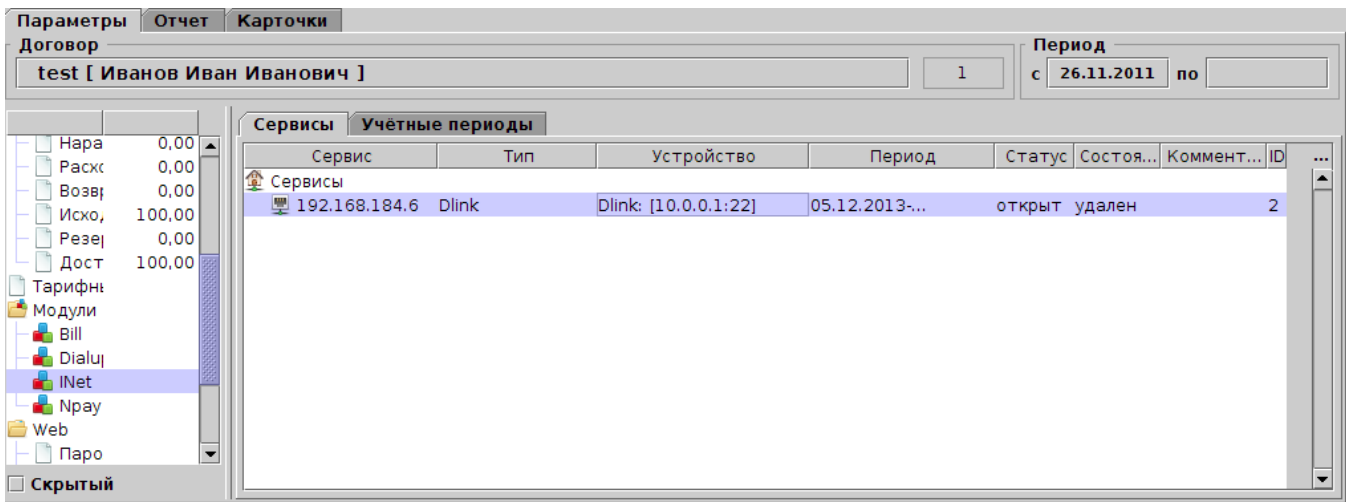


И не забываем id этой категории прописать в переменную `ip.resource.categoryId` в конфигурации устройства:



Добавляем сервис на договор, выбираем устройство и ip.





Для управления доступом у нас используется [обработчик активации сервисов](#), написанные на динамическом коде. Есть несколько стандартных обработчиков активации сервисов(доступные в стандартной поставке), такие как:

[ru.bitel.bgbilling.modules.inet.dyn.device.terminal.SSHServiceActivator](#) - Универсальный обработчик активации сервисов по ssh.

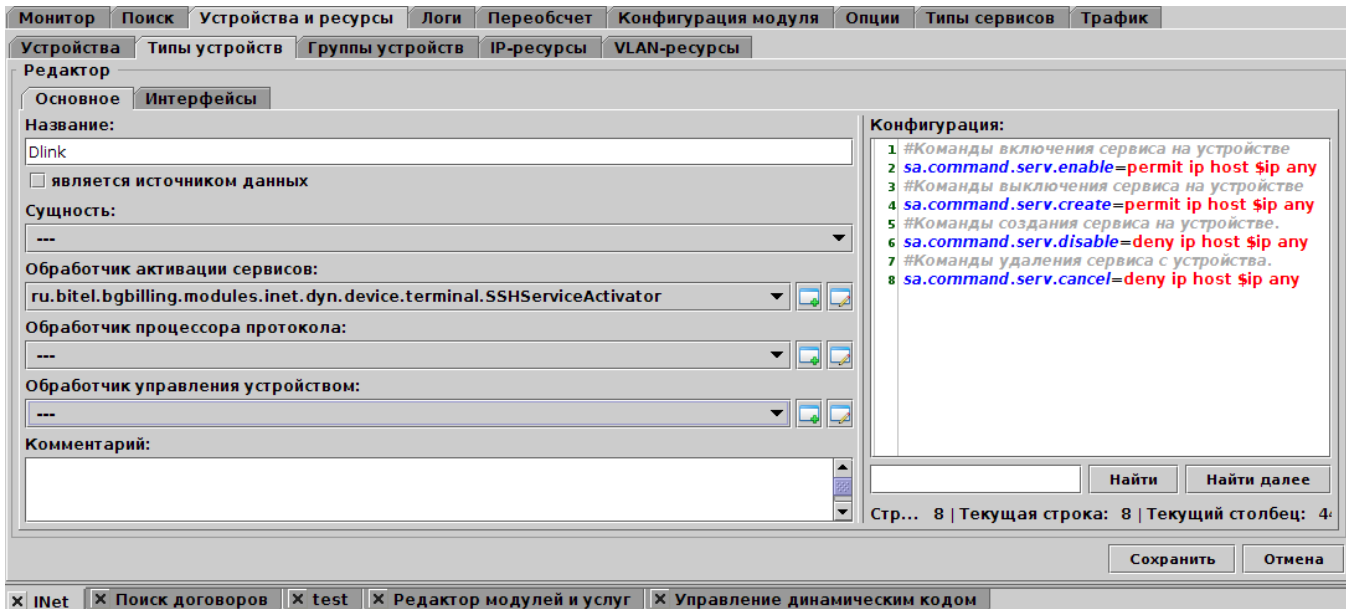
[ru.bitel.bgbilling.modules.inet.dyn.device.terminal.TelnetServiceActivator](#) - Универсальный обработчик активации сервисов по telnet.

[ru.bitel.bgbilling.modules.inet.dyn.device.snmp.SnmpServiceActivator](#) - Универсальный обработчик активации сервисов по snmp.

[ru.bitel.bgbilling.modules.inet.dyn.device.mikrotik.MikrotikServiceActivator](#) - Универсальный обработчик активации сервисов по для Mikrotik, работающий по протоколу Mikrotik Api.

Все они есть в стандартной поставке и описаны в [wiki](#). Есть так же другие .

Пусть для примера мы решили использовать ssh. Тогда мы выбираем [ru.bitel.bgbilling.modules.inet.dyn.device.terminal.SSHServiceActivator](#) в типе устройства.



И конфигурации его указываем команды на подключение и отключение.

```
#
sa.command.serv.enable=permit ip host $ip any
#
sa.command.serv.create=permit ip host $ip any
# .
sa.command.serv.disable=deny ip host $ip any
# .
sa.command.serv.cancel=deny ip host $ip any
```

Это команды указаны просто как пример. В данном случае при отправке команд переменная ip будет заменена на тот Ip адрес, который мы поставили в сервисе. Как задавать команды описано в описании [ru.bitel.bgbilling.modules.inet.dyn.device.terminal.SSHServiceActivator](#).

С этого момента можно попробовать проверить открытие/закрытие доступа на устройстве. Если вы не хотите собирать netflow и обчислять трафик, а просто хотите открывать/закрывать доступ на устройстве по ip или интерфейсу, то этого будет достаточно. Можно проверять доступ, меняя баланс на договоре(например расходами и платежами) так чтобы состояние сервиса изменилось (когда баланс больше лимита, то подключено, когда меньше - отключено) и смотреть на устройстве правильно ли отработали команды. В этом режиме не создаются никакие сессии, а только управляется доступ в зависимости от баланса. Этот режим можно использовать, например, для безлимитчиков, для которых не нужно собирать трафик. Так же состояние сервиса можно менять вручную из меню вызываемого по правой кнопкой мыши, это бывает полезно для отладки обработчика активации сервиса.

В случае управления по интерфейсу(открывать/закрывать доступ на интерфейсе) нужно будет выбрать его в типе сервиса и поменять команды на типе устройства (макрос **\$iface**). Все макросы [ru.bitel.bgbilling.modules.inet.dyn.device.terminal.SSHServiceActivator](#) активатора описаны в [wiki](#). Если вы хотите использовать например не ssh, а telnet, то все настраивается аналогично, только указывает другой порт в устройстве (23 вместо 22) и другой обработчик активации сервиса - [ru.bitel.bgbilling.modules.inet.dyn.device.terminal.TelnetServiceActivator](#). Для mikrotik есть отдельный обработчик [ru.bitel.bgbilling.modules.inet.dyn.device.mikrotik.MikrotikServiceActivator](#), работающий по mikrotik api(порт 8728). Для протокола snmp есть [ru.bitel.bgbilling.modules.inet.dyn.device.snmp.SnmpServiceActivator](#).

Если кроме управления, вы хотите ещё собирать статистику по netflow, то нужно указать ip-адрес в типе сервиса, указать тип инициации по трафику (чтобы сессия создавались по трафику) и в типе сервиса. И настроить netflow согласно этой [статье](#). Также в этом случае понадобится настроить тариф, в тарифе должны быть цены и услуги для всех типов трафика, и добавить тариф на договор. Если все будет нормально работать, то у вас должны появляться сессии с трафиком netflow.

Если вы хотите выдавать динамический ip с помощью dhcp option 82, то нужно настроить [InetDhcpListener](#) в Access-сервере. Настроить на устройстве переменные. Тип инициации сессии тогда в типе сервиса нужно установить "по сигналу" (она будет создаваться по dhcp запросу).

Сбор NetFlow/sFlow

В этой главе описана настройка сбора netflow для любой схемы.

Для начала нужно завести **типы трафика**:

ID	Название	Ед. измерения
0	Время	секунда
1	Входящий	байт
2	Исходящий	байт

и создать **привязку netflow**:

Редактор

Название: netflow

Тип	Период	Тип трафика	Комментарий
Коллектор	...	Исходящий	
Коллектор	...	Входящий	

Редактор правила

Тип: Коллектор с [] по []

RADIUS

Реалм: [] ServiceName: []

Код вендора: Код атрибута: Префикс: [] [] []

Коллектор

Направление: Диапазон адресов: [исходящее] [0.0.0.0] - [255.255.255.255]

DiffServ/ToS: Диапазон портов: [] [0] - [65535]

Источник: [любой] Интерфейс: [любой]

Тип трафика: [Исходящий]

Ok Отмена

Сохранить Отмена

Монитор Поиск Устройства и ресурсы Логи Переобсчет Конфигурация модуля Опции Типы сервисов Трафик

Привязка типов трафика Типы трафика

Редактор

Название: netflow

Тип	Период	Тип трафика	Комментарий
Коллектор	...	Исходящий	
Коллектор	...	Входящий	

Редактор правила

Тип: Коллектор с [] по []

Комментарий:

RADIUS

Реалм: [] ServiceName: []

Код вендора: [] Код атрибута: [] Префикс: []

Коллектор

Направление: Диапазон адресов: входящее [0.0.0.0] - [255.255.255.255]

DiffServ/ToS: Диапазон портов: [] [0] - [65535]

Источник: любой [] Интерфейс: любой []

Тип трафика: Входящий []

Ok Отмена

Сохранить Отмена

На изображении пример простой привязки, когда абсолютно весь трафик делится на входящий и исходящий.

На этом месте лучше прочитать статью про [InetFlowListener](#) и исходя из этого построить иерархию устройств и прописать привязку flow-агента [f low.agent.link](#). Если у вас устройство, инициирующее сессии, само является источником flow-потока, то можно создать простую иерархию:

Монитор Поиск Устройства и ресурсы Логи Переобсчет Конфигурация модуля Опции Типы сервисов Трафик

Устройства Типы устройств Группы устройств IP-ресурсы VLAN-ресурсы

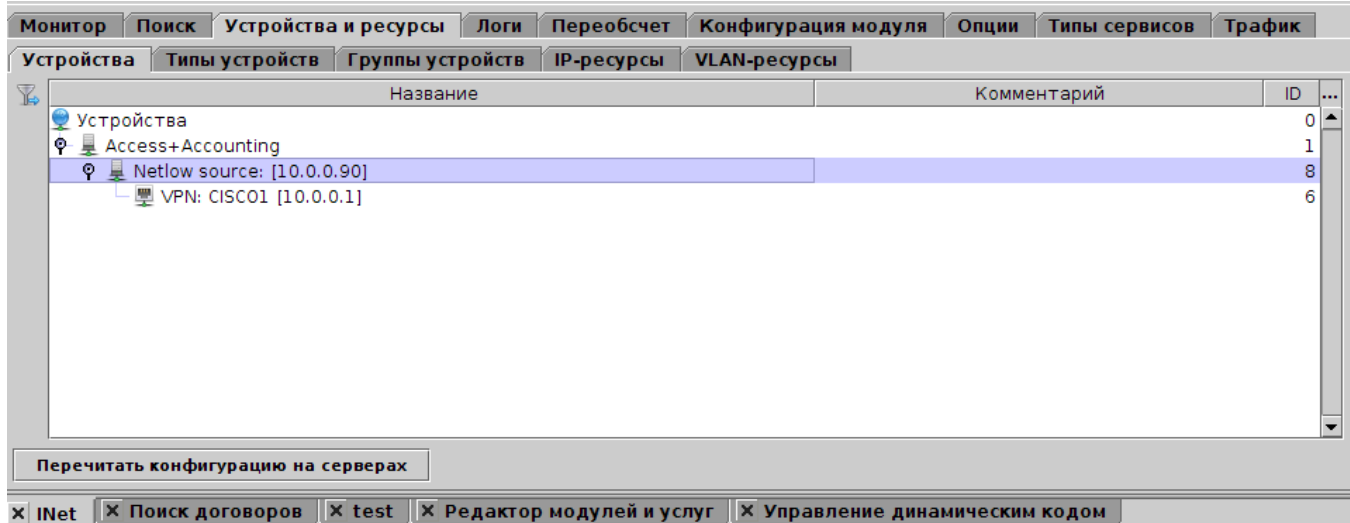
Название	Комментарий	ID
Устройства		0
Access+Accounting		1
VPN: CISCO1 [10.0.0.1]		2

Перечитать конфигурацию на серверах

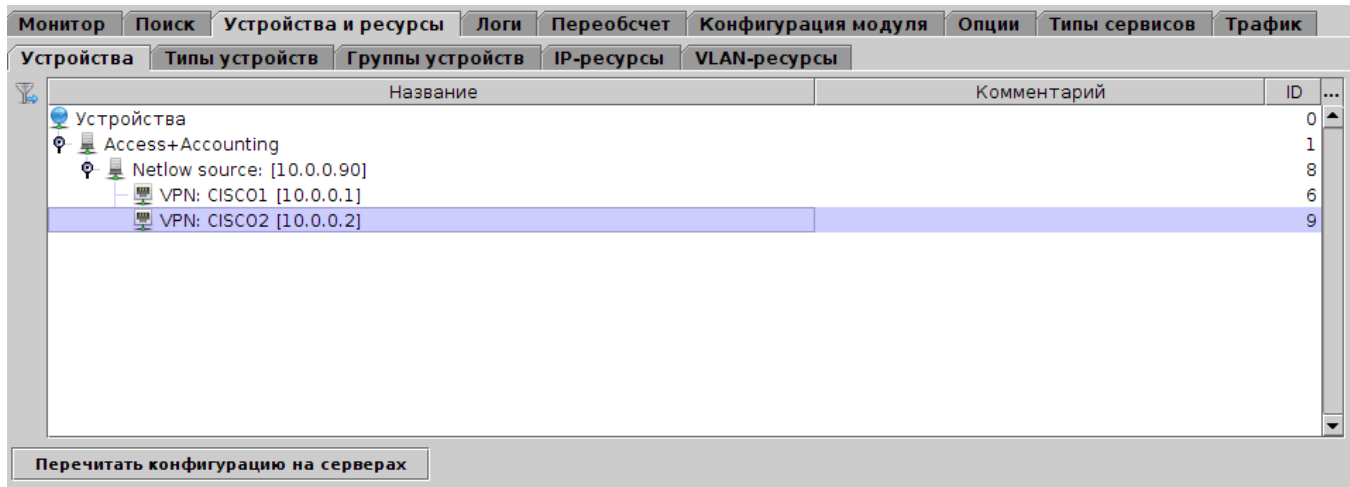
В этом случае в самом устройстве [flow.agent.link](#) можно не прописывать вообще. Если нужно указать конкретный интерфейс (например 10), то нужно прописать на устройстве свой id и конкретный интерфейс:

```
flow.agent.link=2:10
```

Если netflow собирается с какого-то другого IP (т.е. flow-агент это другая машина), то нужно добавить в иерархию выше ещё одно устройство - источник. В данном случае для этого устройства лучше сделать отдельный тип устройства Netflow source и добавить устройство этого типа в иерархию:



Также вполне возможен случай, когда flow-агент один для нескольких устройств, тогда он является промежуточным предком для нескольких устройств:



В случае отдельного flow-агента нужно либо непосредственно в конфигурации устройства flow-агента:

```
flow.agent.link={@deviceId}:-1
```

либо указать привязку flow-агента во всех дочерних устройствах:

```
flow.agent.link=8:-1
```

Netflow описывается одинаково для PPPoE и IPoE схем.

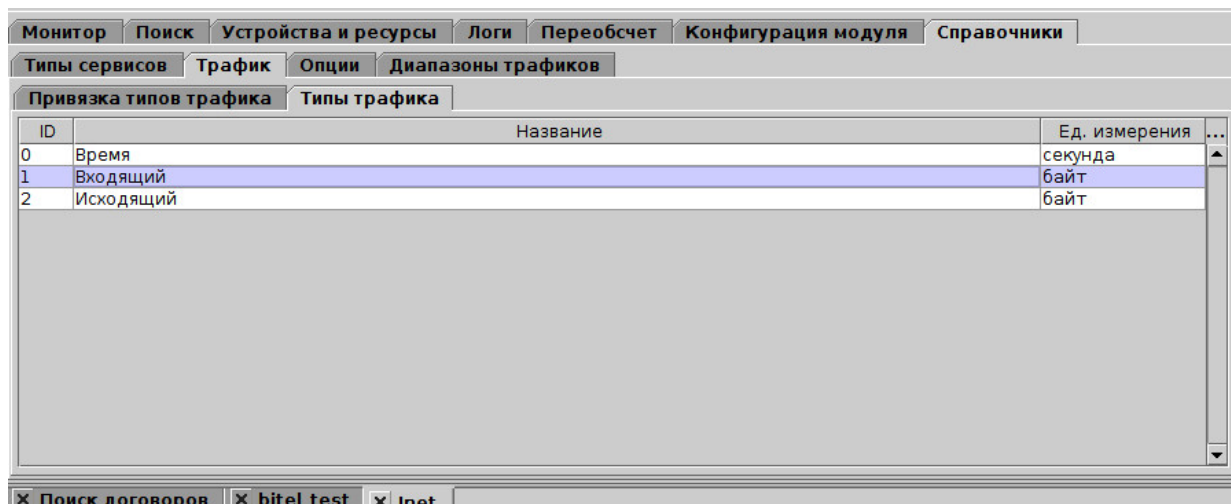
Для сбора netflow на Accounting-сервера нужно не забыть прописать [InetFlowListener](#).

Сбор статистики по snmp

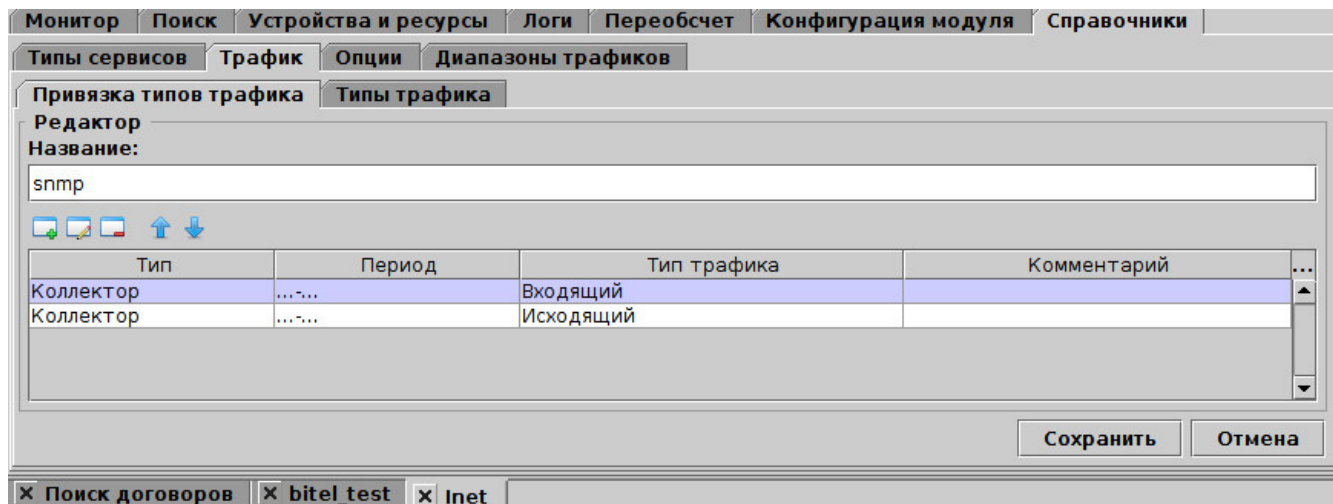
В этой главе описана настройка сбора snmp для любой схемы.

Для начала нужно настроить [InetSnmpWorker](#) в accounting-сервере.

Потом завести [типы трафика](#):



и создать [привязку snmp](#):



Монитор Поиск Устройства и ресурсы Логи Переобсчет Конфигурация модуля Справочники

Типы сервисов Трафик Опции Диапазоны трафиков

Привязка типов трафика Типы трафика

Редактор
Название: snmp

Редактор правила

Тип: Коллектор с по

Комментарий:

RADIUS
Реали: ServiceName:

Код вендора: Код атрибута: Префикс:

Коллектор
Направление: Диапазон адресов:
входящее 0.0.0.0 - 0.0.0.0
DiffServ/ToS: Диапазон портов:
0 - 0
Источник: Интерфейс:
любой любой интерфейс

Тип трафика
Входящий

Ok Отмена
Сохранить Отмена

X Поиск договоров X bitel_test X Inet

Монитор Поиск Устройства и ресурсы Логи Переобсчет Конфигурация модуля Справочники

Типы сервисов Трафик Опции Диапазоны трафиков

Привязка типов трафика Типы трафика

Редактор
Название: snmp

Редактор правила

Тип: Коллектор с по

Комментарий:

RADIUS
Реали: ServiceName:

Код вендора: Код атрибута: Префикс:

Коллектор
Направление: Диапазон адресов:
исходящее 0.0.0.0 - 0.0.0.0
DiffServ/ToS: Диапазон портов:
0 - 0
Источник: Интерфейс:
любой любой интерфейс

Тип трафика
Исходящий

Ok Отмена
Сохранить Отмена

X Поиск договоров X bitel_test X Inet

На изображении пример простой привязки, когда абсолютно весь трафик делится на входящий и исходящий.

Далее в конфигурации [Устройств](#) или [Типов устройств](#) нужно указать флаг

```
flow.agent.type=sntp
```

В [типе устройств](#) нужно добавить все необходимые интерфейсы.

Так же в типе устройства можно указать следующие параметры:

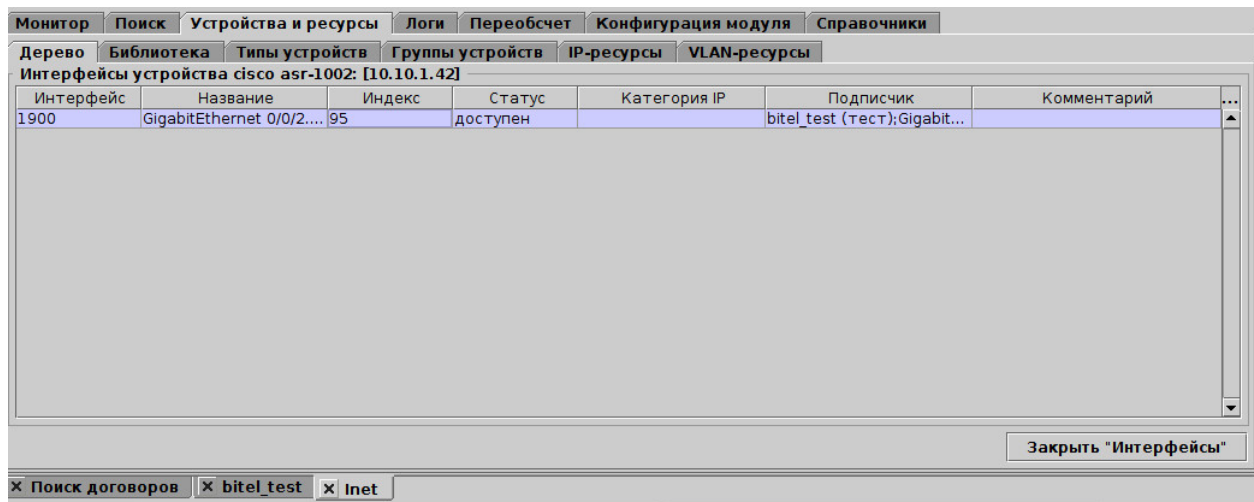

```

# snmp . 1.
snmp.version=1
# ( , /, - 161)
#snmp.port=161

# oid
flow.agent.snmp.oid.ifInOctets=
# oid
flow.agent.snmp.oid.ifOutOctets=

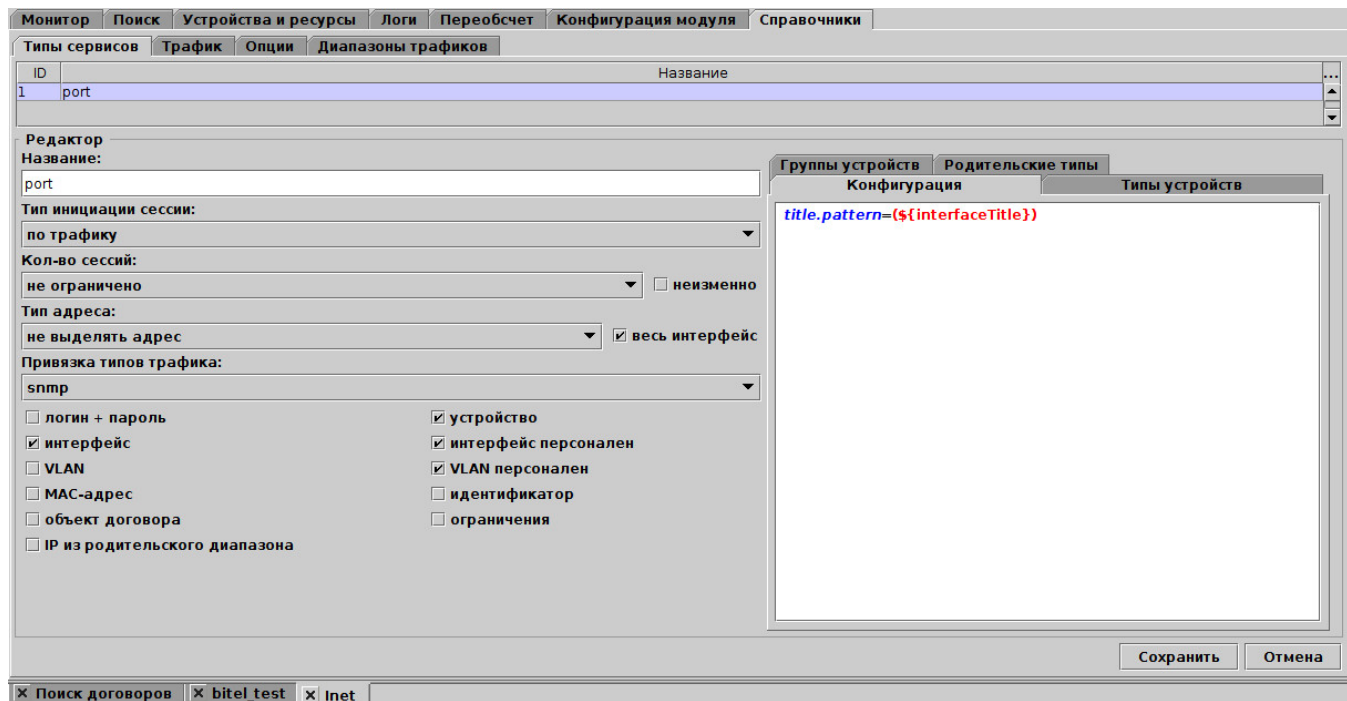
```

Статистика собирается по номерам интерфейсов или их индексам. Индекс можно указать для каждого интерфейса на устройстве:



Если указан индекс, то он используется вместо номера интерфейса.

Тип сервиса нужно добавить вот таким образом :



Тут обратите внимание, что мы указали галочку "весь интерфейс".

Сервис на договоре добавляется обычным образом:

Сервисы **Учетные периоды**

Настройки **Опции** Дополнительно Ошибки

Тип: port

Период: с 09.10.2017 по

Статус: Открыт

Кол-во сессий: не ограничено

Устройство: cisco asr-1002: [10.10.1.42] (2) <<<

Интерфейс: GigabitEthernet 0/0/2.1900 <<<

Комментарий

Состояние: Включен, активные опции: 30 mbit (3)

Сохранить Отмена

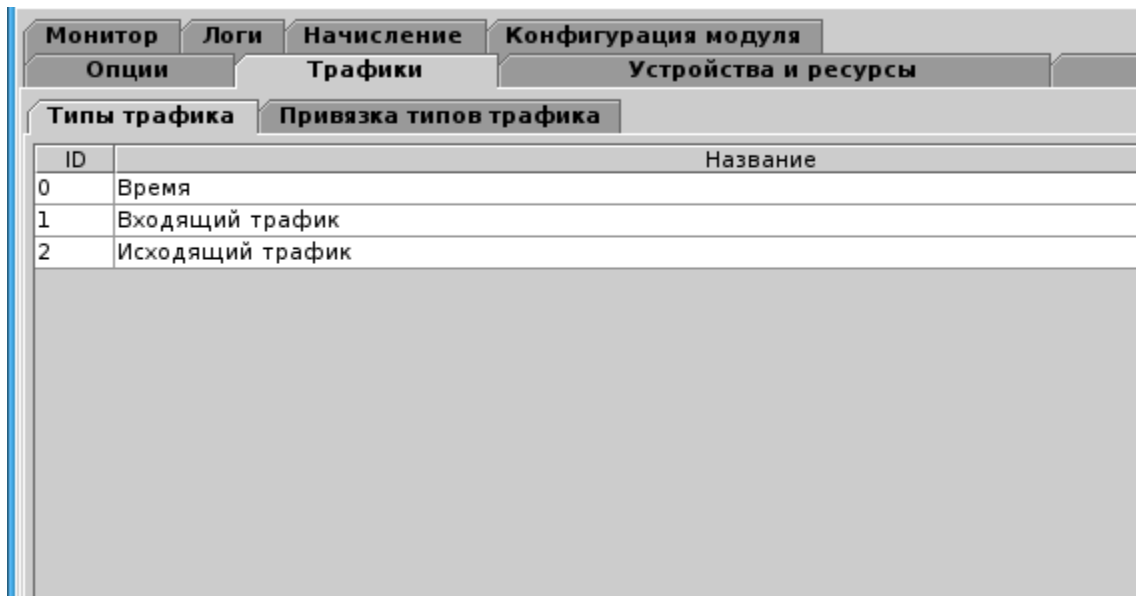
Тут мы выбираем устройство и интерфейс.

Сущности модуля Inet

Трафики

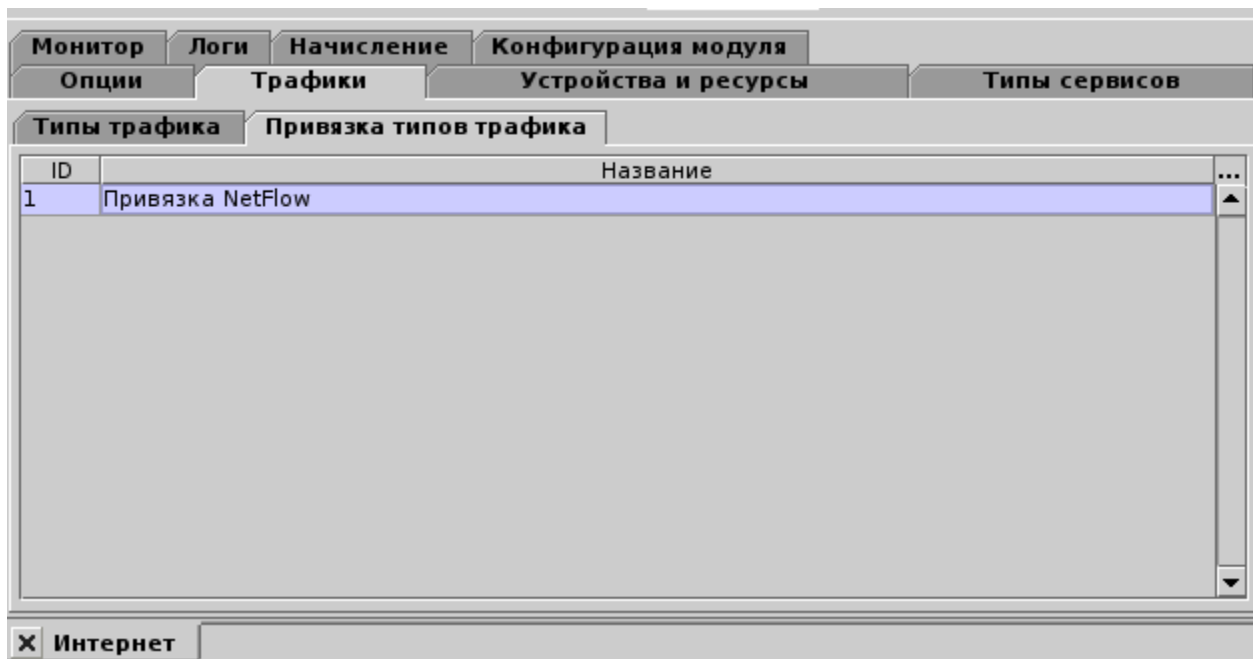
Настройка типов трафика и привязок выполняется на вкладке модуля **Трафики**.

Тип трафика - это категория, необходимая для разделения потребляемого трафика по типам. Например: "Входящий внешний", "Входящий внутренний", "Входящий локальный", "Радио", "Голосовой трафик". Время также является специальным типом трафика с кодом 0, который всегда потребляется в ходе работы. Перечень типов трафика редактируется на подвкладке **Типы трафика**.



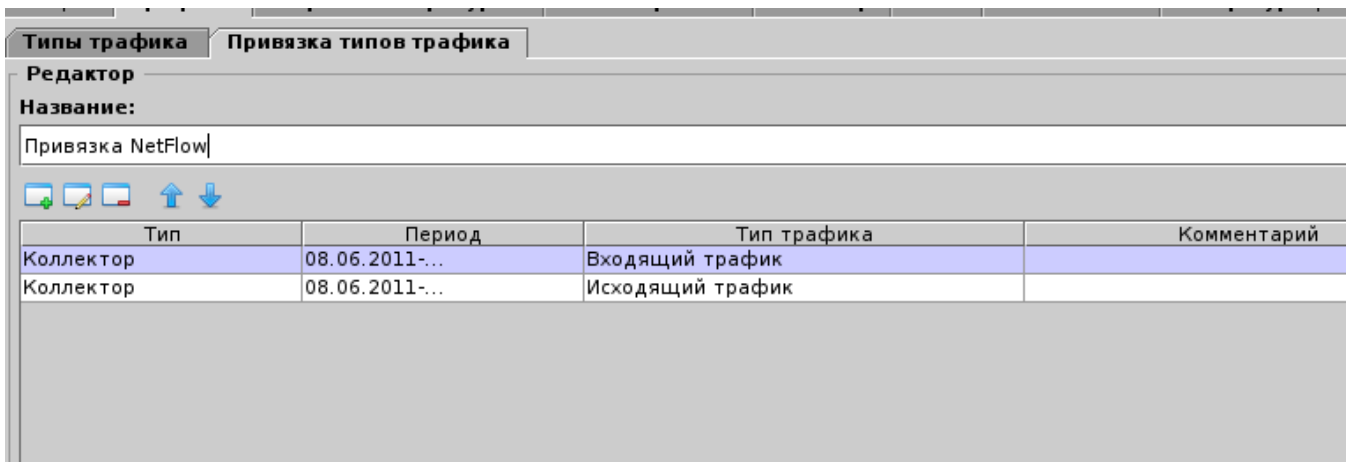
ID	Название
0	Время
1	Входящий трафик
2	Исходящий трафик

Привязка типов трафика определяет правила разделения трафика по типам. Привязок может быть множество. Привязка типов трафика используется в [типе сервиса](#).



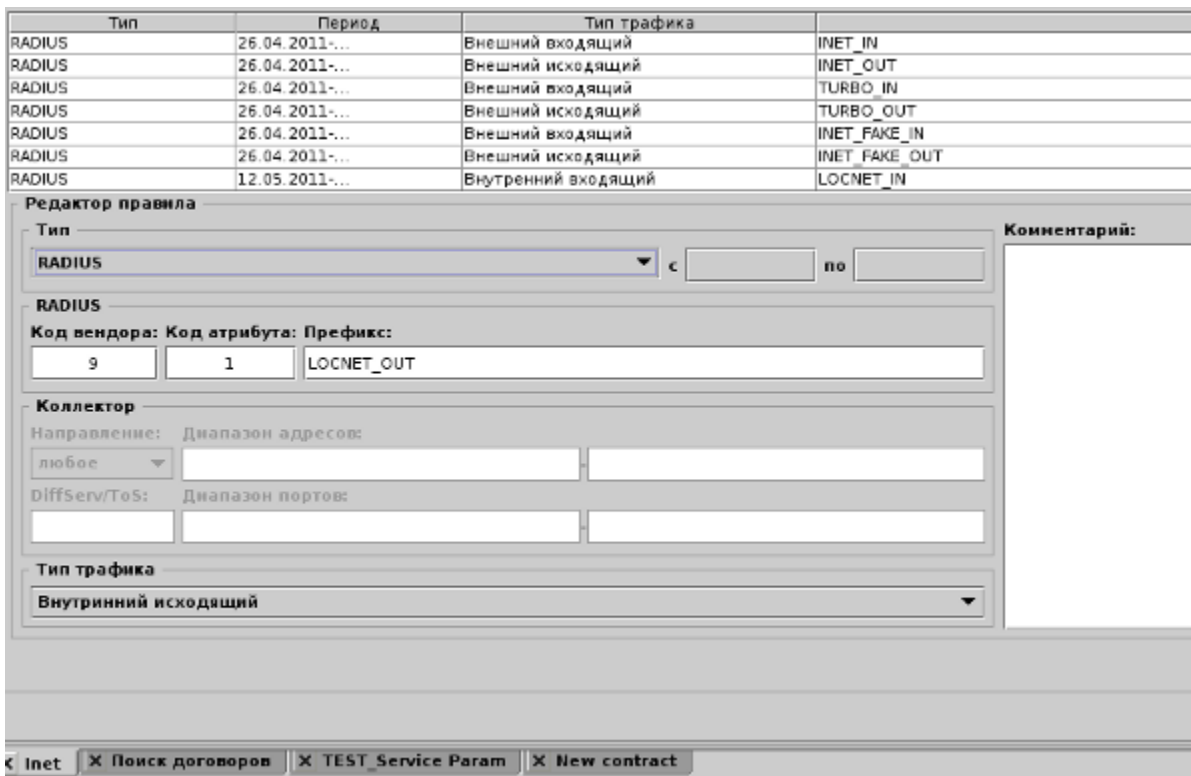
ID	Название
1	Привязка NetFlow

В свойствах привязки определяется название и набор правил, просмотр которых осуществляется последовательно. С помощью панели инструментов можно создавать привязки, изменять их порядок.



В каждом правиле указывается:

- **Тип** - RADIUS, либо Коллектор (для netflow/sflow), в зависимости от протокола данных;
- **Период** - период действия правила;
- **Тип трафика**, к которому относится правило.



Для типа RADIUS необходимо указать **ServiceName**, **Код вендора**, **Код атрибута** и **Префикс трафика**. Код вендора и Код атрибута определяют атрибут, из которого извлекается трафик. Атрибут может быть типа целого типа либо строкового. Для получения трафиков из стандартных атрибутов Acct-Output-Octets+Acct-Output-Gigawords (**входящий** трафик для клиента) необходимо указать в коде вендора и атрибута значения -2 и 1 соответственно. Acct-Input-Octets+Acct-Input-Gigawords (**исходящий** трафик для клиента) - значения -2 и 2.

Префикс может быть использован для извлечения трафика из строкового атрибута, в котором он добавлен с каким-либо префиксом. В этом случае трафик разных типов может идти в одном атрибуте, но с разными префиксами.

ServiceName используется для идентификации/фильтрации RADIUS-аккаунтинга сервиса при посервисном аккаунтинге (Cisco ISG/SmartEdge). В поле можно указать как название сервиса/профиля ISG/SmartEdge, так и регулярное выражение в формате /<regex>/, например /ISG-\d+MBPS/.

Для типа **NetFlow** необходимо указать направление трафика относительно адреса клиента, диапазон адресов (если нужен весь диапазон - поля можно оставить пустыми), а также, если необходимо, DiffServ/TOS биты.

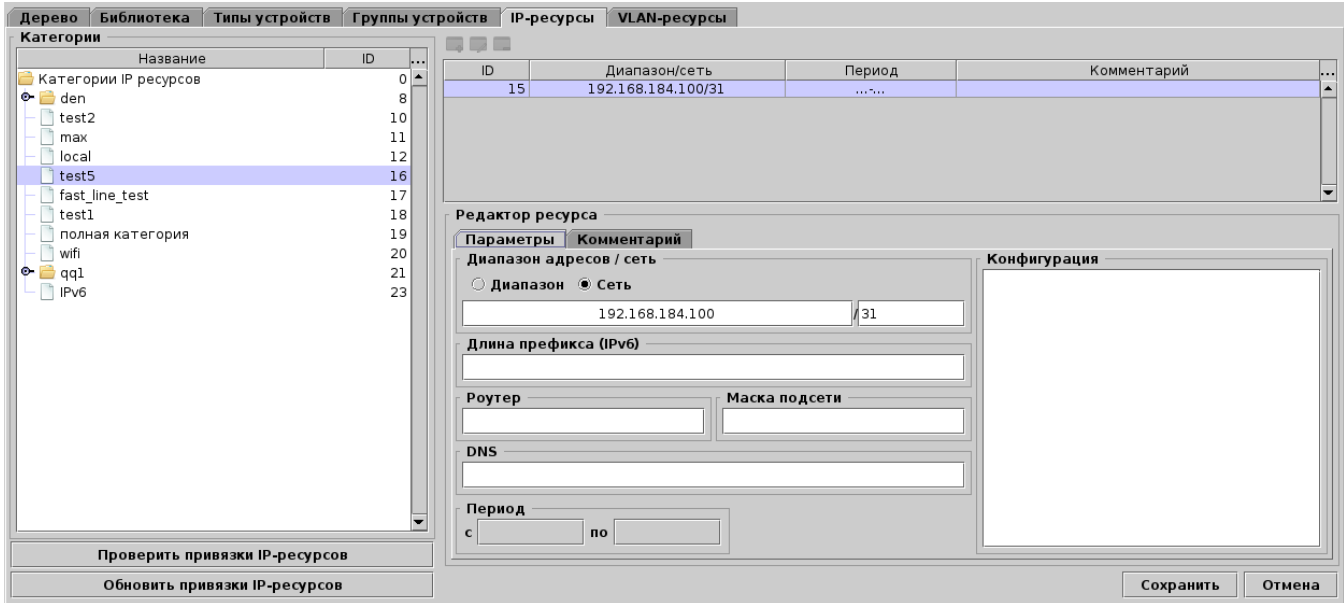
Ресурсы

Ресурсами модуля являются уникальные идентификаторы, количество которых ограничено. В данный момент это IP-адреса, VLAN-ы, порты устройства. Всякий IP-адрес, VLAN или порт, выданный модулем временно, либо постоянно, должен присутствовать в базе ресурсов и помечен там как занятый или свободный.

IP-ресурсы, Пулы IP-адресов

- [Дополнительные параметры IP-ресурса](#)
- [Пулы IP-адресов](#)

Ресурсы IP-адресов группируются в категории. Категории образуют дерево, у каждой категории есть код, отображаемый в столбце ID.



Для редактирования дерева категорий используется общая панель инструментов, для редактирования ресурсов - панель над таблицей с ресурсами. Каждый диапазон адресов уникален внутри категории.

Категории IP-ресурсов можно использовать как для статической привязки в сервисе на договоре, так и для динамической выдачи через RADIUS или DHCP, в том числе одновременно. Например, указать на устройстве категорию для реалма radius и для статической выдачи:

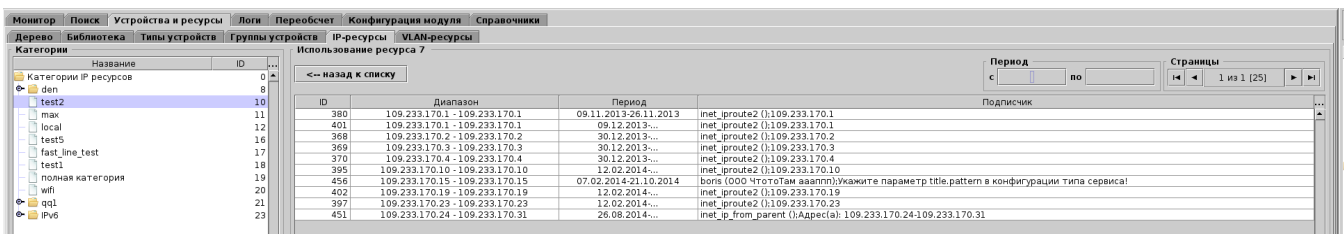
```
radius.realm.default.ipCategories=1
ip.resource.categoryId=1
```

Однако необходимо учитывать, что при статическом назначении в сервисе договора IP-адреса, занятые в данный момент динамически, будут показаны как свободные.

Для того чтобы указать категории статической выдачи отдельно для конкретных типов сервиса можно перечислить их ID через запятую:

```
ip.resource.categoryId.1,3,5=2
```

По правому клику мышки в таблице на конкретном ресурсе во всплывающем меню доступен пункт "Использование". При выборе этого пункта открывается таблица тех, кто использует данный ресурс статически:



Для оперативного перехода к договору, который использует данный ресурс, достаточно щелкнуть правой кнопкой мыши и в появившемся контекстном меню выбрать пункт меню **Договор**.

Дополнительные параметры IP-ресурса

В редакторе IP-ресурса доступны поля Роутер, Маска подсети, DNS, а также поле Конфигурация. Данные поля можно использовать для выдачи соответствующих DHCP-опций с привязкой к IP-ресурсу - когда абоненту по DHCP-будет выдаваться адрес из данного ресурса в OFFER/ACK-пакеты будут добавлены данные опции. В поле конфигурация можно прописать дополнительные DHCP-опции с помощью параметров dhcp.options.<имя опции>=<значение> аналогично [параметрам DHCP-опций в конфигурации устройства](#).

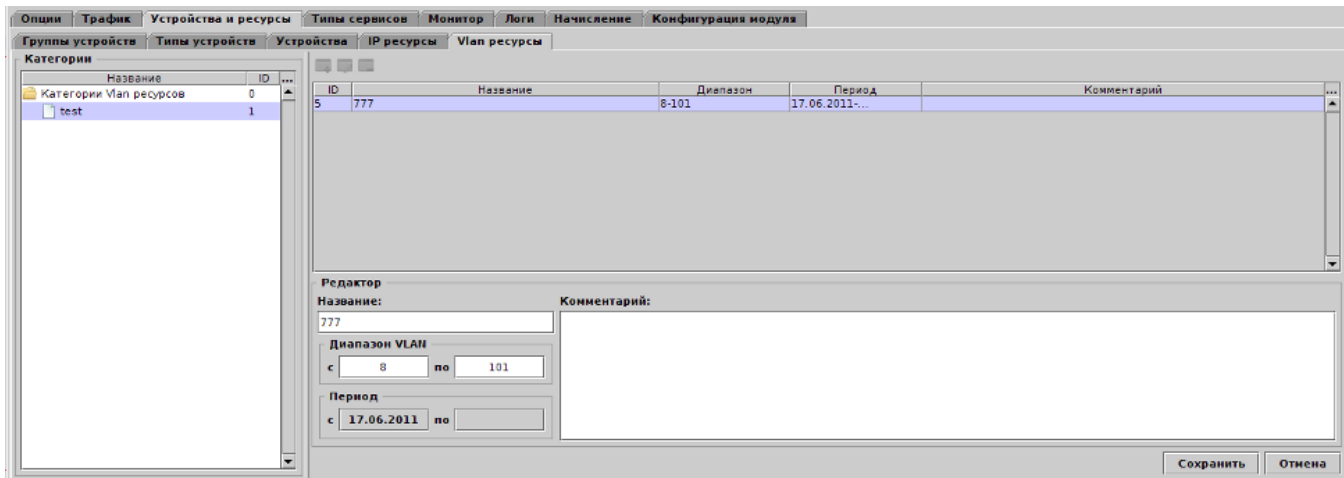
Пулы IP-адресов

Для выдачи динамических IP-адресов можно использовать как просто перечисление категорий IP-ресурсов, так и именованные пулы. Пулы используются для выдачи адресов [в зависимости от активных опций Inet](#). Так же для пула можно указать дополнительную конфигурацию. Для создания пула необходимо прописать его в конфигурации устройства или корневого устройства, как resource.ip.pool.<имяпула>.ipCategories:

```
# IP-,
resource.ip.pool.myPool.ipCategories=1,2
# IP-,      (alarm)
#resource.ip.pool.myPool.alarm.freePercent=10
# - IP-,
#resource.ip.pool.myPool.alarm.freeCount=300
#   IP-
#resource.ip.pool.myPool.alarm.categoryRoundRobin=0
#     IP-
#resource.ip.pool.myPool.alarm.categoryRoundRobinLeastConn=0
```

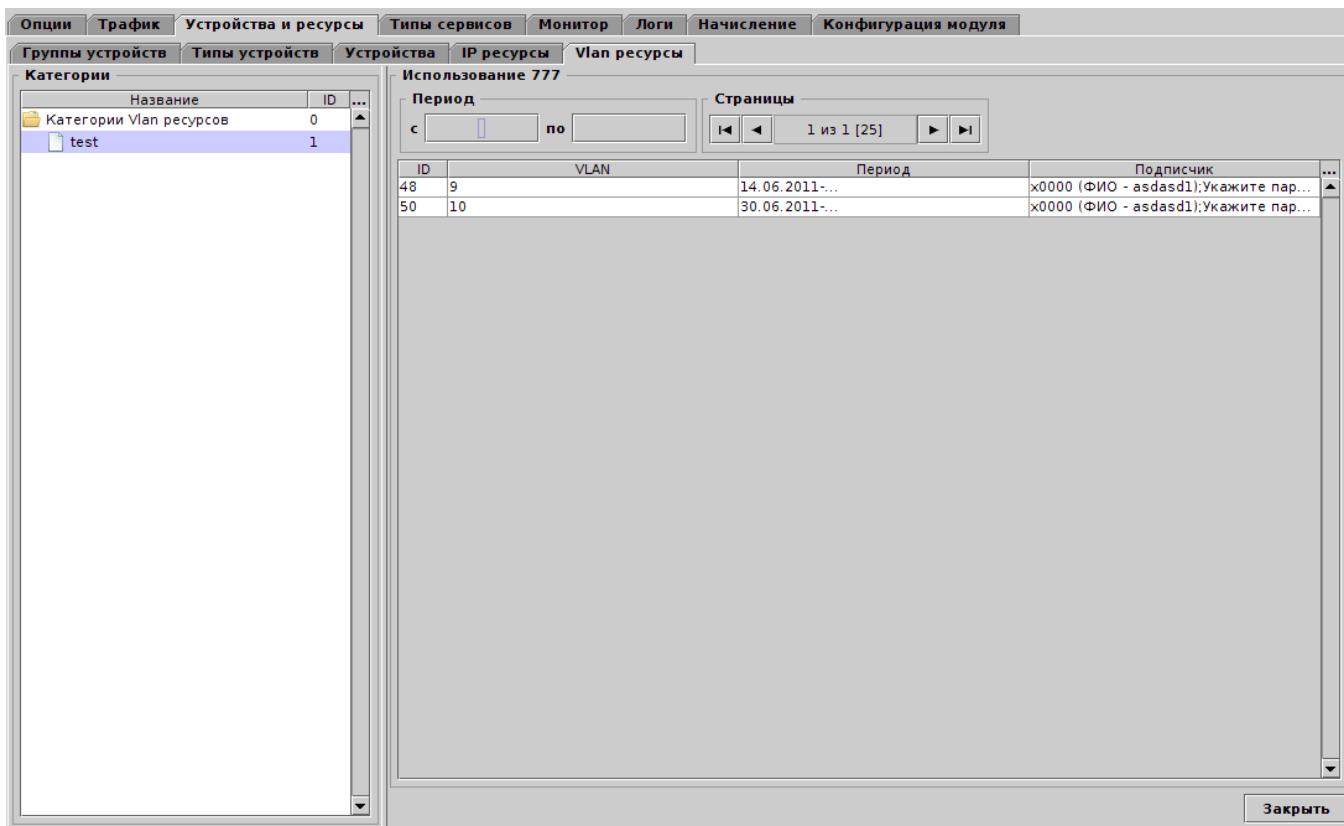

VLAN-ресурсы

Ресурсы VLAN делятся на категории. Категории образуют дерево, у каждой категории есть код, отображаемый в столбце ID.



Для редактирования дерева категорий используется общая панель инструментов, для редактирования ресурсов - панель над таблицей с ресурсами. Каждый диапазон VLAN уникален внутри категории.

По правому клику мышки в таблице на конкретном ресурсе во всплывающем меню доступен пункт "Использование". При выборе этого пункта открывается таблица тех, кто использует данный ресурс:



Чтобы в сервисе договора можно было выбрать VLAN (т.е. назначит абоненту), в конфигурации привязанного к этому сервису устройства или, что бывает чаще, в конфигурации устройства-предка (например, корневого устройства Access+Accounting) необходимо перечислить ID **категории** VLAN-ресурсов:

```
vlan.resource.category=1
```

Данная привязка означает, что все сервисы привязанные к данному устройству или к устройству, которое в дереве является потомком данного, могут получить VLAN из указанных категорий VLAN-ресурсов. До указания данной привязки, при попытке назначить абоненту VLAN, будет происходить ошибка "Не найден подходящий VLAN-ресурс!".

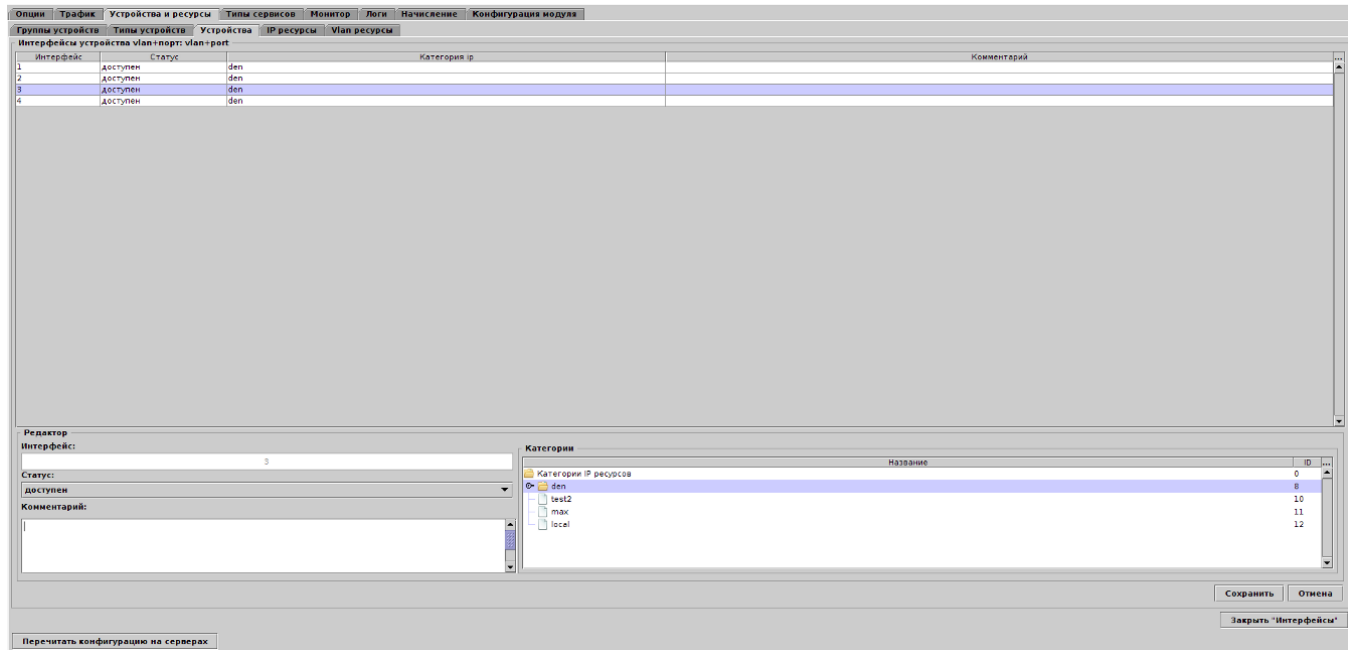
Также в схемах Q-in-Q Вы можете указать SVLAN в устройстве [дерева устройств](#). Для этого в конфигурации устройства-предка (например, корневого устройства Access+Accounting) укажите **категорию** ресурсов SVLAN:

```
svlan.resource.category=2
```

SVLAN может использоваться при обработке RADIUS/DHCP-запросов для поиска агентского устройства, от которого далее будет идти поиск абонента по VLAN.

Ресурсы интерфейсов

Интерфейсы уникальны в пределах одного устройства. Сам список интерфейсов задаётся в [типе устройства](#). Далее в [дереве устройств](#) можно по правому клику мышкой на устройстве выбрать "Интерфейсы", и откроется панель занятости интерфейсов:



Двойным кликом мышки можно открыть на редактирование конкретный интерфейс и изменить его статус. Статусы имеют следующие значение:

Доступен - интерфейс доступен для добавления на договор;

Зарезервирован - интерфейс не доступен для добавления на договор.

Также для каждого интерфейса может быть указана категория адресов, которые могут быть выбраны на этом интерфейсе. Внутри одной категории IP-адреса не пересекаются.

По правому клику мышки в таблице на конкретном интерфейсе во всплывающем меню доступен пункт "Использование". При выборе этого пункта открывается таблица тех, кто использует данный интерфейс:

Опции | Трафик | Устройства и ресурсы | Типы сервисов | Монитор | Логи | Начисление | Конфигурация модуля

Группы устройств | Типы устройств | Устройства | IP ресурсы | Vlan ресурсы

Интерфейсы устройства vlan+порт: vlan+port

Использование интерфейса 3

Период с по Страницы 1 из 1 [25]

ID	Интерфейс	Период	Подписчик	...
44	3	30.06.2011...	x0000 (ФИО - asdasd1);Укажите параметр title.patter...	▲

Закреть "Использование"

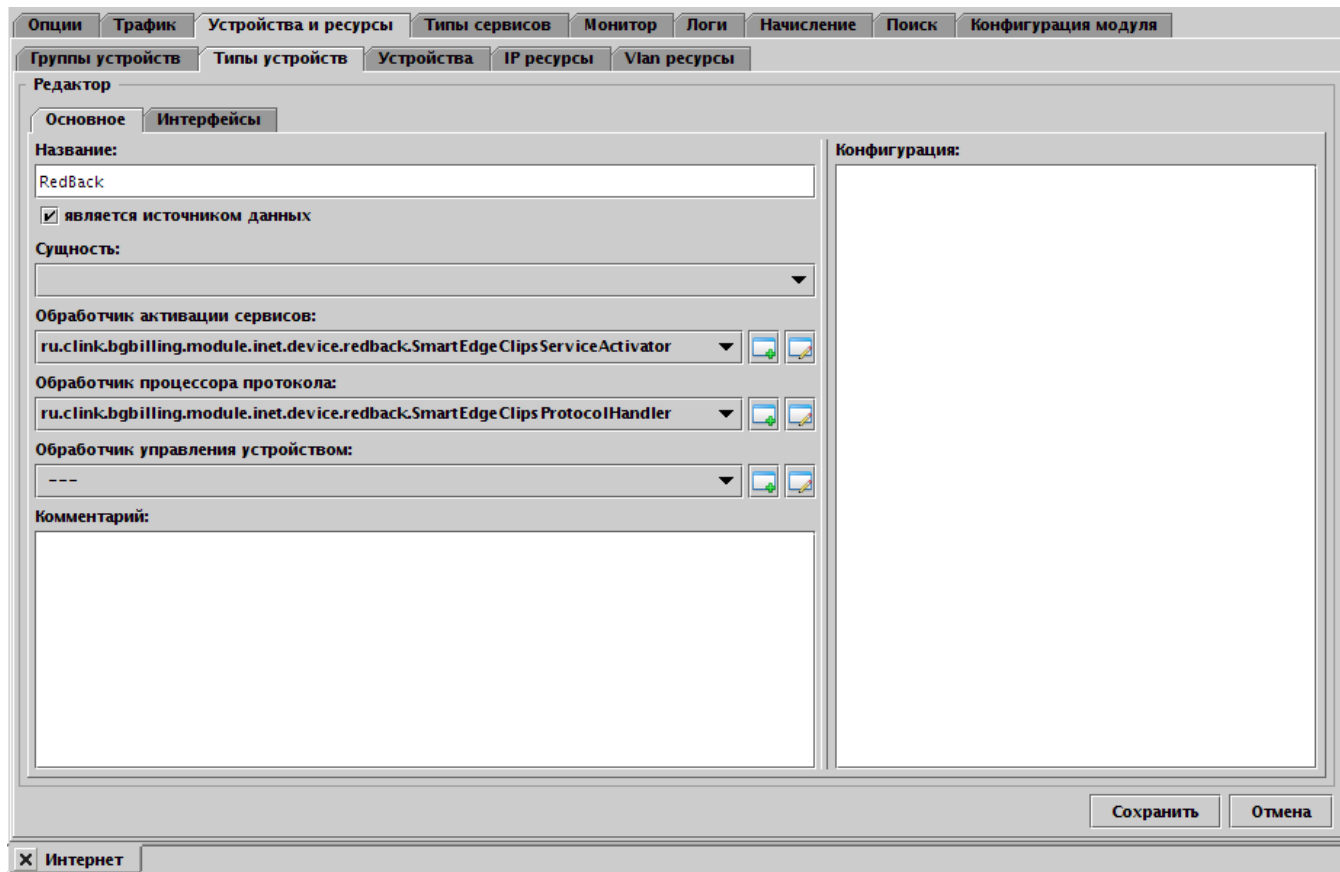
Закреть "Интерфейсы"

Перечитать конфигурацию на серверах

Типы устройств

Дерево устройств представляет из себя отображение структуры сети с точки зрения модуля. При этом устройства не обязательно соотносятся с реальными устройствами сети - это логические единицы. В том числе это могут быть группирующие узлы или, например, физические интерфейсы маршрутизатора. Управление устройствами, их типами и ресурсами адресов производится на вкладке **Устройства и ресурсы** модуля.

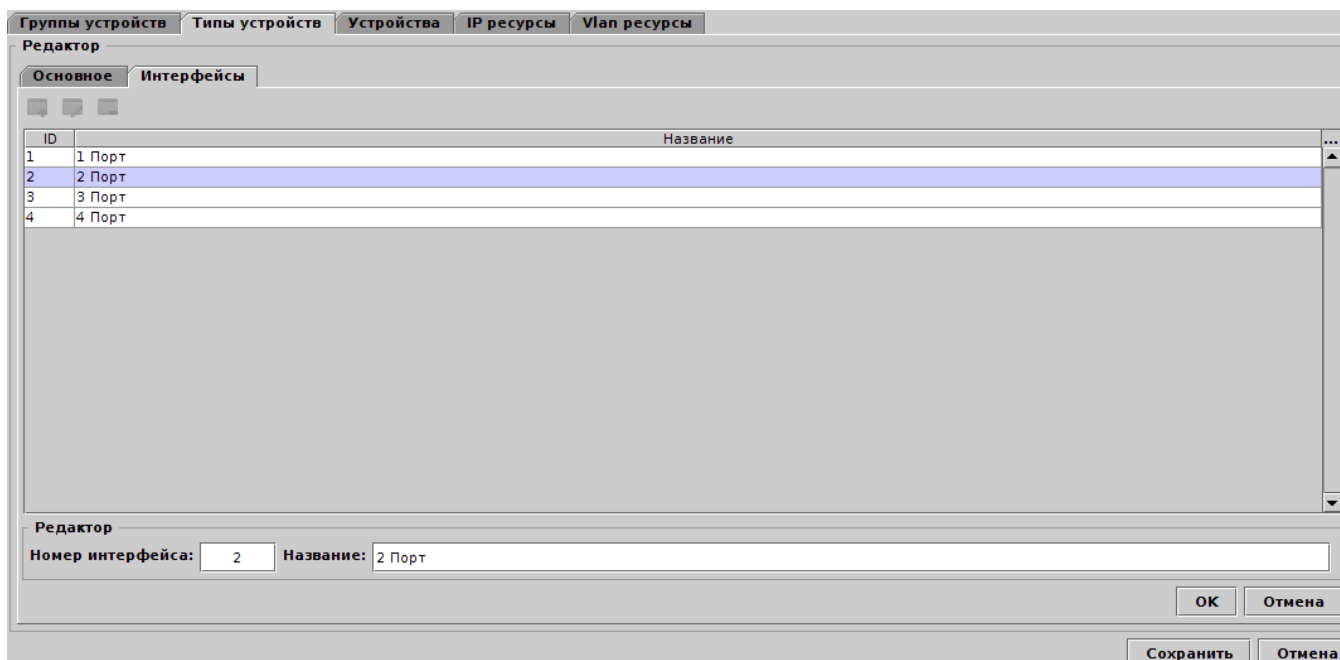
Каждое устройство обладает своим типом. Типы устройств определяются на вкладке **Типы**.



Рассмотрим параметры, которые могут быть указаны в типе устройства:

- **Название** - обозначение типа, используется в т.ч. для формирования имени устройства;
- Флаг **является источником данных** - обозначает, что данный тип устройств генерирует данные о трафике, устройства с данным типом отображаются при переработке логов;
- **Обработчик активации сервисов** - **динамический класс**, реализующий определённый интерфейс по управлению устройством для блокировки/разблокировки сервиса, изменения его параметров;
- **Обработчик процессора протокола** - **динамический класс**, реализующий определённый интерфейс по пред- и постобработке пакетов протокола, связанных с конкретным устройством (например, RAADIUS или DHCP-пакетов).
- **Обработчик управления устройством** - **динамический класс**, реализующий определённый интерфейс по дополнительному управлению устройством (например, получение uptime устройства на текущий момент, для отслеживания перезагрузки устройства).

На вкладке **Интерфейсы** типа устройства указываются его интерфейсы, к которым могут быть привязаны сервисы. Например, это могут быть порты коммутаторов.



После корректировки параметров типов устройств для их перечитывания на серверах Access и Accounting необходимо нажать кнопку **Перечитать конфигурацию на серверах**, расположенную под деревом устройств. Данная кнопка позволяет оповещать сервера об изменениях только, когда конфигурация станет законченной. До тех пор сервера используют сохранённую в памяти конфигурацию.

Обработчик активации сервисов

Обработчик активации сервисов синхронизирует состояние сервиса и сессии на устройстве - именно он производит открытие/закрытие доступа, изменение скорости или других параметров.

Обработчик привязывается к типу устройства и вызывается поочередно от корня дерева устройств до затронутого устройства (при необходимости синхронизировать сервис, привязанный к устройству данного типа или же сессию, начатую на устройстве данного типа).

В поставку модуля Inet входят стандартные реализации для Cisco, Redback, MPD и т.п., производящие интеграцию с соответствующими устройствами. Исходные коды находятся в Управлении динамическим кодом, в пакете ru.bitel.bgbilling.modules.inet.dyn. Данные классы перерегируются при обновлении, поэтому для изменения логики класса необходимо расширить его или создать копию, но не изменять его напрямую.

Код

```
/**
 * .<br/>
 * <pre> :
 * init
 *
 * connect
 *   serviceModify
 *   serviceModify
 *   serviceCancel
 * disconnect
 *
 * connect
 *   serviceCreate
 *   serviceModify
 *   connectionModify
 *   serviceModify
 *   connectionClose
 * disconnect
 *
 * destroy</pre>
 */
public interface ServiceActivator
{
    /**
     * .
     * @param setup
     * @param moduleId
     * @param device
     * @param deviceType
     * @param config
     * @return
     * @throws Exception
     */
    public Object init( Setup setup, int moduleId, InetDevice device, InetDeviceType deviceType,
ParameterMap config )
        throws Exception;

    /**
     * .
     * @return
     * @throws Exception
     */
    public Object destroy()
        throws Exception;

    /**
     * .
     * @return
     * @throws Exception
     */
    public Object connect()
        throws Exception;
}
```

```

/**
 * .
 * @return
 * @throws Exception
 */
public Object disconnect()
    throws Exception;

/**
 * ( )
 * @param e
 * @return
 * @throws Exception
 */
public Object serviceCreate( ServiceActivatorEvent e )
    throws Exception;

/**
 * (// ).
 *
 * @see ServiceActivatorEvent
 * @see {@link ServiceActivatorEvent#getNewState()}
 * @param e
 * @return
 * @throws Exception
 */
public Object serviceModify( ServiceActivatorEvent e )
    throws Exception;

/**
 * ( ).
 * @param e
 * @return
 * @throws Exception
 */
public Object serviceCancel( ServiceActivatorEvent e )
    throws Exception;

/**
 * .
 * .<br/>
 * , {@link ServiceActivatorEvent#getNewState()} == {@link InetServ#STATE_DISABLE}      {@link
#connectionClose(ServiceActivatorEvent)}
 * @param e
 * @return
 * @throws Exception
 */
public Object connectionModify( ServiceActivatorEvent e )
    throws Exception;

/**
 * ( ) .<br/>
 * {@link AccessCodes#TOO_MANY_SESSIONS_ERROR}      {@link #connectionModify(ServiceActivatorEvent)}
 * @param e
 * @return
 * @throws Exception
 */
public Object connectionClose( ServiceActivatorEvent e )
    throws Exception;

/**
 * .
 * @param event
 * @return
 * @throws Exception
 */
public Object onAccountingStart( ServiceActivatorEvent event )
    throws Exception;

/**
 * .

```



```

    * @param event
    * @return
    * @throws Exception
    */
    public Object onAccountingStop( ServiceActivatorEvent event )
        throws Exception;
}

```

Описание интерфейса ru.bitel.bgbilling.modules.inet.access.sa.ServiceActivator доступно в документации по API.

Параметры работы обработчика активации сервисов (прописываются в конфиге типа устройства или конфиге устройства):

Код

```

# -
#sa.batch.size=20
#
#sa.batch.pause.millis=0
#
sa.batch.wait.millis=5000
# , ( disconnect ServiceActivator)
sa.batch.waitNext.millis=5000

#
sa.error.pause.millis=60000
# - , , (alarm)
sa.error.alarm.count=20
# - ,
sa.error.redelivery.count=100
# , -
sa.error.redelivery.timeout=86400

sa.device.sync.allPath=

```

Параметры пересинхронизации устройства (например, по определению перезагрузки устройства по uptime, который опрашивает обработчик управления устройством):

Код

```

# -
#sa.device.sync.allPath=1
# onAccountingStart
#sa.device.sync.onAccountingStart=0
# serviceCancel serviceCreate
#sa.device.sync.cancelBeforeCreate=1
# connectionModify connectionClose
#sa.connection.modifyInsteadClose=0
# connectionClose connectionModify
#sa.connection.closeInsteadModify=0

```

По умолчанию onAccountingStart/onAccountingStop не вызываются для сессий по трафику. Но может быть ситуация, что Вам необходимо выполнять какие-то команды при появлении трафика от абонента или при его долгом отсутствии (когда сессия в биллинге закрывается по таймауту); или же Вы хотите генерировать RADIUS-Accounting биллингом для старта и стопа сессии - тогда укажите в конфигурации устройства /типа устройства:

Код

```

# onAccountingStart/onAccountingStop
sa.connection.auto.accountingEvent=1

```

Обработчик процессора протокола

Обработчик процессора протокола позволяет произвести пред/постобработку RADIUS или DHCP-запросов - например, изменить RADIUS-запрос перед его обработкой системой или установить дополнительные опции, которые поменяют логику обработки запроса.

Обработчик привязывается к типу устройства, т.е. обработчик будет работать с запросами с устройств данного типа. При этом поочередно вызываются все обработчики в цепочке дерева, начиная от корня дерева до устройства, с которого пришел запрос.

В поставку модуля Inet входят стандартные обработчики для Cisco, Redback, MPD и т.п., позволяющие расширить интеграцию и функционал простым изменением конфигурации. Исходные коды находятся в Управлении динамическим кодом, в пакете ru.bitel.bgbilling.modules.inet.dyn. Данные классы перетираются при обновлении, поэтому для изменения логики класса необходимо расширить его или создать копию, но не изменять его напрямую.

Обработчик процессора протокола состоит из двух интерфейсов, которые он расширяет:

Код

```
/**
 * RADIUS-
 */
public interface RadiusProtocolHandler
{
    /**
     * RADIUS- Access-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void preprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;

    /**
     * RADIUS- Access-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void postprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;

    /**
     * RADIUS- Accounting-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void preprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;

    /**
     * RADIUS- Accounting-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void postprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;
}
```

```

/**
 * DHCP-
 */
public interface DhcpProtocolHandler
{
    /**
     * DHCP-
     * @param request
     * @param response
     * @throws Exception
     */
    public void preprocessDhcpRequest( DhcpPacket request, DhcpPacket response )
        throws Exception;

    /**
     * DHCP-
     * @param request
     * @param response
     * @throws Exception
     */
    public void postprocessDhcpRequest( DhcpPacket request, DhcpPacket response )
        throws Exception;
}

```

```

/**
 * .
 * @see RadiusProtocolHandler
 * @author amir
 */
public interface ProtocolHandler
    extends RadiusProtocolHandler, DhcpProtocolHandler
{
    public void init( Setup setup, int moduleId, InetDevice inetDevice, InetDeviceType inetDeviceType,
        ParameterMap config )
        throws Exception;
}

```

Например, такой обработчик будет подменять значение атрибута User-Name на значение идентификатора устройства + значение атрибута Nas-Port-Id, а значение пароля на "Password":

Код

```
import ru.bitel.bgbilling.kernel.network.radius.RadiusDictionary;
import ru.bitel.bgbilling.kernel.network.radius.RadiusPacket;
import ru.bitel.bgbilling.modules.inet.access.sa.ProtocolHandler;
import ru.bitel.bgbilling.modules.inet.access.sa.ProtocolHandlerAdapter;
import ru.bitel.bgbilling.modules.inet.api.common.bean.InetDevice;
import ru.bitel.bgbilling.modules.inet.api.common.bean.InetDeviceType;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.ParameterMap;
import ru.bitel.common.sql.ConnectionSet;

public class SomeProtocolHandler
    extends ProtocolHandlerAdapter
    implements ProtocolHandler
{
    private String deviceIdentifier;

    @Override
    public void init( Setup setup, int moduleId, InetDevice inetDevice, InetDeviceType inetDeviceType,
ParameterMap deviceConfig )
        throws Exception
    {
        this.deviceIdentifier = inetDevice.getIdentifier();
    }

    @Override
    public void preprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception
    {
        String nasPortId = request.getStringAttribute( -1, RadiusDictionary.NAS_Port_Id, null );

        request.setStringAttribute( -1, RadiusDictionary.User_Name, this.deviceIdentifier + ":" +
nasPortId );
        request.setStringAttribute( -1, RadiusDictionary.User_Password, "Password" );
    }
}
```

Обработчик управления устройством

Для дополнительного управления у типа устройства можно указать обработчик управления устройством.

Uptime и обнаружение перезагрузки устройства

На текущий момент главной функцией обработчика является получение текущего uptime устройства. Uptime необходим для определения перезагрузки устройства, чтобы при наступлении такого события Access мог синхронизировать заново все сервисы на этом устройстве.

В конфигурации типа устройства (или в самом устройстве) можно задать параметры выполнения команд:

```
#
#manage.error.pause=5
#  uptime
#manage.uptime.pause=120
#  ,  uptime
#manage.uptime.error.pause=120
```

С модулем Inet поставляется динамический класс `ru.bitel.bgbilling.modules.inet.dyn.device.snmp.SnmpDeviceManager`, у него реализован метод `uptime`, который по SNMP у устройства получает его текущий uptime в секундах.

```
#  SNMP- ( , /)
#snmp.host=
#  SNMP- ( 161)
#snmp.port=
#  SNMP ( 1)
#snmp.version=
#  community ( community/Secret)
#snmp.community=public
#  SNMP OID,  uptime (1.3.6.1.2.1.1.3.0)
#snmp.uptimeOid=1.3.6.1.2.1.1.3.0
```

После привязки обработчика управления устройством с реализованным методом `uptime`, все устройства с периодичностью `manage.uptime.pause` будут опрашиваться на время работы после перезагрузки. Для того, чтобы при определении очередной перезагрузки устройства Access выполнил синхронизацию сервисов, в конфигурации типа устройства (или в устройстве) необходимо прописать:

```
#  , 0 - ( ), 1 -
# ( )
#sa.device.sync.onReboot=1
#  , 0 - serviceCreate 1 ( ) -
#  serviceCancel,  serviceCreate
#sa.device.sync.cancelBeforeCreate=1
```

Это может быть полезно для коммутаторов, которые сбрасывают настройки разрешенных портов после перезагрузки, а также в некоторых других случаях.

Также еще есть специальный метод, который вызывается при перезагрузке:

```
/**
 *
 * @param e
 * @return
 * @throws Exception
 */
public Object onReboot( InetDeviceManageEvent e )
{
    return null;
}
```

Этот метод не обязателен. Его вы добавляете только если вам нужно что-то сделать при перезагрузке устройства.

Выполнение команд

В контекстном меню дерева устройств доступен пункт выполнить команду. Данный пункт вызывает метод у Обработчика управления устройством (DeviceManager), указанном в типе устройства. Таким методом, в том числе, является uptime() из ru.bitel.bgbilling.modules.inet.dyn.device.snmp.SnmpDeviceManager, хотя нужен он не для ручного вызова из контекстного меню.

```
@Override
public Object uptime()
    throws Exception
{
    return snmpClient.get( uptimeOid, -1, Long.class );
}
```

Команды-методы можно именовать с помощью аннотации ru.bitel.oss.systems.inventory.resource.server.DeviceManagerMethod, в этом случае они будут сразу доступны в контекстном меню:

```
import ru.bitel.oss.systems.inventory.resource.server.DeviceManagerMethod;

@DeviceManagerMethod(title = "")
public Object reboot()
    throws Exception
{
    return snmpClient.set( new AsnObjectId( "1.2.3.4.5.6.7.8.9" ).getOid(), -1, Long.class );
}
```

После или вместо выполнения можно вернуть строку определенного вида, чтобы клиент биллинга попытался открыть браузер с указанным URL:

```
@DeviceManagerMethod(title = "")
public Object status()
    throws Exception
{
    return "browse:http://google.ru";
}
```

Или окно с подключением по telnet:

```
@DeviceManagerMethod(title = "Telnet")
public Object telnet()
    throws Exception
{
    return "telnet:google.ru 80";
}
```

Как аргумент в методе можно указать ru.bitel.bgbilling.modules.inet.access.manage.event.InetDeviceManageEvent, чтобы узнать, например, userId пользователя, выполняющего команду:

```
import ru.bitel.bgbilling.modules.inet.access.manage.event.InetDeviceManageEvent;

@DeviceManagerMethod(title = "Telnet")
public Object telnet2( InetDeviceManageEvent e )
    throws Exception
{
    logger.info( e.getUserId() );
    return "telnet:google.ru 80";
}
```

Пример:

```

package ru.provider.bgbilling.modules.inet.dyn.device;

import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.ParameterMap;
import ru.bitel.oss.systems.inventory.resource.common.DeviceManagerMethodType;
import ru.bitel.oss.systems.inventory.resource.common.bean.Device;
import ru.bitel.oss.systems.inventory.resource.common.bean.DeviceType;
import ru.bitel.oss.systems.inventory.resource.server.DeviceManagerMethod;

public class SnmpDeviceManager
    extends ru.bitel.bgbilling.modules.inet.dyn.device.snmp.SnmpDeviceManager
{
    protected ParameterMap deviceConfig;

    @Override
    public Object init( Setup setup, int moduleId, Device<?, ?> device, DeviceType deviceType, ParameterMap
deviceConfig )
    {
        super.init( setup, moduleId, device, deviceType, deviceConfig );

        this.deviceConfig = deviceConfig;

        return null;
    }

    @DeviceManagerMethod(title = "", types = { DeviceManagerMethodType.DEVICE, DeviceManagerMethodType.
ACCOUNT })
    public Object monitor()
    {
        return "browse:" + deviceConfig.get( "monitor.url", "http://zabbix.intranet.provider.ru/latest.
php?hostid=$monitorHostId" )
                .replaceAll( "\\$monitorHostId", deviceConfig.get( "monitor.hostId", ""
) );
    }

    @DeviceManagerMethod(title = "Telnet")
    public Object telnet()
    {
        return "telnet:" + deviceConfig.get( "telnet.host", this.host ) + " " + deviceConfig.getInt(
"telnet.port", 23 );
    }
}

```

Без SNMP/uptime:

```

package ru.provider.bgbilling.modules.inet.dyn.device;

import java.util.List;

import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.ParameterMap;
import ru.bitel.oss.systems.inventory.resource.common.DeviceManagerMethodType;
import ru.bitel.oss.systems.inventory.resource.common.bean.Device;
import ru.bitel.oss.systems.inventory.resource.common.bean.DeviceType;
import ru.bitel.oss.systems.inventory.resource.server.DeviceManagerAdapter;
import ru.bitel.oss.systems.inventory.resource.server.DeviceManagerMethod;

public class DeviceManager
    extends DeviceManagerAdapter
    implements ru.bitel.oss.systems.inventory.resource.server.DeviceManager
{
    protected ParameterMap deviceConfig;

    protected String host;

    @Override
    public Object init( Setup setup, int moduleId, Device<?, ?> device, DeviceType deviceType, ParameterMap
deviceConfig )
        throws Exception
    {
        super.init( setup, moduleId, device, deviceType, deviceConfig );

        this.deviceConfig = deviceConfig;

        final List<String[]> hosts = device.getHostsAsString();
        final String[] host = (hosts != null && hosts.size() > 0) ? hosts.get( 0 ) : null;

        this.host = deviceConfig.get( "snmp.host", host != null ? host[0] : device.getHost() );

        return null;
    }

    @DeviceManagerMethod(title = "", types = { DeviceManagerMethodType.DEVICE, DeviceManagerMethodType.
ACCOUNT })
    public Object monitor()
    {
        return "browse:" + deviceConfig.get( "monitor.url", "http://zabbix.intranet.provider.ru/latest.
php?hostid=$monitorHostId" )
                .replaceAll( "\\$monitorHostId", deviceConfig.get( "monitor.hostId", ""
) );
    }

    @DeviceManagerMethod(title = "Telnet")
    public Object telnet()
    {
        return "telnet:" + deviceConfig.get( "telnet.host", this.host ) + " " + deviceConfig.getInt(
"telnet.port", 23 );
    }
}

```

DeviceManagerMethodType.ACCOUNT в **types** означает, что пункт будет доступен на вкладке сервисов договора, в контекстном меню сервиса, если сервис привязан к устройству данного типа.

Устройства

Устройства - физические или логические устройства, так или иначе участвующие в предоставлении доступа, управлении доступом, тарификации; устройства, которыми биллинг будет управлять и/или которые будут оповещать о наработке (RADIUS/Netflow/sFlow/SNMP), устройства-DHCP-Relay-агенты и т.д.

База устройств модуля представлена в виде дерева, которое отражает логическую модель для управления и связи различных устройств между собой биллингом. Для редактирования используется стандартная панель инструментов, по правому клику мыши на устройство доступны функции вырезки и вставки, для изменения предка узла устройства.

Название	Комментарий	ID
Устройства		0
ProcessGroup		3
RedBack: Redback [172.16.19.1]		4

Редактор устройства.

Идентификатор: Redback
Хост/порт: 172.16.19.1
Логин:
Пароль:
Community/secret:
Тип устройства: RedBack
Группы устройств: Группы устройств

Конфигурация:

```
const.access.attributes={@const.access.attributes}  
redirect.attributes={@redirect.attributes}  
option.1.attributes={@option.1.attributes}  
option.4.attributes={@option.4.attributes}  
@  
nas.radius.realm.default.pool=1  
@  
nas.radius.inetOption.1.attributes={@const.access.attributes};{@option.1.attributes}  
nas.radius.inetOption.4.attributes={@const.access.attributes};{@option.4.attributes}  
@  
realm.reject.error=1,2,3,4,10,11,12  
@  
nas.radius.realm.reject.pool=3  
nas.radius.realm.reject.attributes=Acct-Interim-Interval=600;IP-Interface-Name=NOAUTH;DHCP-  
Max-Leases=1;{@redirect.attributes}  
@  
#@Свой порт  
nas.radius.port=3799  
nas.secret=CQbjcod6  
@  
dhcp.option.serverIdentifier=0.0.0.0  
dhcp.option.leaseTime=600  
@  
@dhcp.net.option.172.16.23.0:255.255.255.0.gate=172.16.23.1  
dhcp.net.option.172.16.23.0:255.255.255.0.dns=172.16.23.1  
@  
flow.agent.link=1;-1
```

Сохранить Отмена

Перечитать конфигурацию на серверах

В устройстве определяются поля: **Идентификатор**, **Хост/порт**, **Логин**, **Пароль**, **Community/secret**. Назначение данных полей зависит от того, какой процессор использует данное устройство. Например, поле **Идентификатор** может быть использовано в качестве значения Nas-Identifier при обработке RADIUS-пакетов или в качестве agentRemoteId/MAC-адреса при обработке DHCP-пакетов. В конфигурации устройства указывается текстовая конфигурация, которая также зависит от процессора, использующего устройство.

Тип устройства позволяет задавать общие параметры для группы одинаковых устройств.

Конфигурация каждого устройства **наследует все параметры конфигурации своего типа устройства, а затем устройства-предка** (конфигурация которого также унаследована). Т.е. параметр, указанный в предке будет доступен во всех потомках и его можно переопределить в конфигурации типа устройства потомка и ещё раз - в конфигурации самого потомка. Это свойство можно использовать для определения одинаковых параметров для множества устройств с одним предком.

После корректировки параметров устройств для их перечитывания на серверах Access и Accounting необходимо нажать кнопку **Перечитать конфигурацию на серверах**, расположенную под деревом устройств. Данная кнопка позволяет оповещать сервера об изменениях только, когда конфигурация станет законченной. До тех пор сервера используют сохранённую в памяти конфигурацию.

В контекстном меню на каждом устройстве доступны следующие пункты:

Договоры - открывает список договоров, на которых добавлен сервис, привязанный к данному устройству.

Синхронизировать сервисы - синхронизирует все сервисы на устройстве, для каждого сервиса вызываются команды удаления и создания сервиса на устройстве.

Выполнить команду - вызывает команду для обработчика управления устройством. В данном случае нужно вбивать имя конкретного метода, которое нужно вызвать.

Удалить событие из очереди - Удаляет событие activemq для этого устройства. Бывает полезно, когда из-за ошибки в обработке активации сервисов событие все время падает при обработке, и не может быть обработано и попытки его обработки продолжаются.

Интерфейсы устройства - интерфейсы устройства. Описаны [тут](#).

Также есть команды - **Вырезать, Вставить, Удалить**, позволяющие редактировать дерево устройств.

Атрибуты устройства

Атрибуты устройства - способ привязать к устройству дополнительные параметры, например, текстовый или адрес. Набор атрибутов связан через объект Сущность, который указывается в типе устройства. Таким образом у всех устройств одного типа один и тот же набор атрибутов (но не значений этих атрибутов).

Чтобы создать сущность нужно открыть вкладку Справочники - Атрибуты. Далее на вкладке Атрибуты добавить необходимые, если были добавлены атрибуты типа "список", то на вкладке Значения списков нужно добавить значения для этих атрибутов-списков. Затем на вкладке сущность создать новую, например, "Коммутатор" и добавить в сущность ранее созданные атрибуты, отсортировать их в нужном порядке.

Договор Модули Плагины Справочники Сервис Утилиты Справка			
Помощник +			
Атрибуты			
ID	Тип атрибута	Название атрибута	
1	Текст	VLAN	
2	Дата	Дата заявки	
3	Число (int)	Ширина канала	
4	Число (int)	Число волокон	
5	Список	Тип клиента	
6	Список	Режим	
7	Список	Статус	
8	Список	Скидка	
9	Число (int)	Ширина	
10	Число (int)	Высота	
11	Адрес	Адрес	
12	Текст	Инвентарный номер	
13	Дата	Принят в эксплуатацию	

Атрибуты			
Значения списков		Сущности	
ID	Тип атрибута	Название атрибута	
5	Список	Тип клиента	
6	Список	Режим	
7	Список	Статус	
8	Список	Скидка	

Атрибуты			
Значения списков		Сущности	
ID	Название	Шаблон	
1	Шлюз 0		
2	Шлюз 1		
3	Шлюз 2		
4	Шлюз 3		sd ad fasdf dsf ads
5	Gate	\$title \$attr(11) \$attr(5) \$attr(13) \$attr(12)	
6	ффф1		
7	Тестовая сущность		комментарий
8	ValerAttrTest		

Название:
Gate

Шаблон имени:
\$title \$attr(11) \$attr(5) \$attr(13) \$attr(12)

Комментарий:

Выбранные атрибуты:
 Инвентарный номер [text]
 Адрес [address]
 Принят в эксплуатацию [date]
 Тип клиента [list]

Доступные атрибуты:
 VLAN [text]
 Дата заявки [date]
 Ширина канала [int]
 Число волокон [int]
 Режим [list]
 Статус [list]
 Скидка [list]
 Ширина [int]
 Высота [int]

Теперь созданная сущность будет доступна в редакторе типа устройства модуля Inet. После выбора сущности и сохранения типа устройства во всех устройствах данного типа будет соответствующий набор атрибутов в виде таблицы аналогично объектам или параметрам договора.

Корневые устройства

Все устройства разделены на группы авторизации, каждая со своим **корневым устройством**. Каждая из групп авторизации управляется отдельным изолированным Access-сервером, либо несколькими Access-серверами, которые работают в режиме кластера. Контроль IP-ресурсов, количества одновременных сессий сервисов производится только в пределах группы. Т.е. группы авторизации позволяют разделить предоставление услуг на не взаимосвязанные сегменты, что повышает масштабируемость.

В пределах группы авторизации может быть одна или несколько групп обработки. Каждая из групп обрабатывается своим Accounting-сервером, либо несколькими Accounting-серверами, работающими в режиме кластера. Корневой узел группы обработки должен быть выделен отдельным типом. Для простоты, корневые узлы группы авторизации и обработки могут быть объединены в один - таким во многих примерах в документации является корневое устройство "Access+Accounting".

Для корректной работы InetAccounting необходимо сконфигурировать в корневом устройстве обработчики тарификации и сброса данных в БД, представленные ниже.

В конфигурации корневого узла группы обработки (BGInetAccounting) **обязательно** нужно указать параметры тарификации и обработки соединений. Для этого нужно задать "worker", указав количество потоков для него, обработчики, которые будут работать в этом worker'e и их параметры. Каждый worker назначается с идентификатором, например, accounting.worker.1. и accounting.worker.2. Каждый обработчик также задаётся с идентификатором. Таким образом, может быть несколько worker'ов и несколько обработчиков в них.

От данных параметров зависит, как часто InetAccounting будет тарифицировать сессии и как быстро будет реагировать на изменения баланса, параметров тарифа и т.п., отправляя задания на изменения скорости/параметров сессий.

Параметры обработчика тарификации (tariffication):

- minDeltaAmount - минимальная сумма нетарифицированного трафика, при которой тарифицировать соединение. Включает в себя все трафики в байтах плюс время в секундах, которое ещё не было протарифицировано;
- delay - пауза между заданиями тарификации;
- batchSize - максимальное количество протарифицированных соединений за одно задание;
- checkAccountingPeriodError - флаг необходимости проверки ошибки учетного периода перед обработкой изменения опций из тарифа. Необходимость может быть вызвана прежде всего тем, что тарификатор не учитывает отсутствие учетного периода на отключенном договоре и при изменении набора опций, посылает команду на включение абонента. Возможные значения: **0** - отключено, **1** - включено. По умолчанию **0**.

Параметры обработчика соединений без трафика (tracking - используется для отслеживания изменения опций модуля для соединений, у которых нет трафика):

- delay (delay.millis) - пауза между заданиями в секундах (в миллисекундах);
- batchSize - максимальное количество протарифицированных соединений за одно задание.

Параметры обработчика сервисов без соединений (serv.tracking - используется для отслеживания изменения опций модуля для сервисов, у которых нет сессии и, соответственно, нет трафика):

- delay (delay.millis) - пауза между заданиями в секундах (в миллисекундах);
- batchSize - максимальное количество протарифицированных соединений за одно задание;
- servTypeIds - типы сервисов, которые нужно обрабатывать (если не указано - то все типы);
- accountingPeriodActivate - нужно ли активировать учётные периоды при проверке. Возможные значения: **1** - активировать, **0** - не активировать. По умолчанию **0**;

Параметры обработчика событий (event.tracking - предназначен для более быстрой реакции на события, например, изменение статуса договора, активация тарифной опции):

- delay (delay.millis) - пауза между заданиями в секундах (в миллисекундах);
- batchSize - максимальное количество обработанных сервисов за одно задание;
- servDeviceState - нужно ли реагировать на событие об изменении текущего состояния сервиса на устройстве после его фиксации в базе данных (InetServDeviceStateAndOptionsModifiedEvent). Возможные значения: **0** - отключено, **1** - включено. По умолчанию **1**;
- raument - нужно ли реагировать на события изменения баланса. Возможные значения: **0** - отключено, **1** - включено. По умолчанию **1**;
- capacity - размер очереди обрабатываемых событий. По умолчанию **10000**;
- processDelay - задержка в миллисекундах перед обработкой каждого из событий. Необходимо для того, чтобы все приложения обновили свои кэши после внесенных изменений, которые порождают события. По умолчанию значение **1500 мс**. В случае обработки события InetServDeviceStateAndOptionsModifiedEvent значение задержки увеличивает на **3100 мс**.

Параметры сброса в базу (flushing):

- minDeltaAccount - минимальная сумма несброшенной наработки соединения, при которой нужно сбрасывать в базу;
- delay (delay.millis) - пауза между заданиями в секундах (в миллисекундах);
- batchSize - максимальное количество сброшенных соединений за одно задание;
- disableServs - Если не null (null возможен, если данный параметр не задан вообще в конфигурации обработчика), то обрабатываются только обычные или только сервисы для неаутентифицированных;
- lockTimeout - величина таймаута блокировки в миллисекундах. Значение по умолчанию - **15000 мс**.

Параметры обработчика завершения сессий (finishing - завершает соединения, а также закрывает соединения по таймауту):

- delay (delay.millis) - пауза между заданиями в секундах (в миллисекундах);
- batchSize - максимальное количество протарифицированных соединений за одно задание;
- finishBatchSize - ограничение числа завершенных соединений. По умолчанию 0;
- lockTimeout - величина таймаута блокировки в миллисекундах. Значение по умолчанию - **15000 мс**.

Пример конфигурации:

```

Конфигурация

# worker
accounting.worker.1.thread.count=2
# :
# ,
accounting.worker.1.tariffication.1.minDeltaAmount=0
#
accounting.worker.1.tariffication.1.delay=10
#
accounting.worker.1.tariffication.1.batchSize=1000
# ( ):
#
accounting.worker.1.tracking.1.delay=10
#
accounting.worker.1.tracking.1.batchSize=1000
# ( , ):
#
accounting.worker.1.serv.tracking.1.delay=20
#
accounting.worker.1.serv.tracking.1.batchSize=100
# ( ):
# ( )
accounting.worker.1.event.tracking.1.delay.millis=200
#
accounting.worker.1.event.tracking.1.batchSize=500

# worker
accounting.worker.2.thread.count=1
# :
# ,
accounting.worker.2.flushing.1.minDeltaAccount=0
# ,
accounting.worker.2.flushing.1.minDeltaAmount=0
#
accounting.worker.2.flushing.1.delay=20
#
accounting.worker.2.flushing.1.batchSize=500

# worker
accounting.worker.3.thread.count=1
# :
#
accounting.worker.3.finishing.1.delay=20
#
accounting.worker.3.finishing.1.batchSize=500

```

Пример конфигурации с несколькими обработчиками тарификации (для более частой тарификации соединений с большим трафиком):

Конфигурация

```
# worker
accounting.worker.1.thread.count=3
# :
# ,
accounting.worker.1.tariffication.1.minDeltaAmount=104857600
#
accounting.worker.1.tariffication.1.delay=10
#
accounting.worker.1.tariffication.1.batchSize=1000
accounting.worker.1.tariffication.2.minDeltaAmount=10485760
accounting.worker.1.tariffication.2.delay=10
accounting.worker.1.tariffication.2.batchSize=1000
accounting.worker.1.tariffication.3.minDeltaAmount=0
accounting.worker.1.tariffication.3.delay=15
accounting.worker.1.tariffication.3.batchSize=1000
# ( ):
#
accounting.worker.1.tracking.1.delay=20
#
accounting.worker.1.tracking.1.batchSize=1000
# ( ):
# ( )
accounting.worker.1.event.tracking.1.delay.millis=200
#
accounting.worker.1.event.tracking.1.batchSize=500

# worker
accounting.worker.2.thread.count=1
# :
# ,
accounting.worker.2.flushing.1.minDeltaAccount=0
# ,
accounting.worker.2.flushing.1.minDeltaAmount=0
#
accounting.worker.2.flushing.1.delay=30
#
accounting.worker.2.flushing.1.batchSize=500

# worker
accounting.worker.3.thread.count=1
# :
#
accounting.worker.3.finishing.1.delay=20
#
accounting.worker.3.finishing.1.batchSize=500
```

Для обработки сервисов без сессий нужно добавить ещё один обработчик (а также можно увеличить количество потоков для accounting.worker.1 или вынести обработчик в отдельный worker):

```
# ( ):
#
accounting.worker.1.serv.tracking.1.delay=60
#
accounting.worker.1.serv.tracking.1.batchSize=500
# , ,
# ( - ! )
accounting.worker.1.serv.tracking.1.servTypeIds=
#
# ( RADIUS/DHCP )
accounting.worker.1.serv.tracking.1.accountingPeriodActivate=1
```

Интерфейсы

Правой кнопкой на любом устройстве можно вызвать интерфейсы из его контекстного меню.

Интерфейс	Название	Индекс	Статус	Категория IP	Подписчик	Комментарий
1	интерфейс 1	1	доступен			

Редактор
Интерфейс:

Индекс:

Название:

Статус:

Комментарий:

Индекс Категория IP-ресурсов

ID	Индекс	Период	...
----	--------	--------	-----

Сохранить Отмена

Тут все зависит от настроек типа устройства. Если в типе устройства стоит галочка "Индивидуальные интерфейсы", тогда их нужно создавать в типе устройства и они автоматически будут созданы на всех устройствах (если вы создали индивидуальные интерфейсы на устройствах, которых нет в типе устройства, то они будут удалены при сохранении типа устройства). Если же галочка "Индивидуальные интерфейсы" указана, то необходимо на каждом устройстве создавать интерфейсы вручную.

В интерфейсе можно задавать индекс.

Редактор
Интерфейс:

Индекс:

Название:

Статус:

Комментарий:

Индекс Категория IP-ресурсов

ID	Индекс	Период	...
0	5	27.01.2014 00:00:00 - ...	

Сохранить Отмена

Индекс интерфейса используется для управления по snmp и сборе статистики по netflow. По умолчанию, если не указан, то индекс приравнивается номеру интерфейса. В тех же случаях, где индекс отличается от номера интерфейса, его можно задать явно. Так же для индекса может быть указан период с точностью до секунд на который данный индекс действует. Может быть полезно, например, в случае, когда cisco после перезагрузки меняет индексы интерфейсов и в Netflow для того же самого интерфейса приходит запись с новым индексом.

Опции

Опции - абстрактные сущности, набор которых привязывается к сервису или сессии через параметры сервиса и тариф договора. Опция может определять скорость или любые другие параметры подключения для клиента.

Название	Конфигурация	ID
Опции		0
Группа: Скорость		1
1024kb (5)		5
128kb (2)		2
256kb (3)		3
512kb (4)		4

Редактор

Название
Скорость

Пересечение в группе возможно

Комментарий

Конфигурация

Сохранить Отмена

Опции можно выстраивать в группы, как для удобства, так и для того, чтобы опции внутри группы не пересекались.

По умолчанию при прохождении тарифного дерева указанные опции просто добавляются в набор, однако, если опции выделить в отдельную группу и **не устанавливать** галочку в поле "Пересечение в группе возможно", то установка очередной опции из группы удалит все другие опции из той же группы. Такую группу, например, можно создать для указания скоростей - тогда при использовании тарифных веток диапазонов (а при тарификации возможна ситуация, когда запрос зайдёт сначала в одну ветку диапазона, заполнив её, а затем в следующую, т.к. есть остаток), опция скорости в итоге всегда будет одна - последняя установленная при прохождении тарифа.

При изменении набора опций в сессии, например, при очередной тарификации или при добавлении/удалении опции в сервисе на договоре автоматически вызывается обработчик активации сервисов - вызывается `serviceModify` и, если в типе сервиса указан тип инициации сессии "по сигналу" (например, для RADIUS-сессий), для каждой сессии сервиса - `connectionModify`.

Типы сервисов

Тип сервиса определяет параметры, которые должны быть указаны в сервисе клиента. Типы сервисов редактируются на одноимённой вкладке модуля с использованием стандартной панели инструментов.

ID	Название
1	IPoE порт
2	IPoE VLAN
3	Абонентское устройство
4	ISG-сервис (ISG_SERVICE)
5	NOAUTH
6	SNMP Collector

Редактор
Название: IPoE порт

Тип инициации сессии: по сигналу

Кол-во сессий: 1 неизменно

Тип адреса: динамический или статический адрес весь интерфейс

Привязка типов трафика: нет

логин + пароль устройство
 интерфейс интерфейс персонален
 VLAN VLAN персонален
 MAC-адрес идентификатор
 объект договора ограничения
 IP из родительского диапазона

Группы устройств: Родительские типы
Конфигурация: Типы устройств

```
# Шаблон имени сервиса в договоре
#title.pattern=Интернет (${deviceIdentifier})@
(${interfaceId})
title.pattern=Интернет ((${interfaceTitle}))
```

Сохранить Отмена

Параметры типа сервиса:

- **Тип инициации сессии** - по сигналу (RADIUS, DHCP-пакет), либо по трафику;
- **Кол-во сессий** - максимальное одновременно возможное количество сессий по одному сервису договора данного типа;
- **Тип адреса** - какой тип адреса соотносится сервису, возможные значения рассмотрены далее;
- **весь интерфейс** - указывает, что при тарификации по Netflow/sFlow/SNMP будет учитываться только интерфейс, независимо от диапазона, указанных или не указанных в сервисе договора (т.е. как если в договоре указан диапазон 0.0.0.0-255.255.255.255), может быть удобно при тарификации по SNMP;
- **Привязка типов трафика** - используемая для сервиса привязка типов трафика.

Тип адреса может принимать следующие значения:

- **Не выделять адрес** - сервис не предполагает использования адреса;
- **Статический диапазон** - статический диапазон адресов указывается непосредственно в сервисе;
- **Статическая сеть** - статическая сеть адресов указывается непосредственно в сервисе;
- **Статический адрес** - статический адрес, указывается непосредственно в сервисе;
- **Динамический адрес** - адрес выдаётся динамически на каждую сессию сервиса из пула;
- **Динамический или статический адрес** - адрес выдаётся динамически на каждую сессию сервиса из пула, но может всё время принимать одно указанное значение;
- **Динамический адрес или статический диапазон** - адрес выдаётся динамически на каждую сессию сервиса из пула, но может выдаваться из указанного непосредственно в сервисе диапазона адресов.

Флаги в нижней области определяют какие параметры должны быть указаны и будут доступны в сервисе договора. Дополнительные флаги:

- **объект договора** - привязка сервиса к **объекту договора**;
- **IP из родительского диапазона** - следует установить, если Вы хотите указывать в дочерних сервисах IP-адреса из диапазона родительского сервиса (например, для последующего использования при синхронизации с коммутатором или иным устройством);
- **интерфейс персонален** - установлено по умолчанию (рекомендуется не изменять, если не знаете для чего Вам это нужно) - интерфейс может быть привязан только к одному сервису, если же флаг не установлен - один и тот же интерфейс можно указать на нескольких сервисах договора;

- **VLAN персонален** - аналогично флагу **интерфейс персонален**, только для VLAN.

В конфигурации типа сервиса могут быть определены переменные:

1. **title.pattern**=(**<шаблон имени сервиса>**), в шаблоне имени возможны переменные:

- **#{login}** - логин;
- **#{deviceId}** - идентификатор устройства, к которому привязан сервис;
- **#{deviceTitle}** - полное название устройства, к которому привязан сервис;
- **#{interfaceId}** - код интерфейса, указанного для сервиса;
- **#{interfaceTitle}** - название интерфейса, указанного для сервиса;
- **#{vlan}** - VLAN, указанный для сервиса;
- **#{addressIp}** - адрес, указанный для сервиса;
- **#{addressRange}** - диапазон адресов, указанные для сервиса;
- **#{addressNet}** - сеть, указанная для сервиса;
- **#{macAddress}** - MAC-адрес(а), указанный для сервиса;
- **#{identifier}** - идентификаторы, указанные для сервиса.

2. **const.device.id**=**<постоянный код устройства для всех сервисов данного типа>**.

3. **Атрибуты**, которые можно выдавать индивидуально сервисам

4. **dhcp.disable.mode**. Режим выдачи ip-адреса, специфичный для типа сервиса. Возможные значения указаны в одноименной опции в разделе **выдачи ip-адреса**. Данная опция имеет БОЛЬШИЙ приоритет над одноименной опцией, указываемой в устройстве/типе устройства.

5. **radius.realm** - список реалмов через запятую, через которые можно подключаться. Аналогично опции **radius.realm** в конфигурации устройства (типа), но имеет БОЛЬШИЙ приоритет.

6. **serv.identifier.auto** - нужно ли устанавливать идентификатор в сервис, если он не был установлен. Аналогично опции **serv.identifier.auto** в конфигурации устройства (типа), но имеет БОЛЬШИЙ приоритет. Возможные значения: **0** - Проверки идентификатора нет, **1** - Автоматическая установка идентификатора в сервис, если еще не был установлен; иначе - проверка, **2** - Автоматическая установка идентификатора в сервис; замена, если текущий идентификатор не совпадает, **3** - Добавление нового идентификатора в сервис. По-умолчанию - 0.

7. **serv.macAddress.auto** - нужно ли проставлять MAC-адрес в сервис, если он еще не был установлен. Аналогично опции **serv.macAddress.auto** в конфигурации устройства (типа), но имеет БОЛЬШИЙ приоритет. Возможные значения: **0** - Проверки MAC-адреса нет, **1** - Автоматическая установка MAC-адреса в сервис, если еще не был установлен; иначе - проверка, **2** - Автоматическая установка MAC-адреса в сервис; замена, если текущий идентификатор не совпадает, **3** - Добавление нового MAC-адреса в сервис. По-умолчанию - 0.

8. **serv.device.link** - нужно ли делать проверку устройства, с которого пришел запрос, и устройства, к которому привязан сервис. Возможные значения: **0** - проверка отключена, **1** - проверять привязку на конкретно этом устройстве, **2** - проверять все дочерние устройства для устройства, на которое пришел запрос абонента. По-умолчанию - 0.

9. **serv.radius.framedIp** - Не используется.

10. **serv.radius.framedRoute** - Устанавливается для дочерних сервисов. Означает, что подсеть сервиса добавляется в атрибут **Framed-Route** radius-ответа. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**. Не путать с **serv.radius.staticSubnet**

11. **serv.radius.delegatedPrefix** или **serv.radius.delegatedIpPrefix** - для выдачи ipv6-адреса, прописанного в дочернем сервисе. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

12. **serv.radius.secondFramedIp** - поддержка вторичной авторизации для схемы с isg. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

13. **serv.search.mac** - поиск сервиса по MAC. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**. Работает с типами поиска, связанными с MAC.

14. **serv.search.address** - поиск сервиса по ip-адресу. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**. Работает с типом поиска по ip-адресу в конфигурации устройства (типа)

15. **serv.search.login** - поиск по логину для дочерних сервисов. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**

16. **serv.radius.staticSubnet** - Выдавать статическую подсеть из сервиса. Означает, что выдает **Framed-IP-Address + Framed-IP-Netmask**. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**. Не путать с **serv.radius.framedRoute**.

17. **serv.flow.staticSubnet** - Всегда добавлять подсеть в отчет Netflow RADIUS-сессии (даже если в аккаунтинге нет атрибутов **Framed-Route** и **Framed-IP-Netmask**). Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

18. **serv.radius.noAddress** - Вообще не выдавать адрес (атрибут **Framed-IP-Address**). Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

19. **serv.dhcp.noAddress** - Вообще не выдавать адрес (для DHCP). Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

20. **serv.radius.skipStaticAddress** - Не выдавать адрес из этого сервиса даже если диапазон/подсеть указан в сервисе. Может быть указан как на родительском, так и на дочернем сервисе. Означает, что диапазон/подсеть сервиса используется для управления. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

21. **serv.radius.password.verification** - Нужно ли проверять пароль при **Access-Request**. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.
22. **serv.radius.connection.checkDuplicate** - проверка на дублирование radius-сессий. Аналогично опциям **radius.connection.checkDuplicate** или **checkduplicate.session** в конфигурации устройства (типа), но имеет БОЛЬШИЙ приоритет. Подробнее смотрите в разделе [InetRadiusProcessor](#).
23. **serv.radius.connection.checkDuplicate.type** - тип проверки на дублирование сессий. Возможные значения **0 (false)** - по атрибуту **Calling-Station-Id**, **1 (true)** - по **User-Name**. По умолчанию **0**
24. **serv.dhcp.connection.checkDuplicate** - проверка на дублирование dhcp-сессий. Подробнее в разделе [InetDhcpProcessor](#).
25. **serv.flow.active.filterIn** - Фильтр активности по входящему трафику - если true то по входящему трафику сессия создаваться не будет и при отсутствии исходящего - закроется по таймауту. Используется для сессий по трафику. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**
26. **serv.flow.active.filterOut** - Фильтр активности по исходящему трафику - если true то по исходящему трафику сессия создаваться не будет и при отсутствии входящего - закроется по таймауту. Используется для сессий по трафику. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.
27. **serv.dhcp.connection.closeOnNew** или **dhcp.connection.closeOnNew** - нужно ли закрывать активные текущие сессии при обработке discover-запросов. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.
- Вместо этого флага рекомендуется проверка на кол-во сессий (**serv.dhcp.connection.checkDuplicate**)
28. **serv.connection.suspend.timeout** - таймаут НЕполучения update-пакетов в секундах для сессий по сигналу (radius/dhcp) - . Если это время выходит, то сессия переключается в suspended. По умолчанию **0**
29. **serv.connection.close.timeout** - для сессий по сигналу (radius/dhcp) - таймаут, после которого suspended сессия будет завершена (аналог **zombie.timeout** в dialup). Для сессий по трафику - таймаут неполучения информации по трафику, после которого сессия будет завершена. Измеряется в секундах. По умолчанию **0**.
30. **serv.connection.disable.suspend.timeout** - таймаут НЕполучения update-пакетов в секундах для **DISABLE** сессий по сигналу (radius/dhcp) - . Если это время выходит, то сессия переключается в suspended. Измеряется в секундах. По умолчанию **0**.
31. **serv.connection.disable.close.timeout** - для сессий по сигналу (radius/dhcp) - таймаут, после которого suspended сессия будет завершена (аналог **zombie.timeout** в dialup). Измеряется в секундах. По умолчанию **0**.
32. **sa.connection.auto.accountingEvent** - нужно ли генерировать start-stop accounting-пакеты для сессий по трафику. Это может понадобиться для генерации фейковых radius- пакетов для СОРМа при помощи сервис-активатора **RadiusFanoutServiceActivator**. Возможные значения **0 (false)** - отключено, **1 (true)** - включено. По умолчанию **0**.

Пример конфигурации типа сервиса:

Конфигурация

```
#
title.pattern=${login}

#
#
#const.device.id=

#
#
serv.login.chars=1234567890ABCDEFGHIJKLMNQRSTUvwxyz
# ,
serv.login.chars.description=

# . , ,
# ( ):
#
#serv.password.length.min=5
#
#serv.password.length.max=16
# ( )
#serv.password.chars=1234567890ABCDEFGHIJKLMNQRSTUvwxyz
# ,
#serv.password.chars.description= .
#
serv.password.length.auto=6
# ( serv.password.chars)
#serv.password.chars.auto=<serv.password.chars>

# RADIUS- . , , .
# default, ..
#radius.realm=default

# MAC-, .
# , , .
#serv.macAddress.auto=0
```

Сервисы

Для предоставления клиенту возможности выхода в сеть необходимо добавить в договор сервис сконфигурированного заранее типа. Все сервисы договора отображаются в сводной таблице.

The screenshot shows a software interface with a top menu bar containing 'Параметры', 'Отчет', 'Карточки', 'HelpDesk', 'Рассылки', 'CRM', and 'Документы'. Below the menu, there is a 'Договор' section with the contract ID 'x0000 [ФИО - asdasd1]' and a 'Период' section with the date '06.02.2005'. The main area is divided into two panes. The left pane shows a tree view of contract details, including 'Статус: Активен', 'Лимит: 200074.00', 'Режим: дебет', 'Лицо: физическое', and a 'Баланс (март 2011)' section with sub-items like 'Входящий остаток: 103762.64', 'Повход: 208.48', 'Начебатея: 0.00', 'Расход: 103970.00', and 'Исходящий остаток: 1.09'. The right pane is titled 'Сервисы' and contains a table with columns: ID, Тип, Устройство, Название, Период, Статус, and Состояние. The table lists several services with their respective details.

ID	Тип	Устройство	Название	Период	Статус	Состояние
1	VPN соединение	Наз. МРО: шпр [81.30.100.255]	Логин: Бава; Ад...	01.02.2011-	открыт	отключен
7	DNS-сервис авториз.	[Бомбонбон]	Укажите парол...	08.02.2011-	открыт	отключен
9	Статическая сеть	Сыктывкар	Адрес(а) 192.16...	17.02.2011-	открыт	отключен
12	Статический для ..	Коммутатор с Орбонс.82 авто...	Адрес(а) 192.16...	20.04.2011-	открыт	отключен
13	статический ф...	Коммутатор с Орбонс.82 авто...	Укажите парол...	26.04.2011-	открыт	отключен

Столбцы таблицы:

- **Тип** - тип сервиса;
- **Устройство** - устройство, к которому привязан сервис;
- **Название** - сгенерированное на основании шаблона из типа сервиса название сервиса договора;
- **Период** - период действия;
- **Статус** - текущий статус сервиса;
- **Состояние** - реальное состояние сервиса на устройстве.

Статус сервиса может принимать следующие значения: **открыт**, **закрыт** и **заблокирован**. Статус сервиса переключается вручную.

Состояние сервиса может быть "подключён", либо "отключен", отображает реальное состояние, которое было установлено на устройстве. Между состояниями "подключен" и "отключен" модуль осуществляет автоматическое переключение в зависимости от состояния баланса (больше, либо меньше лимита), статуса договора, статуса сервиса. Сервис подключен, когда в договоре установлен активный для него статус, сервис в статусе открыт и баланс больше лимита.

Сервисы **Учётные периоды**

Настройки **Опции**

Тип: Статический диапа...
Период: с 20.04.2011 по
Статус: Открыт
Кол-во сессий: не ограничено

Устройство: Коммутатор с Options.82 авторизацией: [127.0.0.1] <<<
Интерфейс: 0 <<<

Адрес: 192.168.184.35 - 192.168.184.255 <<<

Комментарий

Сохранить Отмена

В зависимости от настроек [типа сервиса](#) в редакторе сервиса могут присутствовать различные поля. Обязателен тип, период, статус, количество сессий, устройство. Устройство может быть указано постоянным для всех сервисов одного типа с помощью переменной в конфигурации типа сервиса. Это может быть необходимым в случае идентификации по логину. Привязка к устройствам необходима для возможности разделения базы сервисов модуля. IP-адрес (диапазон) при сохранении проверяется на вхождение в ресурс. Для того, чтобы выбрать свободный адрес из ресурсов, нужно нажать на кнопку "<<<" возле адреса.

Сервисы **Учётные периоды**

Ресурсы адресов

Ресурс: 192.168.184.0–192.168.184.255 (4) Маска/диапазон: 1 Диапазон Максимум: 20

На дату: 16.03.2012 **Вывести**

192.168.184.11–192.168.184.11
 192.168.184.19–192.168.184.19
 192.168.184.20–192.168.184.20
 192.168.184.22–192.168.184.22
 192.168.184.23–192.168.184.23
 192.168.184.24–192.168.184.24
 192.168.184.25–192.168.184.25
 192.168.184.26–192.168.184.26
 192.168.184.27–192.168.184.27
 192.168.184.28–192.168.184.28
 192.168.184.29–192.168.184.29
 192.168.184.30–192.168.184.30
 192.168.184.31–192.168.184.31
 192.168.184.32–192.168.184.32
 192.168.184.33–192.168.184.33
 192.168.184.34–192.168.184.34
 192.168.184.255–192.168.184.255

Ок Отмена

Сохранить Отмена

Тут можно указать диапазон и количество адресов в диапазоне, либо сеть и маску сети. Адрес выбирается из категории ресурса. Алгоритм определения категории такой:

Если интерфейс является обязательным полем и в настройке [интерфейса, с которого идет трафик](#) указана категория IP-адресов, то выбирается эта категория. В противном случае в конфигурации устройства ищется опция

```
ip.resource.categoryId=1
```

Где 1-это код категории IP-адресов. Тут можно указать несколько категорий через запятую.

Нажатие кнопки "<<<" возле поля выбора VLAN (если это поле указано как обязательное в типе сервиса) выбирает первый свободный VLAN из ресурсов. Свободными считаются те, которые выбраны уже на другом сервисе.

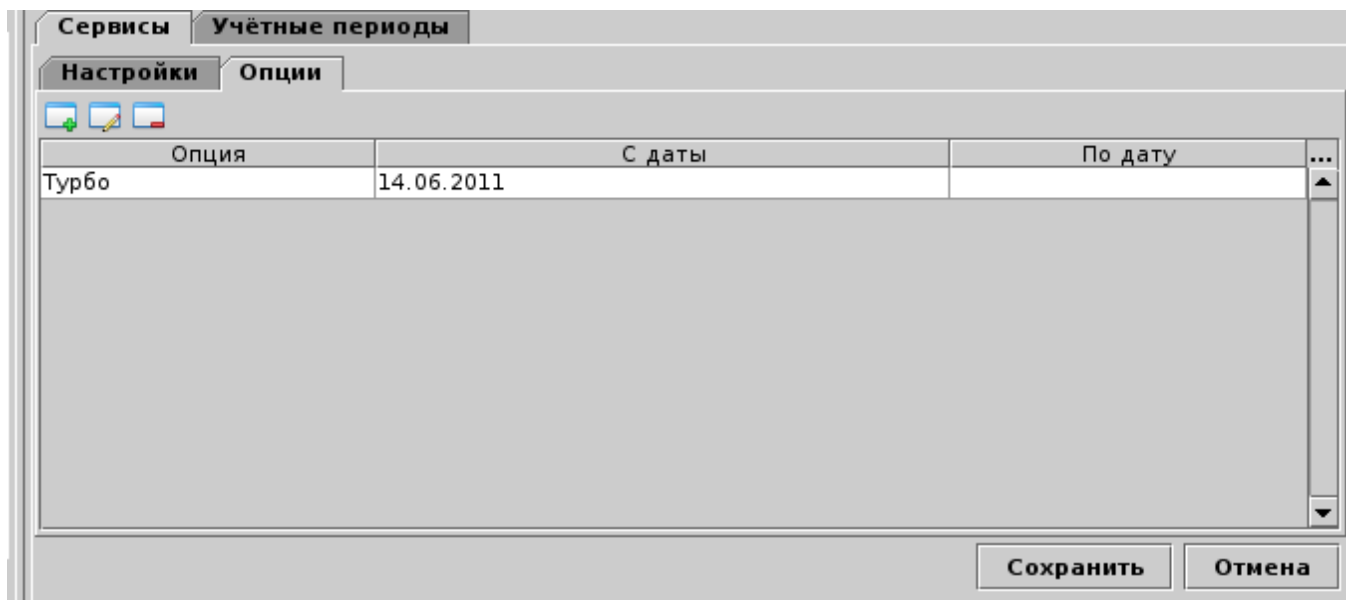
Ресурсы VLAN выбираются из категории, которая определяется опцией в конфигурации устройства:

```
vlan.resource.category=1
```

Где 1-это код категории VLAN.

Нажатие кнопки "<<<" возле поля выбора интерфейса (если это поле указано как обязательное в типе сервиса) выбирает первый свободный интерфейс из ресурсов. Доступны те интерфейсы, которые имеют [статус Доступен на устройстве](#) и не выбраны уже на другом сервисе.

На вкладке **Опции** сервиса указываются статически определённые для данного сервиса опции.



При активации сервиса первыми применяются опции из тарифного плана, после - указанные в сервисе, они могут переопределить или дополнить опции из тарифа. Конкретная реализация опций: RADIUS-атрибуты, либо какие-то правила файрвола, задаются классом обработчиком активации сервиса для конкретного типа устройства, либо процессором Access-сервера.

Вкладка **Учётные периоды** настроек модуля определяет учётные периоды. Периоды активируются скриптами по различным событиям. К периодам могут быть привязаны диапазоны наработки тарифных планов.

Для сервисов, в которых вводится логин возможна генерация логина/пароля автоматически. Ввод логина выглядит так:

Сгенерировать логин или пароль автоматически можно, если проставить галочку **авто**. Параметры для автоматической генерации логина настраиваются в [конфигурации модуля](#). При генерации следующего логина идет поиск максимального числового и к нему прибавляется 1, поэтому при смешении автоматической генерации и ручного ввода рекомендуется при ручном вводе логин начинать с буквы, а не цифры.

Шаблоны договоров Inet

В [шаблоне договора](#) можно указать автоматическое добавление сервиса при создании договора. Для этого, кроме простого добавления модуля, нужно установить галочку **Выполнить** на вкладке **Создать сервис** и указать параметры, которые будут применяться к каждому автоматически добавляемому сервису: тип сервиса, статус по умолчанию, кол-во сессий по умолчанию.

[Тип сервиса](#) необходимо выбирать только из тех, у которых указан параметр **const.device.id**, чтобы при создании сервиса он был привязан к определенному устройству. Исключение составляет только шаблон договора для создания по . В последнем случае, если **const.device.id** не указан, сервис будет привязан к тому устройству, с которого происходила активация карты.

Дополнительно можно указать опции, которые будут привязаны к добавляемому сервису.

Шаблоны

- ADSL — физические лица
- ADSL — юридические лица
- Dial-up по картам
- Dial-up физ. лица
- Dial-up юр. лица
- Ethernet - Inet (Физические)**
- PoN
- PoP - юридические лица
- Тестовый шаблон

Название шаблона: Ethernet - Inet (Физические)

Шаблон имени: ИФ-0\${N4}

Лимит: 0

Статус: Активен

Лицо: Ю Ф

Режим: Д К

Время жизни (дни, 0 - неограничено): 0

Модули Плагины Группы Шаблоны комментариев Параметры Группы тарифов и тарифы Скрипт поведения Объекты

Создать сервис

Выполнить

Тип: SE

Статус: Открыт

Кол-во сессий: 1

Опции

Опция

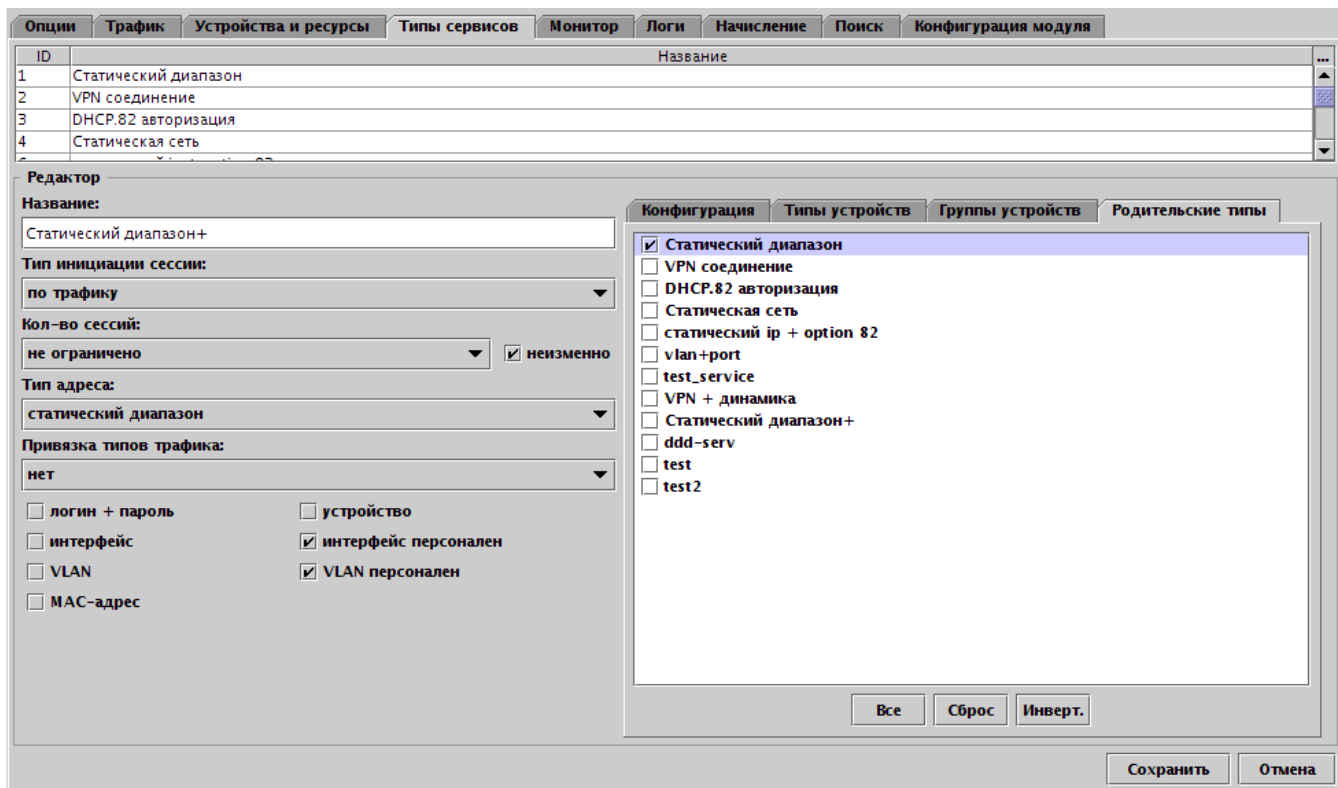
Восстановить

Ок Отмена

Дочерние сервисы

При инициации сессии по трафику, когда обсчет идет по netflow/sFlow, а сессии создаются по наличию трафика и закрываются, когда трафика нет определенное время, возможна ситуация, когда необходимо обсчитывать несколько подсетей, но управление устройством осуществляется в одном месте. Т.е. точка подключения одна, а обсчитываемых диапазонов - несколько. В этом случае понадобятся дочерние сервисы - они предназначены только для обсчета, а не для управления.

Для того, чтобы появилась возможность добавлять дочерние сервисы на родительский необходимо сначала создать тип сервиса, в котором не установлены галочки **устройство, интерфейс/vlan** (все это указывается на родительском сервисе), а на закладке Родительские типы необходимо указать тип (или типы) сервиса, который будет являться родительским для данного.



После создания такого типа сервиса, если добавлять сервис на договор при невыделенном сервисе или при выделенном сервисе, тип которого не проставлен как родительский для другого, то в добавляемом сервисе будут доступны только типы сервиса, у которых родительские типы не указаны и сервис, в итоге, будет добавлен как обычно.

Если же выделен сервис, у типа которого есть дочерние типы (т.е. те, у которых указан родительский тип, тот же что и тип выделенного сервиса), то в редакторе будут доступны для выбора только эти дочерние типы и сервис после сохранения добавится как дочерний к тому, что был выделен перед добавлением.

		Сервисы				
		Сервис	Период	Статус	Состояние	ID
и договора		Сервисы				
договора	5/6	LOGIN:babai	01.02.2011-...	открыт	отключен [Дост...	1
договоров	супер: 2/1	Адрес(а): 192.168.184.35-192.168.184.254	20.04.2011-...	открыт	отключен [Недо...	12
	Активен	Адрес(а): 192.168.184.11-192.168.184.11	19.03.2012-...			66
	0.00	LOGIN:aaaaaa	03.11.2011-...	открыт	отключен [Дост...	27
	кредит	Адрес(а): 192.168.184.7-192.168.184.7	27.01.2012-...	открыт	включен	55
	Физическое					
рт 2012)						
дий остаток	-10.03					
д	10.00					
отка	-0.02					
1	0.00					
ящий остаток	-0.01					
планы						
Ru						
Money						
Crypt2						
о						

Т.к. дочерний сервис не является точкой подключения, у него нет собственного статуса и состояния - они наследуются от родительского сервиса.

Поиск сервиса Inet

В модуле inet возможен поиск по логину, IP-адресу, названию, типу, идентификатору, VLAN, MAC-адресу сервиса. Доступен на вкладке "Поиск".

Монитор	Поиск	Устройства и ресурсы	Логи	Переобсчет	Конфигурация модуля	Опции	Типы сервисов	Трафик		
Логин	<input type="text"/>	X	IP адрес	<input type="text"/>	X	Название	<input type="text"/>	X	Тип сервиса	Любой
Идентификатор	<input type="text"/>	X	MAC-адрес	<input type="text"/>	X	VLAN	<input type="text"/>	X	Найти	

По двойному клику на записи результатов открывается договор с найденным сервисом.

Сессии

- Статус сессии
- Состояние сессии
 - Рекомендуемые параметры для определения состояния сессии
 - Обычный CoA/PoD
 - Cisco ISG/SmartEdge
- Старт сессии по UPDATE-пакету
- Логическое разбиение сессий
- Пример конфигурации устройства

Статус сессии

В ходе жизненного цикла сессия, привязанная к устройству, проходит несколько статусов: ожидание (waiting), активна (working), приостановлена (suspended), закрыта (closed) и завершена (finished). Сессии могут создаваться/закрываться по сигналу (RADIUS, DHCP), либо по наличию трафика (netflow/sflow) и его отсутствию в течение какого-то времени (автосессии).

При типе инициации "по сигналу" соединение может быть приостановлено, а затем закрыто при отсутствии в течение определённого времени пакетов, подтверждающих её активность (RADIUS-Update, DHCP-Request). Таймауты после последнего подобного пакета задаются в секундах переменными конфигурации устройства **connection.suspend.timeout** (таймаут после последнего подтверждающего пакета, после которого соединение будет считаться приостановленным) и **connection.close.timeout** (таймаут после последнего подтверждающего пакета, когда соединение будет закрыто, если оно уже приостановлено). Приход подтверждающего пакета переводит соединение из приостановленного статуса в активный. Значения параметры **connection.suspend.timeout** и **connection.close.timeout** рекомендуется указывать чуть больше, чем 2 * (время между RADIUS/DHCP пакетами). При завершении соединения по сигналу Stop-пакетом (RADIUS-Stop) оно фактически завершается через количество секунд, определяемое переменной **connection.finish.timeout**. Это позволяет, в частности, реализовать сбор "запоздалой" информации о трафике, которая может прийти после Stop-пакета.

После перезапуска InetAccounting временем последней активности сессии считает время запуска InetAccounting (а не время из столбца "Активность" сессии). Это сделано для того, чтобы при возможном долгом простое InetAccounting, после его запуска он не начал завершать все сессии по таймауту.

Для автоматических сессий (сессий по трафику) параметр **connection.close.timeout** определяет время в секундах после последнего поступления информации о трафике данной сессии, по прошествии которого, сессия будет завершена. Для таких сессий не рекомендуется устанавливать слишком маленькое значение **connection.close.timeout**. Параметр **connection.auto.minDuration** указывает минимальную длительность для сессии по трафику. Рекомендуемые значения для сессий по трафику:

```
connection.close.timeout=3600
connection.auto.minDuration=1800
```

Приостановленная сессия не учитывается в ограничении на количество сессий, однако её IP-адрес не выдаётся до тех пор, пока она не завершена.

Для сессий в состоянии "отключена" можно назначить другие значения таймаутов после последнего RADIUS-пакета: **connection.disable.suspend.timeout** (по умолчанию равен значению **connection.suspend.timeout**) и **connection.disable.close.timeout** (по умолчанию равен значению **connection.close.timeout**), т.к. на отдельных маршрутизаторах для сессий в таком состоянии можно увеличить интервал послышки update-пакетов.

Состояние сессии

Также как у сервиса, у сессии может быть состояние: **подключена** или **отключена**. Состояние "**подключена**" означает, что **доступ открыт**, клиент может нормально работать, "**отключена**" - означает, что **доступ закрыт** - используется для схем без физического отключения сессий, например, когда вместо PoD-пакета при нехватке денег на балансе отправляется CoA с запретом всех направлений, кроме сервера статистики; или при инициации сессии по трафику, когда доступом управляет коммутатор: в состоянии "отключена" тарификация сессии не производится.

По умолчанию состояние сессии всегда "**подключена**". Переключить состояние сессии можно в обработчике процессора протокола при приходе update-пакета RADIUS (или же start/stop пакета сервисной сессии), в зависимости от содержимого этого пакета: **request.setOption(InetRadiusProcessor.DEVICE_STATE, InetServ.STATE_ENABLE)** или **request.setOption(InetRadiusProcessor.DEVICE_STATE, InetServ.STATE_DISABLE)**. А также в обработчике активации сервисов, если по RADIUS-пакетам не ясно в каком состоянии сессия: для этого в **connectionModify(e)** при обработке переключения доступа нужно установить **e.setConnectionStateModified(true)**. В стандартных обработчиках этот функционал уже реализован и требует только настройки для соответствующего обработчика - например, **sa.radius.connection.stateModify=1** для CoAServiceActivator меняет состояние после получения ACK на CoA-запрос блокировки или шаблон **radius.disable.pattern.attributes** для CoAProtocolHandler меняет состояние по наличию или отсутствию указанного набора атрибутов, а **radius.serviceName.disable** для ISGProtocolHandler/SmartEdgeProtocolHandler меняет состояние в зависимости от наличия или отсутствия сервисной сессии с указанным именем.

При работе с RADIUS, если по Start-пакету нельзя определить в каком состоянии началась сессия или ни в Start-пакете, ни в Update от NASa не приходит IP-адрес, в конфигурации устройства - NAS'a, или в конфигурации его типа устройства, или в одном из предков следует указать параметр **connection.start.fromAccept=1**. В этом режиме при отправке Access-сервером Access-Accept в ответ на Access-Request в базу будет добавлена запись по сессии в статусе ожидание (waiting), которую считает Accounting-сервер при получении Start-пакета. По умолчанию этот режим отключен.

Рекомендуемые параметры для определения состояния сессии

Суммируя вышесказанное, за определение текущего состояния сессии отвечают параметры:

```
# 1 Access-Accept InetAccess      ()
connection.start.fromAccept=1
# Accounting- ( )
radius.disable.pattern.attributes=
# 1 InetAccess      connectionModify CoAServiceActivator/ISGServiceActivator/SmartEdgeServiceActivator
( e.setConnectionStateModified( true ), CoAServiceActivator)
sa.radius.connection.stateModify=1
# ISG, SmartEdge      ISG/SmartEdge ( )
radius.serviceName.disable=
```

При этом в разных схемах необходимо указывать только некоторые из них (см. далее).

Обычный CoA/PoD

```
# Access-Accept InetAccess
connection.start.fromAccept=1
# InetAccess CoA
sa.radius.connection.stateModify=1
```

Cisco ISG/SmartEdge

```
# Access-Accept InetAccess
connection.start.fromAccept=1
# InetAccess CoA
sa.radius.connection.stateModify=0
# RADIUS-
radius.serviceName.disable=<( ) , >
```

Старт сессии по UPDATE-пакету

Бывают ситуации, когда start-пакет не дошел до Accounting-сервера. В этом случае, при **connection.start.fromUpdate=1** (значение по умолчанию) сессия создастся от текущего момента. При **connection.start.fromUpdate=2** Accounting проверит, что время сессии из update/stop пакета не больше, чем значение **connection.close.timeout** и создаст сессию от ее начала, иначе, если время сессии больше чем **connection.close.timeout**, сессия создастся от текущего момента. При **connection.start.fromUpdate=0** сессия без start-пакета создана не будет.

Логическое разбиение сессий

При наступлении нового дня происходит логическое разбиение сессии, т.е. сессия заканчивается в 23:59:59 предыдущего дня и начинается новая в 00:00:00 нового дня. При этом физического разрыва подключения абонента не происходит. Также для правильного переобсчета сессия логически разбивается при активации или деактивации тарифной опции. При посервисном аккаунтинге (SmartEdge или ISG) необходимость в разбиении отпадает - для того, чтобы отключить этот режим, в конфигурации нужно указать **session.split.onTariffOption=0**.

При использовании схем без физического отключения сессий (исключая посервисный аккаунтинг) или при инициации сессии по трафику (Netflow/sFlow/SNMP) для правильного переобсчета необходимо также логически разбивать сессию при переключении ее состояния (подключена/отключена). Для этого в конфигурации нужно указать **session.split.onDeviceState=1**.

В этом случае сессия, например, при переключении доступа при нехватке средств на балансе с обычного на ограниченный (доступ только к личному кабинету) будет логически разделена в БД на две - завершённую с состоянием "подключена" и новую активную с состоянием "отключена".

Пример конфигурации устройства

```
# Access-Accept ;
# , Reject-to-Accept Start-
#connection.start.fromAccept=0
# Update-, 0 - Start-, 1 - ,
# 2 - , connection.close.timeout
#connection.start.fromUpdate=1

# suspended RADIUS-
connection.suspend.timeout=900
# suspended RADIUS-
# ( connection.suspend.timeout)
#connection.disable.suspend.timeout=900
# RADIUS- , , , flow-
# ( connection.suspend.timeout)
connection.close.timeout=900
# RADIUS- , , , flow-
# ( connection.disable.suspend.timeout, connection.close.timeout)
#connection.disable.close.timeout=1800
#
connection.finish.timeout=5

#
session.split.onDeviceState=0
#
session.split.onTariffOption=1
```

Тарифные планы Inet

Тарифные планы обязательно содержат информацию о отнесении типов трафика к той или иной услуге и стоимости единиц типа трафика. Дополнительно в них могут быть указаны параметры сервиса (опции). Порядок просмотра тарифных планов соответствует [Алгоритму 2](#).

В тарифном запросе передаются следующие параметры:

- идентификатор учётного периода;
- время потребления;
- тарифные опции;
- перечень потреблённых после последней тарификации типов трафика и их объёмы.

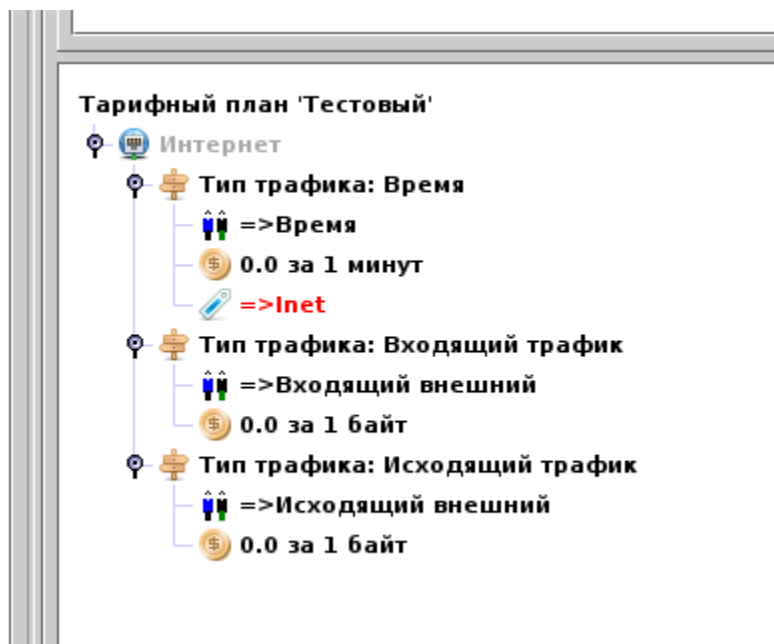
После редактирования тарифного дерева необходимо вызвать контекстное меню у главной ветки модуля в дереве (при редактировании дерева она подсвечивается рыжим цветом) и выбрать **Оповестить об изменениях**, для того чтобы все приложения обновили кэш данного тарифного дерева.

Если с момента последней тарификации изменился час, то вызываются последовательно несколько тарифных запросов, т.к. стоимость может измениться со сменой часа. В результате выполнения в тарифном запросе обязательно появляется следующая информация:

- стоимость каждого из потреблённых объёмов трафика каждого вида;
- услуга, к которой отнесена каждая из стоимостей.

Для каждой цены рекомендуется назначать отдельную услугу для облегчения бухгалтерской отчетности. Например услуга **Включенный трафик** - 0 руб./МБ и услуга **Трафик** - 0.10 руб./МБ.

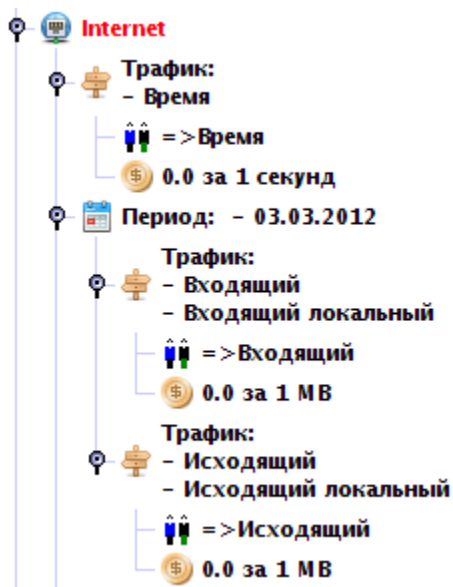
Дополнительно в запрос можно добавить набор опций модуля Inet. Пример простейшего тарифного плана приведён на скриншоте.



Здесь трём типам трафика сопоставляются нулевые стоимости и одноимённые услуги. Кроме того, производится установка опции сервиса "Inet".

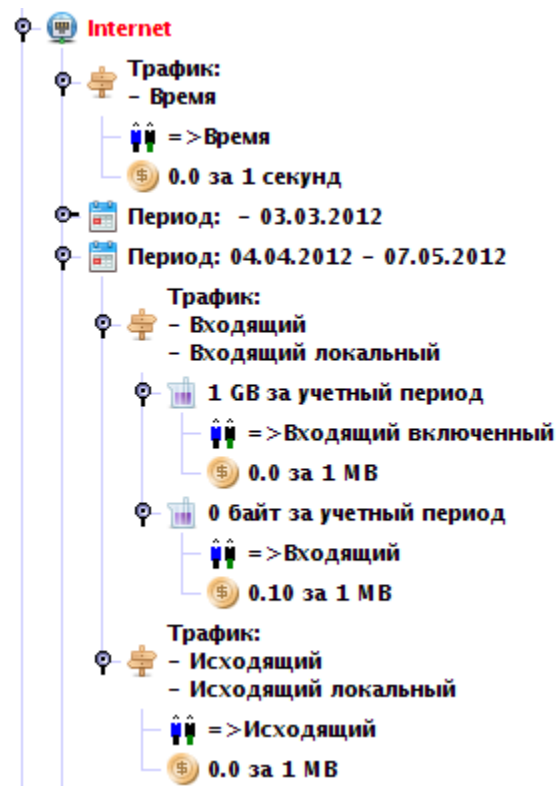
Объединение типов трафика

В ветке **Трафик** можно указать сразу несколько типов трафика:



Диапазоны трафика

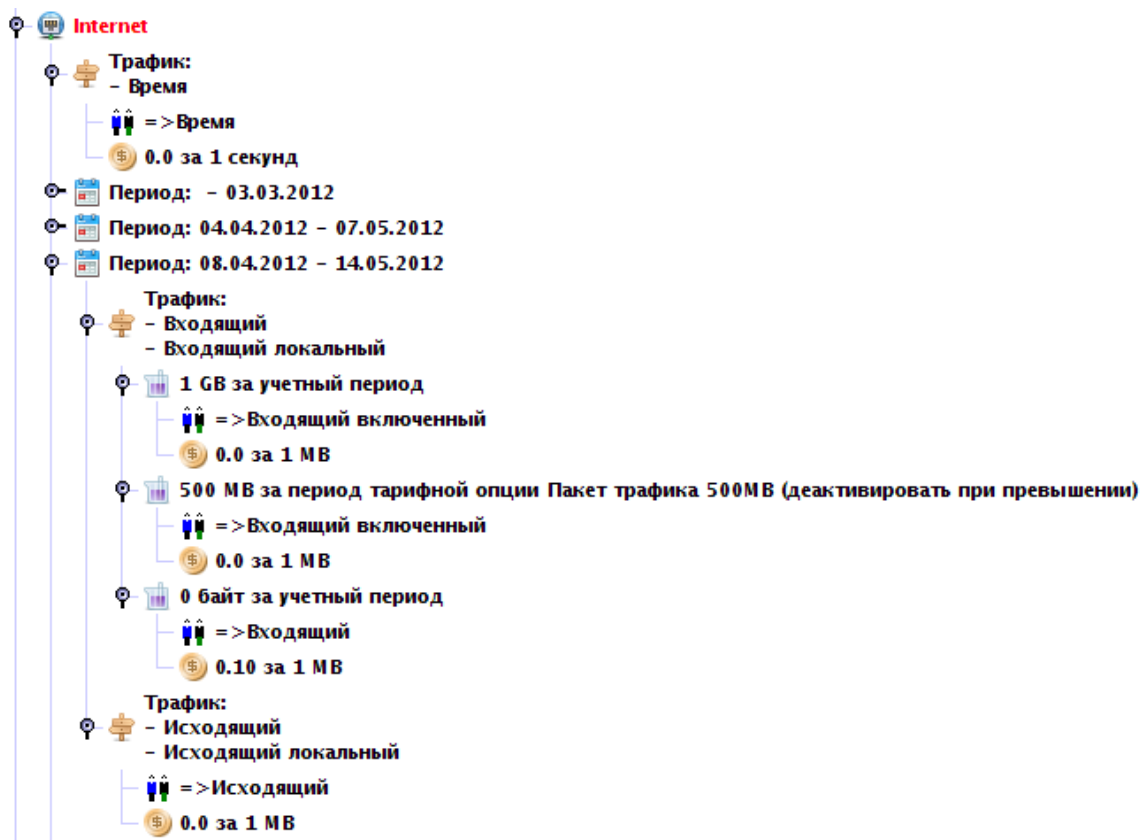
Внутри ветки **Трафик** можно указывать диапазоны, внутри диапазона можно назначить отдельную цену, услугу и/или опции. Диапазон со значением 0 работает как бесконечно большой.



Пакеты трафика

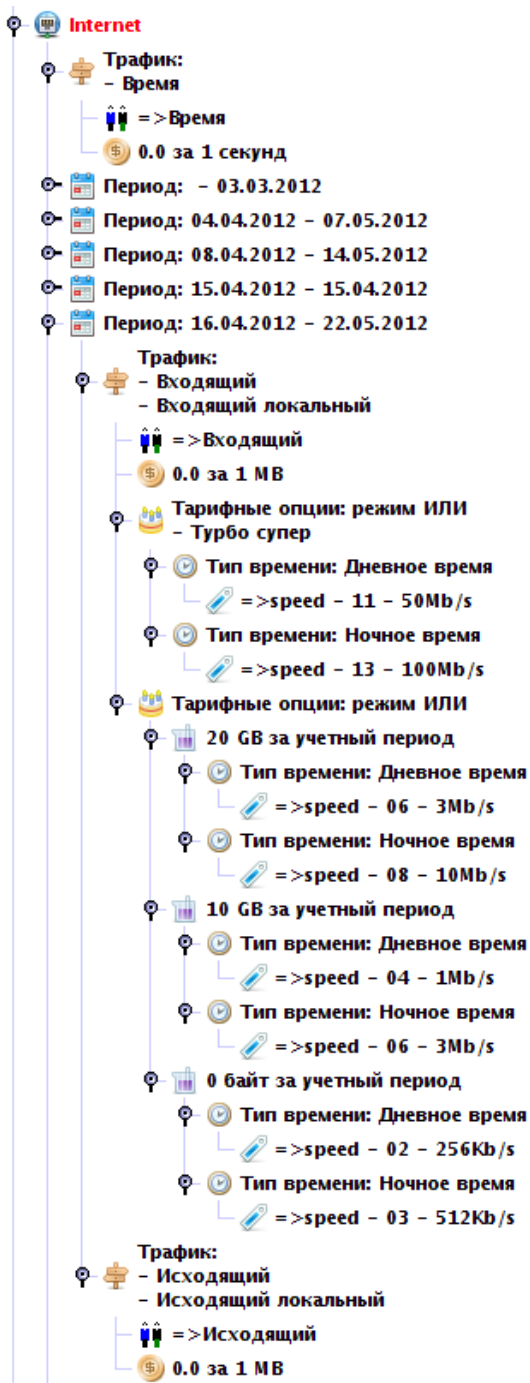
С помощью тарифных опций и диапазонов, зависящих от них, можно создать пакеты трафика. Для этого необходимо создать тарифную опцию с режимом активации, например, на 1 час ровно, добавить диапазон с режимом за период тарифной опции и выбрать тарифную опцию из списка.

Если есть необходимость, в диапазоне (в последнем диапазоне тарифной опции, если их несколько) можно указать "деактивировать при превышении" - тогда при превышении диапазона опция будет деактивирована принудительно текущим временем, даже если время деактивации уже предоставлено. Например, опция активирована на 1 час, но диапазон потрачен за полчаса - время деактивации опции будет перенесено на текущий момент, чтобы клиент мог активировать опцию заново.



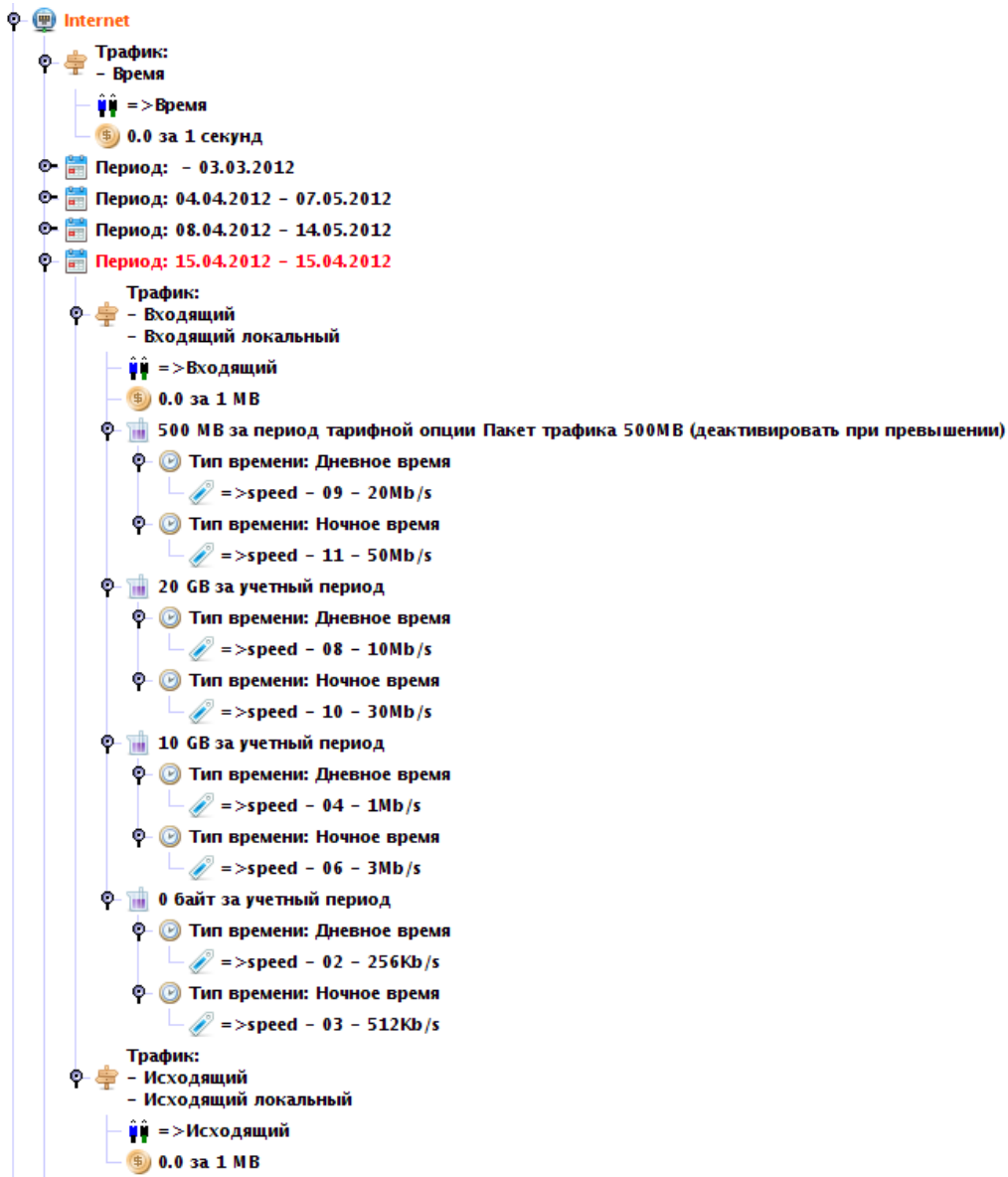
Опции модуля и "турбо-кнопка"

При помощи опций модуля Inet можно, например, регулировать скорость соединения. А при помощи тарифных опций настроить "турбо-кнопку", которая будет действовать на период активированной опции:



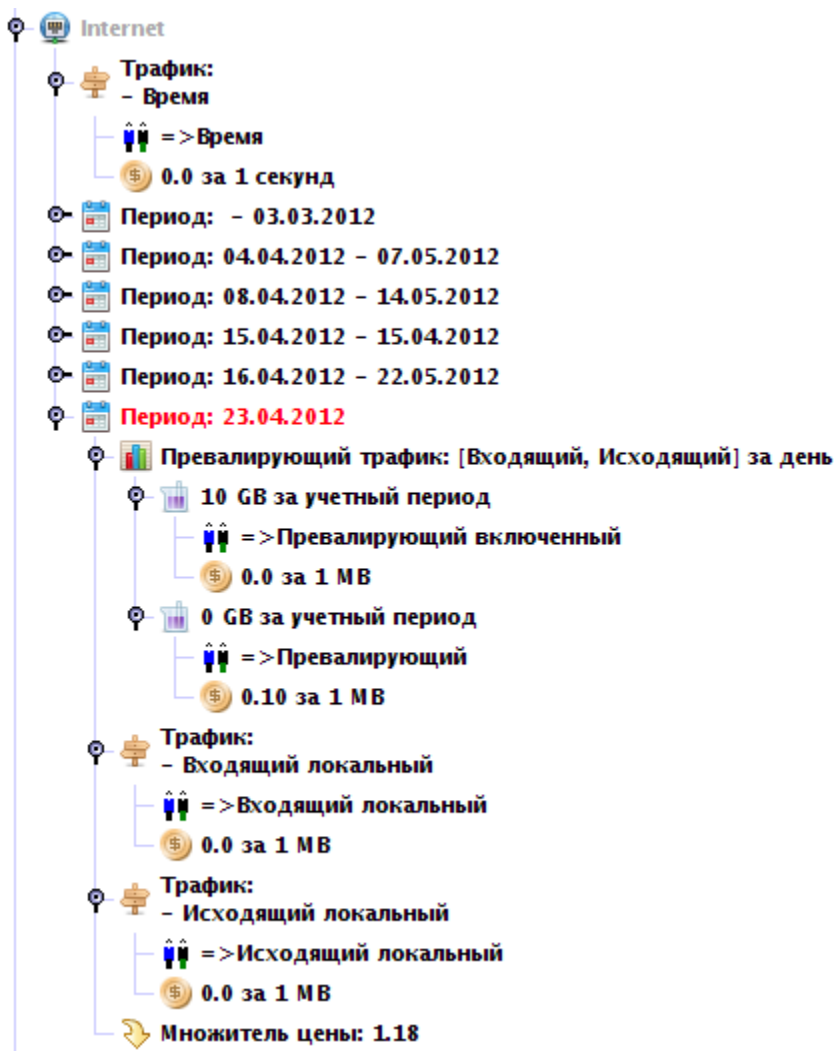
Здесь, при активности тарифной опции "Турбо супер" будут обрабатываться первая ветка **Тарифные опции**, при неактивности - вторая (т.к. она пуста и выше ни одна ветка **Тарифные опции** не работает).

Если же "турбо-кнопка" должна быть ограничена не только по времени действия тарифной опции, а также по объему трафика, необходимо использовать диапазон с привязкой к тарифной опции:



Превалирующий трафик

Для подсчета превалирующего трафика из двух типов трафика нужно использовать ветку **Превалирующий трафик**.



При отсутствии внутри ветки превалирующего трафика зависимостей цены от какого-либо временного промежутка, стоимость за период ветки будет равна стоимости максимального трафика, помноженного на цену.

В других случаях стоимость может "плавать", т.к. в один момент времени превалирующим может быть первый трафик, в другой момент - второй, а за время, пока второй трафик "догоняет" первый наработки соответственно не будет, т.е. трафик будет превалирующим.

Дополнительные ветки тарифного дерева

Ветка **Множитель цены** умножает стоимость на указанное число. При этом, если эта ветка находится внутри веток **Тип трафика** или **Превалирующий трафик**, то умножается только стоимость текущего типа трафика, т.е. ветка отработает относительно ветки **Тип трафика** или **Превалирующий трафик**, иначе (т.е. если ветка **Множитель цены** находится в корневой ветке, в самом низу) будет умножена стоимость по всем типам трафика, которым уже сопоставлена цена.

Ветка **Фильтр по реалму** производит фильтрацию по указанным реалмам, т.е. если текущий REALM (по умолчанию - default) не входит в список, то тарифный запрос не попадет внутрь. Если на том же уровне есть ветка с пустым набором REALMов, а ни в одну из веток **Фильтр по реалму**, которые находятся выше на том же уровне запрос не попал, то эта пустая ветка отработает, т.е. запрос попадет внутрь этой ветки. Для сессий с типом инициации **по трафику** REALM всегда default.

Ветка **Фильтр по группе договоров** фильтрует по указанным группам договоров, т.е. запрос попадет внутрь этой ветки, только, если группы обсчитываемого договора совпадают с фильтром. Если на том же уровне есть ветка с пустым набором групп, а ни в одну из веток **Фильтр по группе договоров** выше на этом же уровне запрос не попал, эта пустая ветка отработает.

Если запрос попал в ветку **Запрещенные устройства**, а авторизация происходит на одном из указанных устройств или на одном из потомков указанных устройств, то в авторизации будет отказано с кодом 40 "Доступ к устройству (NAS'у) закрыт".

Если запрос попал в ветку **Разрешенные устройства**, а авторизация не происходит на одном из указанных устройств или на одном из потомков указанных устройств, то в авторизации будет отказано с кодом 40 "Доступ к устройству (NAS'у) закрыт".

Если запрос попал в ветку **Отказать в авторизации**, то в авторизации будет отказано с кодом 44 "Доступ приостановлен".

Монитор соединений Inet

Для отслеживания сессий и ошибок авторизации можно использовать Монитор соединений.

ConnID	SessID	Дого...	Устройство	Идент...	Сервис	С ном...	IP	Начало	Окончан...	Активность	Стоимость	Статус	Состояние
545...	708...	ИФ...	RedBack...	0100F...	1cbdb9...	889ff...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
634...	710...	ИФ...	RedBack...	0100F...	LOGIN:...	0026...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
634...	710...	ИФ...	RedBack...	0100F...	LOGIN:...	90fba...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
633...	708...	ИФ...	RedBack...	0100F...	1cbdb9...	0011...	193...	11.05.2...		11.05.2012 ...	0.00	активна	отключено
633...	709...	ИФ...	RedBack...	0100F...	LOGIN:...	0019...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
634...	709...	ИФ...	RedBack...	0100F...	LOGIN:...	001a...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
631...	708...	ИФ...	RedBack...	0100F...	LOGIN:...	84c9...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
633...	709...	ИФ...	RedBack...	0100F...	00265a...	0000...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
633...	709...	ИФ...	RedBack...	0100F...	LOGIN:...	e046...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
633...	709...	ИФ...	RedBack...	0100F...	LOGIN:...	e81...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
610...	707...	ИФ...	RedBack...	0100F...	LOGIN:...	84...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
634...	709...	ИФ...	RedBack...	0100F...	LOGIN:...	001...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
634...	709...	ИФ...	RedBack...	0100F...	LOGIN:...	001...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
633...	709...	ИФ...	RedBack...	0100F...	001e58...	90f...	193...	11.05.2...		11.05.2012 ...	0.00	активна	отключено
610...	708...	ИФ...	RedBack...	0100F...	001e58...	b8a3...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено
633...	709...	ИФ...	RedBack...	0100F...	001e58...	d027...	193...	11.05.2...		11.05.2012 ...	0.00	активна	подключено

Тут возможен просмотр текущих и завершённых сессий, ошибок авторизации с фильтрацией по времени, устройству. По нажатию правой кнопки мыши во всплывающем меню доступны следующие операции:

Открыть договор - открыть договор, к которому принадлежит выбранная сессия.

Закрыть - Физически закрывает выбранную сессию. Происходит попытка закрыть сессию на устройстве.

Завершить - Логически завершает выбранную сессию. Сессия закрывается только в биллинге.

Сервисные сессии - показывает список сервисных сессий для выбранной сессии.

По двойному нажатию мыши на сессию отображаются логи для выбранной сессии (возможно только для RADIUS и DHCP).

Личный кабинет (web-статистика)

В личном кабинете клиент может просмотреть отчет по сессиям, отчет по трафикам или сменить пароль сервиса.

Доступ к личному кабинету по логину/паролю сервиса

Для того, чтобы клиент мог зайти в личный кабинет по логину и паролю сервиса, в конфигурации ядра необходимо изменить параметр web.auth.modes, добавив авторизацию модуля Inet. Например, если код модуля Inet 5:

```
#
web.auth.modes=0:1;5:1
```

Затем необходимо редактировать страницу авторизации webroot/xsl/login.xsl. Можно сделать так, чтобы в одной форме авторизация шла и по номеру договора и по логину Inet - для этого в hidden-поле midAuth после запятой код модуля Inet:

```
<form method="post" action="{ $CONTEXT_PATH }/{ $WEBEXECUTER }">
  <input type="hidden" name="midAuth" value="0,5"/>
  <table class="filter" >
    <tr>
      <td align="left" nowrap="1"> <br/> :</td>
      <td><input type="text" name="user" size="15" class="inputLogin" /></td>
    </tr>
    <tr>
      <td align="left" nowrap="1">:</td>
      <td><input type="password" name="pswd" size="15" class="inputPassword" /></td>
    </tr>
    <tr>
      <td></td>
      <td style="text-align: right"><xsl:call-template name="submit"><xsl:with-param
name="title" select="''"/></xsl:call-template></td>
    </tr>
    <tr>
      <td colspan="2" align="right">
        <a href="{ $CONTEXT_PATH }/{ $PUBEXECUTER }?action>PasswordForgot&
module=admin" style="font-size: x-small"> ?</a>
      </td>
    </tr>
  </table>
</form>
```

Или же просто скопировать форму и вставить ниже, после формы авторизации по номеру договора, в поле midAuth оставив только код модуля Inet:

```

<form method="post" action="{CONTEXT_PATH}/{WEBEXECUTER}">
  <input type="hidden" name="midAuth" value="5"/>
  <table class="filter" >
    <tr>
      <td align="left" nowrap="1">:</td>
      <td><input type="text" name="user" size="15" class="inputLogin" /></td>
    </tr>
    <tr>
      <td align="left" nowrap="1">:</td>
      <td><input type="password" name="pswd" size="15" class="inputPassword" /></td>
    </tr>
    <tr>
      <td></td>
      <td style="text-align: right"><xsl:call-template name="submit"><xsl:with-param
name="title" select="''"/></xsl:call-template></td>
    </tr>
    <tr>
      <td colspan="2" align="right">
        <a href="{CONTEXT_PATH}/{PUBEXECUTER}?action>PasswordForgot&
module=admin" style="font-size: x-small"> ?</a>
      </td>
    </tr>
  </table>
</form>

```

В последнем случае в окне авторизации будут две формы на выбор: в одной, как обычно, ввод номера договора и пароля к личному кабинету (статистике), в другой - ниже - ввод логина и пароля сервиса.

Отчет по сессиям

Клиент может просмотреть отчет по сессиям сервиса за месяц или определенный период месяца, выгрузить его в csv или html. Для каждой сессии можно просмотреть детализацию по наработке, если таковая в сессии есть.

Просмотр сессий

Текущие сессии **История сессий**

Сервис: LOGIN:babai | Ед. измерения: Мб.
Месяц: февраль | Год: 2012 | Числа месяца: с 24 по 29 | Вывести

Отчет в виде: CSV, HTML, CSV/ZIP, HTML/ZIP

Страница << < 1 из 1 > >>

С номера/На номер	IP	Время входа	Время выхода	Длительность	Сумма, руб.	
545454/345566	192.168.184.22	24.02.2012 16:19:39	24.02.2012 18:41:54	02:22:16 [8536]	0.00019	>>
Услуга		Входящий трафик				
Наработка		0.000				
Сумма, руб.		0.00019				
545454/345566		24.02.2012 17:39:37	24.02.2012 18:41:36	01:02:00 [3720]	0.00000	
545454/345566	192.168.184.22	24.02.2012 18:42:07	24.02.2012 18:55:31	00:13:25 [805]	0.00000	
545454/345566	192.168.184.22	24.02.2012 18:55:37	24.02.2012 18:56:18	00:00:42 [42]	0.00000	
545454/345566	192.168.184.22	24.02.2012 19:16:23	24.02.2012 21:23:02	02:06:40 [7600]	0.00191	>>
545454/345566	192.168.184.22	24.02.2012 19:20:46	24.02.2012 23:59:59	04:39:14 [16754]	0.00000	
545454/345566	192.168.184.22	25.02.2012 00:00:00	25.02.2012 23:59:59	24:00:00 [86400]	0.00000	
545454/345566	192.168.184.22	26.02.2012 00:00:00	26.02.2012 23:59:59	24:00:00 [86400]	0.00000	
545454/345566	192.168.184.22	27.02.2012 00:00:00	27.02.2012 23:59:59	24:00:00 [86400]	0.00000	
545454/345566	192.168.184.22	28.02.2012 00:00:00	28.02.2012 23:59:59	24:00:00 [86400]	0.00000	
545454/345566	192.168.184.22	29.02.2012 00:00:00	29.02.2012 03:32:32	03:32:33 [12753]	0.00000	
Итого сессий: 11				109:56:50 [395810]	0.00210	

Отчет по трафикам

В отчете по трафикам отображается наработка по трафикам по уже завершенным сессиям. По умолчанию он не отображается - чтобы его показывать, необходимо в конфигурации модуля установить параметр:

```
web.menuItem3= Inet
```

Данные можно получить за месяц, количество в день:

Просмотр трафиков

Сервис: Адрес(а): 192.168.184.35-192.168.184.254

Трафик:

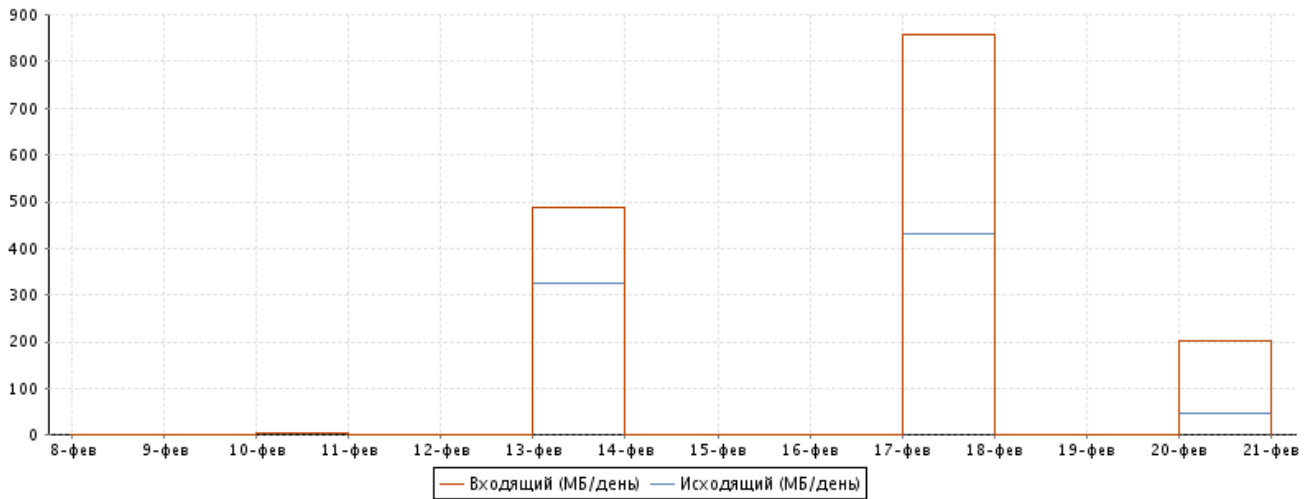
Входящий
 Исходящий

День: -- | Месяц: февраль | Год: 2012

Ед. измерения: МБ. | Вывести

Для выделения сразу нескольких трафиков держите нажатой клавишу 'Ctrl'

Обратите внимание, что в данном отчете выводится информация по трафикам только завершенных сессий.



1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	3.308	0	9.573
11	12	13	14	15	16	17	18	19	20
0	0	813.511	0	0	0	1 289.904	0	0	247.187
21	22	23	24	25	26	27	28	29	
0	0	0	0	0	0	0	0	0	

Итого: 2 363.484 МБ.

Или же за день - количество в час:

Просмотр трафиков

Сервис: Адрес(a): 192.168.184.35-192.168.184.254

Трафик:

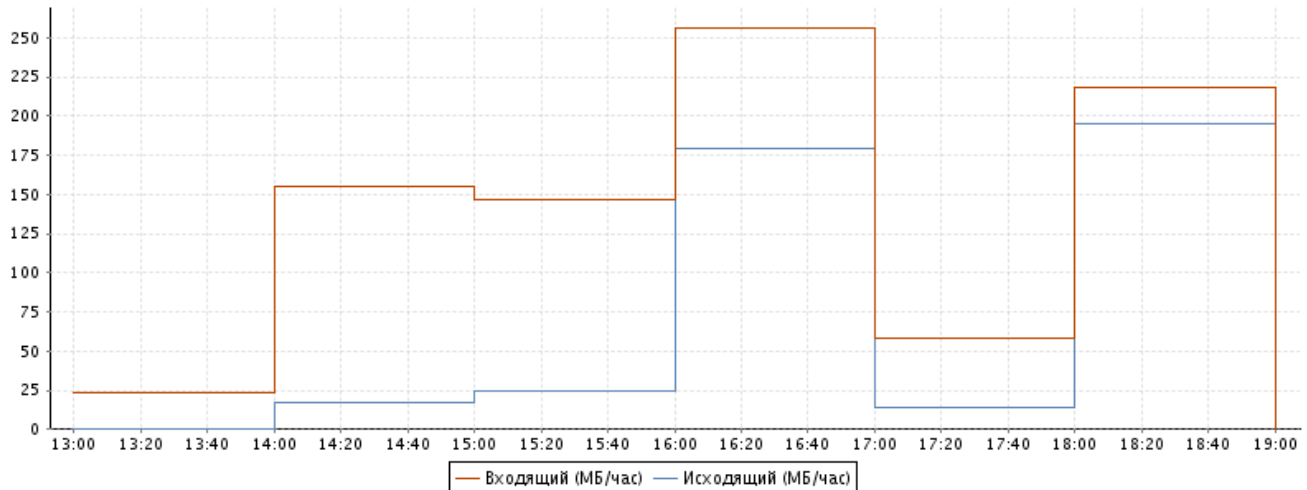
Входящий
Исходящий

День: 17 Месяц: февраль Год: 2012

Ед. измерения: МБ. Вывести

Для выделения сразу нескольких трафиков держите нажатой клавишу 'Ctrl'

Обратите внимание, что в данном отчете выводится информация по трафикам только завершенных сессий.



0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0
8	9	10	11	12	13	14	15
0	0	0	0	0	24.415	172.888	171.817
16	17	18	19	20	21	22	23
435.866	71.657	413.262	0	0	0	0	0

Итого: 1 289.904 МБ.

Смена пароля сервиса

В личном кабинете клиент может сменить пароль для сервиса, если в типе сервиса используется логин:

Смена пароля на логины

Логин:

Старый пароль:

Новый пароль:

Повторите пароль:

Учетные периоды

В личном кабинете клиент может просмотреть список своих учетных периодов.

Учетные периоды inet

Начало	Окончание
08.12.2010	31.12.2010
09.01.2011	31.01.2011
01.02.2011	28.02.2011
04.03.2011	31.03.2011
01.04.2011	30.04.2011
11.05.2011	31.05.2011
13.07.2011	28.07.2011
29.07.2011	31.07.2011
12.08.2011	31.08.2011
01.09.2011	30.09.2011

Переобработка логов

В модуле Inet возможно переобработать логи. В результате переобработки меняется количество трафика на сессиях. Эта необходимость может возникнуть в тех случаях, когда что-то было настроено неверно, например привязка трафиков и нужно это исправить задним числом. Обработать можно radius и netflow-логи. Сделать это можно на вкладке "Логи":



Тут с левой стороны находятся устройства, на которых собираются логи. А с правой - данные о логах для выбранного устройства. По оси абсцисс идут дни, по оси ординат идут часы. Каждый квадратик - это информация о логах за конкретный день и час для выбранного устройства, подкрашивается определенным цветом в зависимости от выбранных галочек. Значения галочек:

Есть данные - означает наличие логов за конкретный час;

В обработке - означает то, что данный час находится в обработке.

Пересоздание сессий - означает то, что для данного часа добавлена задача на пересоздание сессий.

Правой кнопкой мыши можно вызвать всплывающее меню, в котором доступны следующие пункты:

Добавить в обработку (текущее устройство) - добавление в обработку выбранного дня;

Удалить из обработки (текущее устройство) - удаление из обработки выбранного дня.

Пересоздание сессий(текущее устройство) - Пересоздание сессий(текущее устройство) добавление задания на пересоздание сессий для выбранного дня.

Удалить задание на пересоздание сессий(текущее устройство) - удаление задания на пересоздание сессий для выбранного дня.

Оперировать (добавлять и удалять из обработки) можно только днями. Это связано с особенностью работы модуля inet. Следует учитывать, что обработка текущего дня приостанавливает процессы обседа текущих сессий и поэтому стоит прибегать к этому только тогда, когда это действительно необходимо. При этом, если произойдет split сессий во время переобработки (split сессий происходит на границе суток и например при активации тарифной опции на договоре), то детализация сессии может потеряться и чтобы ее получить потребуется повторная переобработка. Поэтому лучше дождаться следующего дня и запустить сегодняшний день задним числом.

В текущий момент поддерживаются такие схемы.

1) Один собирающий логи accounting-сервер. Netflow и Radius могут идти с разных устройств, могут с одного. Если netflow и radius идет с одного устройства, то типы трафика должны быть различными для netflow и radius (иначе они будут перетирать друг друга). Для разных устройств можно заводить одинаковые типы трафика.

2) 2 accounting-сервера, собирающих логи (например, один собирает netflow, другой radius) . Netflow и Radius обязательно идут с разных устройств, причем коренные устройства этих accounting-серверов (значение параметра **rootDeviceId** в xml-конфигурации приложений) должны указывать у каждого на разные объединяющиеся устройства, которые не являются предками друг друга. Типы трафика для устройств netflow и radius могут быть разными, могут одинаковыми, это значения не имеет, так как в случае разных устройств они не пересекаются. В этом случае каждый accounting-сервер обрабатывает только те задачи, которые добавлены для устройств, являющихся предками его корня (значение параметра **rootDeviceId** в xml-конфигурации приложений). Тут важно, чтобы не было пересечений, иначе чужой accounting-сервер может удалить всю детализацию по устройству, если логи этого устройства находятся на другом accounting- сервере.

В общем случае еще могут быть accounting сервера, не собирающие логи. В этом случае у них в настройке должно стоять:

```
<param name="processLogs" value="false" />
```

По умолчанию там true. Аналогично так можно пометить сервера, собирающие логи, но игнорирующие задания на их переобработку. Схема с общим устройством для netflow и radius и 2-мя accounting-серверами, собирающими логи (один собирает netflow, другой radius)- не поддерживается. В этом случае результаты переобработки логов непредсказуемы (возможно, что один accounting-сервер будет перетирать работу другого). Такая схема запрещена в настройке.

Также есть такие опции :

```
<!-- radius- >  
<param name="processRadiusLogs" value="true" />  
<!-- netflow- >  
<param name="processFlowLogs" value="true" />
```

По умолчанию эти опции имеют значение true. Рассмотрим случай: accounting-сервер собирает radius-логи, считает по ним сессии и у него подключена директория с netflow-логами другого коллектора для детализации. Тогда имеет смысл отключить обработку netflow-логов. Желательно отключать все лишнее. Если, например, какие-то логи не собираются вообще, то нужно их отключить.

Сессии пересоздаются по факту наличия трафика, старые сессии при этом будут потеряны. К пересозданию сессий нужно прибегать в крайнем случае, когда произошли какие-то сбои в работе системы и сессии не создавались. В остальных случаях вам, скорее всего, хватит обработки логов или переобсчета. После пересоздания сессий нужно запустить обработку логов, чтобы на них появился трафик. При этом на новых сессиях не будет наработки по времени. Пересоздание сессий для текущего дня игнорируется.

Переобсчет

В модуле Inet возможно переобсчитать наработку, стоимость сессий и баланс. Необходимость в этом может возникнуть, если, например, тариф был настроен неверно. Это можно сделать на вкладке "Переобсчет":

The screenshot shows a web-based configuration interface for the 'Переобсчет' (Recalculation) tab. At the top, there is a navigation bar with tabs: 'Опции', 'Трафик', 'Устройства и ресурсы', 'Типы сервисов', 'Монитор', 'Логи', 'Переобсчет', 'Поиск', and 'Конфигурация модуля'. The 'Переобсчет' tab is active. Below the navigation bar, there are two main sections: 'Месяц' (Month) and 'Договоры' (Contracts). The 'Месяц' section includes a calendar-like interface with a left arrow, a text box containing 'Май 2012', a right arrow, and a checkbox labeled 'День'. The 'Договоры' section has a dropdown menu with the text 'Выберите договор(ы)...' and a close button 'X'. Below these sections is a 'Пуск' (Start) section with a text box for 'Выслать протокол на:' containing 'king@bitel.ru', a text box for 'Комментарий:', and a 'Запуск' (Start) button. At the bottom of the window, there is a title bar with a close button 'X' and the text 'Интернет'.

Тут выбирается месяц переобсчета (опционально можно выбрать день), договоры. Нужно учитывать, что при переобсчете текущего дня происходит остановка всех процессов тарификации на accounting-серверах и поэтому не следует этим злоупотреблять. Текущий час при этом игнорируется. При этом если произойдет split сессий во время переобсчета текущего дня (split сессий происходит на границе суток и, например, при активации тарифной опции на договоре), то наработка сессии может потеряться и чтобы ее получить потребуется повторный переобсчет.

Если accounting-серверов несколько, то нужно указать в конфигурации модуля inet переменную :

```
#id accounting-  
accounting.application.ids=
```

Переобсчет в этом случае распараллеливается по нескольким accounting-серверам (на уровне договоров: одна часть на одном, другая - на другом). Если переменная не указана, то переобсчет происходит на первом попавшемся accounting-сервере. Следует избегать, по возможности, переобсчета текущего дня в текущем месяце (при переобсчете всего текущего месяца текущий день тоже переобсчитывается), так как при этом останавливаются все процессы тарификации на всех accounting-серверах и запускаются после окончания переобсчета. При этом если какой-то accounting-сервер не перечислен в переменной **accounting.application.ids**, то он не будет остановлен и не получит данных об изменениях (например, после переобсчета пользователь уже вышел за границу диапазона в тарифа, но accounting-сервер ничего про это не знает).

Отчеты модуля Inet

В договоре для модуля Inet доступен один отчет по сессиям на вкладке **Отчеты**:

The screenshot displays the 'Отчеты' (Reports) tab in the Inet module. The interface is divided into several sections:

- Интернет** (Internet) section with filters for 'Ед. измерения' (Units of measurement) in bytes, KB, MB, or GB.
- Период** (Period) section set to 'Май 2012' (May 2012).
- Сервис** (Service) section with a filter for 'LOGIN:001e589e5e5e1:0a'.
- Типы трафика** (Traffic types) section with checkboxes for 'Входящий трафик' (Incoming traffic) and 'Исходящий трафик' (Outgoing traffic).
- Логи** (Logs) section with a table of session data and a context menu.
- Итого** (Total) and **Детализация** (Details) summary sections.

ConID	SessID	Устройство	Идентификат...	Сервис	С номера/н...	IP	Начало	...
611325	682508	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.88.130	01.05.2012 00:...	01
611565	682749	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.88.130	01.05.2012 07:...	01
612246	683430	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.88.234	01.05.2012 13:...	01
614352	686011	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.128	02.05.2012 14:...	01
614406	686065	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.89	02.05.2012 15:...	01
614744	686403	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.75	02.05.2012 16:...	01
614876	686535	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.79	02.05.2012 18:...	01
615308	686967	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.169	02.05.2012 23:...	01
615308	687262	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.169	03.05.2012 00:...	01
615867	687959	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.152	03.05.2012 08:...	01
616306	688398	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.223	03.05.2012 12:...	01
616576	688668	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.159	03.05.2012 16:...	01
617186	689278	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.201	03.05.2012 19:...	01
617318	689410	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.150	03.05.2012 21:...	01
617377	689469	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.120	03.05.2012 21:...	01
618039	690570	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.139	04.05.2012 08:...	01
618596	691127	RedBack: Redbac...	0100FFFF7800...	LOGIN:001e589...	0019660cc4...	193.106.89.166	04.05.2012 14:...	01

Summary statistics:

- Сессий: 38
- Время: 154:23:03 [555783]
- Сумма: 0.00000
- Входящий трафик/Исходящий трафик: 14520045600/883460713

Тут возможно отображение текущих, завершенных сессий с возможностью фильтрации по месяцу/дням, сервисам, типам трафика. Во всплывающем меню по правому клику мышки доступны функции:

Закреть - Физически закрывает выбранную сессию. Происходит попытка закрыть сессию на устройстве;

Завершить - Логически завершает выбранную сессию. Сессия закрывается только в биллинге;

Получить детализацию - выслать детализацию по сессии на e-mail;

Сервисные сессии - показывает список сервисных сессий для выбранной сессии;

Получить лог - отобразить логи для выбранной сессии (возможно только для radius и dhcp).

Коды ошибок авторизации

Таблица. Коды ошибок

Код ошибки	Описание
0	Авторизация успешна.
1	Логин не найден.
2	Неверный пароль.
3	Превышен лимит сессий.
4	Учётный период не активирован.
10	Сервис заблокирован.
11	Договор заблокирован.
12	Недостаточно средств.
20	Статус сервиса закрыт/заблокирован.
30	Карта просрочена.
31	Карта заблокирована.
34	Картой был пополнен баланс.
40	Доступ к устройству (NAS'у) закрыт.
43	Реалм запрещён.
44	Доступ приостановлен.
46	MAC-адрес запрещен.
62	Не определён тарифный план.
63	Не определена цена в тарифном плане.

Задачи планировщика модуля Inet

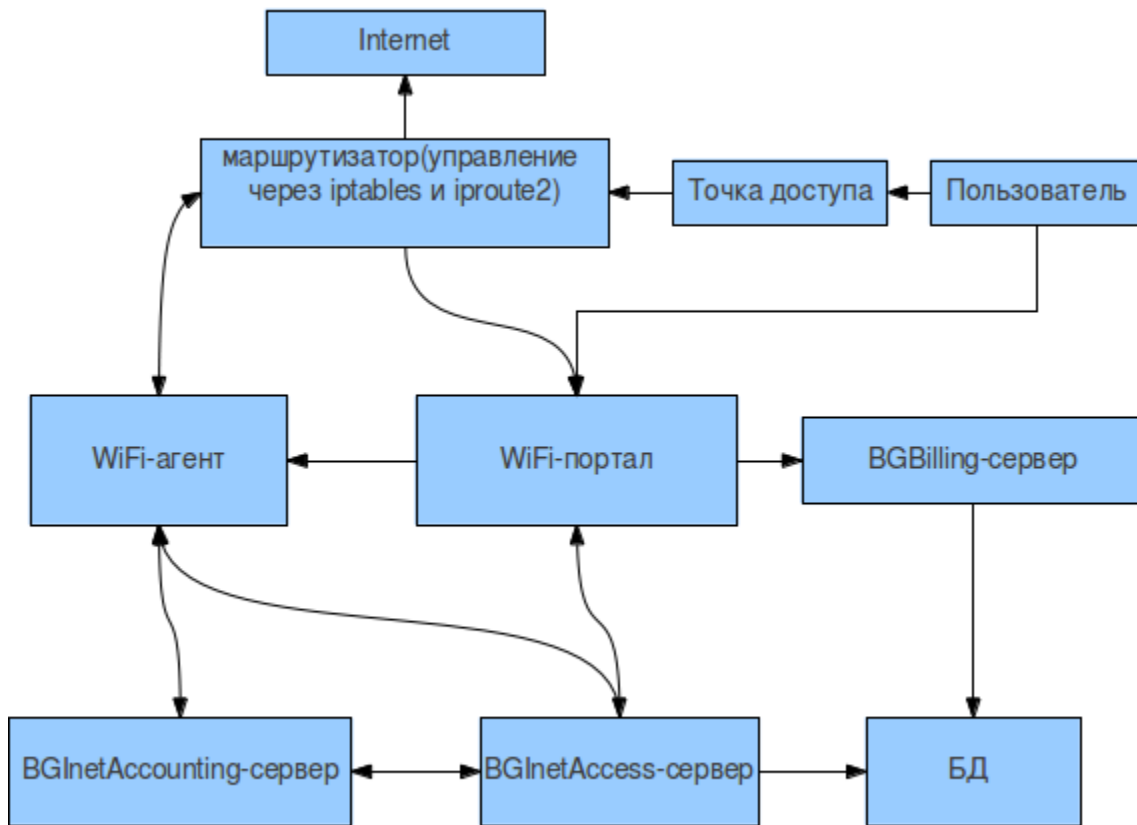
Активация/деактивация сервисов по периоду

Задача предназначена для переключения сервиса договора из состояния **удален** в состояние **включен/отключен** при наступлении периода действия сервиса и переключения из состояния **включен/отключен** в состояние **удален** при окончании периода действия сервиса. Т.е. если добавить сервис с датой начала в будущем - состояние у него после создания будет **удален**, синхронизация не будет вызвана, т.к. сервис еще не активен. При срабатывании данной задачи в день, когда период действия сервиса станет активным, задача отправит команду на синхронизацию сервиса, переключив его из состояния **удален** в состояние **включен/отключен**. Аналогично, если дата закрытия периода сервиса выставлена в будущем - при выполнении задачи в день, когда период действия будет уже не активным - она отправит команду на переключение состояния в **удален**.

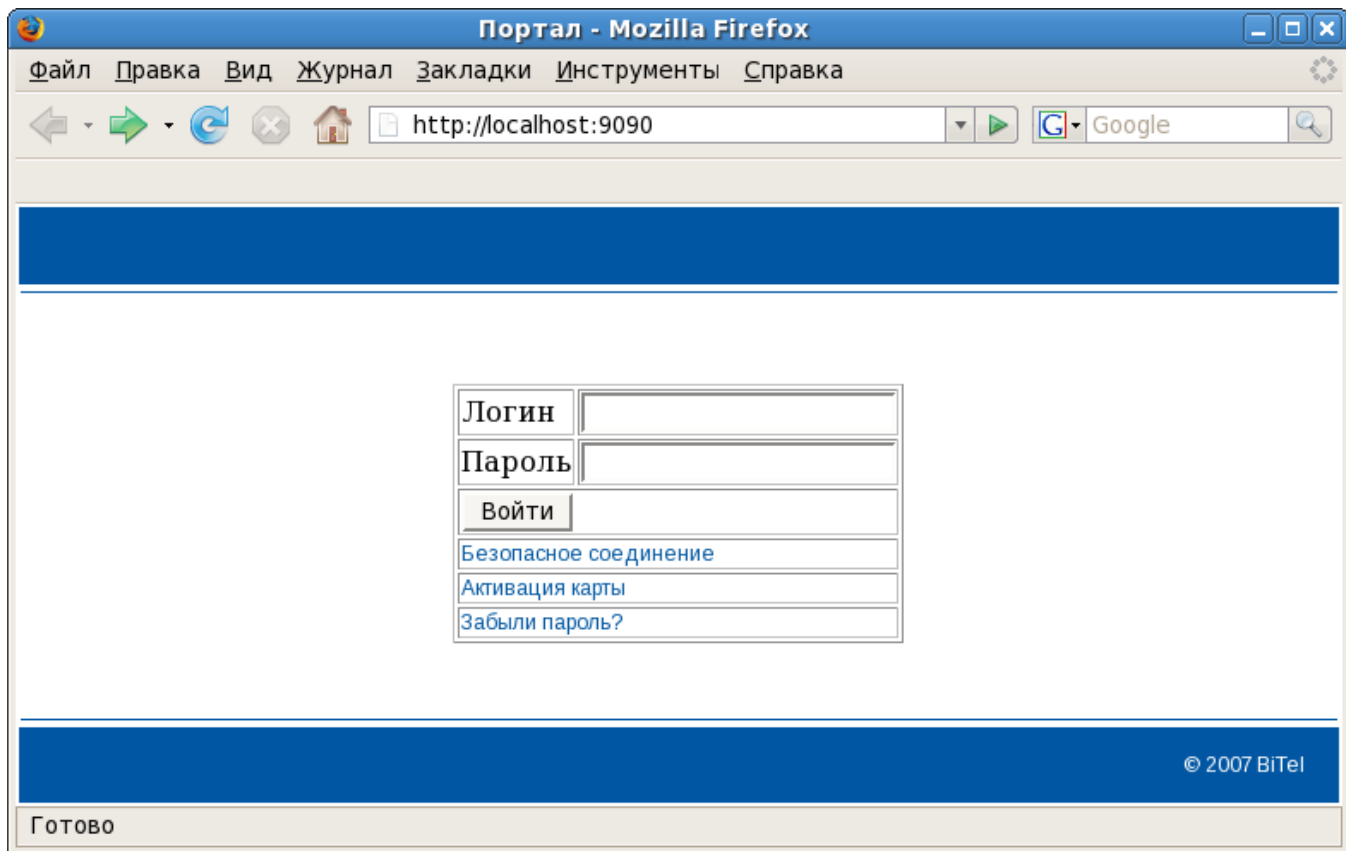
Настройка WiFi-агента для работы с модулем Inet

Описание WiFi-агента

WiFi-агент предназначен для манипулирования клиентами в WiFi-сетях. Он взаимодействует с сервером BGBilling и Access и Accounting-серверами по RADIUS-протоколу для передачи информации о появлении клиента, его уходе и актуальном времени сессии. Также он управляет маршрутизатором. Общая схема взаимодействия может быть представлена на следующей диаграмме :

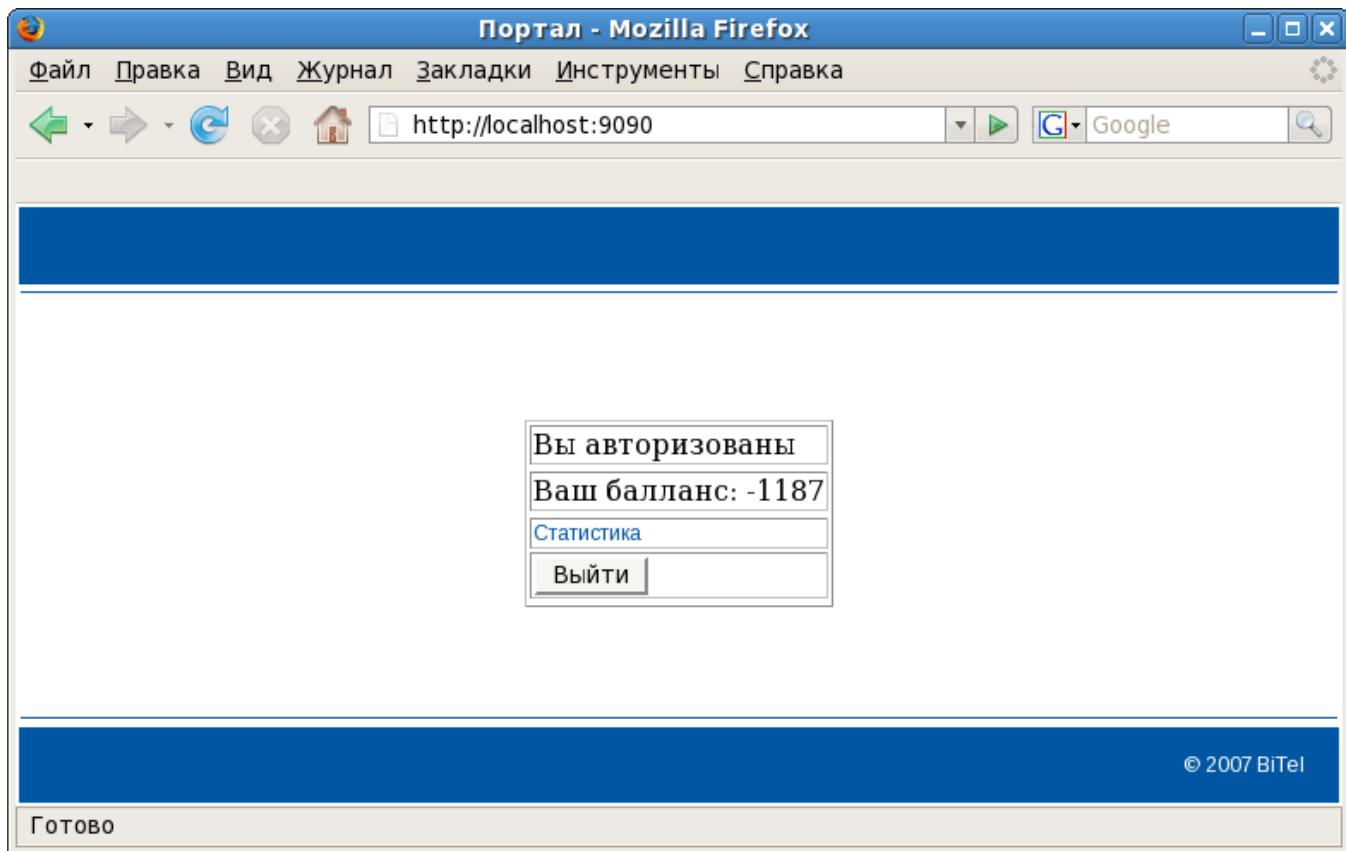


В данный момент WiFi-агент и WiFi-портал реализованы одним приложением и поднимаются на машине маршрутизатора. Это решение работает только для ОС семейства Linux (под FreeBSD и т.п. возможен запуск, если адаптировать скрипты). Пользователь выходит в WiFi-сеть, по DHCP получает IP-адрес с DHCP-сервера, установленного в этой сети, и через точку доступа попадает на маршрутизатор, на котором стоит ОС Linux со службой iptables. Iptables настраивается таким образом, что по умолчанию для клиента закрыты все порты кроме порта, на котором висит WiFi-портал. Все запросы на 80-ый порт перебрасываются на страницу портала, на которой пользователь вводит логин и пароль для доступа к WiFi-сетям. Страница авторизации имеет такой вид:



Пользователь вводит логин/пароль.

Далее Портал обращается к Access-серверу и шлёт на него авторизационный RADIUS-пакет. Если авторизация проходит успешно (в ответ получен RADIUS-пакет подтверждения авторизации), то портал шлёт об этом запрос WiFi-агенту. WiFi-агент, в свою очередь, шлёт стартовый RADIUS-пакет на Access-сервер и меняет правила iptables, разрешая пользователю выйти в интернет. После успешной авторизации пользователя перекидывают на сервисную страницу :



На эту же страницу пользователь может попасть всегда (даже после авторизации), зайдя на <http://192.168.184.39:9090> (путь зависит от настроек). Здесь он видит свой баланс, ссылку на первоначальный ресурс, который он набирал до того, как его перекинули на страницу авторизации. Также есть альтернативная ссылка на эту страницу через [https](https://).

Сессия клиента может быть завершена следующими способами :

1. Клиент зашёл на страницу портала и нажал кнопку "выйти";
2. Клиент исчерпал свой баланс и Access-сервер послал сообщение WiFi-агенту о завершении работы клиента с данным ip;
3. WiFi-агент, периодически проверяющий (в текущий момент через каждый 60 сек) состояние счётчиков iptables, определит, что клиент был неактивен в течении некоторого времени.

В случае завершения работы (не важно каким из выше перечисленных способов) WiFi-агент пошлёт Stop-пакет Accounting-серверу и удалит разрешающие правила iptables для нужного IP-адреса.

С точки зрения модуля Inet WiFi-агент выступает как NAS-устройство в дереве устройств. В текущий момент он поддерживает следующие возможности :

1. Работа с Accounting-сервером по RADIUS-протоколу (Start, Stop, Update).
2. Тарификация с помощью любых тарифов (по времени, по трафику netflow, по RADIUS-пакетам), которые можно завести в системе BGBilling и которые поддерживаются модулем Inet. Для учёта трафика на маршрутизаторе нужно поднять [NetFlow-агент](#) . Также клиент имеет возможность выбора [REALM](#)-ов сразу на странице авторизации.
4. Доступ к статистике клиента и все возможности, которые предоставляет ядро BGBilling и модуль Inet для Web-интерфейса (просмотр баланса, мониторинг сессий и т.п).
4. Возможно в зависимости от RADIUS-атрибутов, переданных Access-сервером;
5. Вызов внешних скриптов при входе и выходе пользователя в сеть, что позволяет сжимать каналы в зависимости от тарифа или сбрасывать установленные соединения или выполнять любые другие действия;
6. Вызов внешних скриптов для получения исходящего/входящего трафика клиента (по умолчанию используются счетчики iptables);
7. [Защита WiFi-сети от ARP-спуффинга](#);

8. Функция восстановления пароля через почтовый ящик клиента;

9. Возможность взаимодействия с модулем Карточки, т.е. возможность активации карты и последующей авторизацией по этой карточке в сети WiFi;

10. Возможность клиента работать с https-версией портала.

Установка, настройка и запуск

В текущий момент WiFi-агент и WiFi-портал реализованы одним приложением, которое и называется WiFi-агент. Для его установки его нужно скачать с сайта и распаковать в какую-нибудь папку. Например в папку /user/local. Далее для простоты будем считать, что мы установили в эту папку.

При установке рекомендуется идти от простого к сложному. Т.е. вначале заставить работать более простую конфигурацию, а потом уже пытаться добавлять дополнительные возможности.

Далее идут шаги, необходимые, для установки.

- 1) Для работы WiFi-агента нужна [Java-машина](#);
- 2) Все настройки агента хранятся в файле inet_wifi_agent.properties. Вот пример этого файла :

Конфигурация

```
#wifi agent class
wifi.agent.class=ru.bitel.bgbilling.modules.inet.wifi.InetWiFiAgent

#mq options
mq.url=failover:(nio://127.0.0.1:61616?socketBufferSize=1000000)
mq.user=bill
mq.pswd=bgbilling

#radius options
radius.auth.host=127.0.0.1
radius.auth.port=1812

radius.account.host=127.0.0.1
radius.account.port=1813

radius.nasId=den_nas
radius.secret=hello

# Update- Accounting- (1 )
radius.update.send=1

#billing server options
billing.server.login=admin
billing.server.passwd=admin
billing.server.http.url=http://localhost:8080/bgbilling
#billing.server.https.url=https://localhost:8443/bgbilling
billing.server.moduleId=XXX

billing.server.show.statistics=0
billing.server.password.remind=0

#portal options
portal.http.port=9090
#portal.https.port=9091
#portal.https.keystore.password=bgbilling
#portal.card.link=http://127.0.0.1:8080/bgbilling/pubexecuter?
action=CreateContract&module=card&mid=5&activateType=1

#tariff options
#portal.use.realm=1
#portal.tarif.1.realm=wifi_128_128
#portal.tarif.1.title=128 kb/s
#portal.tarif.2.realm=wifi_256_128
#portal.tarif.2.title=256 kb/s

portal.http.url=http://localhost:9090
#portal.https.url=https://localhost:9091

#wifi agent options
wifi.agent.port=5555
wifi.agent.port.admin=5556
```



```

wifi.agent.radius.live.time=60000
wifi.agent.client.live.time=24000000

wifi.agent.arp.command=/sbin/arp
#wifi.agent.server.https=1

#radius attributes
#wifi.agent.radius.attribute.1.vendor.code=1111
#wifi.agent.radius.attribute.1.attr.code=1
#wifi.agent.radius.attribute.1.type=integer
#wifi.agent.radius.attribute.2.vendor.code=1111
#wifi.agent.radius.attribute.2.attr.code=2
#wifi.agent.radius.attribute.2.type=integer

#dhcp options
#dhcp=1
#dhcp.server.host=192.168.184.254
#dhcp.server.port=67
#dhcp.agent.host=192.168.184.39
#dhcp.minThreadCount=10
#dhcp.maxThreadCount=10

#
#portal.allow.linking=1
# ssi ( shtml )
#portal.use.ssi=1

```

Установите переменную `billing.server.moduleId=<числовой код экземпляра модуля Inet>`. При необходимости скорректируйте параметры доступа к базе данных и к [MQ-серверу](#).

А теперь остановимся более подробно на каждой настройке .

radius.auth.host, radius.auth.port - хост и порт, на котором поднят RADIUS-слушатель на Access-сервере;

radius.auth.port, radius.account.port - хост и порт, на котором поднят RADIUS-слушатель на Accounting-сервере;

radius.nasId - идентификатор NAS'а (устройства, которое предствляет WiFi-агент в модуле inet);

radius.secret - секретный ключ для NAS'а (как настроить NAS для WiFi-агента будет описано ниже);

radius.update.send - Слать Update-пакеты на Accounting-сервер. По умолчанию включено;

billing.server.login, billing.server.passwd - логин и пароль доступа к серверу (те же самые, что используются в клиенте биллинга). Нужны для получения баланса пользователя. Данный пользователь должен быть заведён в системе и обладать правами на действия : "Основной модуль->Договор->Просмотр договора" и "Основной модуль->Договор->Баланс-Просмотр Баланса". О том как администрировать пользователей читайте [здесь](#);

billing.server.http.url, billing.server.https.url - URL сервера биллинга (тот же самый, по которому обращается клиент биллинга). Соответственно для http и https. Если https-соединение не нужно, то не указывайте **billing.server.https.url**.

billing.server.moduleId - код модуля Inet на BGBilling-сервере, с которым будет работать этот агент;

billing.server.show.statistics - показывать ли ссылку на статистику сервера (1- показывать; 0 - не показывать, стоит по умолчанию). При этом если в текущий момент клиент работает с порталом по http, то для него это будет ссылка на http-версию статистики, а если он работает по https с порталом, то это будет ссылка на https-версию статистики;

portal.http.port, portal.https.port - порты портала для обычного и безопасного соединения. Если https-соединение не нужно, то не указывайте `portal.https.port`;

portal.https.keystore.password - пароль для https-соединения. Для работы https-соединения нужен файл .keystore (основанный на этом пароле), который необходимо положить в папку агента. Как получить этот файл описано [здесь](#). Если https-соединение не нужно, то не указывайте этот параметр;

portal.http.url, portal.https.url - URL портала для http и https соединений. Это URL, по которому будут обращаться клиенты WiFi-сети, чтобы попасть на сервисную страницу. Если https-соединение не используется, то параметр **portal.https.url** не указывается;

wifi.agent.port - это порт, на котором будет подниматься WiFi-агент и Access-сервер будет обращаться к этому порту для сбрасывания клиента.

wifi.agent.port.admin - порт для управления WiFi-агентом;

wifi.agent.client.live.time - время жизни (в миллисекундах) клиента . Т.е. это время неактивности клиента, через которое WiFi-агент считает, что клиента больше нет, сбрасывает сессию клиента (отсылает Stop-пакет Accounting-серверу, очищает iptables, вызывает внешние скрипты). По умолчанию стоит 40 минут. Активность клиента проверяется по изменению счётчиков iptables в цепочке WIFI (о том как настроить iptables читайте ниже);

wifi.agent.arp.command - путь к команде arp в ОС Linux. Она нужна для получения mac-адреса клиента и манипулирования arp-таблицами (в случае настройки [защиты от ARP-спуффинга](#)). По умолчанию обычно /sbin/arp;

wifi.agent.server.https - использовать или нет при взаимодействии между WiFi-агентом и сервером протокол https (1 - https; 0-http, стоит по умолчанию).

3) Для работы внешних скриптов нужно настроить файл conf.sh (часть этих настроек дублируется в inet_wifi_agent.properties). Пусть у нас имеется Linux-маршрутизатор с двумя сетевыми интерфейсами: eth0 - локальный (сеть 172.16.1.0/24, через него выходя клиенты WiFi), eth1 - внешний интерфейс для выхода в интернет (имеет внешний ip - 81.30.199.220).

```
# Java- ,
JAVA_HOME=/opt/java/jre
#
PORTAL_HTTP_PORT=9090
PORTAL_HTTPS_PORT=9091
#   WiFi- Access- .
WIFI_AGENT_PORT=5555
# iptables, .
WIFI_CHAIN_NAME=WIFI
#,   WiFi-
WIFI_INTERFACE=eth0
#
EXTERNAL_INTERFACE=eth1
# IP
EXTERNAL_IP=81.30.199.220
# WIFI
WIFI_NET=172.16.1.0/24
```

4) Необходимо сделать запускаемыми все скрипты в папке агента . Для этого надо перейти в эту папку и выполнить команду :

```
chmod 755 *.sh *.pl
```

5) Настроить скрипты входа/выхода абонента.

login.sh - сюда добавляются команды открытия доступа для авторизовавшегося клиента . В это скрипт в качестве параметра передаётся ip пользователя и RADIUS-атрибуты, полученные в Ассерт-пакете (настройка атрибутов описана ниже).

Скрипт имеет такой вид в стандартной поставке

```
#!/bin/sh
cd ${0%${0##*/}}.
. ./conf.sh

IP=$1
DOWNSTREAM_SPEED=$2
UPSTREAM_SPEED=$3

date >> ./log/manad.out
echo `/sbin/iptables -I $WIFI_CHAIN_NAME 1 -t nat -j ACCEPT -s $IP` >> ./log/manad.out 2>&1
echo `/sbin/iptables -I $WIFI_CHAIN_NAME 1 -t nat -j ACCEPT -d $IP` >> ./log/manad.out 2>&1

if [ $USE_MANAD -eq 1 ]; then
./tell_manad.pl "add $IP $DOWNSTREAM_SPEED $UPSTREAM_SPEED" $MANAD_PORT
fi

#use it for shaping
#./shape.sh $IP $PARAM1 $PARAM2
```

logout.sh - сюда добавляются команды закрытия доступа для клиента . В этот скрипт в качестве параметра передаётся ip пользователя. Скрипт имеет такой вид в стандартной поставке:

```
#!/bin/sh
cd ${0%${0##*/}}.
. ./conf.sh

IP=$1

date >> ./log/manad.out

echo `/sbin/iptables -D $WIFI_CHAIN_NAME -t nat -j ACCEPT -s $IP` >> ./log/manad.out 2>&1
echo `/sbin/iptables -D $WIFI_CHAIN_NAME -t nat -j ACCEPT -d $IP` >> ./log/manad.out 2>&1

if [ $USE_MANAD -eq 1 ]; then
./tell_manad.pl "remove $IP" $MANAD_PORT
fi
```

6) Настроить скрипт проверки активности клиента - ip_counts.pl. Скрипт поставляется в стандартной поставке и рассчитан на парсинг цепочки с разрешающими правилами . На выходе скрипт выдает данные вот в таком формате

```
192.168.185.10 13423 6878
192.168.185.20 133423 6878
```

Тут для каждого ip, найденного в цепочке, выводится информация о входящих (первый столбец) и исходящих байтах (второй столбец) на этот адрес . Столбцы разделены символом табуляции. Стандартный скрипт рассчитан на работу со стандартным файлом login.sh, считывает счетчики iptables из цепочки WIFI. Скрипт можно менять, главное, чтобы выходной формат оставался такой же, как описан выше.

7) Необходимо настроить iptables. По умолчанию WiFi-агент при старте системы инициализирует правила в системе с помощью скрипта iptables.sh. Рекомендуется все ваши настройки iptables тоже помещать в это скрипт. Этот скрипт также может использовать администратор для очистки правил и сбрасывания всех текущих клиентов. Вот пример этого скрипта:

Конфигурация

```
#!/bin/sh
cd ${0%${0##*/}}.
. ./conf.sh

/sbin/iptables -F -t nat
/sbin/iptables -F -t filter

/sbin/iptables -P PREROUTING DROP -t nat

#external interface #####
#ssh
/sbin/iptables -A PREROUTING -s ! $WIFI_NET -t nat -p tcp --dport 22 -d $EXTERNAL_IP -j ACCEPT
#drop others from external interface
/sbin/iptables -A PREROUTING -s ! $WIFI_NET -t nat -j DROP
#end of external interface #####

#internal interface
#####

#before wifi chain we must add redirects for authorized users
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 80 -d $EXTERNAL_IP -j REDIRECT --to-ports $PORTAL_HTTP_PORT
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 443 -d $EXTERNAL_IP -j REDIRECT --to-ports
$PORTAL_HTTPS_PORT

#chain for WiFi (accept rules for authorized users)

/sbin/iptables --delete-chain $WIFI_CHAIN_NAME -t nat
/sbin/iptables -N $WIFI_CHAIN_NAME -t nat
/sbin/iptables -A PREROUTING -j $WIFI_CHAIN_NAME -t nat

#below is rules for internal not authorized users :

#dns
/sbin/iptables -A PREROUTING -t nat -p udp --dport 53 -j ACCEPT

#http
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 80 -j REDIRECT --to-ports $PORTAL_HTTP_PORT
/sbin/iptables -A PREROUTING -t nat -p tcp --dport $PORTAL_HTTP_PORT -j ACCEPT

#https
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 443 -j REDIRECT --to-ports $PORTAL_HTTPS_PORT
/sbin/iptables -A PREROUTING -t nat -p tcp --dport $PORTAL_HTTPS_PORT -j ACCEPT

#statistics
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 8080 -d $EXTERNAL_IP -j ACCEPT

# NAT
iptables -A POSTROUTING -o $EXTERNAL_INTERFACE -s $WIFI_NET -j SNAT -t nat --to-source $EXTERNAL_IP

#RST packets for dropping established connections
/sbin/iptables -A FORWARD -p tcp -j REJECT --reject-with tcp-reset
```

Этот скрипт нужно поправить под ваш случай. В процессе работы правила будут иметь вид :

Конфигурация

```
# iptables -L -t nat -n
Chain PREROUTING (policy DROP)
target      prot opt source                destination
ACCEPT     tcp  -- !172.16.1.0/24        81.30.199.220      tcp dpt:22
DROP       all  -- !172.16.1.0/24        0.0.0.0/0
REDIRECT   tcp  -- 0.0.0.0/0            81.30.199.220      tcp dpt:80 redir ports 9090
REDIRECT   tcp  -- 0.0.0.0/0            81.30.199.220      tcp dpt:443 redir ports 9091
WIFI       all  -- 0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  -- 0.0.0.0/0            0.0.0.0/0          udp dpt:53
ACCEPT     tcp  -- 0.0.0.0/0            0.0.0.0/0          tcp dpt:9090
ACCEPT     tcp  -- 0.0.0.0/0            0.0.0.0/0          tcp dpt:9091
ACCEPT     tcp  -- 0.0.0.0/0            81.30.199.220      tcp dpt:8080

Chain POSTROUTING (policy ACCEPT)
target      prot opt source                destination
SNAT       all  -- 172.16.1.0/24        0.0.0.0/0          to:81.30.199.220

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination

Chain WIFI (1 references)
target      prot opt source                destination
ACCEPT     all  -- 172.16.1.105         0.0.0.0/0
ACCEPT     all  -- 172.16.1.57          0.0.0.0/0
ACCEPT     all  -- 172.16.1.94          0.0.0.0/0

# iptables -L -t filter -n
Chain INPUT (policy ACCEPT)
target      prot opt source                destination

Chain FORWARD (policy ACCEPT)
target      prot opt source                destination
REJECT     tcp  -- 0.0.0.0/0            0.0.0.0/0          reject-with tcp-reset

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination

Chain RH-Firewall-1-INPUT (0 references)
target      prot opt source                destination
```

Очень не рекомендуется заносить правила в цепочку WIFI, т.к. она редактируется автоматически и могут возникнуть проблемы. Свои дополнительные правила вы можете заносить в любые другие цепочки и таблицы (учитывая логику работы iptables и WiFi-агента).

Например, если Access-сервер находится на другой машине, то ему надо разрешить 5555-ый (в данном случае) порт для обращения к WiFi-агенту. Аналогично можно разрешить порты для ssh и т. п., если это необходимо. На шлюзовой машине, где ставиться агент, скорее всего, будет, как минимум, один внешний интерфейс и один внутренний интерфейс, через который будут работать клиенты WiFi. В этом случае, например, если по ssh будут обращаться только через внешний интерфейс, то можно повесить разрешающие правила на внешний интерфейс. Вариантов много, но главное, чтобы выход во внешнюю сеть был закрыт для клиентов локальной сети по умолчанию. Одним из вариантов организации сети может быть набор виртуальных (vlan) интерфейсов, которые будут заведены на данной шлюзовой машине (на интерфейсе eth0) и агент будет добавлять правила для всех клиентов этих виртуальных сетей. В этом случае агент может управлять сразу многими локальными сетями, которые могут быть физически разнесены далеко друг от друга.

Отметим, что для правильной работы сети кроме правила NAT, добавленного выше, в случае ОС Linux необходимо ещё включить ipforwarding. Для дистрибутива Fedora необходимо поставить `net.ipv4.ip_forward = 1` в `/etc/sysctl.conf` и выполнить команду `echo 1 >> /proc/sys/net/ipv4/ip_forward` (чтобы изменения немедленно применились). Рекомендуется вначале проверить работу точки доступа и выход в интернет через неё, прежде чем применять запрещающие правила iptables, описанные выше.

8) Установить скрипт `service/bgwifiaagent_inet` как службу. Для этого нужно скопировать файл `bgwifiaagent_inet` в `/etc/rc.d/init.d` и потом вызвать следующие команды

```
chmod 755 /etc/rc.d/init.d/bgwifiagent_inet
chkconfig --add bgwifiagent_inet
chkconfig --level {lev} bgwifiagent_inet on
```

Где {lev}- это уровень запуска в вашей системе. Узнать его можно так:

```
[root@king ~]# runlevel
N 5
```

Т.к. агент изменяет правила iptables, то в системе он должен запускаться с правами root'a.

9) В файле setenv.sh нужно прописать JAVA_HOME.

10) Перед запуском агента нужно запустить скрипт

```
./update.sh
```

в папке агента. Это обновит все библиотеки на нем (скачает с сервера).

11) Для запуска агента можно выполнить команду:

```
service bgwifiagent_inet start
```

Агент не запуститься если не будет связи с сервером BGBilling, или там не будут установлены лицензии на модуль Inet или сам портал .

12)Подсоединиться клиентом к серверу биллинга, настроить [устройство](#) NAS для работы с нашим агентом. В конфигурацию NAS нужно добавить:

Идентификатор (такой какой мы указали в **radius.nasId**), секретное слово (такое же как и в **radius.secret**), Хост/порт в формате (host:port). host - хост, на котором работает wifi-агент, port - то же самый что и в **wifi.agent.port** (5555). На этот хост и порт Access сервер будет и слать сигнал о завершении работы в случае ухода баланса клиента в минис, ручном сбросе сессии и т.п. В типе сервиса нужно установить обработчик активации сервисов **ru.bitel.bgbilling.modules.inet.dyn.device.wifi.WiFiServiceActivator**(входит в стандартную поставку).

13) Установить [Access и Accounting сервера для модуля Inet](#). Настройки серверов должны совпадать с параметрами **radius.auth.host**, **radius.auth.port**, **radius.account.host**, **radius.account.port** в файле inet_wifi_agent.properties.

Связь WiFi-агента с модулем "Карточки"

Если у вас установлен модуль "Карточки", то портал может отображать ссылку на активацию карты. Для это в конфигурационном файле `inet_wifi_agent.properties` вы должны прописать строку

```
portal.card.link=http://127.0.0.1:8080/bgbilling/pubexecuter?action=CreateContract&module=card&mid=5
```

Формат этой ссылки описан в документации по модулю [Карточки](#).

Защита WiFi-сети от ARP-спуффинга

ARP-spoofing (ARP-poisoning) — техника сетевой атаки, применяемая преимущественно в Ethernet, но возможная и в других, использующих протокол ARP-сетях, основанная на использовании недостатков протокола ARP и позволяющая перехватывать трафик между узлами, которые расположены в пределах одного широковещательного домена. Суть её состоит в том, что любой желающий может посылать ARP-запросы, подменять таким образом ARP-таблицы на других компьютерах и сопоставлять свой mac-адрес с чужим ip-адресом

Одним из методов защиты является использование статических ARP-таблиц, т.е. запрет на изменение ARP-таблицы. Именно это решение мы предлагаем для использования в нашем WiFi-агенте. Для этого внутри WiFi-агента реализуется ещё одно приложение: DHCP relay-агент. По описанию протокола DHCP (RFC 2131) сервер DHCP может отвечать не только на запросы клиентов на получение ip-адреса, но и на запрос relay-агентов, которые пробрасывают запросы клиентов на сервер (проставляя свой адрес, чтобы сервер мог им ответить), а ответ отправляют клиенту. Таким образом обычно реализуется возможность получения ip-адресов, если DHCP сервер находится в другой сети и часто роль relay-агентов выполняют аппаратные шлюзы. Мы используем программный Relay агент для того, чтобы при получении IP-адреса от DHCP-сервера редактировать arp-таблицу на шлюзе (где установлен WiFi-агент) и для удаления записи из arp-таблицы по истечению срока аренды IP-адреса (который задаётся DHCP-сервером и проставляется клиенту, а клиент, в свою очередь, обязан до истечения срока аренды послать запрос на продление IP-адреса). Динамическое обновление arp-таблицы отключается и никто другой не может править arp-таблицу кроме DHCP relay агента.

Для запуска DHCP-агента нужно в файл `inet_wifi_agent.properties` добавить следующие настройки:

```
#dhcp options
dhcp=1

dhcp.servers.1.host=192.168.184.254
dhcp.servers.1.port=67

dhcp.servers.2.host=192.168.184.253
dhcp.servers.2.port=67

dhcp.agent.host=192.168.154.39
dhcp.minThreadCount=10
dhcp.maxThreadCount=10

dhcp.servers.1.host=192.168.184.254
dhcp.servers.1.port=67
```

Здесь

dhcp=1 - это флаг, говорящий о том, что нужно запускать DHCP-агент.

Далее идут настройки DHCP-серверов. Поддерживается несколько серверов (<id> - 1,2,3 и т.п.):

dhcp.server.<id>.host - IP-адрес сервера DHCP. Он нужен Relay-агенту;

dhcp.server.<id>.port - порт, на котором запускается сервер DHCP. Вообще, стандартным портом для сервера DHCP является 67-ой. Но в данном случае, возможно, вам понадобится изменить этот порт. Например, если DHCP-агент и DHCP-сервер находится в одной локальной сети. В этом случае они оба будут получать широковещательные DHCP-запросы клиентов на получение IP-адреса и сервер будет отправлять ответы клиентам напрямую. Нам же нужно, чтобы запрос приходил только к DHCP-агенту, а агент его уже пробрасывал на сервер. Этого можно добиться с помощью организации сети или ещё какими-либо другими способами, но один из вариантов - это поднять сервер на другом порту, чтобы он не получал запросы клиентов на 67-ом. Стандартный сервер DHCPd позволяет это сделать. В этом случае вам понадобится изменить этот параметр.

hcp.agent.host - IP-адрес DHCP-агента, на который потом будет отвечать DHCP-сервер. Этот адрес проставляется в DHCP-запрос, который проходит через relay-агент;

dhcp.minThreadCount, dhcp.maxThreadCount - это минимальное и максимальное количество потоков в пуле при обработке запросов от клиентов и серверов DHCP. Эти параметры можно оставить по умолчанию;

dhcp.arp.command - путь к команде arp в ОС Linux. Она нужна для манипулирования arp-таблицами. По умолчанию обычно `/sbin/arp`.

Для отключения динамического обновления arp-таблицы нужно выполнить команду

```
ifconfig eth0 -arp
```


При этом нужно посмотреть какие адреса уже попали в эту таблицу и, при необходимости, её отредактировать (например, добавить адреса каких-либо серверов в локальной сети, которые имеют статический ip).

Настройка ограничения скорости (шейпинг) для трафика WiFi-сети

Шейпинг осуществляется с помощью iproute2. Его реализация происходит через внешний скрипты и настраиваемые RADIUS-атрибуты, получаемые в Ассерт-пакете от Access-сервера. Вы можете реализовать свой вариант или изменить наш под ваши нужды. Общий принцип такой :

1. Скрипт init.sh (вызывается при старте системы) - в нем можно проводить инициализацию правил шейпинга;
2. login.sh - сюда добавляются правила шейпинга для нового клиента, появившегося в сети. В это скрипт в качестве параметра передаётся ip пользователя и атрибуты радиуса, полученные в ассерт-пакете(настройка атрибутов описана ниже).
- 3.logout.sh - здесь удаляются правила шейпинга при выходе клиента из сети. В этот скрипт в качестве параметра передаётся ip пользователя.

Мы вам предлагаем свой вариант реализации этих скриптов. Для корректной работы этого варианта в системе должен быть установлен perl. Для его конфигурации надо добавить в файл conf.sh следующие строчки:

```
USE_MANAD=0
#MANAD_INTERFACE_IN=eth0
#MANAD_INTERFACE_OUT=eth1
#MANAD_PORT=4567
```

Здесь USE_MANAD=1 обозначает, что будет использоваться шейпинг, MANAD_INTERFACE_IN - это интерфейс, на котором будет контролироваться входящий трафик клиента, MANAD_INTERFACE_OUT - это интерфейс, на котором будет контролироваться исходящий трафик клиента, MANAD_PORT - порт, на котором будет слушать perl-скрипт wifi_manad.pl, управляющий шейпингом. Этот скрипт слушает на определённом порту команды на удаление и добавление нового клиента . При добавлении клиента например с ip 192.168.184.33, скоростью входящего трафика (downstream) - 128 кбит/сек, скоростью исходящего трафика (upstream) - 256 кбит/сек, он добавляет для него следующие правила(eth0 - интерфейс для входящего трафика, eth1 для исходящего) :

```
/sbin/tc class add dev eth0 parent 1:0 classid 1:3 htb rate 128kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:3 handle 3: sfq perturb 10 quantum 1500
/sbin/tc class add dev eth1 parent 1:0 classid 1:3 htb rate 256kbit burst 4k prio 1
/sbin/tc qdisc add dev eth1 parent 1:3 handle 3: sfq perturb 10 quantum 1500
/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio 3 u32 match ip dst 192.168.184.33 flowid 1:3
/sbin/tc filter add dev eth1 parent 1:0 protocol ip prio 3 u32 match ip src 192.168.184.33 flowid 1:3
```

А при удалении клиента 192.168.184.33 скрипт выполняет следующие команды:

```
/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio 3
/sbin/tc filter del dev eth1 parent 1:0 protocol ip prio 3
/sbin/tc class del dev eth0 parent 1:0 classid 1:3 htb rate 128kbit burst 4k prio 1
/sbin/tc class del dev eth1 parent 1:0 classid 1:3 htb rate 256kbit burst 4k prio 1
```

Для правильной работы данного скрипта нужно настроить RADIUS-атрибуты, которые мы хотим получить из Ассерт-пакета от Access-сервера . В данном случае нас интересует два атрибута: ограничение входящей скорости (downstream) и значение исходящей скорости (upstream). Для этих атрибутов в файл в inet_wifi_agent нужно добавить следующие настройки:

```
wifi.agent.radius.atrubute.1.vendor.code=1111
wifi.agent.radius.atrubute.1.attr.code=1
wifi.agent.radius.atrubute.1.type=integer
wifi.agent.radius.atrubute.2.vendor.code=1111
wifi.agent.radius.atrubute.2.attr.code=2
wifi.agent.radius.atrubute.2.type=integer
```

Формат добавления атрибутов следующий:

agent.radius.atrubute.X. - общий вид

X - это код атрибута. Нумерация должна идти по порядку - 1, 2 и т.д. vendor.code - код производителя, attr.code - код атрибута, type - тип атрибута. RADIUS-атрибуты настраиваются в файле dictionary.xml Access-сервера и WiFi-портала (там тоже есть такой файл), и для данного примера можно добавить, например, такие атрибуты:

```
<vendor code="1111" name="linuxWiFi">
  <attribute add="no" name="WiFi-Downstream-Speed-Limit" type="integer" code="1"/>
  <attribute add="no" name="WiFi-Upstream-Speed-Limit" type="integer" code="2"/>
</vendor>
```

Для задания атрибутов возможны 2 варианта

1) Привязка атрибутов к realm'у. Например для realm'a default в конфигурации устройства можно указать:

```
radius.realm.default.attributes=WiFi-Downstream-Speed-Limit=256;WiFi-Upstream-Speed-Limit=128
```

2) Привязка атрибутов к опциям модуля Inet в конфигурации устройства:

```
radius.inetOption.1.attributes=WiFi-Downstream-Speed-Limit=128;WiFi-Upstream-Speed-Limit=128
radius.inetOption.2.attributes=WiFi-Downstream-Speed-Limit=256;WiFi-Upstream-Speed-Limit=256
```

Сами опции можно задавать в тарифе и в сервисе. Вкладка "[Атрибуты RADIUS](#)"#contract_radius_attributes

Значения атрибутов, описанных в файле inet_wifi_agent.properties, в точно в таком же порядке (после ip-адреса) подаются в качестве параметров в скрипт login.sh, вызываемый после установки пользователем соединения.

Настройка REALM'ов

Клиент при авторизации может использовать . Список доступных REALM'ов может отображаться клиенту в виде выпадающего списка :

Логин	den
Пароль	•••••
Услуга	128 kb/s ▾
<input type="button" value="Отправить"/>	
Безопасное соединение	

Пусть мы хотим использовать 2 REALM'а: входящий трафик 128 кбит/с и 256 кбит/с. Исходящий же трафик в обоих случаях пусть будет 128 кбит/сек. Для добавления этой возможности нужно добавить следующие настройки в файл inet_wifi_agent.properties:

```
#tariff options
portal.use.realm=1
portal.tarif.1.realm=wifi_128_128
portal.tarif.1.title=128 kb/s
portal.tarif.2.realm=wifi_256_128
portal.tarif.2.title=256 kb/s
```

В конфигурацию устройства для этого случая нужно добавить :

```
radius.realm.wifi_128_128.attributes=Wifi-Downstream-Speed-Limit=128;Wifi-Upstream-Speed-Limit=128
radius.realm.wifi_256_128.attributes=Wifi-Downstream-Speed-Limit=256;Wifi-Upstream-Speed-Limit=128

realmgr.default=default;wifi_128_128;wifi_256_128
radius.realm=default,wifi_128_128,wifi_256_128
```

Для разных realm'ов можно настроить различную [привязку трафика](#) к услугам.

Web-интерфейс WiFi-портала

Web-интерфейс WiFi-портала сделан с помощью [jsp](#) и располагается в каталоге portal. Также используются технологии [jstl](#) и [struts](#).

Активация на портале

В WiFi-портале есть возможность активации на портале с помощью sms или любым другим способом(логика реализуется в динамическом скрипте на стороне сервера биллинга). Для этого в конфигурации должна быть добавлена опция

```
portal.activate=1
```

В этом случае на странице авторизации появится ссылка "Активация", нажав на которую попадаем на новую страницу активации.

Логин	<input type="text"/>
Пароль	<input type="password"/>
Войти	<input type="button" value="Войти"/>
Активация	

фио	<input type="text"/>
телефон	<input type="text"/>
<input type="button" value="Активировать"/>	

Вид страницы активации зависит от настройки. Вот пример :

```
portal.activate.parameter.1.type=text
portal.activate.parameter.1.name=fio
portal.activate.parameter.1.title=
portal.activate.parameter.2.type=text
portal.activate.parameter.2.name=phone
portal.activate.parameter.2.title=
```

Тут type - это тип параметра (на текущий момент поддерживается только text). name - это некоторый уникальный идентификатор параметра, может быть использован в скрипте на сервера(об это дальше). title - имя параметра, как оно будет отображаться на странице активации.

Далее при нажатии на кнопку активации происходит отправка всех параметров на сервер биллинга. На сервере биллинга возникает глобальное событие "WiFi - активация"(класс ru.bitel.bgbilling.kernel.wifi.common.event.WiFiActivateEvent). И все параметры передаются в WiFiActivateEvent. Скрипт делает все необходимые проверки и действия(создает договор, сервис и т.п.). В случае каких-либо ошибок скрипт сообщает о них с помощью установки метода setError в событие WiFiActivateEvent. В случае успешного выполнения скрипт устанавливает сообщение об успешном выполнении в помощью метода setSuccessMessage. Оба сообщения (о успехе или неудаче) отображаются на активации.

Так же поддерживается другой механизм - активация с переходом по некоторому URL. В этом случае нужно вместо настройки представленной в начале нашей статьи поставить другую:

```
portal.activate.with.forward=1
```

В этом случае логика обработки скриптом аналогично алгоритму описанному выше, за исключением того, что в случае успеха вызывается не setSuccessMessage, а метод setPayUrl, в который передается UR для оплатыL. Портал в случае успеха переходит для оплаты по этому URL.

Например с помощью этого механизма можно сделать переход на станицу оплаты assist в личный кабинет.

Интеграция модуля Inet

Ericsson (RedBack) SmartEdge RSE

Управление через telnet, ssh, tcp, shell, mikrotik-api

IPoE с Mikrotik в качестве шейпера

СКАТ DPI BRAS

RDP IPoE BRAS (EcoNAT)

Интеграция с модулем Card

Ericsson (RedBack) SmartEdge RSE

Пример конфигурации для SmartEdge RSE.

В типе устройства указывается SmartEdge*ServiceActivator в поле "Обработчик активации сервисов" и SmartEdge*ProtocolHandler в поле "Обработчик процессора протокола", в зависимости от используемой схемы - просто PPPoE - SmartEdgeServiceActivator, CLIPS - SmartEdgeClipsServiceActivator, dot1q - SmartEdgeDot1qServiceActivator, static CLIPS - SmartEdgeStaticClipsServiceActivator как "Обработчик активации сервисов" и аналогично в поле "Обработчик процессора протокола".

В конфигурации типа устройства нужно указать:

Конфигурация типа устройства

```
# ( ) : 0 ( ) - , 1 - (
# AGENT_REMOTE_ID AGENT_CIRCUIT_ID INTERFACE_ID), 2 - VLAN (
# AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID), 4 - VLAN
# ( AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID),
# 5 - MAC- ( MAC_ADDRESS), 6 - MAC-
# ( MAC_ADDRESS).
#radius.servSearchMode=0
# : 0 - ( , IPoE), 1 ( ) - .
#radius.password.verification=1

# PoD CoA- ( - , /)
radius.port=3799

# Access-Accept .
# H, Reject-To-Accept Start-
connection.start.fromAccept=1

# Update- IP- (Framed-Ip-Address).
# Update- (0).
#connection.start.fromUpdate.ignoreFramedIpLack=0

# suspended
connection.suspend.timeout=3600
# c ( connection.suspend.timeout)
connection.close.timeout=3600
# , 0 ( ) - ,
# 1 - NAS' ( connectionClose )
#connection.close.timeout.forceClose=1

# , default (default - )
radius.realm.default.attributes=Acct-Interim-Interval=900

# , Access-Accept disable (rejectToAccept)
radius.disable.accessCodes=
# Access-Accept disable:
# 0 ( ) - radius.disable.ipCategories, 1 - , ( )
#radius.disable.mode=0

# , Access-Accept disable
radius.disable.attributes=Acct-Interim-Interval=900

# Id , , ( : 1, ).
# , radius.disable.accessCodes 1
#radius.disable.servId=

# . , .
sa.optionMode=1

# - (1 - -, -)
radius.inetOption.1.template=Service-Name:1=rse-inet;Service-Parameter:1=in=$in inb=$inb out=$out outb=$outb;
Service-Options:1=0
radius.inetOption.14.attributes=Service-Name:2=limited;Service-Options:2=1;Service-Parameter:
2=url=http://unpaid.provider.ru/691-4-31.html
# ID , , ( ) ( )
sa.radius.service.disable.optionIds=14
```



```

# RSE-, ( , )
radius.serviceName.disable=limited

# , CoA ( , SE , )
sa.radius.connection.attributes=Acct-Session-Id

# / ( )
sa.radius.connection.withoutBreak=1

# ServiceActivator ( , Access-Accept)
sa.radius.option.attributesPrefix=radius.inetOption.
# Access-Request. , NAS ( ) ,
# Stop- , .
# callingStationId : 1 - (connectionClose),
# 2 - (connectionClose) , , 3 - .
#radius.connection.checkDuplicate=0

# . - (0).
#radius.username.ignoreCase=0

#
# ,
#sa.error.pause=60
#
#sa.batch.size=20
# ( ) ( )
#sa.batch.wait=5
# ( )
#sa.batch.pause=0
# ( ) disconnect.
#sa.batch.waitNext=5

```

В конфигурации устройства нужно указать:

Конфигурация устройства

```

# PoD CoA- ( - , /)
#radius.host=< >
# PoD CoA- ( - , /)
radius.port=3799

# secret RADIUS- ( - Community/secret )
#radius.secret=<secret >

```

В данном примере используются [опции Inet](#). Необходимо создать опцию-папку "Скорости", в конфигурации типа устройства прописать её ID (например, 1) в имени параметра **radius.inetOption.1.template**:

Конфигурация устройства

```

radius.inetOption.1.template=Service-Name:1=rse-inet;Service-Parameter:1=in=$in inb=$inb out=$out outb=$outb;
Service-Options:1=0

```

Данный параметр представляет собой [шаблон](#), который будет применен к опциям, расположенным внутри опции "Скорости (1)". Поэтому в конфигурации опций-скоростей нужно будет указать параметры, которые были указаны в данном шаблоне (через \$), например:

Конфигурация устройства

```

in=1000000
inb=125000000
out=1000000
outb=125000000

```

Описание параметров SmartEdgeServiceActivator

Для обратной совместимости некоторые значения по умолчанию не изменяются в новых версиях, поэтому в таблице представлен также столбец "Рекомендуемое" значение.

Параметр	Значение по умолчанию	Рекомендуемое значение	Описание
radius.host	Хост из поля "Хост/порт" устройства		Хост, на который будут отправляться CoA/PoD-запросы.
radius.port	Порт из поля "Хост/порт" устройства		Порт, на который будут отправляться CoA/PoD-запросы.
radius.sourceHost			Интерфейс, с которого будут отправляться CoA/PoD-запросы.
radius.sourcePort			Порт, с которого будут отправляться CoA/PoD-запросы.
radius.identifier	Значение из поля "Идентификатор" устройства		Значение, которое будет подставлено в CoA/PoD-запрос в атрибут NAS-Identifier, если NAS-Identifier присутствует в значении параметра sa.radius.connection.attributes
sa.radius.secret	Значение из параметра radius.secret		RADIUS-секрет, который будет использован для создания CoA/PoD-запроса.
radius.secret	Значение из поля "Community/secret" устройства		RADIUS-секрет, который будет использован для создания CoA/PoD-запроса. Значение может быть переопределено в параметре sa.radius.secret Параметр также используется при штатной обработке Access-Request (переопределяет значение из поля "Community/secret").
sa.radius.log	1		Логировать (1) или не логировать (2) CoA/PoD-запросы сессии (возможность просматривать RADIUS-лог сессии в клиенте).
sa.optionMode	0	1	Режим работы с сервисами SmartEdge. При 0 - просто отправляются атрибуты из параметров, при 1 - выполняется привязка опций к имени сервиса SmartEdge по атрибуту Service-Name.
sa.radius.connection.withoutBreak	0		0 - при необходимости переключить состояние сессии (ограничить доступ или наоборот, включить) будет произведен сброс сессии на оборудовании. Может быть нужно, если Вы выдаете IP-адреса из разных ресурсов для тех, у кого ограничен доступ в Интернет и у кого не ограничен. 1 - при необходимости переключить состояние сессии будет произведена смена сервисов RSE.
sa.radius.connection.stateModify	0		Нужно ли сразу после отправки CoA на переключение доступа сменить состояние сессии в БД. Состояния сессии также может определяться по сервисным сессиям RSE с помощью SmartEdgeProtocolHandler. См. далее. См. также Сессии и FAQ модуля Inet
sa.radius.connection.modify.forceOnServiceNak	0		Нужно ли считать, что CoA/PoD отработал, если в ответ пришел NaK. При 0 и NaK запрос будет повторен.
sa.radius.service.disable	Значение из параметра radius.serviceName.disable		Список имен сервисов RSE, по которым будет определен список опций, которые нужно "включить" при необходимости ограничить доступ абоненту.
radius.serviceName.disable			Список имен сервисов RSE, по которым будет определен список опций, которые нужно "включить" при необходимости ограничить доступ абоненту. Значение может быть переопределено в параметре sa.radius.service.disable Параметр также используется в SmartEdgeProtocolHandler.
sa.radius.connection.close.mode	2		При необходимости сбросить соединение: 1 - не отправлять PoD-пакет (подразумевается, что биллинг также обрабатывает DHCP-пакеты и на следующий DHCP-пакет он ответит NAK, см. параметр sa.radius.connection.close.removeFromKeyMap) 2 - отправлять PoD-пакет

sa.radius.connection.close.enableMode	2		<p>При необходимости сбросить соединение для последующего включения доступа:</p> <p>1 - не отправлять PoD-пакет (подразумевается, что биллинг также обрабатывает DHCP-пакеты и на следующий DHCP-пакет он ответит NAK, см. параметр sa.radius.connection.close.removeFromKeyMap)</p> <p>2 - отправлять PoD-пакет</p>
sa.radius.connection.coa.onEnable	0		<p>При необходимости сбросить соединение для последующего включения доступа нужно ли послать CoA на включение доступа (подразумевается, что доступ будет включен, PoD-пакет не отправлен, а на следующий DHCP-пакет биллинг ответит NAK и абонент переполучит адрес).</p>
sa.radius.connection.close.disableMode	Значение параметра sa.radius.connection.close.mode		<p>При необходимости сбросить соединение для последующего отключения доступа:</p> <p>1 - не отправлять PoD-пакет (подразумевается, что биллинг также обрабатывает DHCP-пакеты и на следующий DHCP-пакет он ответит NAK, см. параметр sa.radius.connection.close.removeFromKeyMap)</p> <p>2 - отправлять PoD-пакет</p>
sa.radius.connection.close.disableServices	0		<p>При необходимости сбросить соединение (не для последующего включения доступа) нужно ли послать CoA на отключение доступа (подразумевается, что доступ будет отключен, PoD-пакет не отправлен, а на следующий DHCP-пакет биллинг ответит NAK и абонент переполучит адрес).</p>
sa.radius.service.closeAttributes	<p>При sa.optionMode=1: <пусто></p> <p>При sa.optionMode=0: Deactivate-Service-Name: 1=RSE-SVC-EXT</p>	<пусто>	<p>Атрибуты, добавляемые в CoA при необходимости переключить скорость, необходимые для отключения текущих сервисов RSE. При sa.optionMode=1 имена сервисов для отключения автоматически вычисляются из текущих активных опций Inet, поэтому в этом режиме данное поле можно не указывать.</p>
sa.radius.connection.close.removeFromKeyMap	0		<p>При необходимости сбросить соединение, нужно ли перестать выдавать DHCP-ACK для данной сессии, даже если она еще "живая".</p>
sa.radius.service.disable.optionIds			
sa.radius.connection.close.disableServices			
sa.radius.connection.attributes			

Управление через telnet, ssh, tcp, shell, mikrotik-api

- TerminalServiceActivator
 - Параметры подключения
 - Команды для инициации подключения и отключения
 - Параметры работы
 - Выполнение команд
 - Возможные наборы команд для событий
 - Макросы
 - Вызовы других наборов команд через макросы
 - Управление с помощью Mikrotik API
 - Новый режим
 - Старый режим

TerminalServiceActivator

Обработчик предназначен для выполнения команд по telnet, ssh, tcp, shell. Команды для разных событий задаются в конфигурации типа устройства.

Параметры подключения

```
# ( , /)
#sa.terminal.host=
# ( , /)
#sa.terminal.port=
# (telnet/ssh/tcp/mikrotik/shell/mikrotik-api)
sa.terminal.protocol=telnet
# ( , )
#sa.terminal.username=
# ( , )
#sa.terminal.password=
# (telnet/ssh/tcp)
# ssh regexp, '\', , "\(\#|(~\$)\\"
sa.terminal.endSequence=#

# sa.terminal.protocol=mikrotik - ,
#sa.terminal.sourceHost=
#sa.terminal.sourcePort=

# . 1, ,
# - 0 ( ), connect
sa.terminal.lazyConnect=0
```

При подключении по ssh можно использовать RSA/DSA ключ. При этом в конфигурации необходимо указать путь к приватному ключу:

```
#
sa.terminal.protocol=ssh
# ( , )
#sa.terminal.username=
# ( , )
#sa.terminal.password=
#
sa.terminal.privateKeyFile=id_rsa
```

Параметр **sa.endSequence** используется для определения окончания вывода от удаленного оборудования и приглашения ввода команд. Например, когда консоль ssh выглядит так:

```
amir@ts01:~$ date
. . 16 15:58:55 YEKT 2016
amir@ts01:~$
```

Мы отправили команду, получили ответ и можем отправлять новую.

Во время выполнения команд есть возможность подменить значение **sa.endSequence** на другое, с помощью макроса **#{setEndSequence('::~')}**. Это может быть полезно, т.к. на некотором оборудовании строка приглашения во время выполнения команд может меняться. Для того, чтобы вернуть значение на значение **sa.endSequence** из конфига, можно вызвать макрос без аргумента: **#{setEndSequence()}**.

Команды для инициации подключения и отключения

При появлении задачи синхронизации, если на данный момент соединение не установлено, обработчик подключается к устройству, используя параметры подключения, затем выполняет команды, прописанные в параметре **sa.terminal.connect**. Далее выполняются команды синхронизации, возможно последовательно для нескольких договоров. Когда задания выполнены и новых нет в течении **sa.batch.waitNext.millis**, то происходит отключение от устройства, но перед этим выполняются команды, указанные в **sa.terminal.disconnect**. Перед самым отключением выполняется команда **sa.terminal.exit** (в отличие от **sa.terminal.disconnect** после этой команды нет ожидания ответа от устройства).

```
# ,
sa.terminal.connect=
# ,
sa.terminal.disconnect=
# ( )
sa.terminal.exit=exit
```

Параметры работы

```
#      connectionModify ( 1)
# (      Inet - )
#sa.terminal.connection.stateModify=1

# ( / ISG/SmartEdge)
# .. onAccountingStart/onAccountingStop ( 1)
#sa.terminal.connection.skipServiceAccounting=1

# ,      onAccountingStart/onAccountingStop
#sa.terminal.connection.deviceIds=

# ( - - ((- ) + 1), 100)
#sa.terminal.semaphorePermits=0

# ( )
#sa.inetOption.root=

# Regex , (alarm)
#sa.terminal.alarm.pattern=
# Regex ,
#sa.terminal.error.pattern=
```

Выполнение команд

Возможные наборы команд для событий

Для каждого события есть три набора команд: "перед вызовом обычного набора", "обычный набор", "после вызова обычного набора". Например, **sa.terminal.serv.modify**, **sa.terminal.serv.modify.before**, **sa.terminal.serv.modify.after**. Наборы команд **.before** и **.after** в основном предназначены для вызова других наборов команд и изменения логики работы, в зависимости от ситуации.

```

# ,
sa.terminal.serv.create=
# ,
sa.terminal.serv.cancel=
# ,      (/)
sa.terminal.serv.modify=
# ,      (  ")
sa.terminal.serv.modify.enable=
# ,      (  ")
sa.terminal.serv.modify.disable=
# ,      ID=x (      ID=x)
sa.terminal.serv.inetOption.x.enable=
# ,      ID=x (      ID=x)
sa.terminal.serv.inetOption.x.disable=
# ,      (/)
sa.terminal.connection.modify=
# ,      (  ")
sa.terminal.connection.modify.enable=
# ,      (  ")
sa.terminal.connection.modify.disable=
# ,      ID=x (      ID=x)
sa.terminal.connection.inetOption.x.enable=
# ,      ID=x (      ID=x)
sa.terminal.connection.inetOption.x.disable=
# ,
sa.terminal.connection.close=
# ,
sa.terminal.connection.onAccountingStart=
# ,
sa.terminal.connection.onAccountingStop=

```

Также можно использовать отдельные наборы команд для разных типов сервисов. Для этого нужно использовать префикс **sa.terminal.set.имя_набора_команд**, и указать его типы сервисов в параметре **sa.terminal.set.имя_набора_команд.servTypeIds**:

```

# ID ,      mySet
sa.terminal.set.mySet.servTypeIds=1,2,3
# ,
sa.terminal.set.mySet.serv.create=
# ,
sa.terminal.set.mySet.serv.cancel=
# ,      (/)
sa.terminal.set.mySet.serv.modify=
...
...
...

```

Команды опций **.serv.inetOption.x** и **.connection.inetOption.x** автоматически наследуются для дочерних опций к опции с ID равным **x**.

Синхронизация всегда выполняется для родительского сервиса. Для работы с информацией из дочерних сервисов необходимо использовать макрос **loopServ()**. При изменении IP-адреса, интерфейса, VLAN или набора дочерних сервисов вызываются команды удаления сервиса (с использованием старых параметров), а затем добавления (с использованием новых).

Макросы

Макросы вызываются с помощью последовательности **\$(имямакроса)**. Внутри фигурных скобок может быть логическое выражение, однако в этом случае, оно должно быть оформлено как правильный java-код, например: `$(newState()==1)?'вкл':'выкл'`, `$(newState()==1&&oldState()!=1)?(connectionEnable()+optionsEnable()):''`.

Доступны следующие макросы:

- **ip()** - IP-адрес сессии или IP-адрес сервиса или начало диапазона IP-адресов сервиса;
- **ipList()** - список IP-адресов диапазона сервиса, через запятую с пробелом, например: 192.168.1.1, 192.168.1.2, 192.168.1.3, 192.168.1.4, 192.168.1.5, ... ;
- **ipList(',')** - список IP-адресов диапазона сервиса, через запятую, например: 192.168.1.1,192.168.1.2,192.168.1.3,192.168.1.4,192.168.1.5,... ;
- **net()** - подсеть сервиса в виде строки, например, 192.168.1.1/24;
- **mask()** - маска подсети сервиса, например, 24;
- **netmask()** - маска подсети сервиса, например, 255.255.255.0;

- **netmaskWildcard()** - wildcard подсети, например, 0.0.0.255;
 - **ipNetHostMin()** - первый адрес подсети, например, 192.168.1.1;
 - **ipNetHostMax()** - последний адрес подсети, например, 192.168.1.254;
 - **ipNetBroadcast()** - 192.168.1.255;
 - **ipFrom()** - первый адрес диапазона сервиса;
 - **ipTo()** - последний адрес диапазона сервиса;
-
- **contractId()** - ID договора
 - **contractTitle()** - название договора
 - **servId()** - ID сервиса договора
 - **servTitle()** - название сервиса договора
 - **servDeviceId()** - ID устройства, к которому привязан сервис договора
 - **vlan()** - VLAN сервиса
 - **svlan()** - SVLAN устройства данного обработчика;
 - **svlan(id_устройства)** - SVLAN устройства с указанным ID;
 - **svlan(servDeviceId())** - SVLAN устройства сервиса;
 - **iface()** - интерфейс сервиса
 - **ifaceTitle()** - название интерфейса сервиса
 - **ifaceTitleBeforeColon()** - название интерфейса до символа ':'
 - **ifaceTitleAfterColon()** - название интерфейса после символа ':'
 - **mac()** - MAC-адрес, в виде 0a:1b:2c:3d:4e:5f
 - **macBytes()** - MAC-адрес, в виде 0a1b2c3d4e5f
 - **macDotted()** - MAC-адрес, в виде 0a1b.2c3d.4e5f
 - **identifier()** - идентификатор сервиса
 - **servParam(имя_параметра)** - параметр сервиса
 - **servParam(имя_параметра, значение_по_умолчанию)** - параметр сервиса
-
- **option()** - возвращает активную опцию (используя фильтр sa.inetOption.root, если он установлен);
 - **option(id_опции)** - возвращает активную опцию, которая будет потомком к указанной id_опции (т.е. таким образом можно отфильтровать до опции из определенной подгруппы);
 - **param(опция, имя_параметра, значение_по_умолчанию)** - параметр конфигурации опции (или одного из предка опции, т.к. параметры наследуются) или значение_по_умолчанию, если опция==null или параметр в опции не указан;
-
- **param(имя_параметра, значение_по_умолчанию)** - параметр конфигурации устройства (или одного из предка устройства, т.к. параметры наследуются) или значение_по_умолчанию, если параметр в устройстве не указан;
-
- **ipGate()** - шлюз из IP-ресурса
 - **ipDns()** - DNS из IP-ресурса
 - **ipSubnetMask()** - маска подсети из IP-ресурса
 - **ipParam(имя_параметра)** - параметр из конфигурации IP-ресурса
-
- **deviceId()** - ID устройства данного обработчика;
 - **deviceIP()** - IP-адрес устройства (из поля Хост/порт) данного обработчика;
 - **deviceIP(id_устройства)** - IP-адрес устройства (из поля Хост/порт) с указанным ID;
 - **deviceIdIdentifier()** - идентификатор устройства данного обработчика;
 - **deviceIdIdentifier(id_устройства)** - идентификатор устройства с указанным ID;
 - **deviceIdIdentifier(servDeviceId())** - идентификатор устройства сервиса;
 - **deviceAttr(id_атрибута)** - значение атрибута устройства данного обработчика;
 - **deviceAttr(id_устройства, id_атрибута)** - значение атрибута устройства с указанным ID;
 - **deviceAttr(id_устройства, id_атрибута, значение_по_умолчанию)** - значение атрибута устройства с указанным ID, если значения нет - возвращается значение по умолчанию;
-
- **connectionAgentDeviceId()** - ID агентского устройства, к которому привязано соединение;
 - **connectionDeviceId()** - ID устройства, к которому привязано соединение;
-
- **translit(строка)** - транслитерированное значение аргумента;
 - **format(шаблон, arg1, ...)** - форматирование, используя java.text.MessageFormat.format(String, Object...);
 - **now()** - текущее время, например, \${format('{0,date,yyyy}', now())};
 - **nowString()** - текущее время, в формате dd.MM.yyyy HH:mm:ss ;

- **setEndSequence(строка)** - установка последовательности для ожидания приглашения ввода новой команды;
- **newState()** - новое (требуемое) состояние сервиса/сессии;
- **oldState()** - старое (текущее) состояние сервиса/сессии;
- **skipIfEmpty(val)** - пропускает выполнение команды, если val - null или пустая строка;
- **skipIf(val)** - пропускает выполнение команды, если val - true;
- **mikrotikLastIds()** - результаты вызова предыдущей команды (для Mikrotik, все параметры id, через запятую);
- **lastResponse()** - получение ответа на последнюю выполненную команду;
- **lastResponse(regex)** - получение значения по regex из ответа на последнюю выполненную команду;

```
lastResponse(/\b\w+\.\d+\.\d+\b/)
lastResponse().replaceAll(/(?ms).*\b(\w+)\.\d+\.\d+\b.*/, '$3')
```

- **var('имя', 'значение')** - установка временной переменной;
- **var('имя')** - получение значения временной переменной;

```
sa.terminal.connection.modify.enable.1=:put [/interface bridge host get [/interface bridge host find mac-
address= [ /ip arp get [/ip arp find address="{ip()}"] mac-addr
>ess ]] on-interface]
# ether3.1003.311
sa.terminal.connection.modify.enable.2=${var('vlan', lastResponse(/\b\w+\.\d+\.\d+\b/))}
sa.terminal.connection.modify.enable.3=...${var('vlan')}...
```

- **loopServ(имя параметра шаблона команды, использовать в цикле родительский сервис, выполнять как отдельную команду или построить строку)** - способ выполнения команд с использованием дочерних сервисов; выполняется цикл для всех дочерних сервисов, а также для родительского, если второй аргумент равен true; шаблоном команды является значение параметра, имя которого прописано в первом аргументе; выполняются как отдельные команды (true) или возвращается конкатенированная строка с указанным разделителем

```
m_ipLoop=${net()}
sa.commandSet.staticIP.serv.create.1=/queue/simple/remove\n=numbers=${contractId()}:${servId()}
sa.commandSet.staticIP.serv.create.2=/queue/simple/add\n=name=${contractId()}:${servId()}\n=target=127.
0.0.1/32,${loopServ('m_ipLoop', false, ',')}
```

Вызовы других наборов команд через макросы

Для каждого события есть три набора команд: "перед вызовом обычного набора", "обычный набор", "после вызова обычного набора". Например, **sa.terminal.serv.modify**, **sa.terminal.serv.modify.before**, **sa.terminal.serv.modify.after**. Наборы команд **.before** и **.after** предназначены в основном для вызова других наборов команд и изменения логики работы, в зависимости от ситуации. Например, стандартная предопределенная конфигурация выглядит так:


```

#      (serv.enable=)    (inetOption.x.enable=),
#      (serv.disable=),
sa.terminal.serv.create.after=${(newState()==1)?(serviceEnable()+optionsEnable()):serviceDisable()}

#      (serv.modify=),      -      (serv.modify.enable=)    (inetOption.x.enable=)
sa.terminal.serv.modify.before=${(newState()==1&&oldState()!=1)?(serviceEnable()+optionsEnable()):''}
#      (serv.modify=),      -      (inetOption.x.disable=),      -      (serv.modify.disable=);
#      ,      -      (inetOption.x.disable= (inetOption.x.enable=))
sa.terminal.serv.modify.after=${(newState()!=1)?(optionsDisable()+serviceDisable()):''};${(newState()
==1&&oldState()==1)?(optionsSwitch()):''}

#
sa.terminal.serv.cancel.before=${optionsDisable()+serviceDisable()}

#      ,      -
sa.terminal.connection.modify.before=${(newState()==1&&oldState()!=1)?(connectionEnable()+optionsEnable()):''}
#      ,      -      ,      -      ;      -
sa.terminal.connection.modify.after=${(newState()!=1)?(optionsDisable()+connectionDisable()):''};${(newState()
==1&&oldState()==1)?(optionsSwitch()):''}

#      (sa.terminal.connection.onAccountingStart),      -
#      (sa.terminal.connection.modify.disable),      -      (sa.terminal.connection.modify.enable)
sa.terminal.connection.onAccountingStart.before=${(newState()!=1)?(connectionDisable()):connectionEnable()}
#      ,      -
sa.terminal.connection.onAccountingStart.after=${(newState()==1)?(optionsEnable()):''}

#      (sa.terminal.connection.onAccountingStop),      -
sa.terminal.connection.onAccountingStop.before=${(oldState()==1)?(optionsDisable()):''}

```

Макросы **serviceEnable()**, **serviceDisable()**, **connectionEnable()**, **connectionDisable()**, **connectionClose()**, **optionsEnable()**, **optionsDisable()** вызывают соответствующие наборы команд. **optionsSwitch()** - вызывает наборы команд на отключение для старых опций и включение для новых. **newState()** и **oldState()** возвращают текущее и новое состояние сервиса или соединения, значения -1 - удален (для сервиса), 0 - отключен, 1 - включен.

Управление с помощью Mikrotik API

Оборудование Mikrotik предоставляет протокол для управления: <https://wiki.mikrotik.com/wiki/Manual:API>

Основное отличие от обычного консольного синтаксиса CLI состоит в том, что в командах заменяются пробелы на "/" ("interface vlan print" превращается в "/interface/vlan/print"), параметры передаются по другому и нет возможности вызывать вложенные команды (однако есть запоминание id, которые вернул предыдущий вызов команды, и есть возможность использовать их в следующей команде).

Новый режим

Для управления с помощью Mikrotik API нужно указать:

```
sa.terminal.protocol=mikrotik-api
```

Для управления с помощью Mikrotik API-SSL нужно указать:

```
sa.terminal.protocol=mikrotik-api-ssl
```

В обоих случаях используется библиотека mikrotik-java. Она поддерживает как api так и api-ssl. Синтаксис несколько отличается от того, который описан в протоколе Mikrotik, примеры можно посмотреть здесь: <https://github.com/GideonLeGrange/mikrotik-java>.

Пример конфигурации:

```

# ( , /)
sa.terminal.port=8728
# (telnet/ssh/tcp/mikrotik-api/shell)
sa.terminal.protocol=mikrotik-api

sa.port=8728
# ( )
sa.terminal.exit=
# , ( "" )
sa.terminal.serv.modify.enable.1=/interface/vlan/print where name=${ifaceTitle()}
sa.terminal.serv.modify.enable.2=/interface/vlan/enable .id=${mikrotikLastIds()}
# , ( "" )
sa.terminal.serv.modify.disable.1=/interface/vlan/print where name=${ifaceTitle()}
sa.terminal.serv.modify.disable.2=/interface/vlan/disable .id=${mikrotikLastIds()}

# , ID=x ( ID=x)
sa.terminal.serv.inetOption.1.enable.1=/queue/simple/print where target=${ifaceTitle()}
sa.terminal.serv.inetOption.1.enable.2=/queue/simple/set .id=${mikrotikLastIds()} max-limit= ${param(option() ,
'mikrotik_rate', '20M/20M')}

```

Еще пример:

```

# , ID=x ( ID=x)
sa.terminal.connection.inetOption.1.enable.1=/ip/firewall/address-list/print .proplist=.id where address=${ip()}
and comment=bgb${servId()}
sa.terminal.connection.inetOption.1.enable.2=${skipIfEmpty(mikrotikLastIds())}/ip/firewall/address-list/remove .
id=${mikrotikLastIds()}
sa.terminal.connection.inetOption.1.enable.3=/ip/firewall/address-list/add address=${ip()} list=${param(option
(1), 'list', '512k_ip')} comment=bgb${servId()}
# , ID=x ( ID=x)
sa.terminal.connection.inetOption.1.disable.1=/ip/firewall/address-list/print .proplist=.id where
address=${ip()} and comment=bgb${servId()}
sa.terminal.connection.inetOption.1.disable.2=${skipIfEmpty(mikrotikLastIds())}/ip/firewall/address-list/remove
.id=${mikrotikLastIds()}

```

Старый режим

Так же есть вот такой режим работы:

```
sa.terminal.protocol=mikortik
```

Это тоже протокол Mikrotik API, но другой синтакс команд в конфигурации. В данный момент он объявлен устаревшим и возможно будет удален в следующих версиях. Не рекомендуем его использовать.

Здесь команды задаются немного другим образом:

```

#
sa.terminal.serv.modify.enable.1=/interface/vlan/print\n?name=${ifaceTitle()}
sa.terminal.serv.modify.enable.2=/interface/vlan/enable\n.id=${mikrotikLastIds()}
#
sa.terminal.serv.modify.disable.1=/interface/vlan/print\n?name=${ifaceTitle()}
sa.terminal.serv.modify.disable.2=/interface/vlan/disable\n.id=${mikrotikLastIds()}

```

IPoE с Mikrotik в качестве шейпера

Данный пример предполагает выдачу абоненту биллингом адреса по DHCP и управление Mikrotik'ом с помощью Mikrotik-API для назначения скорости и ограничений данному IP-адресу.

- [Настройка](#)
- [Примечание](#)
- [Вариант конфигурации команд](#)

Настройка

1. На вкладке **Устройства и ресурсы** - **IP-ресурсы** создайте категорию IP-ресурса, а затем IP-ресурс в этой категории. Можете также указать роутер, маску подсети и DNS - они будут выданы в DHCP (данные параметры можно также указывать в конфигурации устройства-коммутатора).
2. На вкладке **Устройства и ресурсы** - **VLAN-ресурсы** создайте категорию VLAN-ресурса и VLAN-ресурс.
3. На вкладке **Справочники** - **Опции** создайте опцию-папку **Скорость**, создайте дочерние опции-скорости, укажите в их конфигурациях address list'ы Mikrotik'a, например:

```
list=512k_ip
```

4. На вкладке **Устройства и ресурсы** - Типы устройств создайте тип устройства с названием **Access+Accounting**.
5. Создайте тип устройства **RouterOS**, укажите в качестве Обработчика активации сервисов **TerminalServiceActivator**. Укажите в конфигурации данного типа устройства:

```
# ( , /)
sa.terminal.port=8729
# (telnet/ssh/tcp/mikrotik-api/mikrotik-api-ssl/shell)
sa.terminal.protocol=mikrotik-api-ssl

# ()
sa.terminal.exit=

# Regex , (alarm)
sa.terminal.alarm.pattern=input does not match any value of queue
# Regex ,
#sa.terminal.error.pattern=input does not match any value of queue

# , ID=x ( ID=x)
sa.terminal.connection.inetOption.1.enable.1=/ip/firewall/address-list/print .proplist=.id where
address=${ip()} and comment=bgb${servId()}
sa.terminal.connection.inetOption.1.enable.2=${skipIfEmpty(mikrotikLastIds())}/ip/firewall/address-list
/remove .id=${mikrotikLastIds()}
sa.terminal.connection.inetOption.1.enable.3=/ip/firewall/address-list/add address=${ip()} list=${param
(option(1),'list','512k_ip')} comment=bgb${servId()}
# , ID=x ( ID=x)
sa.terminal.connection.inetOption.1.disable.1=/ip/firewall/address-list/print .proplist=.id where
address=${ip()} and comment=bgb${servId()}
sa.terminal.connection.inetOption.1.disable.2=${skipIfEmpty(mikrotikLastIds())}/ip/firewall/address-list
/remove .id=${mikrotikLastIds()}
```

Обратите внимание на макрос `${param(option(1),'list','512k_ip')}` - он берет текущую активную опцию, дочернюю по отношению к опции с ID=1 и получает значение параметра list из ее конфигурации (третий параметр - значение по умолчанию).

При работе через SSH настройки подключения будут такими (но также нужно будет поменять команды, т.к. в Mikrotik SSH и API синтаксис различается):

```

# ( , /)
sa.terminal.port=22
# (telnet/ssh/tcp/mikrotik-api/mikrotik-api-ssl/shell)
sa.terminal.protocol=ssh

# Regex
sa.terminal.endSequence=\\[[\\w-_]+@[\\w-_]+\\]\\s+>\\

# ( )
sa.terminal.exit=quit

```

При работе через SSH рекомендуется добавлять к логину суффикс **+ct**. Тогда SSH-соединение с Mikrotik будет работать без дополнительного форматирования и цветового оформления (ради которого посылается множество дополнительных спецсимволов).

- Создайте тип устройства-relay, например, **Cisco Catalyst**, укажите в конфигурации [параметры разбора Agent-Remote-Id из Option 82](#), например:

```

dhcp.option82.removeHeader=2
# agentRemoteId
# agentRemoteId: 0 ( ) - , 1 -
#dhcp.option82.agentRemoteId.type=0
# 82, ,
dhcp.option82.agentRemoteId.code=2
dhcp.option82.agentRemoteId.position=0
dhcp.option82.agentRemoteId.length=6

# 82, VLAN,
#dhcp.option82.vlanId.code=1
#dhcp.option82.vlanId.position=0
#dhcp.option82.vlanId.length=2
# 82, SVLAN,
#dhcp.option82.agentSvlanId.code=1
#dhcp.option82.agentSvlanId.position=0
#dhcp.option82.agentSvlanId.length=2

```

- Создайте тип устройства с названием **SA-Rerouting**, с обработчиком активации сервисов RecipientListServiceActivator, в конфигурации укажите:

```

# onAccountingStart
sa.device.sync.onAccountingStart=1

```

Устройство данного типа будут перенаправлять задания синхронизации на другую папку дерева устройств.

- Создайте тип устройства коммутатора доступа, например **DGS-1100-06/ME**, на вкладке Интерфейсы создайте интерфейсы (порты) коммутатора.
- В дереве устройств создайте устройство типа **Access+Accounting**, оно будет "корневым" для приложений InetAccess и InetAccounting. Укажите в нем конфигурацию:

```

# DHCP- (IP- InetAccess)
dhcp.serverIdentifier=10.0.25.35

# VLAN
vlan.resource.category=1
# IP-
dhcp.ipCategories=1
# lease time,
dhcp.option.leaseTime=1800
# DHCP(RADIUS/Netflow)-
connection.suspend.timeout=1300
connection.close.timeout=1300

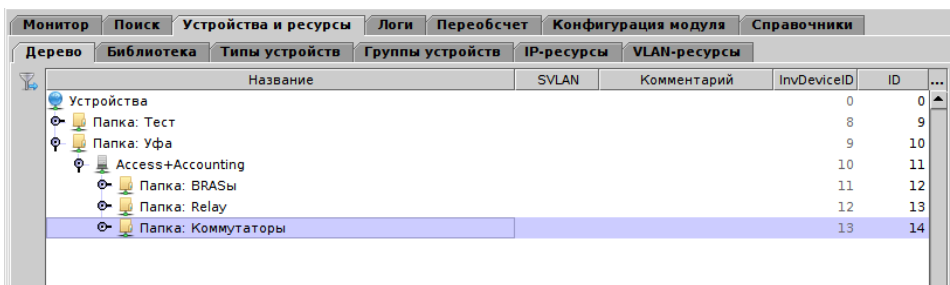
# , , VLAN
dhcp.servSearchMode=4

#
accounting.worker.1.thread.count=1
accounting.worker.1.tariffication.1.minDeltaAmount=0
accounting.worker.1.tariffication.1.delay=10
accounting.worker.1.tariffication.1.batchSize=500
accounting.worker.1.tracking.1.delay=10
accounting.worker.1.tracking.1.batchSize=500
accounting.worker.2.thread.count=1
accounting.worker.2.flushing.1.minDeltraAccount=0
accounting.worker.2.flushing.1.delay=20
accounting.worker.2.flushing.1.batchSize=500
accounting.worker.3.thead.count=1
accounting.worker.3.finishing.1.delay=20
accounting.worker.3.finishing.1.batchSize=500

# Access-Accept
connection.start.fromAccept=1

```

10. Создайте дочерние устройства-папки **BRASы, Relay, Коммутаторы**:



11. В папке **BRASы** создайте устройство типа RouterOS, укажите в поле Хост/порт IP-адрес данного BRAS'a, в полях Логин и Пароль - соответственно логин и пароль для управления данным устройством.
12. В конфигурации папки **Relay** укажите в конфигурации, что для всех устройств в данной папке агентскими устройствами будут устройства в папке **Коммутаторы**, указав ID папки **Коммутаторы**:

```

# ( )
device.agentDeviceRootId=14

```

13. В папке **Relay** создайте устройство типа Cisco Catalyst, в поле Хост/порт укажите IP-адрес данного устройства.
14. В папке **Коммутаторы** создайте устройство типа **SA-Rerouting** и укажите в его конфигурации в параметре **sa.recipientList.rootDeviceId** ID папки **BRASы**, чтобы задания синхронизации дублировались на устройства данной папки:

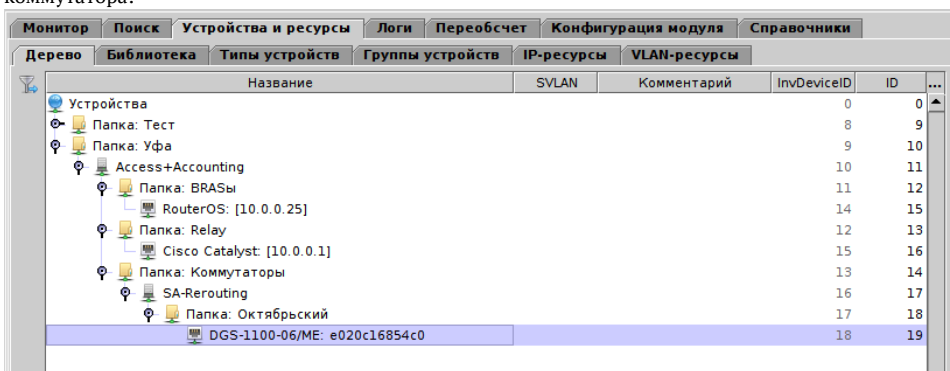
```

# (BRAS)
sa.recipientList.rootDeviceId=12

```

15. В устройстве SA-Rerouting создайте дочернюю папку (например, для группировки по районам).

16. Дочерним к устройству типа SA-Rerouting создайте устройство типа **DGS-1100-06/ME**, укажите в поле идентификатор MAC-адрес коммутатора:



17. На вкладке **Справочники - Типы сервисов** создайте тип сервиса с названием, например IPoE, установите галочки устройство, интерфейс, интерфейс персонален, VLAN, VLAN персонален, MAC-адрес. Типа адреса: динамический или статический. В конфигурации:

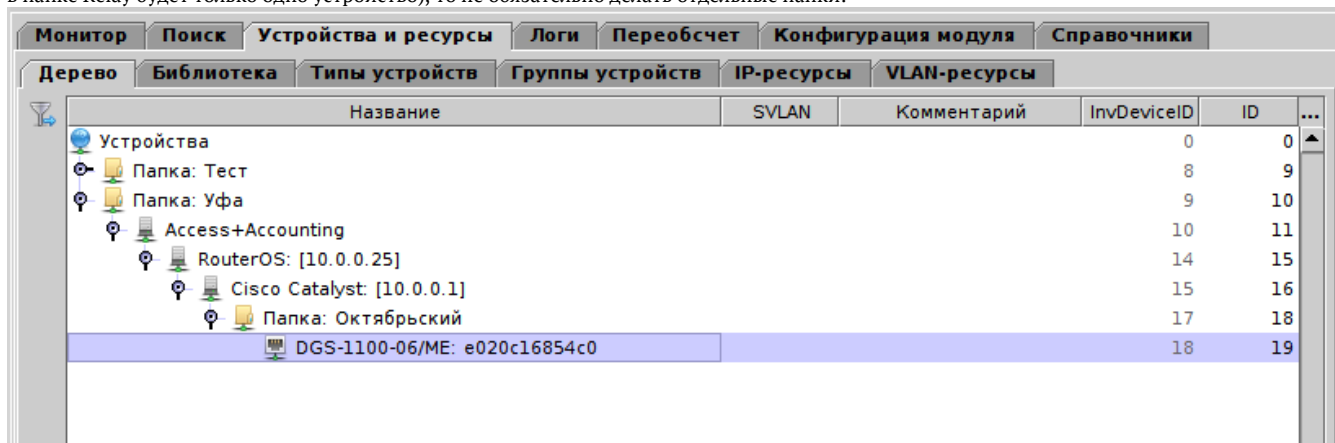
```
#
title.pattern=({deviceIdentifier})/({interfaceId})
```

18. В договоре создайте сервис данного типа, укажите коммутатор, порт, VLAN и MAC-адрес.

Примечание

Указание всех полей в типе сервиса не является обязательным - важны лишь те поля, по которым идет поиск. Например, при поиске по VLAN обязательно нужно лишь указать устройство-коммутатор (не обязательно коммутатор доступа, это может быть коммутатор агрегации, если у вас VLAN-per-User) и VLAN. Остальные поля могут быть использованы при управлении устройствами (коммутаторами).

Дереву устройств не обязательно быть именно таким. Если у вас один BRAS (шейпер), то SA-Rerouting не нужен. Если релей не дублируются (т.е. в папке Relay будет только одно устройство), то не обязательно делать отдельные папки:



Вариант конфигурации команд

```
# ( , /)
#sa.terminal.host=
# ( , /)
sa.terminal.port=22
# (telnet/ssh/tcp/mikrotik-api/mikrotik-api-ssl/shell)
sa.terminal.protocol=ssh
# Regex
sa.terminal.endSequence=\\[\\w+@\\w+\\]\\s+>\\

# ()
sa.terminal.exit=quit

# {@key} , , alias'
# ${key()}
```

```

acl=ACL-accepted
sif=ether3
# SVLAN calledStationId ( SVLAN DHCP-)
svlan=${calledStationId().replaceAll( /\^(d+)\./+$/, '$1' )}
vlan=${vlan()}
interface="@sif}.${svlan}.${vlan}"
#
speed=${param( option(), 'speed', '1' )}

# onAccountingStop sa.terminal.connection.modify.disable
sa.terminal.connection.onAccountingStop.after=${connectionDisable()}

# , ( ""),
sa.terminal.connection.modify.enable.1=/ip firewall address-list add address=${ip()} list={acl}
# , ( ""),
sa.terminal.connection.modify.disable.1=/ip firewall address-list remove [/ip firewall address-list find
address=${ip()}]
sa.terminal.connection.modify.disable.2=/queue simple remove [/queue simple find name=("queue{@vlan}up")]
sa.terminal.connection.modify.disable.3=/queue tree remove [/queue tree find name=("queue{@vlan}down")]
sa.terminal.connection.modify.disable.4=/interface bridge filter remove [/interface bridge filter find out-
interface=({@interface})]

# , ID=x ( ID=x)
sa.terminal.connection.inetOption.1.enable.1=/interface bridge filter add action=mark-packet chain=output out-
interface=({@interface}) new-packet-mark=("{@speed}down")
sa.terminal.connection.inetOption.1.enable.2=/queue simple add name=("queue{@vlan}up") queue=("up-{@speed}M
/default-small") target=({@interface})
sa.terminal.connection.inetOption.1.enable.3=/queue tree add name=("queue{@vlan}down") packet-mark=("{@speed}
down") parent=({@interface}) queue=("down-{@speed}M")
# , ID=x ( ID=x)
sa.terminal.connection.inetOption.1.disable.1=/queue simple remove [/queue simple find name=("queue{@vlan}up")]
sa.terminal.connection.inetOption.1.disable.2=/queue tree remove [/queue tree find name=("queue{@vlan}down")]
sa.terminal.connection.inetOption.1.disable.3=/interface bridge filter remove [/interface bridge filter find
out-interface=({@interface})]

```

СКАТ DPI BRAS

В СКАТ DPI используются RADIUS-атрибуты вендора, которые необходимо добавить в dictionary.xml InetAccess и InetAccounting, если они там отсутствуют:

```
<vendor name="VasExperts" code="43823">
  <attribute name="VasExperts-Policing-Profile" code="1" type="string" />
  <attribute name="VasExperts-Service-Profile" code="2" type="string" />
  <attribute name="VasExperts-Enable-Service" code="3" type="string" />
  <attribute name="VasExperts-Multi-IP-User" code="4" type="octets" />
  <attribute name="VasExperts-UserName" code="5" type="string" />
  <attribute name="VasExperts-Service-Type" code="6" type="integer" />
  <attribute name="VasExperts-Restrict-User" code="7" type="integer" />
  <attribute name="VasExperts-DHCP-RelayRemoteId" code="38" type="octets" />
  <attribute name="VasExperts-DHCP-RelayCircuitId" code="39" type="octets" />
  <attribute name="VasExperts-DHCP-Option" code="40" type="octets" />
  <attribute name="VasExperts-DHCP-DNS" code="41" type="ipaddr" />
  <attribute name="VasExperts-DHCP-Gateway" code="42" type="ipaddr" />
  <attribute name="VasExperts-BOOTP-SName" code="43" type="string" />
  <attribute name="VasExperts-BOOTP-File" code="44" type="string" />
</vendor>
```

- **VasExperts-Service-Profile** - имя профиля услуги fastDPI,
- **VasExperts-Enable-Service** - включение/выключение услуги по ее коду,
- **VasExperts-Policing-Profile** - имя профиля policy,
- **VasExperts-Restrict-User** - флаг блокировки абонента.

Коды услуг СКАТ DPI:

- 1 - бонусная программа
- 2 - реклама
- 3 - блокировка рекламы
- 4 - черный список
- 5 - белый список
- 6 - уведомление
- 7 - кэширование
- 8 - пройдена ddos защита
- 9 - сбор netflow статистики для биллинга
- 10 - услуга защиты от DDOS
- 11 - CGNAT

Например, для того, чтобы включить отправку Netflow для сессии, в Access-Accept (radius.realm.default.attributes=) необходимо выдать "**VasExperts-Enable-Service=9:on**":

Конфигурация типа устройства

```
radius.realm.default.attributes=Acct-Interim-Interval=900;VasExperts-Enable-Service=9:on
```

Если для услуги необходимо включить настроенный профиль, то указываем **VasExperts-Service-Profile=<код услуги>:<имя профиля услуги>**:

Конфигурация типа устройства

```
radius.realm.default.attributes=Acct-Interim-Interval=900;VasExperts-Enable-Service=9:on;VasExperts-Service-Profile=11:nat_profile;VasExperts-Service-Profile=5:white_list
```

Скорость регулируем, указывая имя профиля policy: **VasExperts-Policing-Profile=rate_10M**.

InetAccess/InetAccounting не поддерживают запросы RADIUS Status-Server, поэтому в [конфигурации](#) СКАТ (radius-main.conf) необходимо указать status_server=0

- [Настройка типа устройства](#)
- [IPoE](#)
 - [IP Subscriber](#)
 - [DCHP-RADIUS Proxy](#)
 - [Статический IP-адрес](#)

- [PPPoE и IPoE одновременно](#)

Настройка типа устройства

Создайте тип устройства "СКАТ DPI", укажите CoAServiceActivator в поле "Обработчик активации сервисов" и CoAProtocolHandler в поле "Обработчик процессора протокола". В конфигурации типа устройства укажите:

Конфигурация типа устройства

```
#      ( ) : 0 ( ) - , 1 -      (
#      AGENT_REMOTE_ID AGENT_CIRCUIT_ID INTERFACE_ID), 2 - VLAN      (
#      AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID), 4 - VLAN
#      (      AGENT_REMOTE_ID AGENT_CIRCUIT_ID VLAN_ID),
# 5 - MAC-      (      MAC_ADDRESS), 6 - MAC-
#      (      MAC_ADDRESS).
#radius.servSearchMode=0
#      : 0 - ( , IPoE), 1 ( ) - .
#radius.password.verification=1

#      PoD CoA- ( - ,      /)
#radius.host=
#      PoD CoA- ( - ,      /)
radius.port=3799

#      Access-Accept      .
# H,      Reject-To-Accept      Start-
connection.start.fromAccept=1

#      suspended
connection.suspend.timeout=3600
#      c      (      connection.suspend.timeout)
connection.close.timeout=3600
#      , 0 ( ) - ,
# 1 -      NAS' (      connectionClose      )
#connection.close.timeout.forceClose=1

#      ,      default (default -      )
# Acct-Interim-Interval -      Accounting-Update      NAS'
radius.realm.default.attributes=Acct-Interim-Interval=900
#radius.realm.default.attributes=Acct-Interim-Interval=900;VasExperts-Enable-Service=9:on

#      ,      Access-Accept      disable (rejectToAccept)
radius.disable.accessCodes=
#      Access-Accept      disable:
# 0 ( ) -      radius.disable.ipCategories, 1 -      ,      (      )
#radius.disable.mode=0

#      ,      Access-Accept      disable
radius.disable.attributes=Acct-Interim-Interval=900;VasExperts-Restrict-User=1
#      ,      CoA
sa.radius.disable.attributes=VasExperts-Restrict-User=1
#      ,      CoA
sa.radius.enable.attributes=VasExperts-Restrict-User=0

# Id      ,      ,      ( : 1,      ).
#      ,      radius.disable.accessCodes      1
#radius.disable.servId=

#      - (1 - -,      -Service-Profile, 2 - -,      -Policing-Profile)
#      ,      serviceProfile=      policingProfile=
#radius.inetOption.1.template=VasExperts-Service-Profile=$serviceProfile;VasExperts-Policing-
Profile=$policingProfile
radius.inetOption.1.template=VasExperts-Service-Profile=$serviceProfile
radius.inetOption.2.template=VasExperts-Policing-Profile=$policingProfile

#      ,      CoA ( ,      ,      )
sa.radius.connection.attributes=Acct-Session-Id
```

```

# / ( , CoA)
sa.radius.connection.withoutBreak=1

# ServiceActivator ( , Access-Accept)
sa.radius.option.attributesPrefix=radius.inetOption.
# Access-Request. , NAS ( ) ,
# Stop- , .
# callingStationId : 1 - (connectionClose),
# 2 - (connectionClose) , , 3 - .
#radius.connection.checkDuplicate=0

# . - (0).
#radius.username.ignoreCase=0

#
# ,
#sa.error.pause=60
#
#sa.batch.size=20
# ( ) ( )
#sa.batch.wait=5
# ( )
#sa.batch.pause=0
# ( ) disconnect.
#sa.batch.waitNext=5

```

О выдаче IP-адреса и других параметрах при работе с RADIUS [читайте здесь](#).

IPoE

IP Subscriber

Схема аналогична Cisco IP Subscriber с инициацией сессии по IP-пакету - Unclassified source MAC address (For Layer 2-connected IP subscribers, a new IP session is triggered when an IP packet with an unclassified source MAC address (which means that an IP session does not yet exist for that MAC address) is received). Абонент получает динамический или статический адрес по **DHCP** от InetAccess-сервера биллинга, далее в InetAccess от СКАТ приходит Access-Request с **User-Name**=<IP-адрес абонента>, биллинг находит по указанному <IP-адрес абонента> и выдает параметры доступа в RADIUS-ответе СКАТ'у.

Для работы необходимо настроить [InetDhcpProcessor](#), чтобы InetAccess выступал DHCP-сервером и абонент получал IP-адрес от биллинга.

В inet-access.xml и inet-accounting.xml необходимо вместо:

```

<!-- C radius- -->
<bean name="radiusProcessor" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor"/>

```

указать:

```

<!-- C radius- -->
<bean name="radiusProcessor" class="ru.bitel.bgbilling.modules.inet.radius.InetRadiusHelperProcessor"/>

```

В конфигурации корневого устройства указать в параметре **radius.key.deviceTypeIds** ID типа устройства СКАТ DPI:

Конфигурация типа устройства

```
radius.key.deviceTypeIds=<ID DPI>
```

Чтобы отключить проверку пароля при RADIUS-аутентификации, в типе устройства "СКАТ DPI" укажите:

Конфигурация типа устройства

```
# : 0 - (, IPoE), 1 ( ) - .
radius.password.verification=0
```

DCHP-RADIUS Proxy

СКАТ может выступать как DHCP-Proxy, конвертируя DHCP-запросы от абонентов в RADIUS-запросы к биллингу:

RADIUS-запрос Access-Request имеет следующие атрибуты:

User-Name - MAC-адрес из DHCP-запроса в формате XX:XX:XX:XX:XX:XX. Для Q-in-Q-сетей в качестве User-Name возможно использовать QinQ-теги, см. ниже.

User-Password - значение fastprcf.conf-параметра dhcp_user_psw. Этот параметр задает пароль именно для DHCP Radius proxy режима. Если параметр не задан - используется параметр user_password Radius-сервера.

NAS-IP-Address - если DHCP-запрос содержит IP-адрес Relay-агента, то в данный атрибут подставляется этот адрес. Если Relay-агента нет - атрибут содержит виртуальный IP-адрес СКАТ из fastdpi.conf-параметра bras_arg_ip. По значению данного атрибута можно определить, из какой подсети пришел Radius-запрос (от какого Relay-агента).

NAS-Port-Type - содержит значение fastprcf.conf-параметра radius_attr_nas_port_type для данного Radius-сервера.

NAS-Port - только для VLAN-сетей (с одним VLAN): номер VLAN.

NAS-Port-Id: только для QinQ-сетей (с двойным VLAN): содержит VLAN-ы в строковом виде через '/', например: «123/67».

Framed-IP-Address - этот атрибут содержит IP-адрес абонента, присутствует только если IP-адрес абонента известен.

Session-Timeout – опциональный атрибут, задает время действия авторизации в секундах. Значение 0 игнорируется. По истечении этого времени статус авторизации пользователя устанавливается в "неизвестен", что приводит к отправке запроса на авторизацию Access-Request.

При этом в Access-Акцепт необходимо выдать User-Name, который далее будет передаваться в RADIUS-Accounting от СКАТ. Поэтому в типе сервиса необходимо установить галочку "логин", чтобы у сервиса было какое-то значение в поле **Логин** (автоматически сгенерированное или указанное вручную).

При использовании Q-in-Q СКАТ может передавать VLAN-ы в поле User-Name, поэтому в поле **Логин** можно просто указывать значение которое он присылает, например, 123.1000. Справочник ресурсов VLAN при этом не используется, проверяется только уникальность логинов.

```
# Access-Request
radius.connection.accessRequest=1

#
radius.servSearchMode=0

radius.realm.default.attributes=Acct-Interim-Interval=60;VasExperts-Enable-Service=9:on;MS-Primary-DNS-Server=8.8.8.8;MS-Secondary-DNS-Server=8.8.4.4
radius.realm.default.attributes.macros=User-Name=$login;VasExperts-DHCP-Gateway=$ipGate;Framed-IP-Netmask=$ipSubnetMask;VasExperts-DHCP-DNS=$ipDns
```

Если же в биллинге Вы указываете SVLAN в дереве устройств, а VLAN - в сервисе (используя VLAN-ресурсы), то можно настроить поиск по **SVLAN** (указанный в устройстве дерева) агентского устройства и по VLAN - сервиса договора (абонента). Для извлечения номера VLAN из Access-Request в типе устройства в поле **Обработчик процессора протокола** должен быть указан **CoAProtocolHandler**, а в RADIUS-Accounting будет использоваться поиск по логину:

```
# Access-Request
radius.connection.accessRequest=1

# VLAN, -
radius.servSearchMode=4,0

# , Access-Accept
#
radius.realm.default.attributes=Acct-Interim-Interval=60;Session-Timeout=86400;VasExperts-Enable-Service=9:on;MS-Primary-DNS-Server=8.8.8.8;MS-Secondary-DNS-Server=8.8.4.4
#
radius.realm.default.attributes.macros=User-Name=$login;VasExperts-DHCP-Gateway=$ipGate;Framed-IP-Netmask=$ipSubnetMask;VasExperts-DHCP-DNS=$ipDns

# VLAN
# vendor radius-
radius.vlan.vendor=-1
# (Nas-Port-Id)
radius.vlan.type=87
```

```
# ,
radius.vlan.prefix=
# regex
radius.vlan.pattern=^(\d+)/(\d+)

# SVLAN
radius.agentSvlan.vendor=87
radius.agentSvlan.type=-1
radius.agentSvlan.prefix=
radius.agentSvlan.pattern=^(\d+)/(\d+)
```

При использовании VLAN (без Q-in-Q) можно настроить поиск сервиса по VLAN (для извлечения номера VLAN из RADIUS-пакета в типе устройства в поле **Обработчик процессора протокола** должен быть указан **CoAProtocolHandler**):

```
# Access-Request
radius.connection.accessRequest=1

# VLAN, -
radius.servSearchMode=4,0

# , Access-Accept
#
radius.realm.default.attributes=Acct-Interim-Interval=60;Session-Timeout=86400;VasExperts-Enable-Service=9:on;
MS-Primary-DNS-Server=8.8.8.8;MS-Secondary-DNS-Server=8.8.4.4
#
radius.realm.default.attributes.macros=User-Name=$login;VasExperts-DHCP-Gateway=$ipGate;Framed-IP-
Netmask=$ipSubnetMask;VasExperts-DHCP-DNS=$ipDns

# VLAN
# vendor radius-
radius.vlan.vendor=-1
# (Nas-Port)
radius.vlan.type=5
# ,
#radius.vlan.prefix=
# regex
#radius.vlan.pattern=
```

При выдаче Access-Accept можно использовать [макросы \\$login, \\$identifier, \\$ipGate, \\$ipSubnetMask, \\$ipDns](#):

```
radius.realm.default.attributes.macros=VasExperts-DHCP-Gateway=$ipGate;Framed-IP-Netmask=$ipSubnetMask;
VasExperts-DHCP-DNS=$ipDns
```

В случае, если абонент сам назначил себе адрес, то в Access-Request придет с атрибутом Framed-IP-Address. Для того, чтобы биллинг проверил, что данный адрес есть в сервисе абонента и выдал в Access-Accept его же, укажите в конфигурации устройства:

```
# Access-Accept Framed-IP-Address Access-Request',
radius.address.fromRequest=1
# Framed-IP-Address Access-Request'
radius.address.fromRequest.check=1
```

Session-Timeout – опциональный атрибут, задает время действия авторизации в секундах. Значение 0 игнорируется. По истечении этого времени статус авторизации пользователя устанавливается в “неизвестен”, что приводит к отправке запроса на авторизацию Access-Request.

Для поддержки Access-Request для активного соединения в конфигурации необходимо указать:

```
# Access-Request
radius.connection.accessRequest=1
```

Статический IP-адрес

Если абонентам назначается статический IP-адрес, то в этом случае биллинг не обязательно должен быть DHCP-сервером (но может им быть и выдавать одни и те же адреса, указанные в сервисе договора). Адрес (или подсеть) должен быть указан в сервисе на договоре абонента. Для того, чтобы для сервисов работал поиск по IP-адресу, в конфигурации типа сервиса нужно указать:

Конфигурация типа устройства

```
# IP-  
serv.search.address=1
```

При добавлении этого параметра в уже существующий тип сервиса, требуется перезапуск Access и Accounting, чтобы уже созданные сервисы подгрузились и находились по их IP-адресу.

Чтобы при RADIUS-аутентификации [поиск происходил по IP-адресу](#), в конфигурации типа устройства SKAT DPI укажите:

Конфигурация типа устройства

```
# IP-  
radius.servSearchMode=7
```

PPPoE и IPoE одновременно

В этом режиме SKAT будет присылать RADIUS-пакеты для обоих режимов с одними и теми же NAS-Identifier и NAS-IP-Address, но скорее всего для разных типов доступа Вам нужно будет выдавать разные RADIUS-атрибуты и использовать разные настройки биллинга.

Чтобы решить эту проблему, мы предлагаем настроить два типа доступа как два разных устройства-NAS'a. Создайте три типа устройства, например, **FastDPI**, **FastDPI.IPoE** и **FastDPI.PPPoE**. Для типа устройства **FastDPI** укажите обработчик процессора протокола **SkatOverrideNasProtocolHandler**. Создайте устройство данного типа, укажите в поле **Хост** IP-адрес NAS'a, в поле **Идентификатор - NAS-Identifier** (например, **VasExperts.FastDPI**). Запрос будет приходить на это устройство, а далее, в зависимости от типа услуги обрабатываться устройством с **Идентификатором** равным идентификатору данного устройства плюс суффикс (.IPoE или .PPPoE).

Создайте два устройства-NAS'a, настройте их, как описано выше, но поле **Хост** в них оставьте пустым, а в поле **Идентификатор** укажите идентификатор с соответствующим суффиксом (.IPoE или .PPPoE). Т.е., если **Nas-Identifier** у нас **VasExperts.FastDPI**, то в устройстве типа **FastDPI** в поле Идентификатор указано **VasExperts.FastDPI**, в устройстве типа **FastDPI.IPoE** - **VasExperts.FastDPI.IPoE**, в устройстве типа **FastDPI.PPPoE** - **VasExperts.FastDPI.PPPoE**.

Т.к. поле **Хост** мы оставляем пустым (чтобы не было путаницы с одинаковыми NAS-IP при обработке RADIUS-пакета), чтобы CoA/PoD запросы отправлялись - в конфиге устройств **FastDPI.IPoE** и **FastDPI.PPPoE** нужно указать IP-адрес SKAT в параметре:

Конфигурация типа устройства

```
# PoD CoA- ( - , / )  
radius.host=x.x.x.x
```

Чтобы по RADIUS-Accounting атрибутам также можно было определить тип услуги, в Access-Акцепт устройства **FastDPI.IPoE** дополнительно выдаем атрибут class="IPoE":

```
#  
radius.realm.default.attributes=Class=IPoE;Acct-Interim-Interval=60;VasExperts-Enable-Service=9:on;MS-Primary-DNS-Server=8.8.8.8;MS-Secondary-DNS-Server=8.8.4.4
```

Теперь для биллинга запросы будут обрабатываться как от двух разных NAS'ов и настройки для них могут быть абсолютно разными. Устройствам **FastDPI.IPoE** и **FastDPI.PPPoE** в дереве не обязательно быть потомками устройства **FastDPI**.

Так как NAS-Identifier мы подменяем, а в CoA для SKAT он не нужен, указываем также:

```
# CoA,  
sa.radius.connection.attributes=NAS-Port, Acct-Session-Id, User-Name, Framed-IP-Address
```

RDP IPoE BRAS (EcoNAT)

EcoBRAS начинает новую сессию после переключения сервиса (скорости/доступа), текущий включенный сервис приходит в RADIUS-Accounting в атрибуте Cisco-Service-Info.

Создайте тип устройства Econat, укажите в поле **Обработчик активации сервисов**: ru.bitel.bgbilling.modules.inet.dyn.device.radius.CoAServiceActivator, а в поле **Обработчик процессора протокола**: ru.bitel.bgbilling.modules.inet.dyn.device.rdp.EconatProtocolHandler.

В конфигурации данного типа устройства укажите:

```
# PoD CoA- ( - , /)
radius.port=3799
# ( ) : IP- ( )
radius.servSearchMode=7
# : 0 - ( , IPoE), 1 ( ) - .
radius.password.verification=0
# Access-Request
radius.address.fromRequest=1
# Access-Accept disable:
# 0 ( ) - radius.disable.ipCategories, 1 - , ( )
radius.disable.mode=0

# Access-Accept .
# H, Reject-To-Accept Start-
connection.start.fromAccept=1
# suspended
connection.suspend.timeout=3600
# c ( connection.suspend.timeout)
connection.close.timeout=3600
# , 0 ( ) - ,
# 1 - NAS' ( connectionClose )
#connection.close.timeout.forceClose=1

# , default (default - )
# Acct-Interim-Interval - Accounting-Update NAS'
radius.realm.default.attributes=Acct-Interim-Interval=90;Idle-Timeout=4200
# - ,
radius.inetOption.1.template=cisco-SSG-Account-Info=QU; ;$in; ;D; ;$out

# Access-Accept disable:
# 0 ( ) - radius.disable.ipCategories, 1 - , ( )
radius.disable.mode=0
# , reject accept
radius.disable.accessCodes=1,2,3,4,10,11,12,20,44
# , Access-Accept disable
radius.disable.1,2,3,4,11,20,44,62.attributes=Acct-Interim-Interval=90;cisco-SSG-Service-Info=AserviceBlock;
Session-Timeout=86400
radius.disable.10,12.attributes=Acct-Interim-Interval=90;cisco-SSG-Service-Info=AserviceNomoney;Session-
Timeout=86400
radius.disable.attributes=Acct-Interim-Interval=90;cisco-SSG-Service-Info=AserviceBlock
# , CoA
sa.radius.disable.attributes=Acct-Interim-Interval=90;cisco-SSG-Service-Info=AserviceNomoney
# , CoA
sa.radius.enable.attributes=Acct-Interim-Interval=600;cisco-SSG-Service-Info=Aservice300m

# RADIUS-Accounting , ( )
radius.serviceName.disable=serviceNomoney,serviceBlock
# CoA/PoD- ( - RADIUS-)
sa.radius.log=1
# , CoA ( , Econat , )
sa.radius.connection.attributes=User-Name

# / ( , CoA)
sa.radius.connection.withoutBreak=1
# 1 InetAccess connectionModify
# CoAServiceActivator/ISGServiceActivator/SmartEdgeServiceActivator ( e.setConnectionStateModified( true ) ,
# CoAServiceActivator)
# Econat , .. CoA
sa.radius.connection.stateModify=0
```

```
#
# ,
#sa.error.pause=60
#
sa.batch.size=4
# ( ) ( )
sa.batch.wait=10
# ( )
#sa.batch.pause=0
# ( ) disconnect.
#sa.batch.waitNext=5
```

Статические IP-адреса

Если поиск абонента будет происходить по IP-адресу, указанному в сервисе договора, то в конфигурации типа сервиса нужно указать:

```
# IP- RADIUS-
serv.search.address=1
```

При изменении данного параметра в существующем типе сервиса, с которым сервисы договора уже заведены, требуется перезапуск InetAccess /InetAccounting.

Адреса, выданные по DHCP

Для того, чтобы биллинг находил абонента по IP-адресу, ранее выданному ему по DHCP (с помощью InetDhcpProcessor и, возможно, DHCP Option 82), нужно:

1. В inet-access.xml и inet-accounting.xml нужно заменить **radiusProcessor** с "ru.bitel.bgbilling.modules.inet.radius.InetRadiusProcessor" на "ru.bitel.bgbilling.modules.inet.radius.InetRadiusHelperProcessor";
2. В конфигурации корневого устройства (Access+Accounting) указать в параметре **radius.key.deviceTypeIds** ID типа устройства Esonat:

```
# IP- ( )
radius.key.deviceTypeIds=
```

Примеры настройки

управление через **TerminalServiceActivator**, статические диапазоны IP адресов на договорах, управление EsonAT (протоколом EsonAT)

```

12 #-----
13 # протокол подключения (telnet/ssh/tcp/mikrotik/shell/mikrotik-api)
14 sa.terminal.protocol=tcp
15 #команды завершения соединения
16 #sa.terminal.exit=quit
17 #Команды, выполняемые при подключении к терминалу( например configure terminal для cisco)
18 #sa.terminal.connect=statal
19 #Команды, выполняемые перед отключением от терминала.
20 #sa.command.disconnect=
21 sa.terminal.endSequence=
22 #-----
23 #Команды создания сервиса на устройстве.
24 sa.terminal.serv.create=ads ${contractId0}-${servId0} {oid} ${param(option(20,'speed','LIM10M/LIM10M')} ${ipList(',')}, // RULE43
25 #sa.terminal.serv.create.2=statal
26 #sa.command.serv.create.27=do write
27 #Команды включения сервиса на устройстве
28 sa.terminal.serv.modify.enable=ads ${contractId0}-${servId0} {oid} ${param(option(20,'speed','LIM10M/LIM10M')} ${ipList(',')}, // RULE43
29 #sa.terminal.serv.modify.enable.2=statal
30 #sa.command.serv.modify.enable.1=ads $contractId {oid} $param($option,speed,LIM10M/LIM10M) $ipList() // RULE43
31 #sa.command.serv.modify.enable.2=exit
32 #Команды удаления сервиса с устройства.
33 sa.terminal.serv.cancel=remove ${contractId0}-${servId0} {oid} ${param(option(20,'speed','LIM10M/LIM10M')} ${ipList(',')},
34 #sa.terminal.serv.cancel.2=statal
35 #sa.command.serv.cancel.1=remove $contractId {oid} $param($option,speed,LIM10M/LIM10M) $ipList()
36 #sa.command.serv.cancel.2=exit
37 #Команды выключения сервиса на устройстве
38 sa.terminal.serv.modify.disable=remove ${contractId0}-${servId0} {oid} ${param(option(20,'speed','LIM10M/LIM10M')} ${ipList(',')},
39 #sa.terminal.serv.modify.disable.2=statal
40 #
41 # команды, вызываемые при включении сервиса (состояние стало "включено")
42 sa.terminal.serv.modify.enable=ads ${contractId0}-${servId0} {oid} ${param(option(20,'speed','LIM10M/LIM10M')} ${ipList(',')}, // RULE43
43 #sa.terminal.serv.modify.enable.2=statal
44 #
45 # команды, вызываемые при отключении сервиса (состояние стало "отключено")
46 sa.terminal.serv.modify.disable=remove ${contractId0}-${servId0} {oid} ${param(option(20,'speed','LIM10M/LIM10M')} ${ipList(',')},
47 #sa.terminal.serv.modify.disable.2=statal
48 #####
49 #-----
50

```

опции скорости

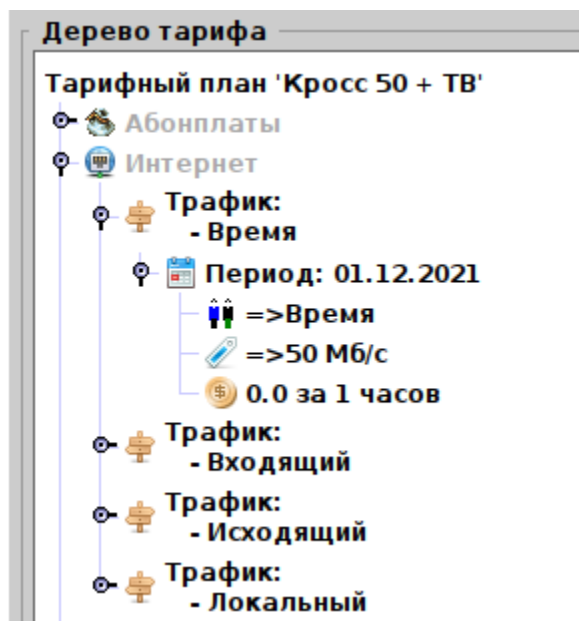
Типы сервисов	Трафик	Опции	Диапазоны трафиков
Название	Комментарий	ID	...
Опции		0	▲
Группа: Скорость		20	
1 Гб/с (14)		14	
10 Мб/с (6)		6	
100 Мб/с (1)		1	
15 Мб/с (8)		8	
20 Мб/с (15)		15	

Типы сервисов	Трафик	Опции	Диапазоны трафиков
Название	Комментарий	ID	...
Опции		0	▲
Группа: Скорость		20	
1 Гб/с (14)		14	
10 Мб/с (6)		6	
100 Мб/с (1)		1	

Редактор

Название	Конфигурация
1 Гб/с	<i>speed=LIM1000M/LIM1000M</i>
<input type="checkbox"/> Пересечение в группе возмож...	
Комментарий	
<input style="width: 100%;" type="text"/>	

тариф



Интеграция с модулем Card

В модуле Inet возможна интеграция с модулем Card для автоматической активации карт при первой авторизации по протоколу RADIUS.

Каждая карточка имеет три параметра: серийный номер (уникальный, не должен повторяться), логин и пароль. При первом подключении по логину и паролю карты после попытки стандартной аутентификации, InetRadiusProcessor запросит поиск карты с таким логином у модуля Card. Если карта будет найдена, ее статус и время действия будут активны, активация разрешена, а пароль совпадет, то модуль Card создаст договор, на основе [шаблона договора](#), указанного в карте.

В шаблон договора нужно [добавить модуль Inet с созданием сервиса](#) и указать тип сервиса, с которым будет создан сервис, а также статус по умолчанию и максимальное кол-во активных сессий. Если в конфигурации выбранного [типа сервиса](#) присутствует параметр **const.device.id**, то созданный при активации карты сервис будет привязан к указанному устройству, иначе он будет привязан к устройству, с которого происходила активация карты (первое подключение).

В конфигурации модуля или же конфигурации устройства нужно указать параметры:

- **card.moduleId** - код модуля Card
- **card.activate.serviceIds** - id разрешенных услуг активации модуля Card. Услуга активации привязана к карте, и если услуга активации карты отсутствует в данном параметре, то карта активирована не будет. Если в параметре указано значение 0, то карты с любыми услугами активации могут быть активированы.
- **card.login.min** - минимальное значение карточного логина, указывается для того, чтобы поиск карты не выполнялся для любого не найденного цифрового логина. Если указано 0 (по умолчанию), то ограничение не действует.
- **card.login.max** - максимальное значение карточного логина, указывается для того, чтобы поиск карты не выполнялся для любого не найденного цифрового логина. Если указано 0 (по умолчанию), то ограничение не действует.

Пример:

```
# card InetRadiusProcessor.
#
# card
#card.moduleId=
# id
#card.activate.serviceIds=
# , , ;
# 0, .
#card.login.min=0
# , , ;
# 0, .
#card.login.max=0
```

Так как при активации карты создаются сервисы с цифровым логином, то для того, чтобы обычные логины не пересекались и, при создании сервиса с автоматическим генерированием логина, сгенерированный логин не попал в диапазон логинов карточек, в конфигурации модуля необходимо правильно указать:

```
#
#
#serv.login.min=1
# (.. 1,2,3 10000000,
# 4, 10000001)
#serv.login.max=9999999
```

FAQ модуля Inet

- Не запускается Access и/или Accounting
- Access/Accounting не обновляются через update.sh
- Соединение с ограниченным доступом (Reject-To-Accept) подключается, но почти сразу отключается или постоянно меняется ограничение соединения
- Абонентов не пускает, команды на коммутаторах не выполняет
- Удаление определенных сообщений из очереди ActiveMQ
- В ActiveMQ копятся и в итоге не обрабатываются сообщения
- Не совпадает состояние сервиса Inet с тем, что должно быть
- Все сессии сбросились на NAS'e, при массовом переподключении - проблемы
- Свободный IP-адрес не найден или Can't reserve ip address
- Access и/или Accounting потребляют много памяти
 - Access и/или Accounting потребляют много памяти постепенно после старта
 - Accounting потребляет много памяти сразу после старта
- Как помочь разработчикам быстрее исправить ошибку
- Как SQL запросом посмотреть IP-адрес, MAC-адрес из inet_serv varbinary(64)?
- Невозможно удалить сервис с активным периодом действия
- Создание Web-сервиса в динамическом коде
- При работе по DHCP и статическом адресе абонент иногда получает динамический адрес
- При работе по DHCP и статическом адресе абонент иногда получает динамический адрес
- Ошибка InetDhcpProcessor - Timeout exceed!
- Пакеты из некоторых сетей не доходят до слушателя radius или dhcp
- Удаление старых данных из БД
- При CoA-запросе переключения ISG-сервиса Cisco отвечает 405 Config apply failed
- При добавлении IP-адреса выдает ошибку "Подходящий IP-ресурс не найден"

Не запускается Access и/или Accounting

Вы только установили Access/Accounting и в access.out или accounting.out ошибка java.lang.ClassNotFoundException, например, для Access:

```
Error on node access
java.lang.ClassNotFoundException: ru.bitel.bgbilling.modules.inet.access.Access
    at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:190)
    at ru.bitel.bgbilling.kernel.application.server.Application.processBean(Application.java:391)
    at ru.bitel.bgbilling.kernel.application.server.Application.process(Application.java:297)
    at ru.bitel.bgbilling.kernel.application.server.Application.processChildren(Application.java:749)
    at ru.bitel.bgbilling.kernel.application.server.Application.application(Application.java:225)
    at ru.bitel.bgbilling.kernel.application.server.Application.<init>(Application.java:161)
    at ru.bitel.bgbilling.kernel.application.server.Application.main(Application.java:803)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at ru.bitel.common.bootstrap.Boot.boot(Boot.java:129)
    at ru.bitel.common.bootstrap.Boot.main(Boot.java:178)
```

Решение

В документации по установке Access/Accounting указан пункт "Обновите как обычные серверные приложения биллинга", т.е. запустите update.sh, который находится в директории биллинга.

Access/Accounting не обновляются через update.sh

При вызове update.sh у Access/Accounting происходит ошибка TimeoutException или NullPointerException:

```
java.lang.NullPointerException
    at bitel.billing.server.installer.library.LibraryUpdate.getLibrariesForUpdate(LibraryUpdate.java:94)
    at bitel.billing.server.installer.library.LibraryUpdate.main(LibraryUpdate.java:119)
```

Решение

Проверьте, что BGBillingServer запущен.

Если Access/Accounting и BGBillingServer запущены на разных машинах - проверьте что время на этих машинах синхронизировано.

Соединение с ограниченным доступом (Reject-To-Accept) подключается, но почти сразу отключается или постоянно меняется ограничение соединения

Обычно это происходит из-за того, что Accounting не знает, в каком состоянии соединение, "подключен" или "отключено", т.е. полный доступ или ограниченный (например, редирект). За это отвечают несколько параметров:

```
# 1 Access-Accept InetAccess      ( )
connection.start.fromAccept=1
# Accounting- ( )
radius.disable.pattern.attributes=
# 1 InetAccess      connectionModify CoAServiceActivator/ISGServiceActivator/SmartEdgeServiceActivator
( e.setConnectionStateModified( true ), CoAServiceActivator)
sa.radius.connection.stateModify=1
# ISG, SmartEdge      ISG/SmartEdge ( )
radius.serviceName.disable=
```

Рекомендуемые параметры для CoAServiceActivator:

```
connection.start.fromAccept=1
sa.radius.connection.stateModify=1
```

Рекомендуемые параметры для ISGServiceActivator/SmartEdgeServiceActivator с посервисным аккаунтингом:

```
connection.start.fromAccept=1
sa.radius.connection.stateModify=0
radius.serviceName.disable=<( ) , >
```

Абонентов не пускает, команды на коммутаторах не выполняет

Возможно в ActiveMQ накопилось большое кол-во сообщений и нет возможности их корректно обработать. Проверьте, если в /opt/activemq/data/kahadb большое кол-во файлов *.log (например, больше 5-ти), то скорее всего причина в [этом](#) или [этом](#).

Удаление определенных сообщений из очереди ActiveMQ

Для того, чтобы через командную строку удалить сообщения, необходимо, чтобы в ActiveMQ был включен JMX. Для этого в conf/activemq.xml должен был пункт managementContext с createConnector="true"

```
<managementContext>
  <managementContext createConnector="true"/>
</managementContext>
```

После изменения необходимо перезапустить ActiveMQ.

Если JMX включен, то команда `activemq-admin purge` позволяет удалить определенные сообщения из очереди. Например, чтобы удалить сообщения о необходимости изменения состояния сервисов для устройства с ID=55, нужно выполнить:

```
./activemq-admin purge --msgsel "deviceId=55" BG.Event.ru.bitel.bgbilling.modules.inet.access.sa.event.InetSaStateModifyEvent
```

<https://activemq.apache.org/activemq-command-line-tools-reference.html#ActiveMQCommandLineToolsReference-purge>

В ActiveMQ копятся и в итоге не обрабатываются сообщения

Для того, чтобы быстрее узнать, почему ActiveMQ копит сообщения лучше включить web-консоль ActiveMQ. Для этого в `/opt/activemq/conf/activemq.xml` раскомментируйте ветку `<import resource="jetty.xml"/>` и перезапустите ActiveMQ.

```
<!-- Enable web consoles, REST and Ajax APIs and demos It also includes Camel (with its web console), see
${ACTIVEMQ_HOME}/conf/camel.xml for more info
    Take a look at ${ACTIVEMQ_HOME}/conf/jetty.xml for more details --
>

<import resource="jetty.xml"/>
```

Проследите, чтобы порт не оказался открыт наружу.

Откройте в браузере web-консоль <http://адрес:8161/admin>, перейдите в **Queues. Number Of Pending Messages** - это кол-во сообщений, которые ActiveMQ получил, но еще не успел или не смог передать приложениям биллинга. При нормальной работе среднее значение 0 или небольшое, т.е. при увеличении очень быстро уменьшается обратно.

Если web-консоль не открывается, при этом `activemq.xml` поправили корректно и порт 8161 открыт - возможно проблема в дистрибутиве `activemq` (старая версия ActiveMQ), попробуйте заменить его.

Если значения **Number Of Pending Messages** у каких-либо очередей большие и не уменьшаются, то на это может быть несколько причин.

Это может быть очередь скриптовых событий (Функции скриптов поведения), возможно:

- в БД (`script_event_type`) по какой-то причине нет информации об этом событии и сервер его не слушает (обычно большая очередь у одного-двух скриптовых событий);
- какой-то скрипт/класс поведения работает слишком медленно (у всех скриптовых событий есть какое-то количество сообщений в очереди).

Это могут быть очереди заданий синхронизации `Inet`, возможно:

- произошел какой-то сбой (например, ниже), очередь переполнилась и теперь не может отработать корректно;
- одно из устройств (NAS' или коммутатор) очень долгое время не отвечало при попытках синхронизации, таким образом сообщения для него постепенно копились и см. первую причину; нужно посмотреть логи `Access`;
- `Access` долгое время не был запущен и см. первую причину;
- конфигурация `Access` некорректна, например, значения `rootDeviceId` или `accounting.deviceTypeIds` неправильные;
- ошибка в `ServiceActivator` при синхронизации с NAS'ом или коммутатором, нужно смотреть логи `Access` (возможно нужно перекомпилировать динамические классы);
- одно из устройств удалили через БД, но к нему были привязаны активные сервисы `Inet`, в итоге `Accounting` генерирует сообщения, которые `Access` не может обработать.

Можно открыть конкретную очередь в web-консоли и посмотреть, какие там сообщения, для каких устройств (поле `deviceId`).

Решение (Inet)

1. Если в `Access` постоянные ошибки с синхронизацией для всех устройств (например, нет подключения к коммутаторам), попробуйте их исправить.
2. Попробуйте перезапустить `Access` и `Accounting`.
3. Если после перезапуска `Access` не начал выполнять синхронизацию или начал выполнять нормально, но очень и очень медленно, т.е. **Number Of Pending Messages** не уменьшается, можно попробовать очистить очередь, нажав на ссылку `Purge` для очереди в web-интерфейсе.
4. Если `Purge` не выполняется вообще или выдает ошибку (старые версии ActiveMQ), можно попробовать удалить очередь через ссылку `Delete`, но после успешного удаления очереди или очередей нужно будет перезапустить `BGBillingServer`, `BGInetAccess` и `BGInetAccounting`.
5. Если `Delete` тоже не выполняется или web-интерфейс не открывается (старые версии ActiveMQ), можно попробовать очистить данные `activeMQ`. Для этого нужно остановить `BGBillingServer`, `Accounting`, `Access`, `ActiveMQ`, переименовать папку `/opt/activemq/data/kahadb` (чтобы осталась резервная копия), запустить `ActiveMQ`, `BGBillingServer`, `Access`, `Accounting`.
6. Если после очистки очередь все равно набирается - возможно проблемы с конфигурацией `Access`.
7. Если с `Access` все в порядке, а за ActiveMQ замечены еще какие-то проблемы - возможно проблема в дистрибутиве `activemq` (например, файлы дистрибутива повреждены), попробуйте заменить его.

Таким образом, можно быстро решить проблему, очистив очередь одним из способов, но это, возможно, будет временное решение, если существует исходная не решенная проблема, из-за которой растет кол-во сообщений в очереди - нужно будет ее найти.

Не совпадает состояние сервиса Inet с тем, что должно быть

После сбоя или после ручного добавления платежей в БД состояние сервисов может не совпадать с необходимым. При этом при пересохранении сервиса состояние меняется на нормальное. Если это произошло по неизвестной причине, следует ее выяснить. Например, при проблеме с обработкой сообщений (см. **В ActiveMQ копятся и в итоге не обрабатываются сообщения**) необходимо сначала устранить ее. Или есть скрипт добавления платежей, который не создает необходимые события - необходимо сначала его исправить.

Для того, чтобы заставить биллинг синхронизировать состояние сервисов согласно балансу нужно добавить в "Глобальные скрипты поведения" - "Классы Java" класс `ru.bitel.bgbilling.modules.inet.dyn.access.InetServStateSync`. Перед запуском этого скрипта нужно перезапустить `BGBillingServer`, чтобы синхронизация гарантировано запустилась для всех договоров.

Все сессии сбросились на NAS'е, при массовом переподключении - проблемы

Если произошел сбой питания или по какой-то другой причине сессии на NAS'е были сброшены - в этой ситуации сессии остаются в биллинге, т.к. от NAS'а не пришел RADIUS-stop-пакет. При этом адрес, привязанный к сессии считается занятым. По умолчанию сессия не будет закрыта, пока не выйдет `connection.close.timeout`. Также следует учитывать, что перезапуск `InetAccounting` приведет к сбросу последней активности на время запуска `InetAccess` (т.к. в другой ситуации пакеты могли приходиться пока `InetAccess` был не запущен). При этом могут возникнуть проблемы, что абонентов не пускает по ограничению кол-ва сессий, либо происходит ошибка "Свободный IP-адрес не найден".

Такие повисшие сессии можно убрать двумя способами:

- С помощью `radius.connection.checkDuplicate`, чтобы при новых Access-Request от одного и того же пользователя старую сессию закрывать в биллинге. Желательно чтобы и при обычной работе этот параметр был указан (но возможно понадобится другое значение, чем при такой аварии).
- С версии 7.0 для устройства в дереве устройств доступен пункт "Завершить сессии в БД" с указанием минимальной длительности сессий, которые нужно завершить.
- Уменьшить в конфигурации устройства `connection.close.timeout` и `connection.suspend.timeout` до минуты или даже 5 секунд, указать в конфигурации корневого устройства более частую [обработку завершения сессий](#):

```
#
accounting.worker.3.finishing.1.delay=10
#
accounting.worker.3.finishing.1.batchSize=5000
```

Перезапустить `InetAccounting`, затем после того как сессии будут завершены в биллинге и пропадут из монитора вернуть правильные значения `connection.close.timeout` и `connection.suspend.timeout` и нажать "Перечитать конфигурацию на серверах". Позже можно вернуть сброс сессий в БД (finishing) в прежние значения (для этого понадобится перезапуск `InetAccounting`).

Также следует учитывать, что при выдаче Access-Асепт вместе с ним выдается IP-адрес, который резервируется на минуту. Таким образом, если в `inet-access.xml` для `radiusListener` указано большое значение очереди `maxQueueSize` возможна ситуация, что `InetAccess` будет выдавать Access-Асепт, который абонент уже перестал ждать и отправил новый Access-Request. В этом случае адреса будут выдаваться в никуда, это также может быть причиной ошибки "Свободный IP-адрес не найден", а также задержке восстановления до нормальной работы. Поэтому если `maxQueueSize` больше 200, укажите 100-200 и перезапустите `InetAccess`. С версии 6.0 `InetAccess` старается не обрабатывать пакеты из очереди, которые он считает устаревшими, поэтому вероятность возникновения проблемы из-за `maxQueueSize` меньше.

Свободный IP-адрес не найден или Can't reserve ip address

Если произошел сброс сессий на NAS'е - смотрите выше.

Если назначенных ресурсов IP-адресов должно хватать, посмотрите текущую информацию об IP-ресурсах в выводе команды `./accounting.sh status'`.

Проверьте, что таблицы в базе данных у вас в InnoDB, а не MyISAM:

```
SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA="bgbilling" AND ENGINE="MyISAM"
```

Если в MyISAM (мы используем InnoDB с 5.2, информация об этом есть в документации и инструкции по обновлению) - сконвертируйте с помощью утилит: http://wiki.bitel.ru/index.php/___MySQL

Далее, даже если таблицы в InnoDB - проверьте целостность данных

```
SELECT INET_NTOA(CONV(HEX(s.address), 16, 10)), s.* FROM inv_ip_resource_subscription_dyn_<mid>_<yyyyMM> as s
LEFT JOIN inet_connection_<mid> as c ON s.ipResourceId=c.ipResourceId AND s.address=c.ipAddress AND c.status=1
WHERE s.timeTo IS NULL AND c.id IS NULL
```

В ууууММ сначала укажите текущий месяц, затем выполните запрос для предыдущих. Если в выводе много строк, возможно какая-то ошибка (в билдах до июля 2014 г. в некоторых случаях могла возникнуть такая проблема) и биллинг считает эти адреса занятыми. Проверьте, действительно ли эти адреса свободны. Если да и записей набирается как раз, чтобы занять до конца пул IP-адресов, попробуйте завершить эту подписку (осторожно!):

```
UPDATE inv_ip_resource_subscription_dyn_<mid>_<yyyyMM> as s
LEFT JOIN inet_connection_<mid> as c ON s.ipResourceId=c.ipResourceId AND s.address=c.ipAddress AND c.status=1
SET s.timeTo=NOW()
WHERE s.timeTo IS NULL AND c.id IS NULL
```

Затем перезапустите InetAccess и InetAccounting, перед этим сохранив их логи для дальнейшего разбирательства.

Access и/или Accounting потребляют много памяти

Access и/или Accounting потребляют много памяти постепенно после старта

Возможно задействовано много источников логов (NAS'ы, которые посылают пакеты RADIUS, коммутаторы, которые посылают DHCP-пакеты, flow-агенты). При получении, обработки и записи DHCP/Radius/Netflow/sFlow пакетов используются буферы. Максимальное кол-во памяти, которые могут забрать буферы - $\text{threadCount} * \text{datalog.chunk.size} * (\text{кол-во устройств-источников данных})$, где threadCount - кол-во потоков слушателя (InetRadiusListener/DhcpListener/InetFlowListener).

Решение

В этом случае не стоит указывать большое кол-во потоков для слушателя threadCount и при большом кол-ве источников логов имеет смысл уменьшить chunk.size:

```
# , ,
datalog.chunk.size=131072
# DHCP
datalog.dhcp.chunk.size=65536
# RADIUS
datalog.radius.chunk.size=65536
# Netflow/sFlow
datalog.flow.chunk.size=262144
```

Эти параметры указываются в inet-access.xml и inet-accounting.xml. Пример для inet-accounting.xml:

```
<!-- radius- -->
<!-- , radius -->
<param name="datalog.radius.dir" value="data/radius" />
<!-- , -->
<param name="datalog.radius.chunk.size" value="524288" />
<!-- radius : 0 - , 1 - zlib -->
<param name="datalog.radius.compression.type" value="1" />
<!-- flow- -->
<!-- , flow -->
<param name="datalog.flow.dir" value="data/flow" />
<!-- , -->
<param name="datalog.flow.chunk.size" value="524288" />
<!-- flow : 0 - , 1 - zlib -->
<param name="datalog.flow.compression.type" value="1" />
```

Accounting потребляет много памяти сразу после старта

При использовании InetFlowListener возможна схема, когда будет множество устройств-коммутаторов с интерфейсами, которые никак не связаны с Flow-агентом и поэтому у них не указан в конфигурации flow.agent.link. Исторически сложилось, что если параметр flow.agent.link не указан, то устройство само считается Flow-агентом с указанными интерфейсами. Для экономии памяти рекомендуется указать для таких устройств flow.agent.link={@deviceId}-1.

Как помочь разработчикам быстрее исправить ошибку

- Предоставить более полный лог ошибки (исключения), со всеми Caused, а не только первую строчку, например:

```
05-20/12:22:29 ERROR [ "http-bio-/0.0.0.0-8080"-exec-5 ] AbstractJaxWsHandler -
java.util.concurrent.ExecutionException: java.security.PrivilegedActionException: java.lang.Exception:
java.lang.NullPointerException
    at java.util.concurrent.FutureTask$Sync.innerGet(FutureTask.java:232)
    at java.util.concurrent.FutureTask.get(FutureTask.java:91)
    at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener$ThreadedScriptEventListener.
runScriptImpl(DynamicScriptEventListener.java:267)
    at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener.runScript
(DynamicScriptEventListener.java:149)
    at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener.notify
(DynamicScriptEventListener.java:117)
    at ru.bitel.bgbilling.kernel.event.LocalEventProcessor.request(LocalEventProcessor.java:240)
    at ru.bitel.bgbilling.kernel.event.EventProcessor.request(EventProcessor.java:848)
    at ru.bitel.bgbilling.kernel.event.EventProcessor.request(EventProcessor.java:817)
    at bitel.billing.server.contract.action.ActionUpdateListParam.doAction(ActionUpdateListParam.java:35)
    at bitel.billing.server.Executer.doModule(Unknown Source)
    at bitel.billing.server.Executer$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:396)
    at bitel.billing.server.Executer.doPost(Unknown Source)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:641)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:722)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:304)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)
    at bitel.billing.server.filters.SetCharacterEncodingFilter.doFilter(SetCharacterEncodingFilter.java:
48)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:243)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)
    at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:240)
    at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:164)
    at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:462)
    at org.apache.catalina.valves.AccessLogValve.invoke(AccessLogValve.java:563)
    at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:164)
    at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:100)
    at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:118)
    at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:403)
    at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:301)
    at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:162)
    at org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:309)
    at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
    at java.lang.Thread.run(Thread.java:662)
Caused by: java.security.PrivilegedActionException: java.lang.Exception: java.lang.NullPointerException
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:396)
    at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener$ThreadedScriptEventListener$1.
call(DynamicScriptEventListener.java:241)
    at java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:303)
    at java.util.concurrent.FutureTask.run(FutureTask.java:138)
    at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
    at java.lang.Thread.run(Thread.java:662)
    at ru.bitel.common.worker.WorkerThread.run(WorkerThread.java:40)
Caused by: java.lang.Exception: java.lang.NullPointerException
    at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener.runScriptImpl
(DynamicScriptEventListener.java:200)
    at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener$ThreadedScriptEventListener.
access$0(DynamicScriptEventListener.java:1)
```



```
at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener$ThreadedScriptEventListener$1$1.run(DynamicScriptEventListener.java:247)
... 9 more
Caused by: java.lang.NullPointerException
at ru.bitel.bgbilling.kernel.script.server.DynamicScriptEventListener.runScriptImpl(DynamicScriptEventListener.java:196)
... 11 more
```

- Сохранить лог-файлы на момент ошибки
- Сохранить информацию по стекам с помощью утилиты jstack из JDK:

```
jstack <id_> > stack1.txt
```

- Если проблема сложная или связана с излишним потреблением памяти - сделать дампы памяти процесса (может быть затратно по ресурсам) с помощью утилиты jmap из JDK (дампы памяти нельзя выкладывать в открытый доступ). Данная операция может быть ресурсозатратна, особенно на приложениях, которые заняли более 1GB оперативной памяти:

```
jmap -dump:format=b,file=dump.hprof <id_>
```

Перед снятием дампа рекомендуется вызвать garbage collector:

```
accounting.sh gc
```

Как SQL запросом посмотреть IP-адрес, MAC-адрес из inet_serv varbinary (64)?

MAC:

```
SELECT HEX(macAddress) FROM inet_serv
```

IP:

```
SELECT INET_NTOA(CONV(HEX(ipAddress), 16, 10)) FROM inet_connection
SELECT * FROM inet_connection WHERE ipAddress=UNHEX(CONV(INET_ATON('10.0.0.1'), 10, 16))
```

Невозможно удалить сервис с активным периодом действия

Нельзя удалять сервис, период действия которого пересекается с сегодняшним днем. Нужно закрыть сервис датой, например, хотя бы вчерашним днем, сохранить, потом удалить.

Создание Web-сервиса в динамическом коде

Тут создается Web-сервис InetDeviceService. Web-сервис для любого другого класса получается по аналогии. INET_MODULE_ID - код экземпляра модуля inet.

```
ServerContext context = ServerContext.get();
InetDeviceService wsDevice = context.getService( InetDeviceService.class, INET_MODULE_ID );
```

При работе по DHCP и статическом адресе абонент иногда получает динамический адрес

Скорее всего это происходит, когда абонент подключает другое устройство. Появляется новый MAC, биллинг не может технически определить, что предыдущая сессия уже закончилась, только по таймауту DHCP-lease, поэтому считает адрес еще занятым и выдает адрес из динамического пула. При схеме один порт/VLAN - один компьютер/роутер поможет параметр конфигурации устройства `Inet dhcp.connection.closeOnNew=1`. С этим флагом при появлении DHCP Discover с того же порта/VLAN с другим MAC-адресом предыдущая сессия будет завершена, таким образом адрес освободится и для новой сессии будет выдан он же.

При работе по DHCP и статическом адресе абонент иногда получает динамический адрес

Скорее всего это происходит, когда абонент подключает другое устройство. Появляется новый MAC, биллинг не может технически определить, что предыдущая сессия уже закончилась, только по таймауту DHCP-lease, поэтому считает адрес еще занятым и выдает адрес из динамического пула. При схеме один порт/VLAN - один компьютер/роутер поможет параметр конфигурации устройства `Inet dhcp.connection.closeOnNew=1`. С этим флагом при появлении DHCP Discover с того же порта/VLAN с другим MAC-адресом предыдущая сессия будет завершена, таким образом адрес освободится и для новой сессии будет выдан он же.

Ошибка InetDhcpProcessor - Timeout exceed!

В логах InetAccess есть ошибки при обработке DHCP-запросов:

```
dhcp 01-10/21:33:31 ERROR [dhcpLstnr-p-10-t-11] InetDhcpProcessor - Timeout exceed!
ru.bitel.bgbilling.common.BGException: Timeout exceed!
    at ru.bitel.bgbilling.modules.inet.access.InetConnectionManager.accountingStart(InetConnectionManager.
java:576)
    at ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor.processOption82RequestImpl0(InetDhcpProcessor.
java:757)
    at ru.bitel.bgbilling.modules.inet.dhcp.InetDhcpProcessor.processOption82RequestImpl(InetDhcpProcessor.
java:216)
    at ru.bitel.bgbilling.modules.inet.dhcp.InetAbstractDhcpProcessor.processOption82Request
(InetAbstractDhcpProcessor.java:504)
    at ru.bitel.bgbilling.modules.inet.dhcp.InetAbstractDhcpProcessor.processRequest
(InetAbstractDhcpProcessor.java:253)
    at ru.bitel.bgbilling.kernel.network.dhcp.DhcpListenerWorker.runImpl(DhcpListenerWorker.java:89)
    at ru.bitel.common.worker.WorkerTask.run(WorkerTask.java:86)
    at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:895)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:918)
    at java.lang.Thread.run(Thread.java:662)
    at ru.bitel.common.worker.WorkerThread.run(WorkerThread.java:40)
```

Решение

- Возможно, проблемы с activemq, проверьте пункт **В ActiveMQ копятся и в итоге не обрабатываются сообщения**.
- Если activemq, InetAccess или InetAccounting находятся на разных машинах, убедитесь, что время между ними **синхронизовано**. Проверьте еще раз, расхождения на секунды достаточно для создания проблем (это же все-таки биллинг).
- Если и там все в порядке, а это первичная установка или недавно менялась конфигурация сети или серверов убедитесь, что имя сервера с ActiveMQ указано в файле `/etc/hosts`. Имя сервера можно получить командой `uname -n`. [ActiveMQ-сервер, http://forum.bitel.ru/viewtopic.php?f=44&t=10031](http://forum.bitel.ru/viewtopic.php?f=44&t=10031)

Пакеты из некоторых сетей не доходят до слушателя radius или dhcp

Иногда бывает что до InetAccess не доходят dhcp-запросы, при это на этой же машине tcpdump их видит(даже может работать стандартный isc сервер если его запустить на том же порте) . Первое что нужно проверить - это настройка файервола.

И одна из возможных причин изложена ниже.

В сетевых подсистемах есть RPF (Reverse Path Filtering)

<http://xgu.ru/wiki/RPF>

Смысл в том, что к серверу прилетает запрос например из сети 10.1.1.0/24 и для сервера этой сети нет в таблицы маршрутизации. Он думает что его надо отправить по маршруту по умолчанию.

Но прилетел он с другой сетевой карты, не там где находится маршрут по умолчанию. Поэтому пакет не проходит проверку.

Есть 2 выхода из положения:

1. Добавить маршрут к сети, кто делает релей, через тот интерфейс, через который все прилетает.
2. Отключить проверку на том интерфейсе, с которого прилетает пакет.
<https://serverfault.com/questions/816393/disabling-rp-filter-on-one-interface>

Удаление старых данных из БД

Можно удалять месячные таблицы за предыдущие месяцы (например, старше двух месяцев):

- `inet_auth_error_x_yyyyMM` - таблица, которая используется для отображения ошибок на вкладке Монитор;
- `connection_log_entry_x_yyyyMM` - таблица, которая используется для получения RADIUS/DHCP-логов при нажатии правой кнопкой мыши на сессии.

Если не нужна история сессий за предыдущие месяцы (например, старше полугода), то можно удалить:

- `inet_session_log_x_yyyyMM` - сессии абонентов
- `inet_session_log_detail_x_yyyyMM` - информация по трафикам сессий (тип трафика - кол-во)
- `inet_session_log_account_x_yyyyMM` - информация по наработке сессий (услуга - кол-во, наработка)

Или, если же информация по сессиям нужна, но не нужна информация по трафикам, а достаточно информации по наработке, то можно выполнить `TRUNCATE inet_session_log_detail_x_yyyyMM` для старой таблицы. Если и детализации по наработке старых сессий не нужна, то выполните также `TRUNCATE inet_session_log_account_x_yyyyMM`

При CoA-запросе переключения ISG-сервиса Cisco отвечает 405 Config apply failed

Возможно конфигурация ISG-сервиса отсутствует в конфигурации Cisco или не верна.

Если конфигурацию ISG-сервиса Cisco запрашивает по RADIUS, то возможно она не получает Access-Асепт на запрос по этому сервису (User-Name=<имя-ISG-сервиса>) - например, на RADIUS-сервере не заведено описание ISG-сервиса с таким именем или RADIUS-сервер не отвечает (не запущен).

При добавлении IP-адреса выдает ошибку "Подходящий IP-ресурс не найден"

Период IP-ресурса должен быть не меньше, чем период сервиса договора.

Модуль IPN

В настоящее время для данного модуля разработана более современная замена - модуль [Inet](#). Модуль IPN оставлен для совместимости и постепенно будет удалён.

Назначение модуля

[Модуль IPN](#) предназначен для учёта постоянных подключений по протоколу IP путём сбора и анализа первичных логов. Первичные логи собираются либо напрямую коллектором BGIPNetFlowCollector, либо сторонней программой, которая "подсовывает" бинарные логи в коллектор, оставляя ему лишь функцию обработки.

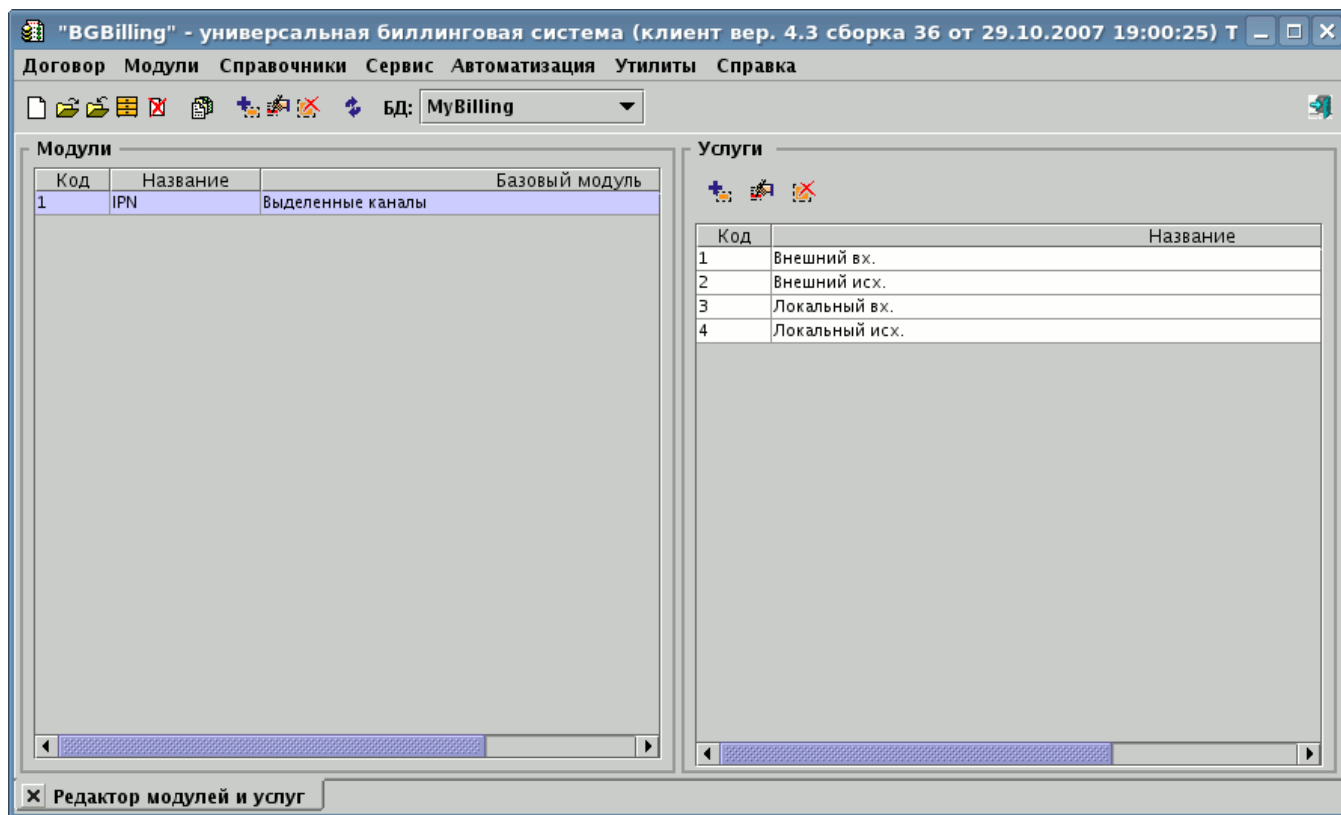
Также модуль может осуществлять автоматическую блокировку пользователей, посылая управляющие команды на шлюз пользователя. В данный момент модуль поддерживает следующие типы шлюзов:

- FreeBSD роутер с файрволом ipfw;
- Linux-роутер с файрволом iptables и системой контроля трафика iproute2;
- DLINK 35xx, 38xx с поддержкой DHCP Options 82;
- любой управляемый коммутатор в режиме простой блокировки порта;
- CISCO как VPN-шлюз с RADIUS-авторизацией (предоставление виртуальных сетей корпоративным клиентам);
- Mikrotik RouterOS;
- Любое ваше оборудование, которое можно поддержать с помощью реализации [шлюза на встроенном языке BeanShell](#).

Идентификатором клиента в системе выступает IP-адрес, либо интерфейс подключения.

Настройка модуля

После инсталляции модуля, создания его экземпляра в **Редакторе модулей и услуг** необходимо завести услуги, определяющие типы трафика в модуле. Например: IPN внешний входящий, IPN внешний исходящий, IPN локальный входящий, IPN локальный исходящий.



На вкладке **Конфигурация** модуля установите конфигурацию по умолчанию.

Конфигурация

```

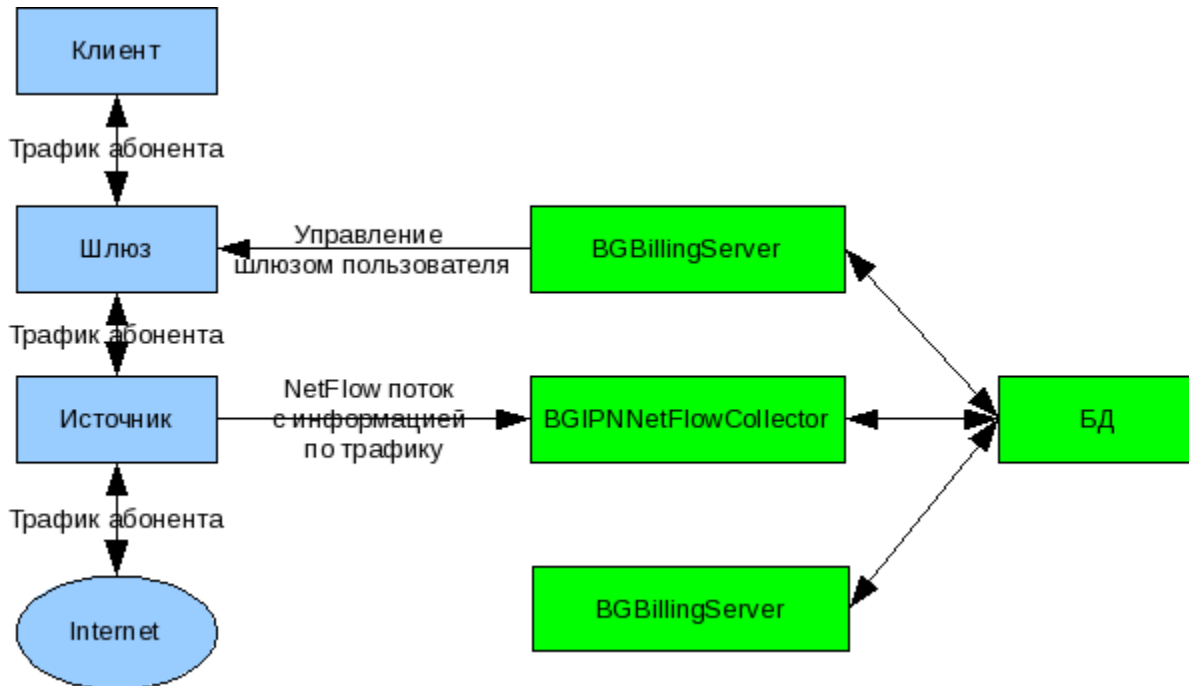
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

web.menuItem1=IP- (IPN)
web.menuItem2= (IPN)
web.menu.gateRules= (IPN)
# 0 - , 1 -
status.after.unlock=1
# , ,
#service.recalc.ignore=
# (0 - , 1 - , 2 - , 3 - , 4 - )
default.contract.status=0
# Web- . 0 - , 1 - .
web.service.allow=1
#
max.periodic.errors=30
#-----
#
#
#closed.date.disabled.ActionMaxRecalculate=1
#
#closed.date.disabled.ActionRecalculate=1
#
#closed.date.disabled.ActionUpdateContractAddress=1
#-----
#
#{ddl}{MM1}{yyyy1} - , , , {dd2}.{MM2}.{yyyy2}-, , , {dd2}.{MM2}.{yyyy2} , , .
#{ip} - ip-
#{contract_title} -
#{user} - ,
#file.detail.template={cid}_{ddl}.{MM1}.{yyyy1}_{dd2}.{MM2}.{yyyy2}_{dd_now}.{MM_now}.{yyyy_now}

```

Базовые понятия и алгоритм работы модуля

Все базовые компоненты модуля изображены на схеме.



Клиент получает доступ в сеть, проходя через машину - **Шлюз**, ограничивающую доступ и машину - **Источник**. В некоторых случаях они могут быть совмещены на одном устройстве.

Функция источника - предоставление информации о трафике абонента посредством передачи NetFlow-потока на коллектор модуля IPN (BGIPNetFlowCollector). Функция шлюза - получение от сервера управляющих команд, закрывающих или открывающих доступ пользователя к сети и их выполнение.

Обработка входных данных разделена на несколько этапов (описание упрощено, более подробно каждый этап рассмотрен далее):

1. Коллектор принимает поток логов и сохраняет их в бинарные файлы определённого формата с разбивкой по часам, в логах содержится сырая информация о том с какого IP-адреса на какой какой количество байт было передано;
2. При переходе часа отдельный поток коллектора обрабатывает файлы, помещая в БД агрегированную наработку по часам, привязывая IP-адреса к договорам и разделяя их по видам услуг;
3. По расписанию планировщик тарифицирует агрегированную наработку, используя тарифные планы абонентов и начисляя наработку на баланс договоров. Переобсчёт производится каждый раз за весь месяц.

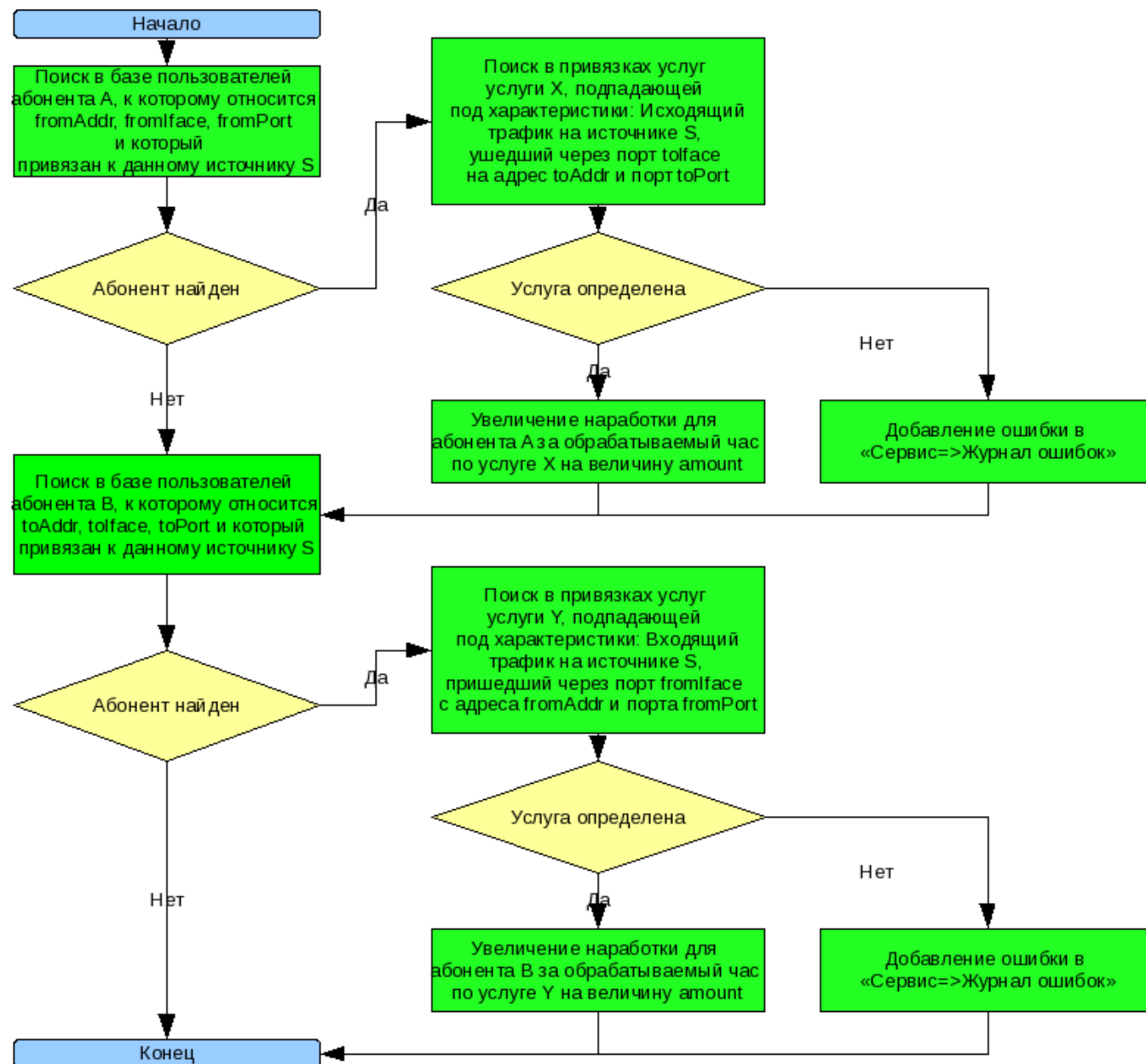
Сервер биллинга, планировщик и коллектор - отдельные процессы. Связь между ними осуществляется исключительно через БД. Так, сервер может передать через БД задание на переобработку логов коллектору, задание на переобсчёт логов планировщику заданий. Коллектор передаёт через базу серверу информацию по статусу часовых логов (загружен/обработан/отсутствует) для отображения в менеджере источников.

Привязки услуг (категории трафика)

При обработке сырых логов с трафиком из бинарных логов для каждой записи, содержащей информацию:

- **fromAddr** - IP-адрес отправителя пакета;
- **toAddr** - IP-адрес получателя пакета;
- **fromIface** - интерфейс роутера, на который вошёл пакет;
- **toIface** - интерфейс роутера, через который пакет покинул роутер;
- **fromPort** - UDP/TCP-порт, с которого пакет ушёл;
- **toPort** - UDP/TCP порт, на который пакет пришёл;
- **amount** - количество байт в пакете;
- **tos/diffServ** - поле DiffServ.

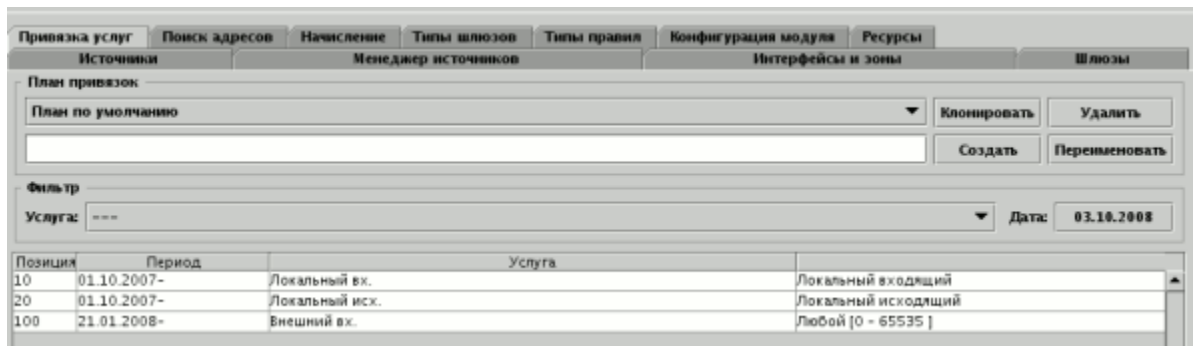
Выполняются следующие шаги, изображённые на схеме:



Из схемы видно, что каждый пакет обрабатывается как бы в "две стороны" и может быть отнесён к двум абонентам провайдера: как исходящий для одного и входящий для другого. Использование TCP/UDP-портов в идентификации абонента позволяет получать выделенный трафик по серверам клиента. Использование портов в идентификации услуги позволяет разделять трафик по типам сервиса (Mail, HTTP, FTP).

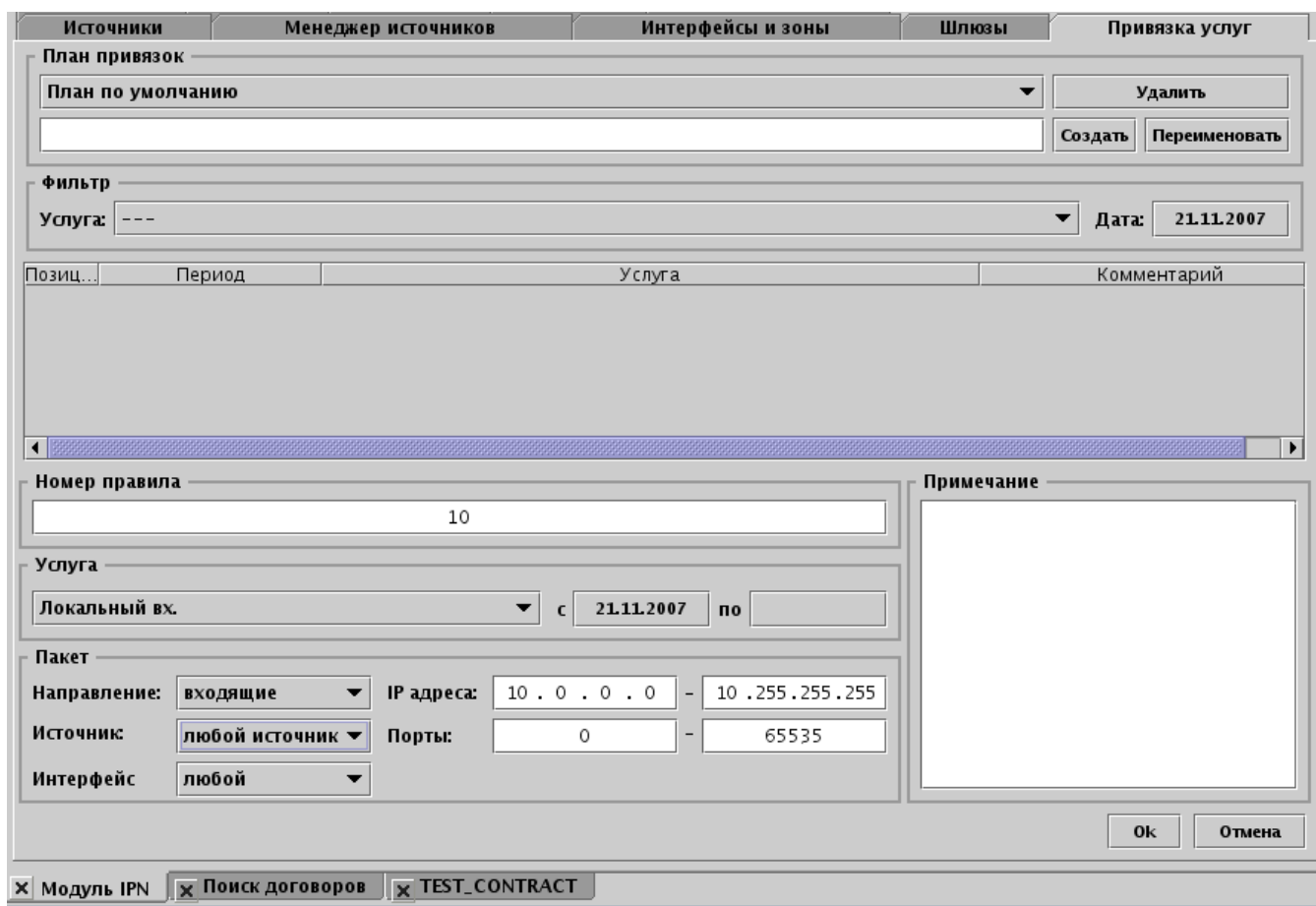
Привязки услуг определяют принципы разделения сырого трафика из логов по услугам. Редактирование осуществляется во вкладке **Привязки услуг** модуля. Каждая привязка отнесена к своему **Плану привязок**. Использование планов привязок позволяет по-разному разделять трафик по услугам для разных категорий абонентов.

Редактирование планов осуществляется в верхней области редактора. С момента установки в системе присутствует **План по умолчанию**.



Для создания нового плана его название вводится в текстовую область и кнопкой **Создать** план создаётся, при этом он также становится активным в выпадающем списке. Соответственно, для переименования плана он выбирается в выпадающем списке, в текстовую область вводится новое название плана и выбирается кнопка **Переименовать**. Удаление плана осуществляется выбором его в выпадающем списке и кнопкой **Удалить**. Кнопка **Клонировать** создаёт новый план привязок, с аналогичным клонируемому содержимым.

План по умолчанию нельзя ни удалить, ни переименовать. В таблице ниже отображаются привязки услуг для текущего плана привязок, редактирование осуществляется общей панелью инструментов. Кнопка **Новый элемент** открывает редактор привязки.



Все правила (привязки) в плане привязок упорядочены по номерам, в порядке которых осуществляется просмотр при определении услуги. Правила с меньшими номерами просматриваются ранее и в них можно определить наиболее жёсткие правила. Например, удобен следующий порядок правил: трафик с выделенных серверов, локальная сеть, прочий интернет. При нумерации правил удобно оставлять интервалы (10, 20, 40..), что позволит в дальнейшем вставить в пробелы другие правила.

Для примера работы правил рассмотрим сеть с выделенным почтовым сервером 10.0.0.10, локальной сетью 10.x.x.x и внешним интернетом через, например, NAT. Трафик входящий из внешнего интернета и с почтового сервера следует отнести к внешнему, а трафик с локальной сети - к внутреннему. Последовательно отображены редакторы добавления правил.

Номер правила <input type="text" value="10"/>	Примечание <input type="text" value="Любой [0 - 65535]"/>
Услуга Внешний вх. с 21.01.2008 по	
Пакет	
Направление: <input type="text" value="входящие"/>	IP адреса: <input type="text" value="10 . 0 . 0 . 10"/> - <input type="text" value="10 . 0 . 0 . 10"/>
Источник: <input type="text" value="любой источник"/>	Порты: <input type="text" value="0"/> - <input type="text" value="65535"/>
Интерфейс: <input type="text" value="любой"/>	ToS/DiffServ: <input type="text"/>
<input type="button" value="Ок"/> <input type="button" value="Отмена"/>	

Номер правила <input type="text" value="20"/>	Примечание <input type="text" value="Любой [0 - 65535]"/>
Услуга Локальный вх. с 21.01.2008 по	
Пакет	
Направление: <input type="text" value="входящие"/>	IP адреса: <input type="text" value="10 . 0 . 0 . 10"/> - <input type="text" value="10 . 255 . 255 . 255"/>
Источник: <input type="text" value="любой источник"/>	Порты: <input type="text" value="0"/> - <input type="text" value="65535"/>
Интерфейс: <input type="text" value="любой"/>	ToS/DiffServ: <input type="text"/>
<input type="button" value="Ок"/> <input type="button" value="Отмена"/>	

Номер правила <input type="text" value="30"/>	Примечание <input type="text" value="Любой [0 - 65535]"/>
Услуга Внешний вх. с 21.01.2008 по	
Пакет	
Направление: <input type="text" value="входящие"/>	IP адреса: <input type="text" value="0 . 0 . 0 . 0"/> - <input type="text" value="0 . 0 . 0 . 0"/>
Источник: <input type="text" value="любой источник"/>	Порты: <input type="text" value="0"/> - <input type="text" value="65535"/>
Интерфейс: <input type="text" value="любой"/>	ToS/DiffServ: <input type="text"/>
<input type="button" value="Ок"/> <input type="button" value="Отмена"/>	

Также возможно делить трафик на услуги используя **DiffServ**. Для этого в правиле необходимо прописать значение вида 000.000.00, т.е. 8 бит, для каждого бита 1, 0 или x - любой. Например, 000.100.xx или 000.1x0.xx. 000.1x0.xx преобразуется в 000.100.00 и маску 111.101.006 сначала наложится маска, результат проверится на соответствие с 000.100.00. Если использовать DiffServ не нужно, просто оставьте поле пустым.

В поле **Примечание** можно ввести краткое описание правила, если поле оставляется пустым система сама генерирует комментарий. Обратите также внимание, что нулевой адрес начала или конца диапазона адресов означает отсутствие ограничений в данном правиле на адрес сверху или снизу. Каждое правило характеризуется периодом действия. При обработке логов за даты, не попадающие в период правила, правило будет игнорироваться.

После добавления данных правил таблица примет следующий вид:

Позиц...	Период	Услуга	Комментарий
10	21.11.2007-	Внешний вх.	Входящий 10.0.0.10 - 10.0.0.10 [0 - 65535]
20	21.11.2007-	Локальный вх.	Входящий 10.0.0.0 - 10.255.255.255 [0 - 65...
30	21.11.2007-	Внешний вх.	Входящий [0 - 65535]

Создание источников и интерфейсов

В терминах модуля IPN **Источник** - это роутер, с которого приходит NetFlow-поток с информацией от трафике, прошедшем через него. К источнику привязываются адреса клиентов. Привязка абонентов к источнику позволяет избежать двойного учёта пакета клиента, прошедшего через несколько роутеров.

Источник характеризуется IP-адресом и названием. IP-адрес сравнивается коллектором с адресами входящих пакетов с информацией о трафике. Название визуально идентифицирует источник в графическом интерфейсе. Доступны типы источников логов:

- **NetFlow** - обработка NetFlow-потока, либо логов flow-tools коллектором, в свойствах указывается адрес хоста, с которого приходит поток;
- **sFlow** - обработка sFlow-потока коллектором, в свойствах указывается адрес хоста, с которого приходит поток;
- **SNMP** - опрос коллектором счетчиков интерфейсов по SNMP-протоколу, в свойствах указывается адрес для опроса и community.

Редактор

Тип источника: NetFlow

Наименование источника: MAIN

Период: с 01.11.2007 по

NetFlow

Адрес хоста: 23 . 11 . 22 . 33

Конфигурация | Комментарий

Восстановить | Ок | Отмена

Модуль IPN | Поиск договоров TEST_CONTRACT

В конфигурации источника типа NetFlow возможно указание опции:

- **ignore.by.zero.dst.iface=1** - пропускать пакеты с нулевым интерфейсом назначения, т.к. на аппаратуре CISCO это означает что пакет не получил клиент
- **use.load=1** - отображать ли пункт меню Добавить в загрузку в менеджере источников

В конфигурации источника типа SNMP возможно указание опций:

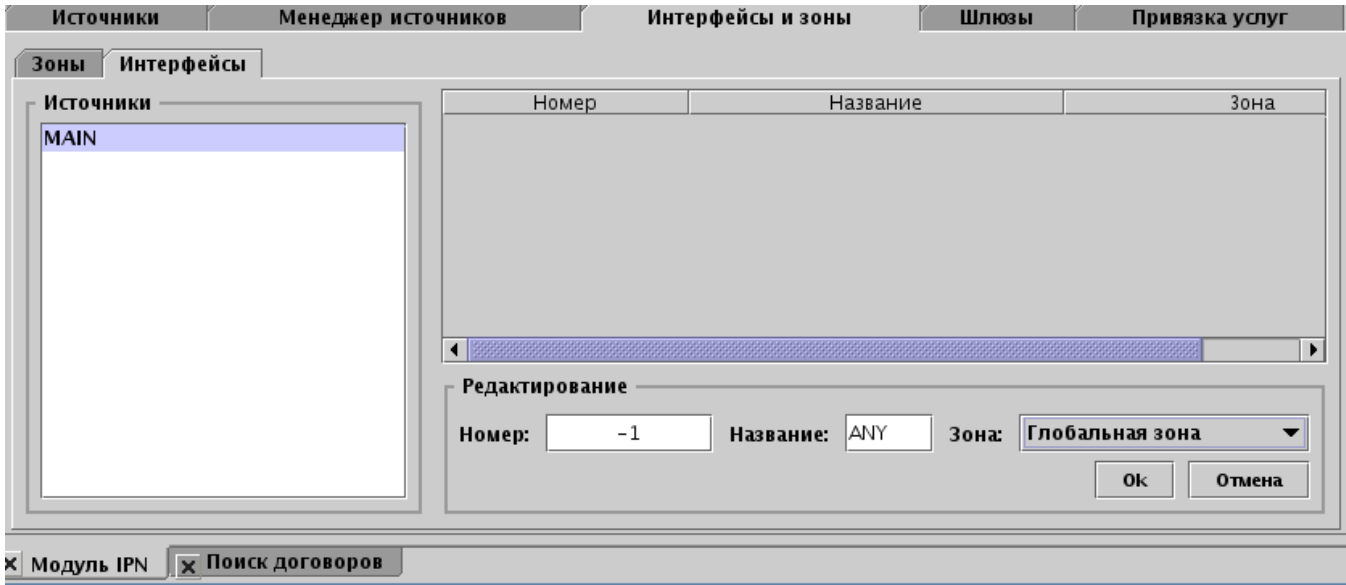
- **snmp.port=<port>** - для источника с типом SNMP - порт опроса, отличный от стандартного 161;
- **snmp.version=<version>** - версия протокола 1, либо 2с, по умолчанию используется версия 1.

Типы источников **FTP** и **Локальная или сетевая** папка использовались в устаревших версиях биллинга, при загрузке логов сначала в БД и дальнейшей обработки. Данная схема не рекомендуется более к использованию как ресурсоемкая и в данном руководстве более не описывается, при необходимости вы можете использовать инструкции к старым версиям биллинга.

После создания источника ему присваивается уникальный **Код источника** (отображается в первом столбце таблицы, см. ниже), который должен быть указан в конфигурации коллектора. Разные источники могут обслуживать разные коллекторы, это позволяет разнести обработку логов по разным машинам, повышая масштабируемость системы.

ID	Название источника	Период	Комментарий
1	MAIN	01.11.2007-	

После создания источника определяются его интерфейсы. **Интерфейс** в терминах биллинга - это сетевые интерфейсы роутера, к которым могут быть привязаны клиенты. Если учёт интерфейсов не требуется, то достаточно создать на вкладке **Интерфейсы и зоны** интерфейс **ANY** с кодом -1 и привязывать клиентов на данном роутере к нему.

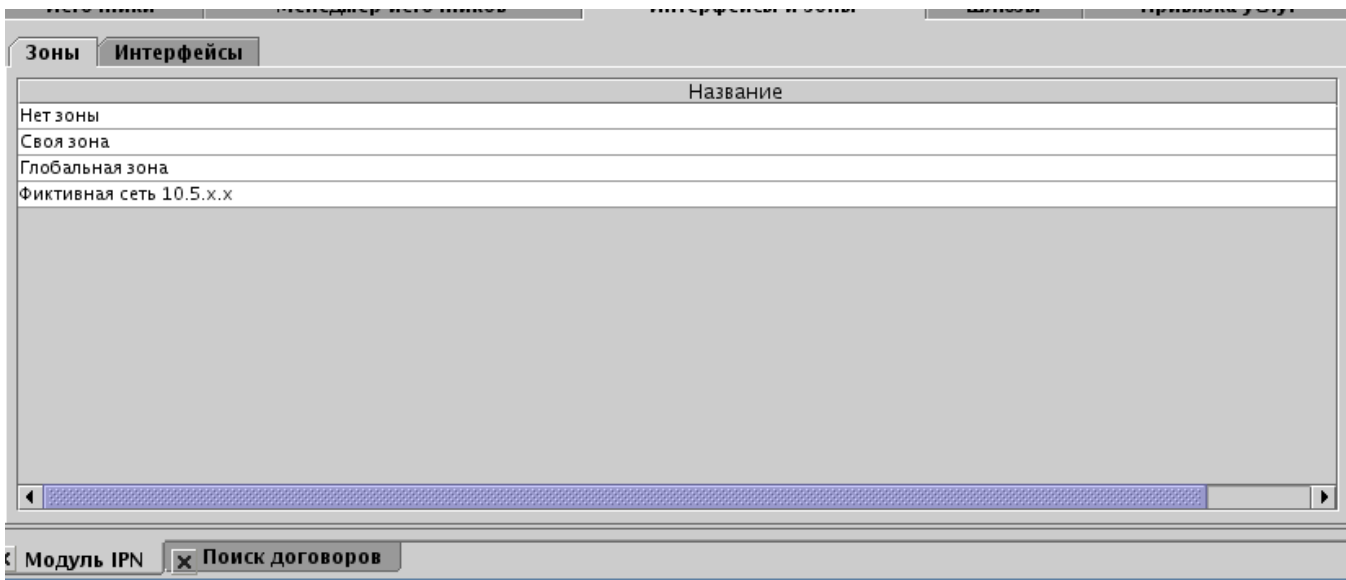


Зона - это область в которой не должны пересекаться адреса клиентов. Зоны сделаны для поддержки в сети провайдера нескольких фиктивных сетей с одинаковыми адресами но на разных интерфейсах.

Зоны нужны лишь для контроля непересечения адресов клиентов. Зона привязывается к интерфейсу. Биллинг предоставляет три предопределённые зоны: **Глобальная зона** - единая зона на всех клиентов, **Своя зона** - зона в пределах интерфейса, **Нет зоны** - нет проверки уникальности.

Если всем клиентам выдаются адреса из непересекающихся сетей, достаточно использовать глобальную зону.

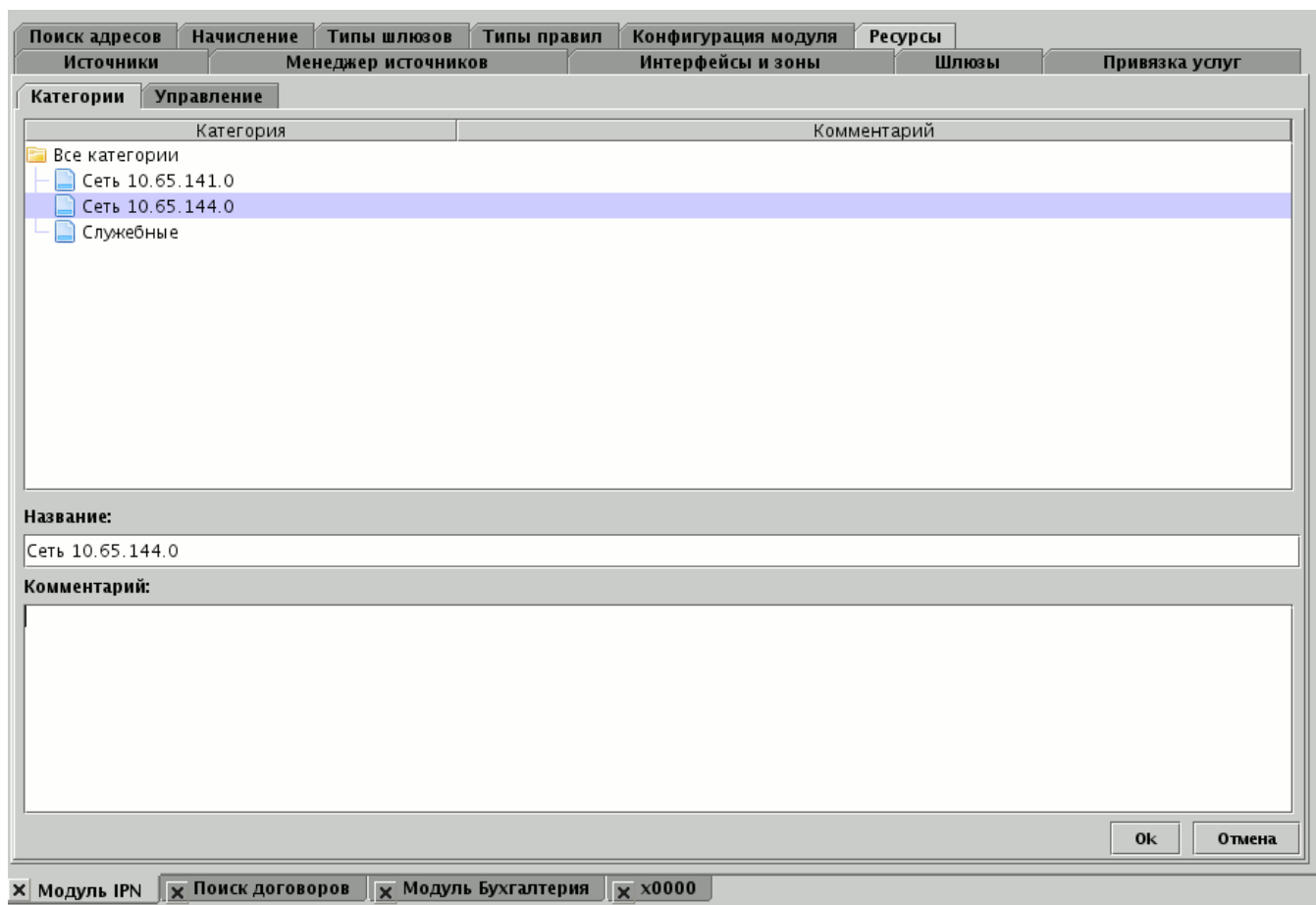
Дополнительно возможно создание своих собственных зон. Например, одна фиктивная сеть может быть разделена на несколько интерфейсов разных роутеров. Для редактирования зон используется вкладка **Зоны**.



Управление ресурсами IP-адресов

Система управления ресурсами позволяет отслеживать состояние доступных IP-адресов, автоматизировать выделение их клиентам. Управление базой адресов производится на вкладке **Ресурсы** модуля.

На подвкладке **Категории** определяются категории ресурсов. В качестве категорий целесообразнее всего определять сети провайдера.



Для редактирования категорий используется общая панель инструментов. Непосредственно просмотр ресурсов осуществляется на вкладке **Управление**.

Привязка услуг | Поиск адресов | Начисление | Типы шлюзов | Типы правил | Конфигурация модуля | Ресурсы | Шлюзы

Источники | Менеджер источников | Интерфейсы и зоны | Шлюзы

категории | Управление

Категория

- Все категории
- Сеть 10.65.141.0
- Сеть 10.65.144.0
- Службные

IP: 10.65.144.0 / 24 Подсеть Любые на дату: Вывести

Период	IP
04.03.2008-	10.65.144.0
04.03.2008-	10.65.144.1
04.03.2008-	10.65.144.2
04.03.2008-	10.65.144.3
04.03.2008-	10.65.144.4
04.03.2008-	10.65.144.5
04.03.2008-	10.65.144.6
04.03.2008-	10.65.144.7
04.03.2008-	10.65.144.8
04.03.2008-	10.65.144.9
04.03.2008-	10.65.144.10
04.03.2008-	10.65.144.11
04.03.2008-	10.65.144.12
04.03.2008-	10.65.144.13
04.03.2008-	10.65.144.14
04.03.2008-	10.65.144.15
04.03.2008-	10.65.144.16
04.03.2008-	10.65.144.17

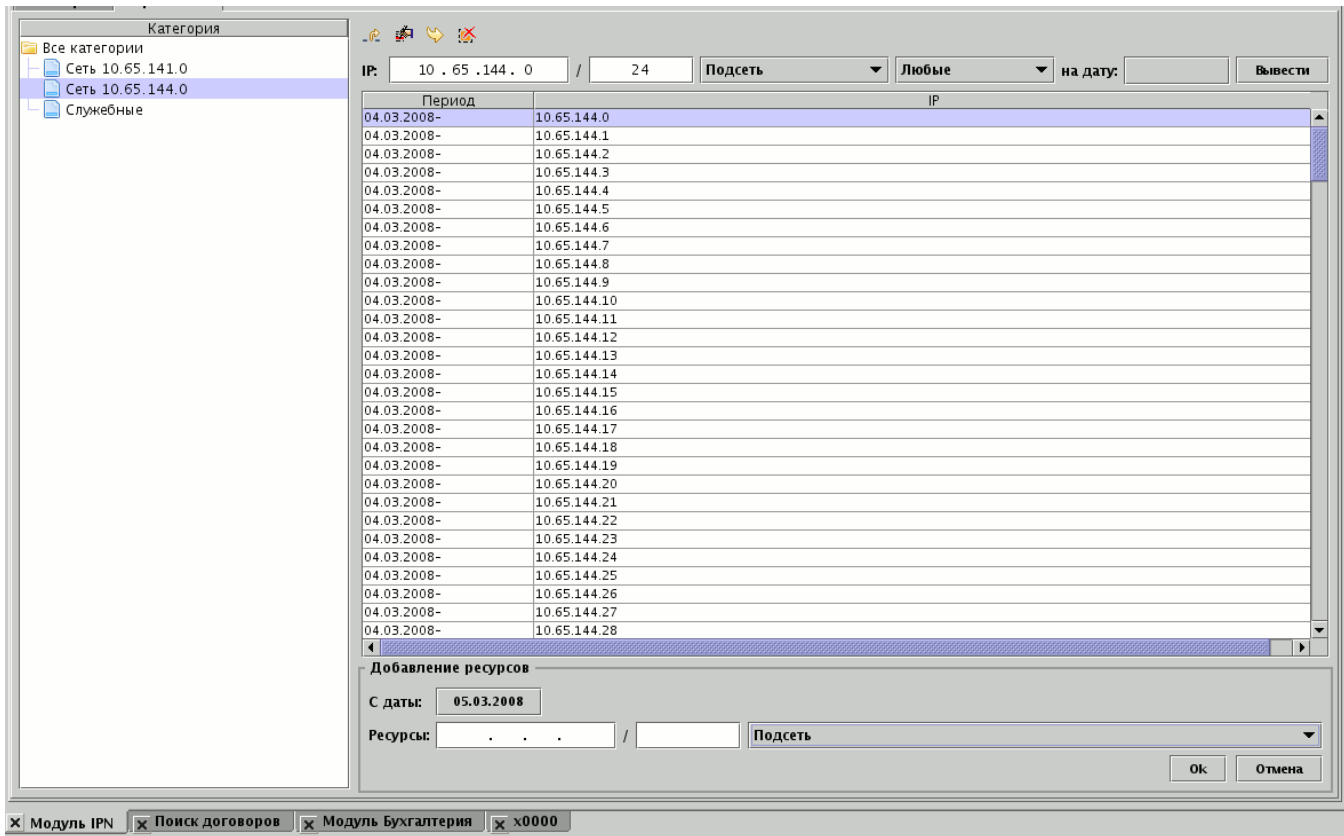
Период	Договор	Комментарий
05.03.2008-	x0000	Иванов Петр Сидорович

Закреть

Модуль IPN | Поиск договоров | Модуль Бухгалтерия | x0000

Возможна установка фильтра по категории, сети, диапазону ресурсов. При двойном клике по IP-адресу отображается история его использования договорами. При двойном клике по строке истории открывается договор.

Для добавления/удаления и изменения периода действия ресурсов используется панель инструментов над таблицей. Для добавления ресурсов необходимо выбрать категорию в дереве и нажать кнопку **Открыть ресурсы**.



Ресурсы можно добавить сетью, либо диапазоном адресов.

Кнопками **Изменить период ресурсов** и **Изменить категорию** возможно изменение периода доступности ресурсов, либо их категории. Предварительно ресурсы нужно выбрать в таблице используя клавиши Shft, Ctrl, либо комбинацию Ctrl+A (предварительно отфильтровав только нужные ресурсы).

Кнопка **Удалить** удаляет выбранные ресурсы, при этом они не должны быть задействованы ни в одном договоре.

Ресурс помечается как занятый только для диапазонов, находящихся в Глобальной зоне.

С версии 4.5 ресурсы модуля IPN хранятся как диапазоны.

После добавления ресурсов их необходимо синхронизировать с уже назначенными диапазонами договоров. Для этого необходимо нажать кнопку **Синхронизировать занятые**.

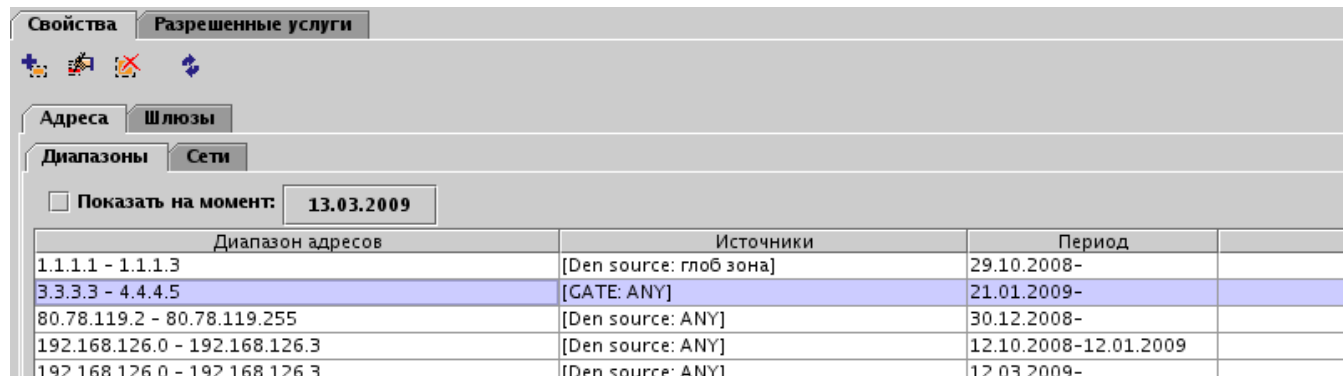
Добавление адресов абонентам

Подключение модуля IPN к договору осуществляется путем выбора узла **Модули** дерева договора и нажатием кнопки **Новый элемент** стандартной панели инструментов.

Добавление разрешённых услуг в договор необходимо:

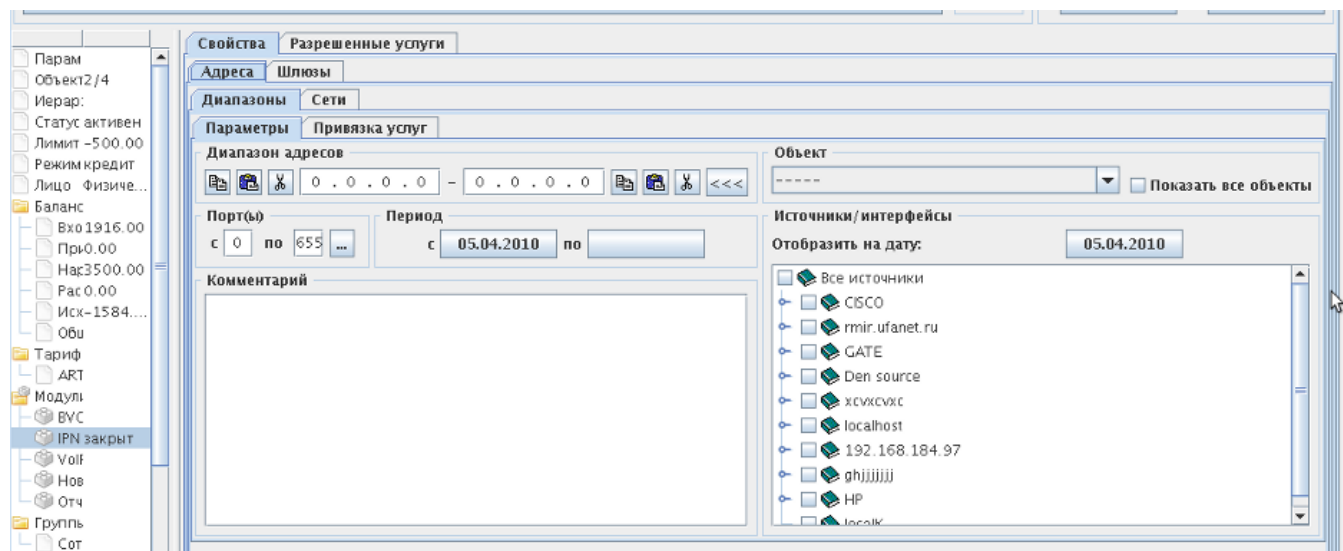
- При использовании в тарифных планах договора узлов **Диапазон** в режиме **пропорционально периоду разрешённой услуги** (см. далее о тарификации). Объем наработки в этом случае определяется пропорционально доле присутствия разрешённой услуги в месяце. Отсутствие разрешённой услуги означает полный объем;
- Для ограничения видимых клиентом в отчёте Web статистики услуг модуля при установленной опции конфигурации модуля **web.service.allow=1**;
- При начислении максимальных трафиков, см. [далее](#).

Управление адресами абонента производится на вкладке **Адреса** редактора свойств модуля IPN.



Диапазон адресов	Источники	Период
1.1.1.1 - 1.1.1.3	[Den source: глоб зона]	29.10.2008-
3.3.3.3 - 4.4.4.5	[GATE: ANY]	21.01.2009-
80.78.119.2 - 80.78.119.255	[Den source: ANY]	30.12.2008-
192.168.126.0 - 192.168.126.3	[Den source: ANY]	12.10.2008-12.01.2009
192.168.126.0 - 192.168.126.3	[Den source: ANY]	12.03.2009-

Здесь можно добавлять сети и диапазоны. Вкладка сети позволяет вводить сети вида xx.xx.xx.xx/xx. Сеть - это частный случай диапазона и работает аналогичным образом. Во многих редакторах вывод сетей и диапазонов происходит единым списком. Сети используются в командах некоторых шлюзов IPN. Добавление адреса (добавление сети имеет аналогичный вид, за исключением формы ввода xx.xx.xx.xx/xx) :



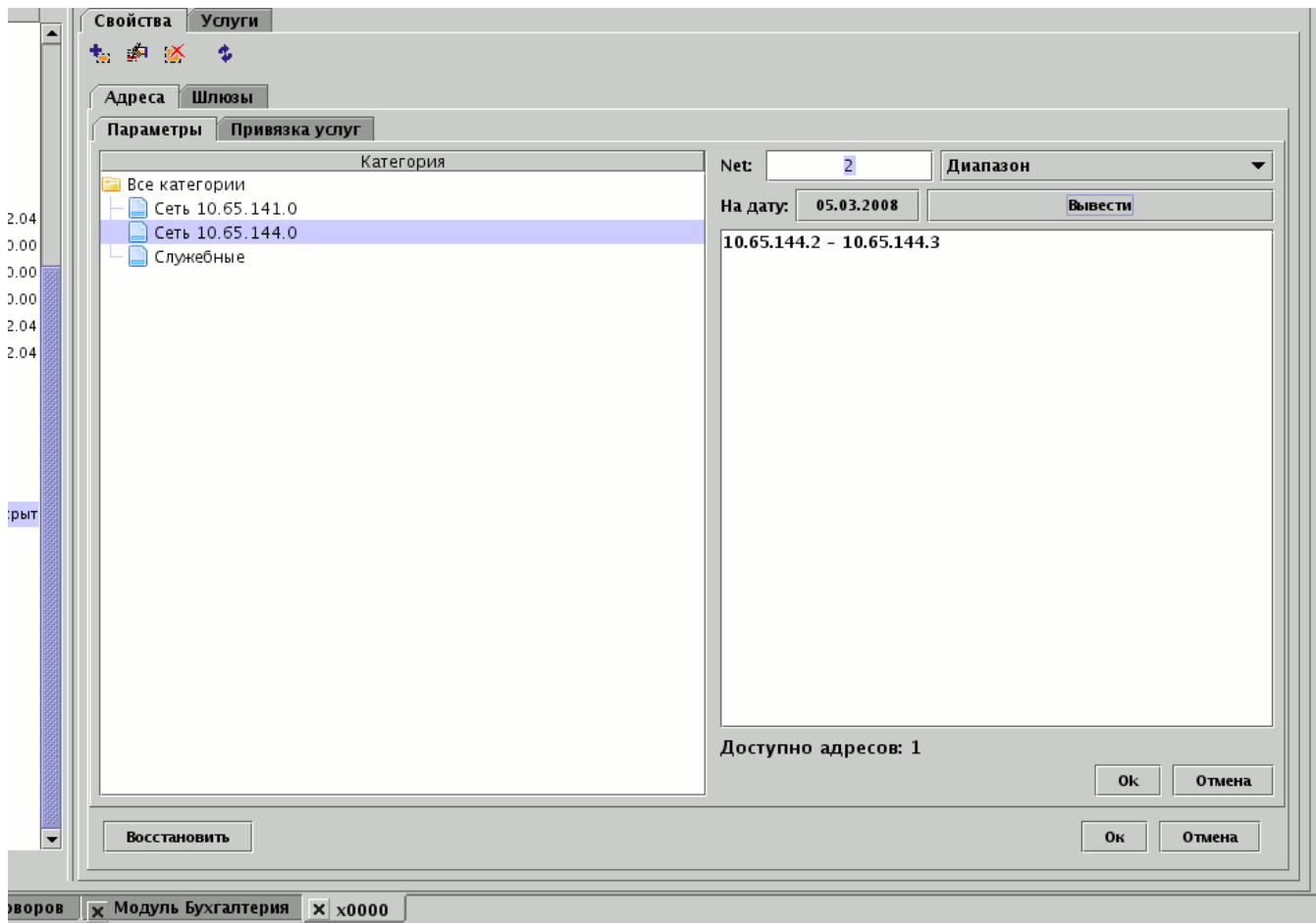
При добавлении адреса указывается договор, интерфейс, к которому привязан адрес на источнике. Опционально может быть указан объект, к которому относится диапазон адресов. Если не установлена галочка **Показать все объекты**, то в выпадающем списке будут только объекты, имеющие активный период, иначе будут выведены все имеющиеся объекты. Для раскрытия списка источников дважды нажмите по верхнему узлу дерева **Все источники**.

В случае, если абонент идентифицируется не по адресу, а по интерфейсу, необходимо выбрать интерфейс в дереве, а диапазон адресов установить в 0.0.0.0-255.255.255.255. Интерфейс с зоной **Своя зона** должен быть предварительно создан на вкладке **Интерфейсы и зоны** модуля.

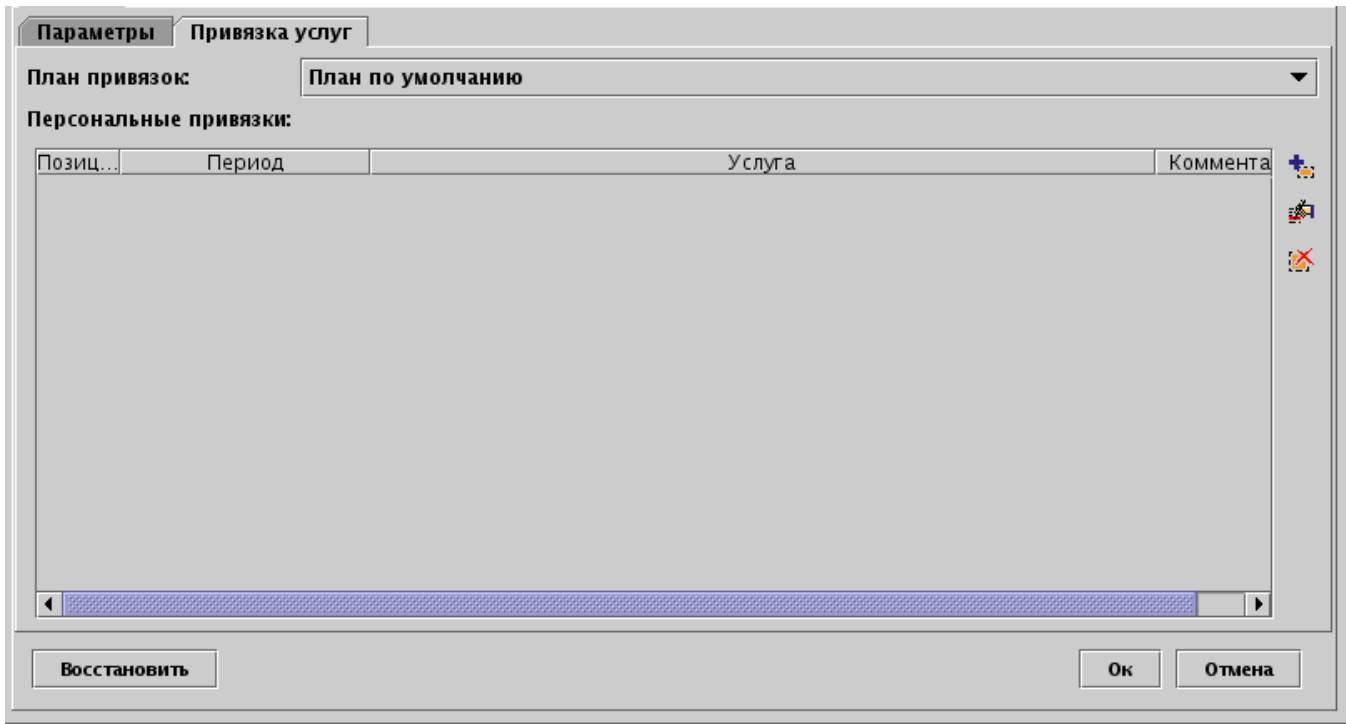
Начиная с 4.4 версии рядом с диапазоном адресов добавлена кнопка **Взять из пула ресурсов**.



При её нажатии открывается окно выбора адресов из пула, где можно выделить диапазон адресов требуемой размерности, либо подсеть.



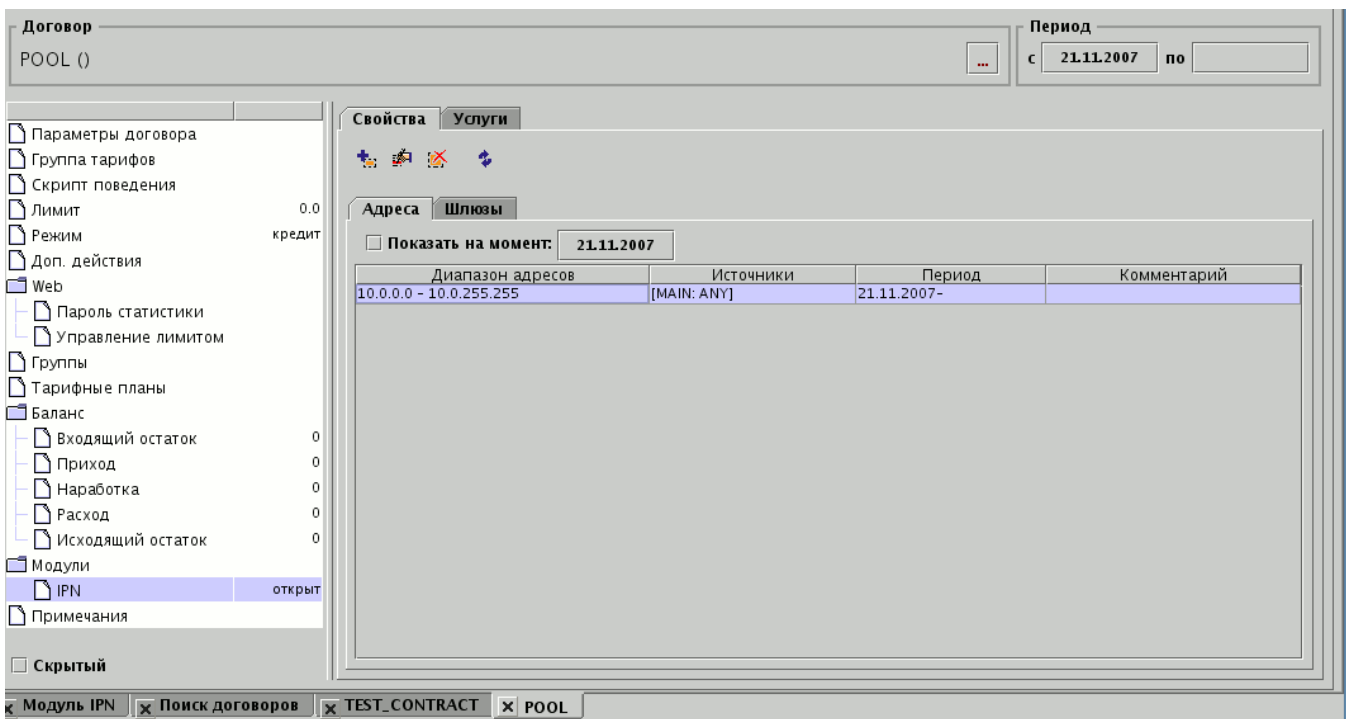
На вкладке **Привязка услуг** редактора адреса можно выбрать план привязок для данного диапазона адресов абонента. Также можно добавить персональные правила определения услуг. При обработке логов для данного диапазона адресов первыми будут просмотрены персональные привязки, далее, если услуга не определена - правила выбранного плана привязок. Редактор персональных привязок полностью идентичен редактору привязок, относящихся к какому-либо плану.



Метод добавления адресов из пула является рекомендуемым к использованию. Также есть возможность учёта номерной ёмкости с помощью специального диапазона на одном из договоров. Это метод использовался в системе до появления ресурсов адресов и на текущий момент считается устаревшим и менее удобным, чем работа через ресурсы.

Метод непосредственного заведения адресов рекомендуется использовать только на момент запуска модуля в эксплуатацию. В дальнейшем для удобства учёта номерной ёмкости работа с адресами выполняется по следующему алгоритму.

Создаётся договор, содержащий все сети адресов, либо несколько сетей, как удобнее. При этом диапазон адресов привязывается ко всем интерфейсам, где возможно использование данных адресов.



Для переноса одного или нескольких адресов с договора-пула на договор абонента выбирается строка с диапазоном и производится двойной клик мышью при нажатой клавише Ctrl. Адрес может быть перенесён только на один из открытых договоров. При переносе указывается источник и интерфейс, к которому относится абонент, а также дата, с которой адрес переходит новому договору

Адреса Шлюзы

Текущий диапазон: 10.0.0.0 - 10.0.255.255 [0 - 65535]

Новый диапазон: 10.0.0.0 - 10.0.0.1

Дата: 21.11.2007

Новый диапазон портов: 0 - 65535

Комментарий:

Договор: TEST_CONTRACT POOL

Источники/интерфейсы:

Отобразить на дату: 21.11.2007

- Все источники
- MAIN
- ANY

Восстановить

Ок Отмена

TEST_CONTRACT POOL

После переноса система автоматически изменит диапазон адресов договора-пула, либо разорвёт его на два диапазона если перенос был выполнен из середины пула.

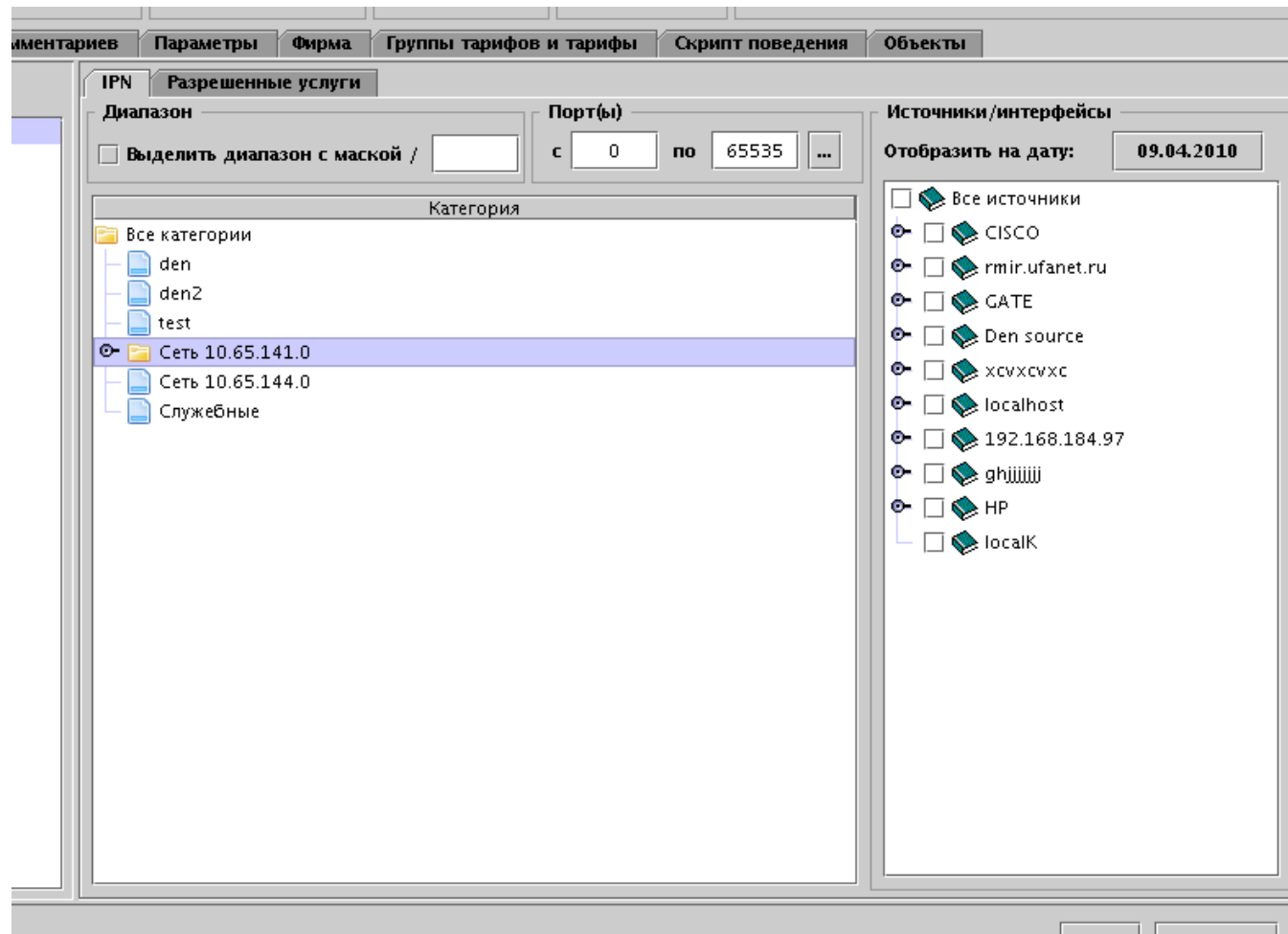
Адреса Шлюзы

Показать на момент: 21.11.2007

Диапазон адресов	Источники	Период	Комментарий
10.0.0.2 - 10.0.255.255	[MAIN: ANY]	21.11.2007-	

Настройка выделения адресов в шаблоне договора

Для того, чтобы настроить автоматическое выделение подсети нужного размера из числа свободных подсетей в глобальной зоне для всех создаваемых договоров, можно воспользоваться настройками модуля IPN при редактировании шаблонов договоров. Во вкладке Модули добавьте модуль IPN в качестве подключаемого для договоров данного шаблона.



Затем во вкладке IPN отметьте галочкой флаг Выделить диапазон с маской и укажите необходимую маску подсети. Укажите категорию ресурсов и интерфейс источника, на котором будет выделена подсеть. Сохраните шаблон.

Данный функционал производит корректный поиск свободных адресов только в глобальной зоне! Попытка создания найденной подсети на интерфейсе в других зонах может привести к конфликтам адресов (договор не будет создан и подсеть не будет выделена).

Настройка сбора и обработки логов

BGIPNNetFlowCollector обновляется как обычное [серверное приложение биллинга](#). Необходимо обновить приложение перед первым запуском.

В данном руководстве пропущена настройка экспорта NetFlow-логов на вашем роутере, вы можете обратиться к его документации для выяснения данного момента. Для PC-роутеров вы можете использовать NetFlow-агент [ipcad](#).

Установите на вашей машине BGIPNNetFlowCollector, который можно загрузить с нашего сайта. Распакуйте архив например на диск **C:** (для MS), либо **/usr/local** для Linux.

Следующий абзац актуален только для MS-систем.

Далее установите переменную среды **BGIPN_NETFLOW_HOME=C:\BGIPNNetflowCollector**. Как устанавливать переменные среды вы можете посмотреть в документации по установке сервера. Также на машине с коллектором должна быть установлена переменная **JAVA_HOME**. Если на этой машине уже установлен сервер биллинга, либо радиус сервер, она уже указана. Вызовите скрипт **netflow_install.bat** для установки службы и можете запускать/останавливать коллектор через консоль управления службами MS. После установки системных переменных перезагрузите машину с коллектором.

Для UNIX-систем укажите в файле **netflow.sh** переменную **JAVA_HOME** - путь к установленной jre, либо jdk (например **/opt/jdk**). Создайте службу запуска коллектора через скрипт **netflow_start.sh** и стопа через **netflow_stop.sh**, либо просто сделайте его автоматически стартующим после mysql-демона. Скрипты для /etc/init.d можно сделать по аналогии со скриптами **bgbilling**, **bgscheduler**.

Коллектор хранит логи в бинарных файлах собственного формата. При этом записи хранятся в своем изначальном виде - т.е. пакеты netflow и sflow хранятся как есть. Часовой лог может храниться в нескольких файлах. Лог совместим с форматом XDR. Лог состоит из заголовка и данных. Данные **могут быть** разбиты на блоки одинаковой длины, при этом, если данных в блоке меньше, чем его длина, то оставшееся место заполнено нулевыми байтами. Данные могут быть сжаты zlib (блоки сжимаются в одном zlib потоке).

Формат файла лога:

```
header {
  int magic = "BGDL" = 0x4247;
  int version = 2;
  int type = 1 (IP);
  int header_params_length;
  param<> {
    int param_type;
    int param_value_length;
    opaque value;
  }
}

data {
  block<> {
    int block_length;
    long reserved;
    int reserved;
    opaque block_data;
  }
}
```

Коллектор может работать как в автономном режиме, принимая и обрабатывая логи, так и взаимодействуя с flow-tools коллектором. Обработка логов производится непосредственно самим коллектором, что позволяет хорошо разносить нагрузку по машинам.

Логи складываются в дерево каталогов в следующем формате:

```
source_< >
+ yyyy/
  + yyyy-MM/
    + yyyy-MM-dd/
      + log_yyyy-MM-dd-NN.nnn.bgdl
```

например, лог за 3 час 25-го мая 2009 года:

```
source_1
+ 2009/
  + 2009-05/
    + 2009-05-25/
      + log_2009-05-25-03.000.bgdl
```

```
+ log_2009-05-25-03.001.bgdl
+ log_2009-05-25-03.002.bgdl
```

Для начала работы необходимо сконфигурировать файл **netflow_ipn.properties** в соответствии с необходимым функционалом.

Конфигурация

```
#
port.admin=2003

#
db.driver=com.mysql.jdbc.Driver
db.url=jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8
db.user=bill
db.pswd=bgbilling
db.maxActive=300
db.maxIdle=100

# IPN
collector.mid=XXX

# ( )
log.dir=/usr/bill/log
#
process=1
# ,
# ,
process.sources=< >

# ,
# .. , - ,
# -
#process.thread.count=1
# -
#
# ( snmp)
#process.datalog.thread.count=1

#
# ,
#generate.minutes=30

# ,
#ipn.collector.detail.folder=tmp
```

В параметре **collector.mid** файла **netflow_ipn.properties** укажите числовой код вашего модуля IPN, вы можете его узнать в редакторе модулей и услуг. Если база расположена на отличной от коллектора машине или вы меняли логин/пароль, скорректируйте опции подключения к БД. В параметре **process.sources** укажите числовые коды источников, логи которых коллектор будет обрабатывать, через запятую. Параметр **log.dir** определяет директорию расположения бинарных логов.

Код источника можно посмотреть на вкладке **Источники** экземпляра модуля IPN в первом столбце таблицы.

Возможно использование коллектора в автономном режиме, при этом он сам будет сохранять логи в файлы и производить их обработку ежечасно или чаще (опция **generate.minutes**). Также коллектор выполняет задания на переобработку логов добавленные пользователем в менеджере источников вручную.

Либо коллектор может работать в связке с flow-tools или иными сторонними коллекторами, при этом он При работе в связке с flow-tools коллектор перестаёт выполнять функции коллектора и остаётся лишь обработчиком.

Далее идёт описание запуска коллектора в автономном режиме и связке с flow-tools.

Настройка коллектора в автономном режиме

IPN collector поддерживает NetFlow версий 1, 5 и 7 (9-ая версия не поддерживается), sFlow версии 5, SNMP версий 1 и 2с.

Добавьте в `netflow_ipn.properties`:

```
#
load=1
```

Для приема NetFlow/sFlow-потока добавьте для каждого прослушиваемого порта. Коллектор может принимать данные на несколько портов, при этом на один порт может принимать данные только одного типа (NetFlow или sFlow).

```
#
collector.capture.flow.port.<listener_id>=<port>
collector.capture.flow.port.<listener_id>.type=<type>
collector.capture.flow.port.<listener_id>.sources=<codes>
collector.capture.flow.port.<listener_id>.thread.count=<count>
#
```

Где:

- **<listener_id>** - уникальный числовой идентификатор слушателя в пределах properties-файла, начинать нумерацию следует с 1, далее последовательно;
- **<port>** - прослушиваемый номер порта;
- **<type>** - тип потока **netflow**, либо **sflow**;
- **<codes>** - числовые коды источников через запятую, поток с которых приходит на этого слушателя;
- **<count>** - количество потоков-обработчиков на данном порту.

Дополнительно могут быть указаны настройки:

```
# / NetFlow/sFlow
datalog.flow.chunk.size=524288
# NetFlow/sFlow : 0 - , 1 - zlib
datalog.flow.compression.type=0
```

Для каждого порта указывается количество потоков, работающих на порту (**collector.capture.flow.port.x.thread.count**). Эти потоки будут существовать все время работы коллектора, обрабатывая приходящие на порт потоки. Каждый поток хранит буферы приема для каждого источника, пакет которого хотя бы раз обрабатывался источником. Размер буфера равен или почти равен размеру блока буфера приема (**datalog.flow.chunk.size**).

Таким образом на порт необходимо памяти **collector.capture.flow.port.x.thread.count * кол-во источников * datalog.flow.chunk.size**. Здесь используется direct buffer memory, а не обычный java heap, по умолчанию максимум 64МБ. Чтобы при необходимости увеличить максимальный объем нужно указать в скрипте запуска **-XX:MaxDirectMemorySize=256M**. При настройках по умолчанию и 2-х источниках на порт необходимо 10 МБ памяти.

Возможно создать два и более источников с одинаковым IP-адресом и разнести их по разным портам, указав на каждом порту свой источник.

Ниже приведен пример настройки портов коллектора для получения логов источника с кодом 244, высылающим NetFlow-поток на порт 2001 коллектора и источника с кодом 255, высылающего sFlow-поток на порт 2002. Компрессия логов отключена.

```
# NetFlow-
#
collector.capture.flow.port.1=2001
# - NetFlow/sFlow
collector.capture.flow.port.1.type=netflow
# ,
collector.capture.flow.port.1.sources=244
# -
collector.capture.flow.port.1.thread.count=10
#
#
collector.capture.flow.port.2=2002
collector.capture.flow.port.2.type=sflow
collector.capture.flow.port.2.sources=255
collector.capture.flow.port.2.thread.count=10
#
```

```
# / NetFlow/sFlow
datalog.flow.chunk.size=524288
# NetFlow/sFlow-: 0 - , 1 - zlib
datalog.flow.compression.type=0
```

Коллектор также может опрашивать источник по SNMP на количество трафика на интерфейсах. Для этого необходимо установить опцию **collector.capture.snmp=1** и указать обслуживаемые источники. Тип обслуживаемых источников должен быть SNMP.

collector.capture.snmp.period=n - период опроса в секундах, т.е. каждые n секунд коллектор будет опрашивать источники. Пример настройки опроса SNMP источников, опрашивается источник с кодом 1:

```
# SNMP-
#collector.capture.snmp=1
# ,
#collector.capture.snmp.sources=1
#
#collector.capture.snmp.period=60
```

При опросе через SNMP известны только сетевые интерфейсы, IP-адреса, породившие и принявшие трафик, неизвестны. Это необходимо учитывать при составлении привязок услуг. Т.е. все привязки услуг, используемые для классификации трафика, полученного путём съёма SNMP-статистики, должны быть с пустым (0.0.0.0-255.255.255.255) фильтром по IP-адресу. Счетчики **CNT_IN** и **CNT_OUT**, полученные с интерфейса **IF** источника преобразуются в две записи о прошедшем трафике:

- с интерфейса -1 адреса 0 на интерфейс IF адрес 0 прошло CNT_IN байт;
- с интерфейса IF адреса 0 на интерфейс -1 адрес 0 прошло CNT_OUT байт.

Эти фиктивные записи о трафике далее обрабатываются по обычному [алгоритму](#), они могут быть получены при выгрузке исходного лога в текстовый файл [коллектором](#).

Сбор и сохранение логов происходит в блоки (chunk), которые при заполнении сбрасываются в файл. Размер блока **datalog.chunk.size**. При обсчете/переходе часа блоки также сбрасываются в файл, при этом пустое пространство заполнено нулями.

При частом обсчете и малом потоке может возникнуть много незаполненного пространства, например: 4 рабочих потока по 524288 собирали логи за 5 минут. Минимальный размер файла будет 2МБ, а нужных данных может быть гораздо меньше. Для оптимизации в таком случае необходимо уменьшить параметр **datalog.chunk.size** (по умолчанию 512 * 1024 для NetFlow/sFlow и 2 * 1024 для SNMP) или же просто использовать сжатие. Если необходимо указать размер отдельно для NetFlow/sFlow и для SNMP, то можно использовать параметры **datalog.flow.chunk.size** и **datalog.snmp.buffer.size** соответственно.

Сжатие логов используется для уменьшения потребления дискового пространства. По умолчанию сжатие выключено. Для включения необходимо указать параметр **datalog.compression.type=1** (**datalog.flow.compression.type** отдельно для NetFlow/sFlow и **datalog.snmp.compression.type** для SNMP).

Настройка коллектора в связке с flow-tools

Построение связки возможно только на UNIX-системах.

В конфигурации необходимо установить опцию **load** в 0.

```
#  
load=0
```

Теперь коллектор не будет слушать порты для приема NetFlow/sFlow-потоков и опрашивать SNMP-источники, а только обрабатывать логи источников.

Далее необходимо настроить связку с flow-tools. Коллектор поддерживает формат логов flow-tools, с логами NetFlow версий 5 и 7, поэтому будет достаточно настроить сбор логов flow-tools в директорию логов источника. `nesting_level` должен быть -3 для совместимости с системой хранения логов коллектора.

Пример: создаем perl файл **rotate_1.pl**, он будет обрабатывать появление нового лог-файла flow-tools с источника с кодом 1

```
#!/usr/bin/perl  
  
my $name = $ARGV[0];  
my $loader='/usr/local/BGIPNNetflowCollector/netflow.sh';  
  
if ( $name =~ /\.*\.(\\d\\d\\d\\d)-(\\d\\d)-(\\d\\d)\\. (\\d\\d)/ )  
{  
    my $logYY = $1;  
    my $logMM = $2;  
    my $logDD = $3;  
    my $logHH = $4;  
  
    ` $loader isload 1 $logYY-$logMM-$logDD-$logHH `;  
}  
else  
{  
    die "unknown format $name";  
}
```

Далее устанавливаем в автозапуск:

```
flow-capture -N -3 -w /usr/local/log/source_1 -n 95 -R rotate_1.pl 0/0/2003
```

Запуск коллектора

Для старта коллектора в UNIX-системах вызовите файл **netflow_start.sh**. Остановка - **netflow_stop.sh**. Для MS-систем запуск осуществляется через оснастку **Службы**.

После запуска проверьте файлы **log/collector.out** и **log/collector.log** на предмет ошибок.

Теперь коллектор подключится к БД, выберет из указанного модуля IPN список источников с кодами из параметра **sources** и будет собирать их логи, сохраняя в бинарные файлы, и обрабатывать их.

При запуске команды **netflow.sh (.bat)** отображается справка по командам коллектора:

```
[bill@flow BGIPNNetflowCollector]$ ./netflow.sh

Usage: [start|stop|status|save|help]
Parameters:
    help                - show this help
    start               - starting NetFlow collector
    stop                - stopping NetFlow collector
    status              - current status
    isload <source_id> <yyyy-MM-dd-HH> - mark source is load
    save <source_id> <yyyy-MM-dd-HH> <path> - save binary log to text on path
    sourcelist          - get ip and source id list for external loader
```

Для вызова команды запустите: **netflow.sh (.bat) <команда>**

Команды **start**, **stop** вызываются скриптами **netflow_start.sh** **netflow_stop.sh** для запуска и остановки коллектора. Рассмотрим остальные команды коллектора:

status - отображение текущего статуса работающего коллектора.

```
[bill@flow BGIPNNetflowCollector]$ ./netflow.sh status
Traffic collector for IPN v 4.6 build 113 from 03.04.2009 14:47:16
Started: 03.04.2009 18:07:37 Uptime: 0 d 00:00:11
Memory total: 92 995 584; max: 1 379 467 264; free: 85 600 872
FlowListener: queue_size: 0; threads_active: 0; largest: 2; core: 4; pool_size: 2; recv_socket_buf_size: 131
071; recv_buf_size: 524 288; packets: 3
FlowListener: queue_size: 0; threads_active: 0; largest: 0; core: 4; pool_size: 0; recv_socket_buf_size: 131
071; recv_buf_size: 524 288; packets: 0
Flow loader: ru.bitel.bgbilling.server.util.ip.datalog.hourly.IPHourlyDataLogger@554d7745 [files: 1]
```

В первой строке отображается версия, номер и время билда коллектора. Во второй - время старта и прошедшее от старта время. На третьей - выделенная память Java-машине, максимально доступная для неё память и свободная в выделенном heap память.

save <source_id> <yyyy-MM-dd-HH> <path> - выгрузка часового лога по источнику с кодом **<source_id>** за час, определённый в формате **<yyyy-MM-dd-HH>** в текстовый файл, путь к которому указан в **<path>**, например:

```
./netflow.sh save 1 2008-01-01-00 /tmp/log
```

Сохранение лога по источнику с кодом 1 за 0 часов 1-го января 2008 года в файл **/tmp/log**

Команды **isload** и **sourcelist** используются для связки IPN коллектора с внешними коллекторами в режиме просто обработчика.

Наладка приёма данных коллектором в автономном режиме

Если коллектор запустился и поток приходит с одного из обслуживаемых источников в каталоге, указанным в переменной **log.dir**, должны появиться подкаталоги source_<код источника>, а в них каталоги с годом, месяцем и бинарные файлы с трафиком.

В случае если этого не произошло:

1. Сделайте сброс кэша коллектора командой **./netflow.sh flush** для UNIX, **netflow.bat flush** для MS;
2. Установите уровень логирования в файле **log4j.xml** из **INFO** в **DEBUG** (атрибут /log4j:configuration/root/priority/@value);
3. Проверьте файл **collector.log** на наличие строки: Starting FlowListener on port 2001. Если строки нет - значит опция **load** в **netflow_ipn.properties** не установлена в 1 или не указаны порты приема данных;
4. Посмотрите файл **loader.log** на наличие ошибок, проверьте опции экспорта NetFlow-потока на источнике, номер порта;
5. Посмотрите файл **loader.log**, если в нем присутствуют строки "Ignore packet from ...", это значит, что код источника с данным адресом не присутствует в переменной sources файла **netflow_ipn.properties**. Список обслуживаемых источников перечислен в логге **collector.log** в виде:

```
Reload source list
IP: X.X.X.X => Y
```

Добившись приёма файлов в лог следует перейти к следующей стадии - обработке первичных логов коллектором.

Настройка обработки данных

При установке опции **process=1** в файле **netflow_ipn.properties** коллектор начинает принимать задания на обработку часовых логов для "своих" источников. Команды передаются в следующих случаях:

1. При переходе часа генерируются задания на обработку логов прошедшего часа;
2. Если количество минут текущего часа кратно параметру **generate.minutes** в файле **netflow_ipn.properties** генерируются задания на обработку логов текущего часа;
3. При передаче сторонним ПО команды **isload** коллектор помечает логи загруженными и запускает их обработку;
4. Обработка, либо переобработка логов может быть запущена администратором системы в **Менеджере источников** модуля.

При повторной переобработке логов за какой-либо час данные не суммируются, переобрабатывать логи можно неограниченное количество раз. Весь контроль за обработкой данных осуществляется через вкладку **Менеджер источников**.

Источники

Менеджер источников

Интерфейсы и зоны

Шлюзы

Период

Ноябрь 2007

Источники

- ATC-53
- ATC-97
- ATS-32
- ATS-65
- ATS-74
- ATS-90
- pagios
- TEST
- test-pproe-server
- АМТС г. Екатеринбург
- АМТС г. Челябинск
- АТС-22 г. Магнитогор
- АТС-26
- АТС-37 г.Магнитогорс
- АТС-41
- АТС-60
- АТС-62/1
- АТС-65 Новый
- БД АльфаЦентр
- БД Карнавал
- БД Магнитострой
- БД Скиф
- БЦ Верона
- БЦ Горки (Сити-Парк)
- БЦ Домэк
- БЦ Звезд

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Загружен В загрузке Обработан В обработке Помечен Нулевой объем

Модуль Выделенные каналы

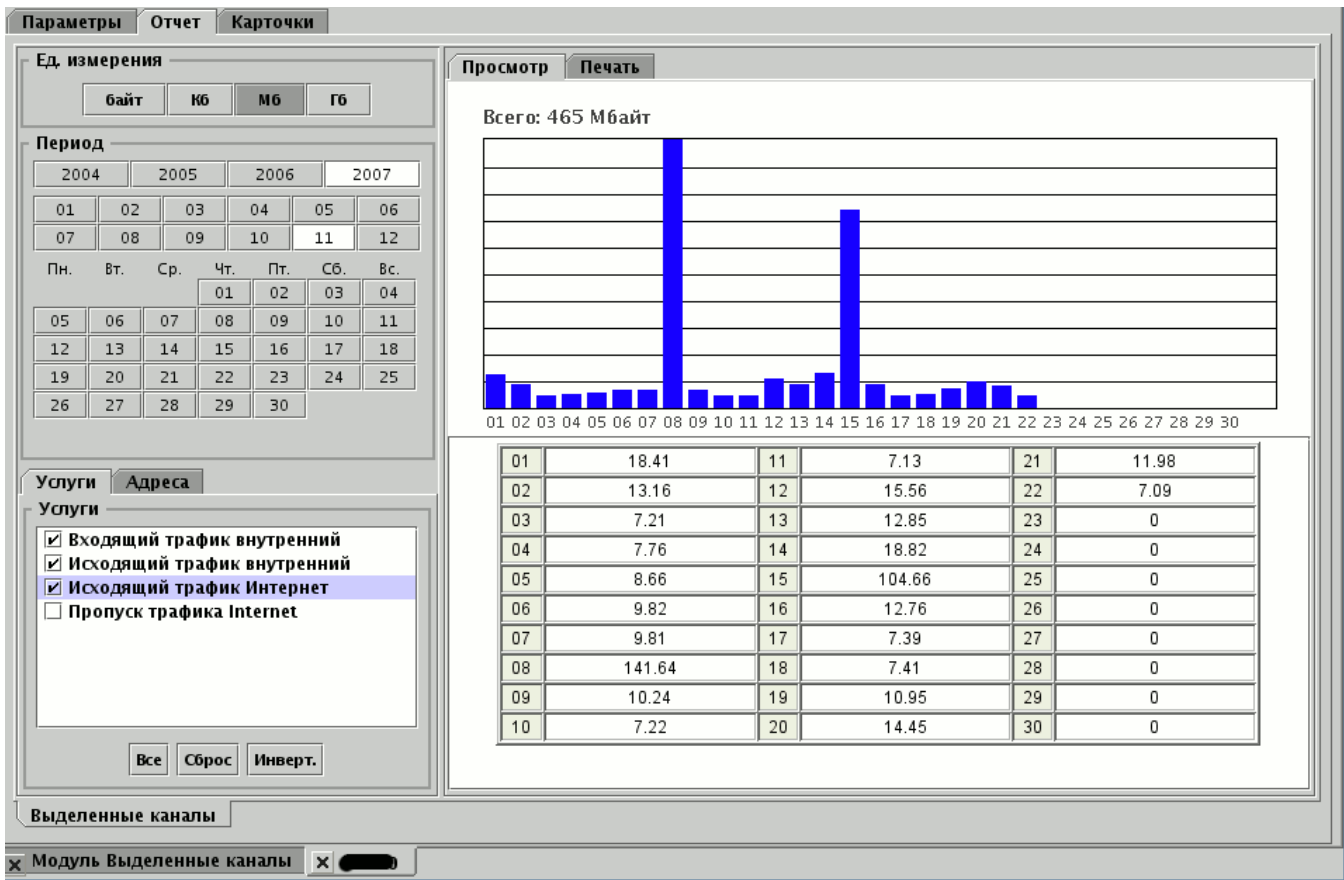
Для просмотра состояния логов по источнику выберите месяц и источник в списке слева. Загруженные и обработанные логи выглядят как на приведенном выше снимке экрана.

Чтобы переобработать логи за какой-либо час - выберите мышью требуемые часы, затем вызовите меню нажатием правой кнопкой мыши. В появившемся меню добавьте в обработку логи, либо для выбранного источника, либо для всех источников модуля. Нажимая кнопку **Обновить** по изменению цвета можно отслеживать обработку логов коллектором.

Если по какой-либо причине необходимо полное перечитывание лог-файла с последующей переобработкой (например, произошла подмена ошибочных лог-файлов на исправленные), то можно добавить логи в загрузку. Аналогично, необходимо выбрать требуемые часы, вызвать контекстное меню нажатием правой кнопкой мыши и далее **Добавить в загрузку**. Данная опция есть в наличии только в случае, если у источника установлен флаг **use.load=1** в конфигурации.

Если логи для какого-либо источника за какой-либо час принимаются коллектором, либо переданы командой **isload** извне, а файл пуст, либо его объем минимален, то квадрат логa помечается красной точкой в центре, что означает **Нулевой объем**.

По результатам обработки в договорах на вкладке **Отчет** при выборе интересующего месяца и услуг должен появиться отчет по трафику.



Отчет может быть просмотрен за день, месяц и год, в байтах, Мб и Гб. Если вопреки ожиданиям трафик договора не отображился, выполните следующие шаги:

1. Проверьте наличие трафика по адресам абонента в бинарных логах за час. Для этого используйте команду `./netflow.sh save` для UNIX-систем, `netflow.bat save` для MS-систем. В параметрах команды передаются код источника, час лога и файл, в который лог будет экспортирован в виде текста. Для уточнения синтаксиса запустите `netflow.sh(.bat)` без параметров;
2. Попробуйте для начала сделать правила в привязках, в которые попадет весь трафик, любой входящий и любой исходящий. Если вам не получается отфильтровать требуемый вам вид трафика - также обратитесь к первичному логу.

Ошибки обработки трафика фиксируются в **Сервис=>Журналы=>Журнал ошибок**. Там отображаются пакеты, для которых не была определена услуга. Для просмотра журнала выберите модуль в выпадающем списке и дату.

Подсистема аудита

Модуль может осуществлять простейшую проверку на предмет нецелевого использования адресной ёмкости. Неучтённые адреса обнаруживаются по признаку отсутствия начисления по этому адресу какому-либо клиенту биллинга.

Есть некий интернет провайдер, обладающий следующими диапазонами адресов: 8.8.8.1-8.8.8.241 и 9.9.9.4-9.9.9.200. Большая часть этих адресов роздана клиентам. Для контроля за неиспользованием оставшимися в конфигурации модуля IPN указывается следующее:

```
#
audit=1
# , ""
audit.email=admin@prov.com
audit.range.1=8.8.8.1-8.8.8.241
audit.range.2=9.9.9.4-.9.9.9.200
```

В результате обработки часового лога источника, если были нарушения аудита, будет сброшено письмо с темой: "Отчет аудита IPN". Содержимое письма будет примерно таким.

```
@21.07.2006 14
: 8.8.8.1-8.8.8.241 => 115740/634314674
```

В первой строке - название источника и время лога. Далее диапазоны, количество строк лога с нарушением и количество байт. Выборка записей лога с нарушениями - задача в данный момент в биллинге не реализованная.

Тарификация

В тарифном плане должна быть определена цена каждой услуги, потребляемой клиентом. Тарификация - отдельный процесс, использующий данные обработки логов и никак с ней не связан. О редактировании тарифных планов и их отнесении к договору вы можете почитать в документации к [основной части программы](#). Там же описана логика работы тарифных деревьев и поведения стандартных узлов тарифных деревьев, общих для всех модулей.

Для тарификации на верхнем уровне в тарифе может находиться либо услуга либо мультиуслуга и ничто другое. В тарифах так же кроме цены могут задаваться Типы правила для шлюза . Они описаны в [следующей главе](#).

На один день в договоре может быть активно несколько тарифных планов с поддеревьями модуля, определяющих цены на разные услуги. Однако описание стоимости для каждой услуги должно быть только в одном из них. Не допускается описание цены одной услуги в нескольких тарифах, которые могут быть установлены в одном договоре параллельно. Логика поиска тарифа соответствует [Алгоритму 2](#).

Если у вас не было тарифного плана, создайте его, создайте для него поддерево, либо расширьте от другого тарифа. После того, как вы откроете дерево, в нем должен отобразиться узел со значком графика и названием экземпляра модуля IPN.

Создание тарифных планов

В тарифном запросе модуля IPN передаются следующие параметры:

- код потребляемой услуги;
- время момента потребления.

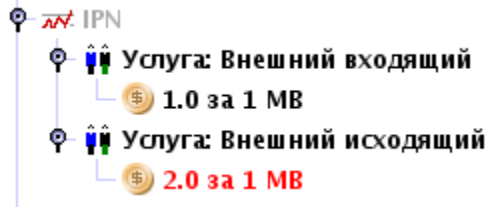
В ответе возвращаются:

- стоимость услуги (обязательно);
- параметры доступа (опционально);
- категория трафика (опционально).

Дополнительные примеры тарифных планов модуля IPN доступны на нашем [Wiki](#).

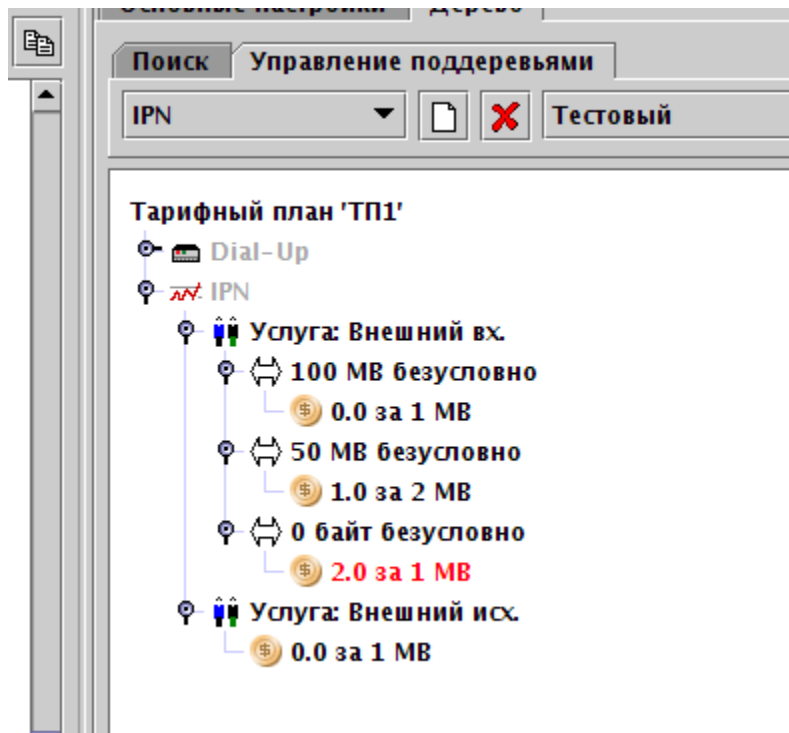
Простейший тариф1

Следующий пример тарифа устанавливает фиксированные платы 1 и 2 рубля за МБ за входящий и исходящий трафики.

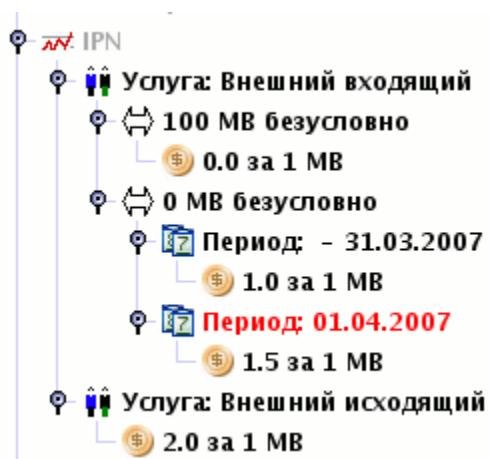


Тариф с зависимостью стоимости от объема

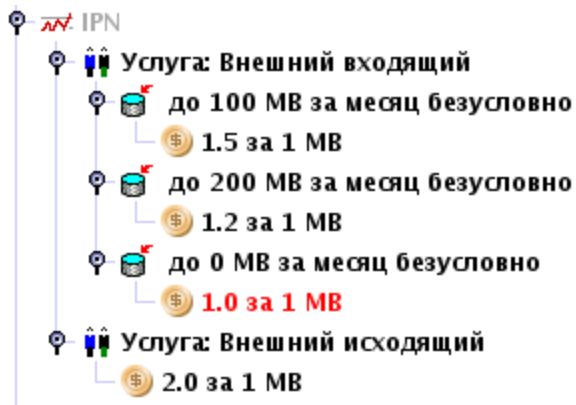
100 МБ бесплатного Внешнего входящего трафика, следующие 50 МБ по цене 1 руб. за МБ, а далее тарификация производится по 2 рубля за МБ. Квота не зависит от того, сколько проработал клиент в данном месяце.



Предположим, что до 31 марта цена мегабайта превышения была 1.0 рубля, а с 1 апреля стала 1.5 рубля. Возможно заранее скорректировать тариф подобным образом:



В случае, если наработка по **Внешнему входящему трафику** до 100 МБ, он тарифицируется по 1.5 руб/МБ, от 100 до 200 - по 1.2 руб/МБ, свыше 200 - по 1 руб/МБ. Т.е. в зависимости от объема меняется цена мегабайта на весь объем.



Как можно понять из примеров 0 означает бесконечность в узлах типа **Ограничение по объёму** и **Диапазон**. Узлы **Диапазон** и **Ограничение по объёму** могут работать в режимах: **безусловно**, **пропорционально периоду разрешённой услуги**, **пропорционально периоду действия тарифа**. При этом количество постоянно, вычисляется пропорционально периоду действия разрешённой услуги в договоре, либо периоду действия тарифного плана соответственно.

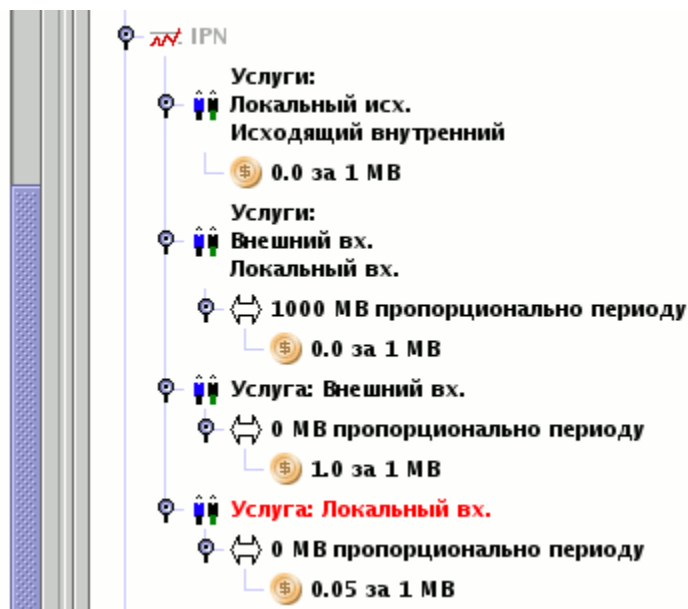
Например, если при режиме **пропорционально периоду разрешённой услуги** месяц содержит 30 дней, а услуга добавлена в разрешенные с 15 го числа, то все объёмы автоматически поделятся пополам. Если разрешенная услуга не добавлена, узел отработает в безусловном режиме.

Использование узла "Мультиуслуга" 1

Вместо нескольких узлов **Услуга** возможно создание узлов типа **Мультиуслуга**. Данный узел аналогичен по функциям узлу **Услуга**, но пропускает в себя запросы цены для нескольких указанных в нем услуг. Использование данного узла целесообразно для:

- указания в одном узле цен для нескольких услуг с одинаковой стоимостью;
- создания тарифов с квотами трафика, в которых начальная квота общая для нескольких услуг.

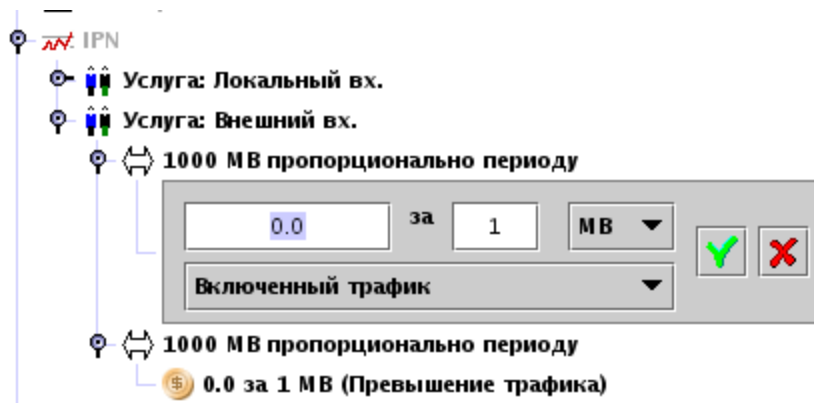
Рассмотрим оба этих случая в едином примере:



В данном примере стоимость услуг исходящих трафиков равна 0. На услуги входящего локального и внешнего трафиков выделена общая бесплатная квота 1000МБ, при превышении которой входящий внешний трафик тарифицируется по 1 руб/МБ, а локальный по 0.05 руб/МБ.

Детализация по тарифу модуля IPN

Детализация по тарифу позволяет в счетах модуля бухгалтерии разделить наработку по какой-либо из услуг, посчитанных по различным ценам или условиям. Например, можно вынести в отдельные строки счета бесплатный входящий трафик и платный трафик. Суть метода состоит в определении категорий услуги в конфигурации модуля и указании их в выпадающем списке узла **Стоимость услуги**. В результате каждая протарифицированная единица услуги относится к той или иной категории.



О детализации по тарифу написано подробно [в документации по настройке позиций модуля бухгалтерии](#).

Комбинированный тариф

Для того, чтобы показать всю возможную сложность создаваемых тарифных планов, создадим следующий тариф.

Таблица. Описание тарифа

Услуга	Правила
внешний исходящий	до 100 МБ в месяц бесплатно, далее - по 0.5 руб за 1 МБ
внешний входящий	трафик делится на ночной с 0 до 8 часов, дневной с 9 до 23 часов. ночной трафик - если до 10 МБ в месяц, то бесплатно, иначе все по (в т.ч. первые 10 МБ) по 1 руб за 1 МБ, дневной трафик - по 0.8 руб за 1 МБ
локальный входящий	с 8 утра до 8 вечера по 1.3 руб за 1 МБ, с 9 вечера до 23 часов по 1.1 руб за 1 МБ, с 0 часов до 7 утра - бесплатно
локальный исходящий	бесплатно

Тарифный план настраивается следующим образом.

Тарифный план 'Выделенка'

- DialUp
 - IPN
 - Услуга: Внешний вх.
 - Набор ограничений: Часы => 0 - 8
 - до 10 МБ за месяц
 - 0.0 за 1 байт
 - до 0 МБ за месяц
 - 1.0 за 1 МБ
 - Набор ограничений: Часы => 9 - 23
 - 0.8 за 1 МБ
 - Услуга: Внешний исх.
 - 100 МБ
 - 0.0 за 0 байт
 - 0 байт
 - 0.5 за 1 МБ
 - Услуга: Локальный вх.
 - Набор ограничений: Часы => 8 - 20
 - 1.3 за 1 МБ
 - Набор ограничений: Часы => 9 - 23
 - 1.1 за 1 МБ
 - Набор ограничений: Часы => 0 - 7
 - 0.0 за 1 байт
 - Услуга: Локальный исх.
 - 0.0 за 1 байт
 - VOIP

В реальности тарифы могут быть и сложнее. С помощью элемента **Набор ограничений** вы можете добавлять тарифы выходного дня, тарифы праздничных дней.

Следующий пример тарифа: клиент получает предоплаченные 500 000 000 байт бесплатно, стоимость остального трафика различается в зависимости от суммарной наработки:

- до 1 500 000 000 - по 0.11 за 1 000 000 байт;
- до 2 500 000 000 - по 0.95 за 1 000 000 байт;
- если выше - по 0.08 за 1 000 000 байт.

Дерево в данном случае будет выглядеть так:

Выделенные каналы

Услуга: Входящий трафик Интернет

- до 500000000 байт за месяц пропорционально периоду
 - 0.0 за 1 байт
- до 1500000000 байт за месяц пропорционально периоду
 - 500000000 байт пропорционально периоду
 - 0.0 за 1 байт
 - 0 байт безусловно
 - 0.11 за 1000000 байт
- до 2500000000 байт за месяц пропорционально периоду
 - 500000000 байт пропорционально периоду
 - 0.0 за 1 байт
 - 0 байт безусловно
 - 0.095 за 1000000 байт
- до 0 байт за месяц пропорционально периоду
 - 500000000 байт пропорционально периоду
 - 0.0 за 1 байт
 - 0 байт безусловно
 - 0.08 за 1000000 байт

Тарифные опции в модуле IPN

В зависимости от активированных опций можно менять логику тарифа. Ветка Тарифная опции отрабатывает, если в момент обсчета одна из указанных в ветке опций активна. В случае срабатывания одной из веток, последующие этого же уровня не отработают, даже если указаны те же опции. Отсутствие указанных опций означает, что ветка отработает в том случае, если не было срабатывания ветки Тарифная опция выше.

Для того, чтобы сделать понижение стоимости трафика на период, можно создать такой тариф:



При отсутствии активированных опций запрос попадет во вторую ветку и цена будет 2.0 . При активации опции Дешевый трафик, запрос попадет в первую ветку и цена будет 1.0.

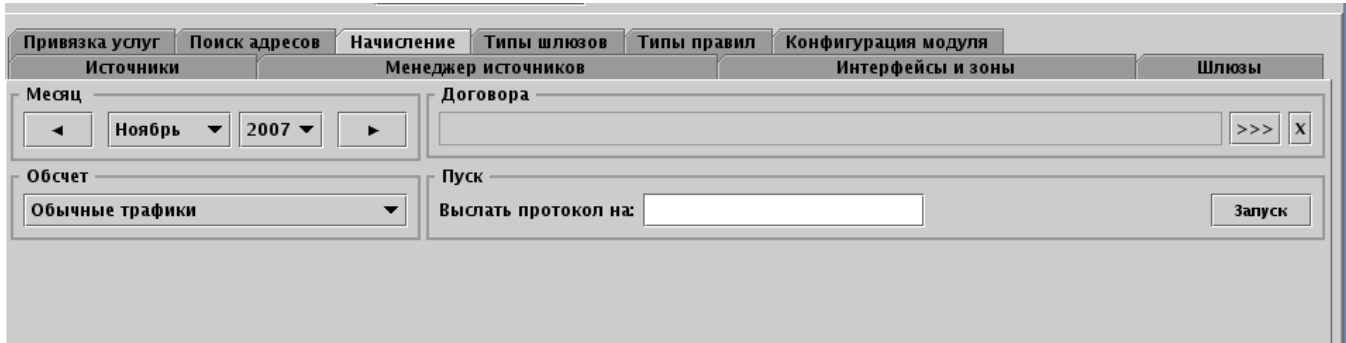
При работе тарифных опций в этом модуле есть такая особенность - тарифные опции при обсчете округляются до часа. Это связано с тем, что наработка в модуле IPN хранится одной цифрой за весь час и точности предоставляемых данных не хватает для более точного реагирования на тарифные опции.

Запуск начисления

Начисление производится планировщиком заданий, он должен быть запущен.

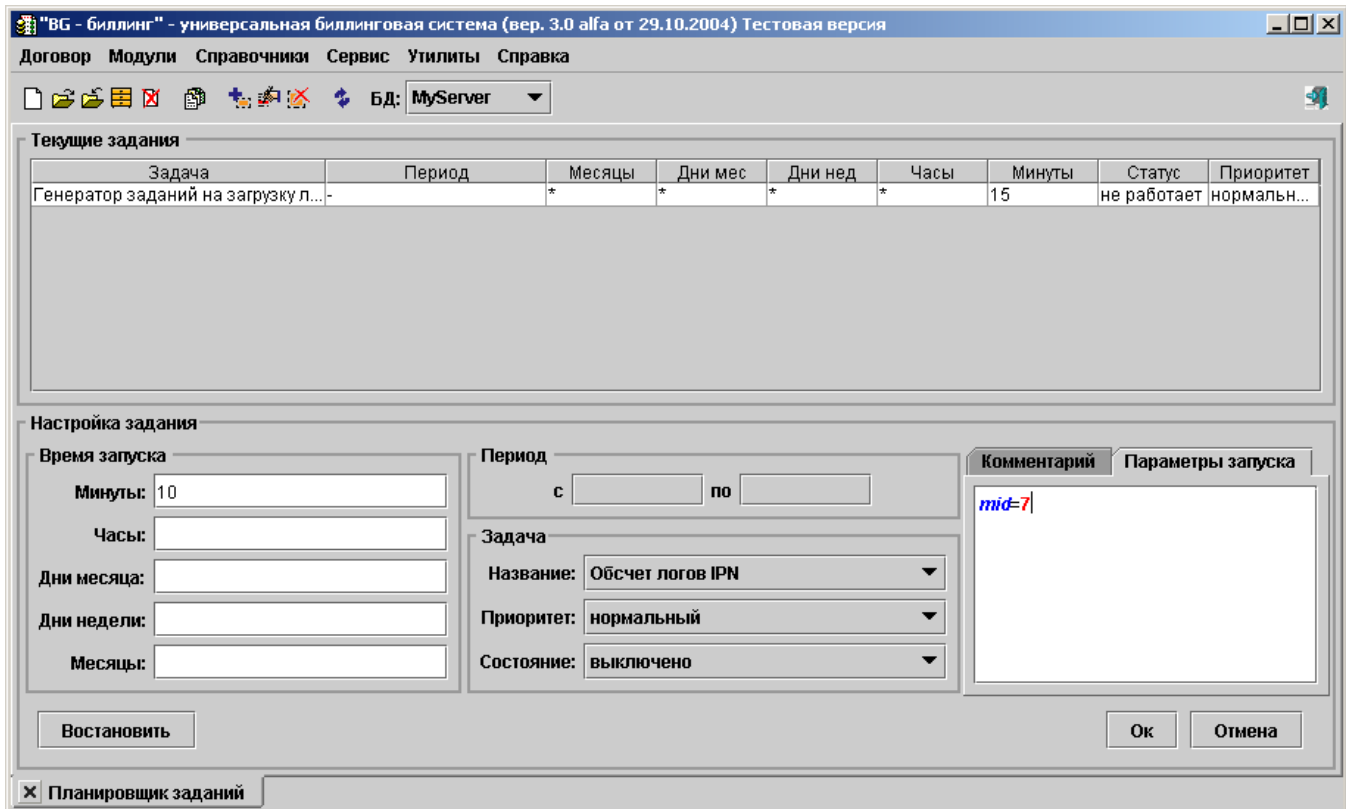
Асинхронное начисление производится на вкладке модуля **Начисление**. В выпадающем списке должны быть выбраны трафики для обчёта. Если в фильтре договоров не выбраны договоры, производится тарификация всех договоров за выбранный месяц.

По окончании начисления будет выслано письмо на указанный ящик. Если письмо не пришло, проверьте логи **scheduler.out**, **scheduler.log** на наличие ошибок. Возможная причина - неверные опции настройки почты в конфигурации сервера биллинга.



Ручной режим начисления используется в период наладки модуля, либо при необходимости перетарификации трафика абонента (абонентов) в связи с ошибками в тарифах, либо с ошибками в привязках услуг и, как следствие, неверной наработкой по услугам в договоре. При каждом обчёте происходит перетарификация всего месяца.

Автоматический режим реализуется добавлением задания **Обсчёт логов IPN** в планировщик.



Задача	Период	Месяцы	Дни мес	Дни нед	Часы	Минуты	Статус	Приоритет
Генератор заданий на загрузку л...		*	*	*	*	15	не работает	нормальн...

Настройка задания

Время запуска: Минуты: 10, Часы: , Дни месяца: , Дни недели: , Месяць:

Период: с по

Задача: Название: **Обсчёт логов IPN**, Приоритет: **нормальный**, Состояние: **выключено**

Комментарий: **mid=7**

Восстановить, Ок, Отмена

Обратите внимание на строку **mid=7** в параметрах задачи. Вместо 7 поставьте код вашего модуля IPN (можно посмотреть в **Модули=>Редактор модулей и услуг**).

При выполнении задачи начисления задача берет предыдущий минуте выполнения час и обсчитывает логи за этот месяц. Сделано это для того, чтобы можно было реализовать "Последний обсчёт" трафика за месяц, после обработки логов за последний час месяца. Кроме задачи периодического обсчёта логов добавьте точно такую же задачу, запускающуюся в 0 часов 50 минут 1 го дня любого месяца.

Если используете детализацию по тарифу в модуле бухгалтерия, то для автоматического переобсчета детализации, в задачу "Последнего обсчета" добавьте параметр `tariff_detail=1`, в таком случае после последнего обсчета, будет произведен переобсчет с вычислением детализации по тарифу.

После выполнения тарификации в балансе договора должна появиться денежная наработка по обчисляемым трафикам.

Начисление наработки за максимальные трафики (IPN)

Начисление наработки за максимальные трафики осуществляется дополнительно к основной тарификации модуля периодически в ручном режиме, либо используя планировщик. В конфигурации модуля должны быть определены зависимости услуг, представляющих собой максимальные трафики, например:

```
max.traffic.74=39,40
```

В данном случае услуга с кодом 74 представляет собой максимум между услугами с кодами 39 и 40.

Начисление осуществляется по следующему алгоритму:

1. выбираются все договоры с разрешённой услугой типа "Максимальный трафик" за обчисляемый месяц;
2. выбираются действующие у клиента тарифные планы в период действия услуги, получая наборы: договор - услуга - тариф - период;
3. для каждого пункта набора осуществляется тарификация, причём дата в тарифном запросе передаётся равной последнему дню набора.

Предположим, что у нас есть договор X, у которого с 3 сентября по 20 сентября разрешена услуга **Макс. трафик 1**. Предположим также, что в течении сентября у договора был **Тариф1** с 1 по 10 и **Тариф2** с 11 по 30.

В данном случае будут обсчитаны две позиции:

1. услуга **Макс. трафик 1**, вычисленная на период с 1 по 10 по тарифу **Тариф1**;
2. услуга **Макс. трафик 1**, вычисленная на период с 11 по 30 по тарифу **Тариф2**.

Для начисления максимальных трафиков можно использовать ручной режим, используя вкладку **Начисление** модуля.

Привязка услуг | Поиск адресов | **Начисление** | Типы шлюзов | Типы правил | Ресурсы | Конфигу

Источники | Менеджер источников | Ин

Месяц: Май 2010

Обсчет: Максимальные трафики

Договора

Пуск: Выслать протокол на: email@server.domain | Комм

Для автоматического начисления необходимо настроить задачу **Максимальные трафики IPN** в планировщике задач. В конфигурации задачи должно быть установлено:

```
mid=< >
```

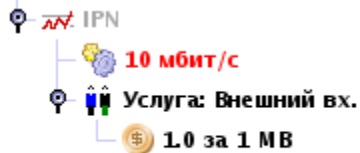
Как и обсчёт логов DialUp задача берет месяц, отнимая час от текущего времени. Это позволит вам обсчитать все трафики по окончанию месяца, если запуск задачи будет установлен на 0 часов 55 минут последующего месяца. При этом необходимо настроить сброс сессий пользователей на границе месяца.

Установка типа правила в тарифе.

В тарифном плане можно изменять правило Тип правила шлюза с помощью узла **Тип правила**. Типы правил, их смена поддерживаются не всеми типами шлюзов. Описаны [тут](#).

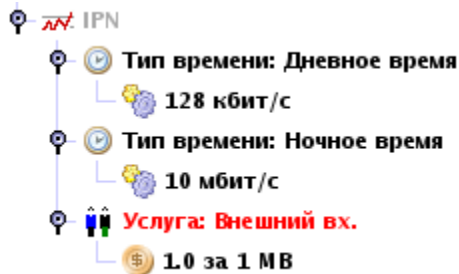
В тарифном плане алгоритм работы узла **Тип правила** похож на алгоритм работы узла **Стоимость услуги**. Основное отличие типа правила от стоимости услуги в том, что тип правила может быть определен не только внутри узла **Услуга** - он может быть на любом уровне в тарифе.

Это 2 разных запроса: получить стоимость, получить тип правила. Минимально в тарифе должна быть определена стоимость для тарификации. Наличие типа правила в тарифе не является необходимым. Для того, чтобы показать простейший тариф с типом правила, возьмем простейший тариф со стоимостью и добавим туда тип правила:



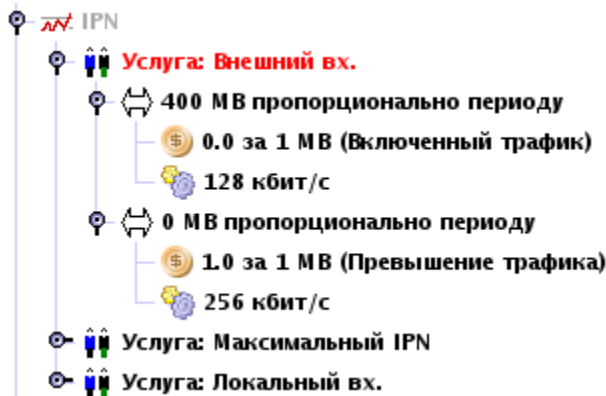
В данном тарифе, сам тариф определяет сразу тип правила.

Можно усложнить этот тариф условиями. Например так:



Тут мы установили разные тарифные опции в зависимости от времени суток. Можно сделать аналогичный тариф с узлами **Период**, **Фильтр** по времени и т.п.

Также можно делать тарифные планы, зависящие от наработки в конкретной услуге. Например вот:



В данном случае добавлены 2 типа правил - 128 кбит/с для первых 400 МБ и 256 кбит/с для остального трафика. Добавлять тип правила в более, чем одну услугу не имеет смысла, т.к может отработать любая из этих веток, если есть трафик по обеим услугам.

Саму процедуру смены правил по тарифу делает задача **Обсчета логов IPN**. Но эта возможность является опциональной и включается с помощью настройки в конфигурации задачи:

```
set.rules=1
```

также есть дополнительный параметр

```
rule.error=1
```

Если его поставить в 1, то если в тарифе не будет найден тип правила, то сообщение об ошибке появится в логах коллектора (и вышлетса на почту) по аналогии с ситуацией, когда цена в трайфе не найдена. По умолчанию эти ошибки не логируются.

Настройка шлюзов

Шлюз - устройство, ограничивающее доступ абонента к интернету. В зависимости от статуса, установленного на вкладке **Шлюз** в договоре, биллинг передаёт управляющие воздействия на добавленные договору шлюзу с целью ограничить или открыть доступ в сеть.

Команды шлюзу передаются либо асинхронно, при смене статуса шлюза администратором, либо периодически по выполнению задания **Проверка шлюзов IPN**. Второй режим нужен в случае, когда статус шлюза изменяется самим модулем по состоянию баланса абонента.

Состояние шлюзов может быть следующим:

- **Открыт** - шлюзы открыты;
- **Закрыт** - шлюз закрыт, но может быть открыт пользователем через страницу статистики (например, абонент уехал на несколько дней);
- **Заблокирован** - шлюз закрыт за задолженность на балансе;
- **Жёсткая блокировка** - шлюз закрыт и может быть открыт только администратором;
- **Удалён** - информация о пользователе удаляется из шлюзов (актуально для шлюзов, которым передаётся информация даже о закрытых абонентах, например, привязка маков к порту коммутатора);

Задача **Проверка шлюзов IPN** должна запускаться периодически, примерно раз в 15 минут. Работа задачи разделена на две стадии: смена статуса должников, синхронизация правил. В параметрах запуска задачи должно быть установлено:

```
mid=< IPN>
```

И опционально:

```
email=< >  
thread.count=< , 100>
```

В первой фазе задача изменяет статусы на **Заблокирован** открытым должникам. Должником считается договор, остаток баланса которого менее допустимого **Лимита**. Эту функцию можно отключить, если поставить в конфигурации задачи :

```
lock.status=0
```

Открытие должника происходит по событию прихода платежа на его счёт, при этом абонент может быть переведён в статус **Открыт**, либо **Закрыт** - это регламентируется переменной **status.after.unlock** конфигурации модуля. Если абонент переводится в статус **Закрыт**, он должен сам открыть шлюз через страницу статистики. По платежу открываются шлюзы договоров только с текущим активным **статусом**.

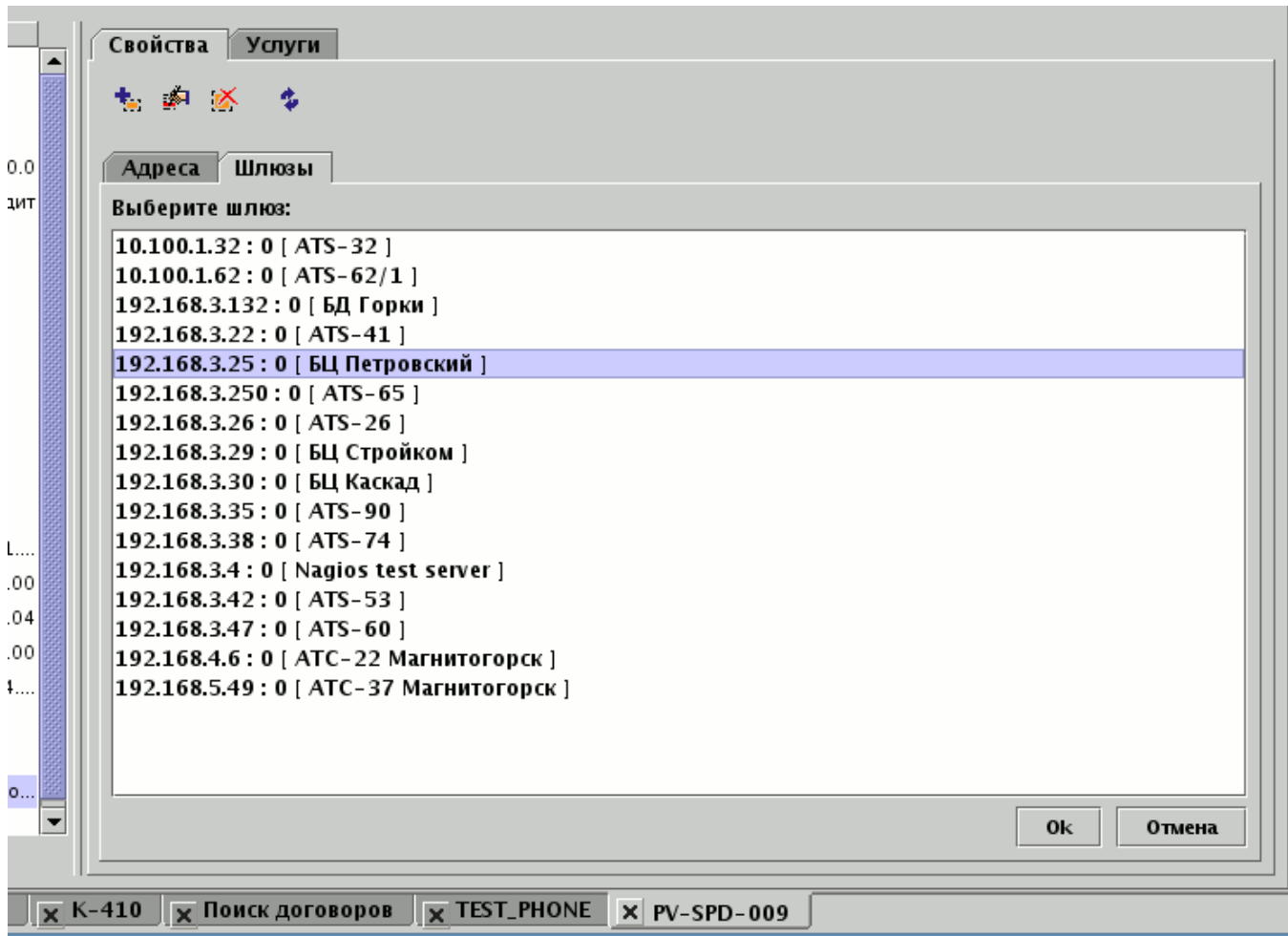
При смене статуса договора изменяется состояние его шлюзов. При всех, кроме активного, статусах, если состояние шлюзов **открыт**, они переходят в состояние **заблокирован**, при изменении статуса договора на **активный**, если состояние шлюза **заблокирован**, то он переходит в состояние, указанное переменной **status.after.unlock**.

Даже если абонент будет открыт, модуль лишь меняет статус шлюза, реальное открытие будет произведено при следующем срабатывании задачи **Проверка шлюзов IPN**, во второй её фазе.

Во второй фазе задача сверяет состояние на шлюзе с состоянием в БД и актуализирует состояние шлюза. При этом абоненты могут как блокироваться, так и открываться.

Состояние шлюза изменяется вслед за статусом.

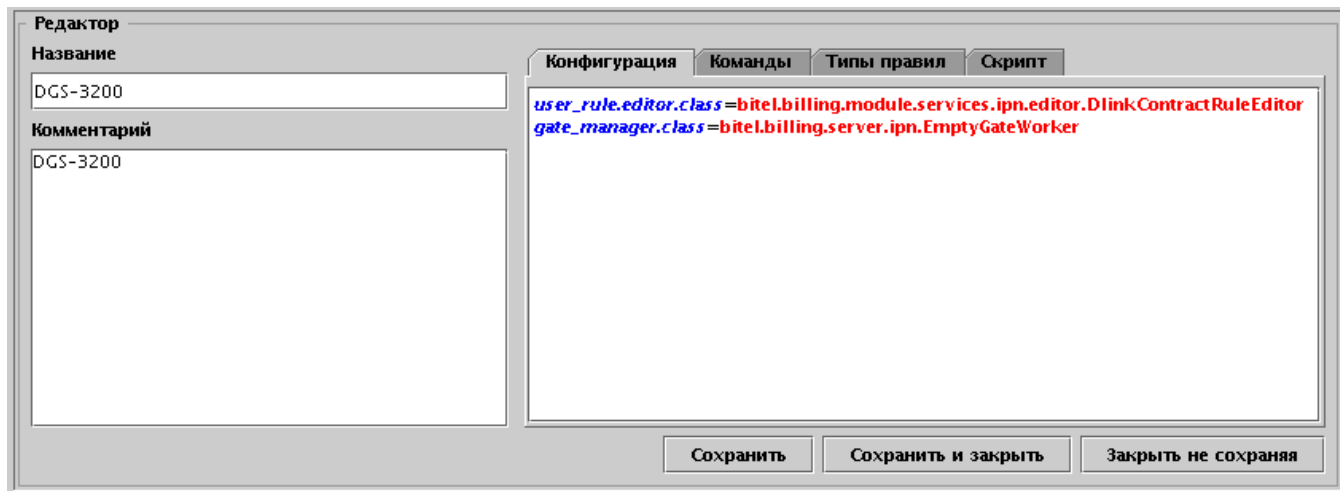
Для добавления шлюза договору выберите вкладку **Шлюзы** в свойствах модуля для данного договора, далее на расположенной чуть выше панели инструментов выберите **Новый элемент**. Выберите шлюз в открывшемся списке.



В зависимости от типа шлюза меняются параметры и приёмы работы с ними, далее подробнее описан процесс работы со всеми поддерживаемыми биллингом типами шлюзов.

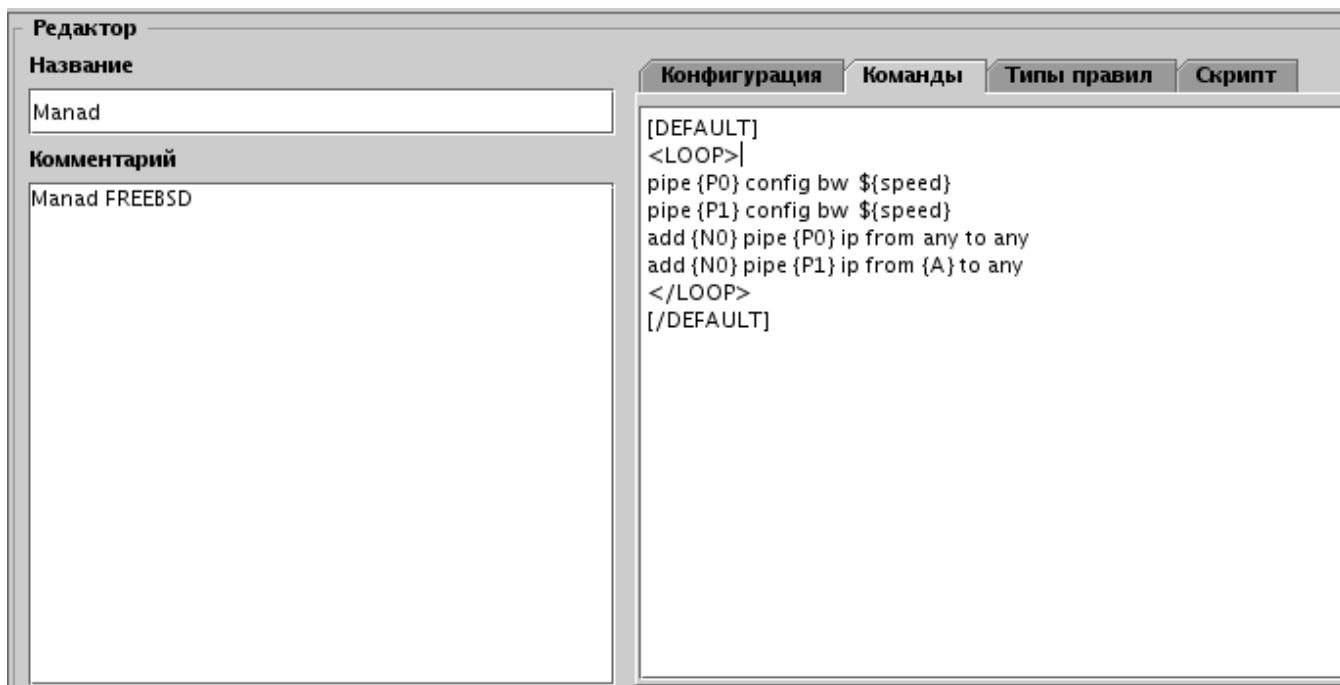
Настройка типов шлюзов

Типы шлюзов настраиваются на вкладке **Типы шлюзов** модуля IPN. Если добавить новый шлюз или редактировать уже добавленный, то откроется редактор типа шлюза. В этом редакторе конфигурация шлюза настраивается на вкладке **Конфигурация** :



The screenshot shows a web-based configuration editor for a gateway. It has a title bar 'Редактор' and a main area with four tabs: 'Конфигурация', 'Команды', 'Типы правил', and 'Скрипт'. The 'Конфигурация' tab is active. On the left, there are two text input fields: 'Название' (Name) containing 'DGS-3200' and 'Комментарий' (Comment) containing 'DGS-3200'. The main configuration area contains two lines of code: `user_rule.editor.class=bitel.billing.module.services.ipn.editor.DlinkContractRuleEditor` and `gate_manager.class=bitel.billing.server.ipn.EmptyGateWorker`. At the bottom, there are three buttons: 'Сохранить', 'Сохранить и закрыть', and 'Закрыть не сохраняя'.

Команды шлюза задаются на вкладке **Команды**:



The screenshot shows the same configuration editor, but with the 'Команды' (Commands) tab active. The 'Название' field contains 'Manad' and the 'Комментарий' field contains 'Manad FREEBSD'. The main configuration area contains a block of shell script code: `[DEFAULT]`, `<LOOP>`, `pipe {P0} config bw ${speed}`, `pipe {P1} config bw ${speed}`, `add {N0} pipe {P0} ip from any to any`, `add {N0} pipe {P1} ip from {A} to any`, `</LOOP>`, and `[/DEFAULT]`. The tabs at the top are 'Конфигурация', 'Команды', 'Типы правил', and 'Скрипт'.

Команды должны иметь следующий формат:

```
[DEFAULT]

[/DEFAULT]

[RULE ID="12,20"]
[/RULE]
```

То, что заключено в теги [DEFAULT] [/DEFAULT] - это команды шлюза по умолчанию. В тегах [RULE ID=""] [/RULE] - команды для конкретных правил, id которых перечислены через запятую в ID. Количество тегов [RULE ID=""] [/RULE] не ограничено. Эти теги остались для перехода с версии 4.4 на версию 4.5 и не рекомендованы к использованию.

Настройка шлюза

Сами шлюзы настраиваются на вкладке шлюзы модуля IPN:

Источники	Менеджер источников	Интерфейсы и зоны	Шлюзы	Привязка услуг
Фильтр				
Хост:порт	Тип	Адрес	Комментарий	
Шлюзы				
1.1.1.1:1	PPPoE_script_test		Ленина 10/2	
1.1.1.1:1	EMPTY		EMPTY!!!!	
1.1.1.1:1	Manad			
1.1.1.1:28	garant_cisco	с. Ново-Талицы, ул. Садовая...	Рарант-Шлюз	
1.1.1.2:111	SCRIPT_TEST		TEST_SRCIPT	
1.1.1.55:22	SCRIPT_EXTERNAL		SCRIPT_EXTERNAL	
10.10.10.10:0	PPPoE	Уфа, Ленина, д. 1	PPPOE	
10.10.20.1:0	PPPoE		PPPoE LOCALgg	
10.10.30.1:0	PPPoE_TEST		PPPoETest	
10.90.90.90:161	Dlink30xx35xxD	Уфа, Ленина, д. 1	DLink30xx35xxD	
10.90.90.91:161	DLink-test		dlink 3562	
1111:23	CISCO		CISCO_TEST	

Для того, чтобы добавить шлюз, нужно нажать кнопку "Добавить" на панели инструментов . Откроется такой редактор:

Хост и порт шлюза 1.1.1.55 : 22	Комментарий	Конфигурация	Адрес
Ключевое слово 	Город: -----	Улица: -----	Дом / Дробь:
Тип шлюза SCRIPT_EXTERNAL	Подъезд 	Этаж 	
Восстановить	Ок	Отмена	

Для каждого шлюза указывается ip-адрес, порт, ключевое слово, тип шлюза. Шлюз можно добавить как потомок другого шлюза. Шлюзы можно объединять в папки для структуризации. На отдельной вкладке можно задать адрес Шлюза. На вкладке конфигурация располагаются настройки шлюза . Как настраивать конкретные шлюзы (шлюз типа Manad и т.п) описано в соответствующих главах.

Также при нажатии правой кнопкой на шлюз в всплывающем меню доступны следующие пункты:

- 1) "вырезать" и "вставить" . Это позволяет копировать шлюз из одного места в другое (все потомки шлюза также копируются);
- 2) "Договоры" - выводит список договоров, на которых добавлен данный шлюз;
- 3) "Мониторинг портов" - какие порты заняты на данном шлюзе .

Для поиска нужного шлюза по заданным параметрам нужно открыть фильтр шлюза нажав кнопку " Фильтр":

Фильтр

Хост: Тип: Любой Комментарий:

Адрес

Город: любой Район: любой Квартал: любой

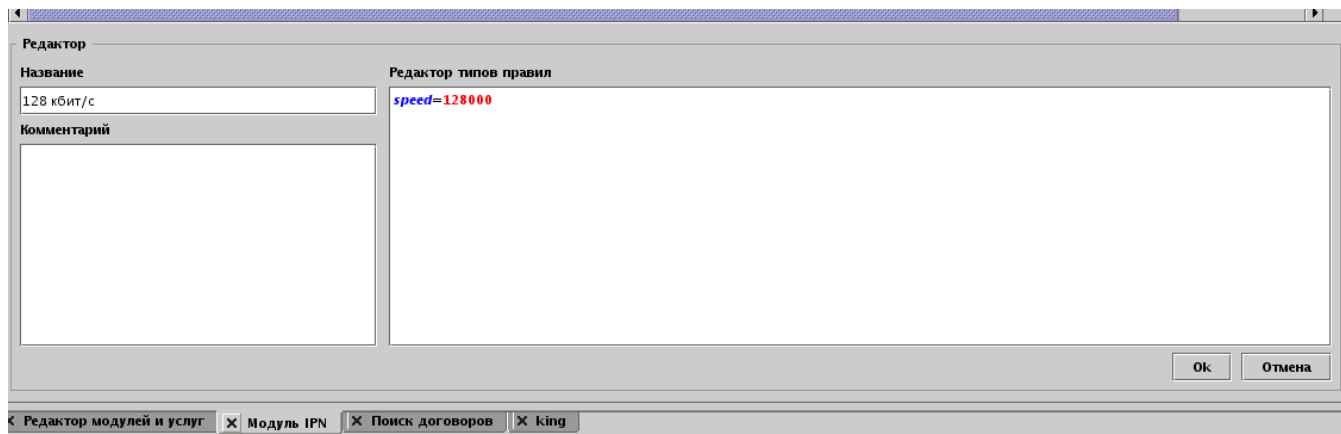
Улица: любой Дом / Дробь: Подъезд: Этаж:

Фильтр

Здесь можно фильтровать шлюзы по хосту, типу, комментарию и адресу .

Типы правил

Типы правил шлюза глобальные, жёстко не привязаны к шлюзам и поддерживаются не всеми типами шлюзов. Они заполняются на вкладке Типы правил. Если добавить новое правило, то откроется редактор правила:



Вот пример кода типа правил :

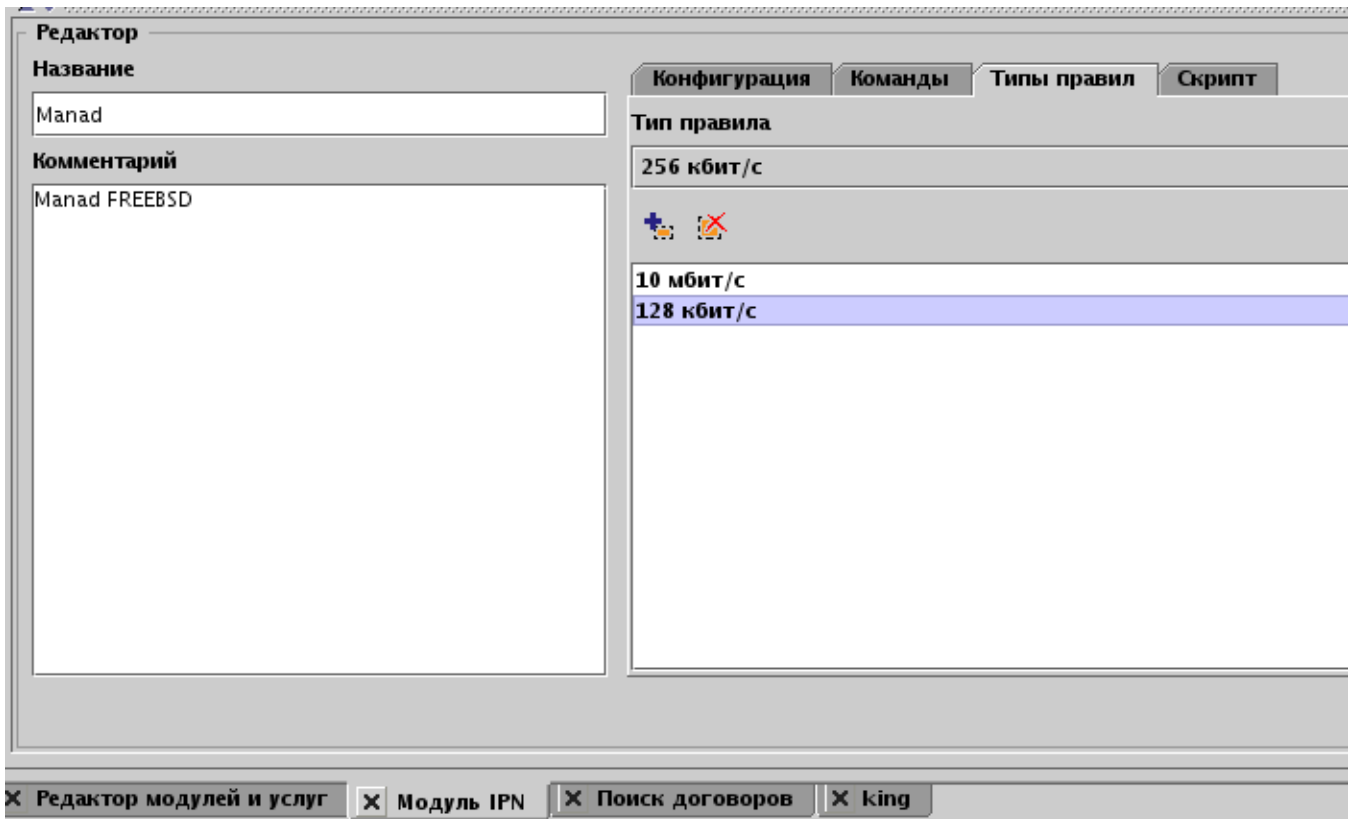
```
speed=128
```

Правила можно воспринимать как параметры, которые потом подставляются в команды шлюзов. Пример команд шлюза :

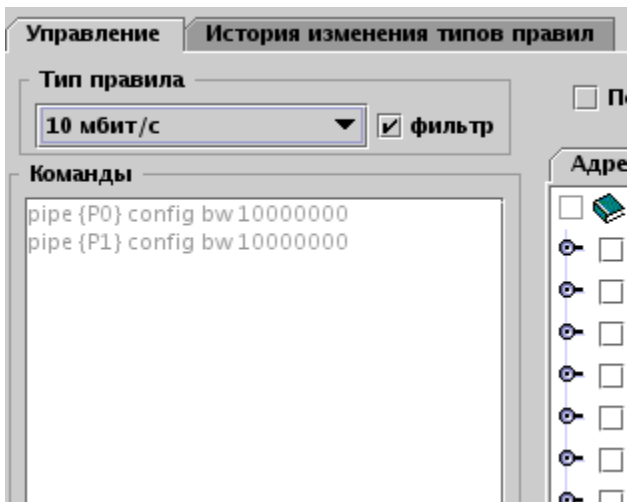
```
[DEFAULT]
ipfw pipe {P0} config bw ${speed}
[/DEFAULT]
```

Все конструкции вида \${param} заменятся на соответствующие значение из правил.

Для каждого типа шлюза можно задать конкретные типы правил на вкладке Правила :



Этот список шлюзов показывается при добавлении [шлюза в договор](#) . Не все шлюзы поддерживают типы правил, более подробно смотрите в главах по конкретным типам шлюзов.



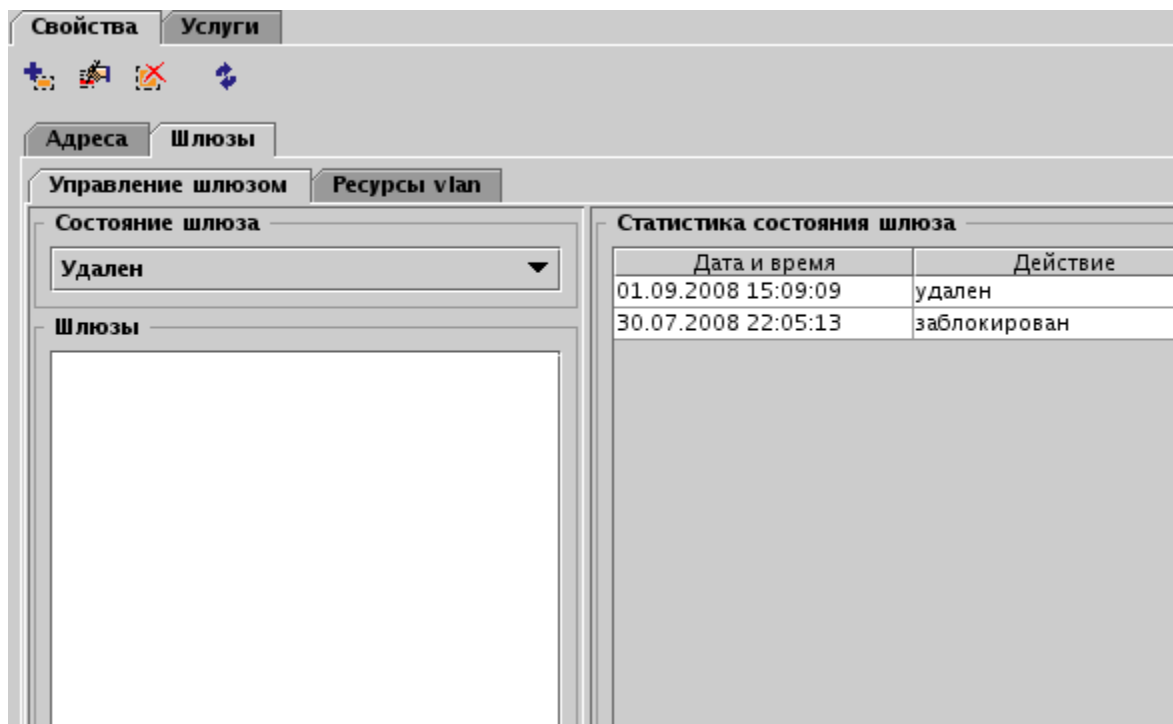
В данном случае в выпадающем списке мы видим те типы правил, которые добавлены для данного типа шлюза . Но если отключить галочку **фильтр**, то мы увидим все типы правил в этом списке. Также тут есть история смены типов правил для данного договора на данном шлюзе :

Управление шлюзом		
Ресурсы vlan		
Порты		
Управление		
История изменения типов правил		
Тип правила	Дата	Пользователь
CISCO_VLAN	24.02.10 12:55:15	Denis Pimenov
Linux_Manad_1024	24.02.10 13:11:45	Denis Pimenov
CISCO-Стандартный к...	06.04.10 16:05:45	Сервер
31	06.04.10 16:14:24	Сервер
10 мбит/с	06.04.10 16:26:46	Пользователь

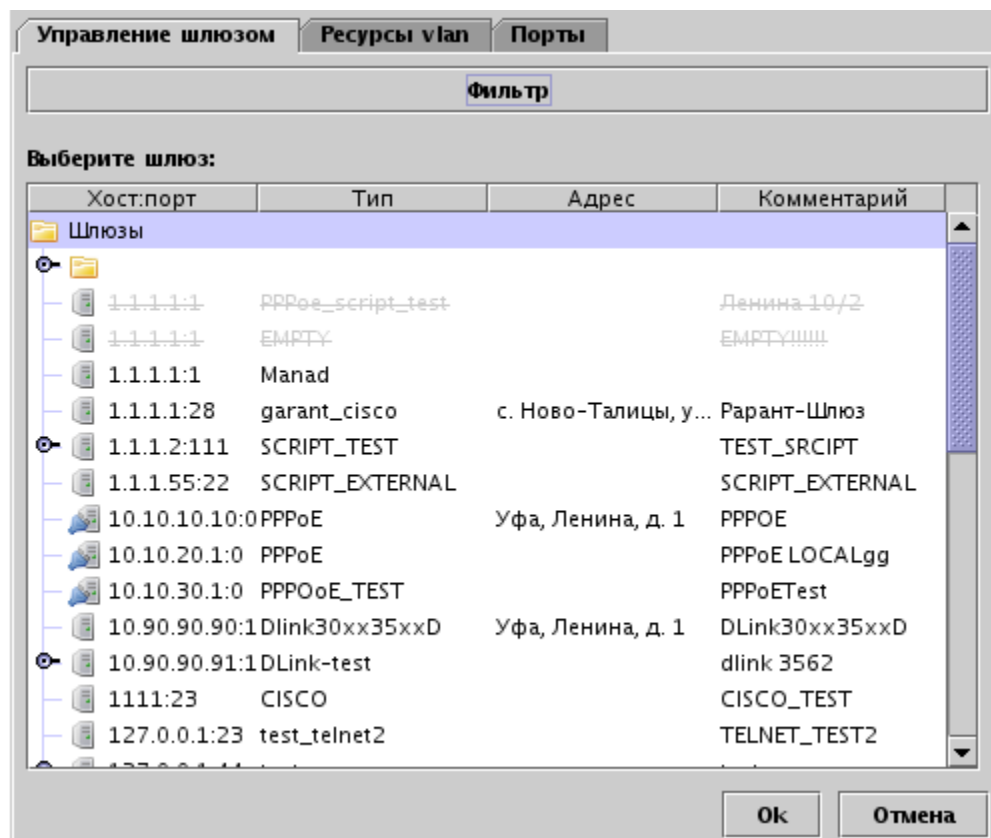
Можно настроить смену типа правил в тарифе. Пример тарифа со сменой типа правил - [тут](#). Тип правила на шлюзе меняет задача "Обсчёт логов IPN". При этом команды на оборудование реально посылаются в задаче "Проверка шлюзов IPN". Смена правил шлюза не поддерживается стандартными встроенными шлюзами, т.к., в общем случае, это задача специфическая. Для того, чтобы воспользоваться этой возможностью, нужно делать аналогичные [скриптовые шлюзы](#).

Добавление шлюза в договор.

Для добавления шлюза в договор выбираем модуль IPN в договоре. Затем вкладку Свойства. Потом вкладку Шлюзы.



Там нажимаем кнопку добавления шлюза и видим список шлюзов.



Выбираем нужный шлюз. Если нужна дополнительная фильтрация, то нажимаем на кнопку "Фильтр", откроется дополнительный фильтр, который в точности повторяет фильтр [настройки шлюзов](#). После выбора открывается диалог редактирования шлюза (зависит от типа выбранного шлюза), в нем нажимаем ОК. Шлюз добавлен.

Выделение ресурса VLAN на шлюз.

Можно выделять ресурс VLAN на договор. Некоторые типы шлюзы могут использовать VLAN. Для этого в конфигурации шлюза прописываем строку :

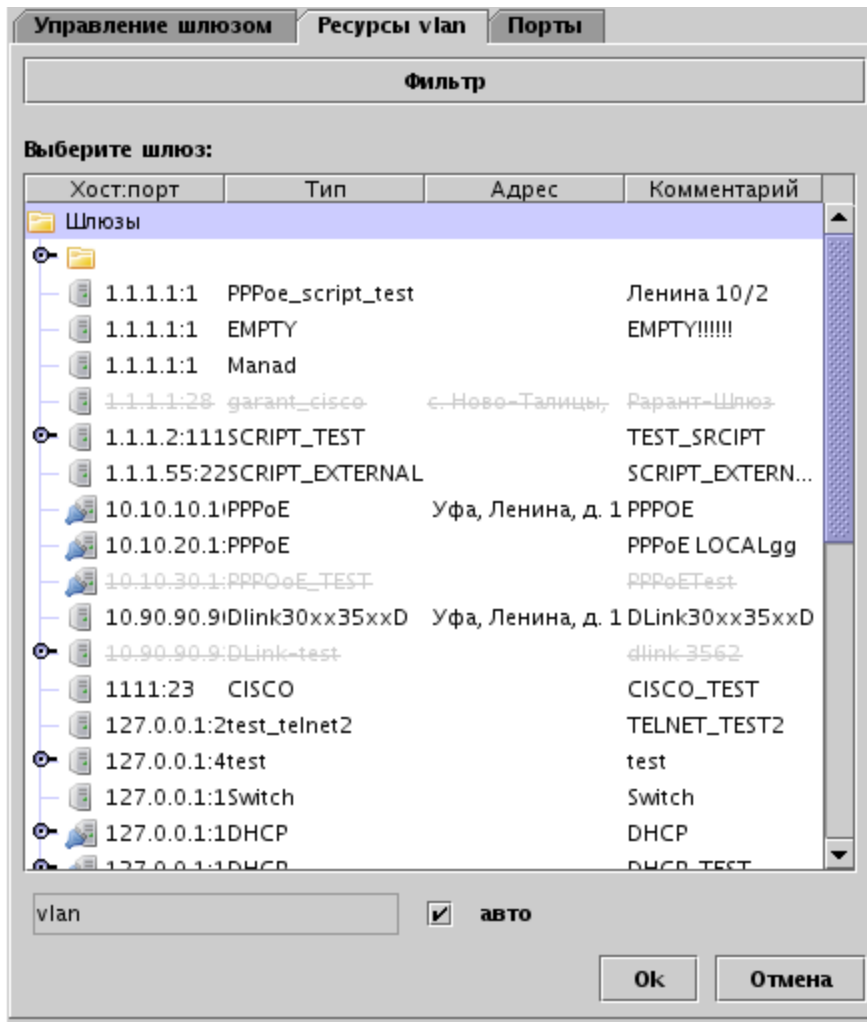
```
range=2-10;15;20-4096
```

Это диапазоны, из которых генерируются свободные VLAN . В данном случае VLAN может быть между 2 и 10, может равняться 15, и быть между 20-4096.

Для выделения VLAN на договор выбираем модуль IPN в договоре ->Свойства->Шлюзы->Ресурсы vlan.

Управление шлюзом	Ресурсы vlan	Порты
Шлюз		
81.200.144.160 : 23 [CISCO_VLAN]		5
127.0.0.1 : 4567 [ffffff]		2
127.0.0.1 : 9023 []		2
1.1.1.1 : 28 [Парант-Шлюз]		6000
10.90.90.91 : 161 [dlink 3562]		3
127.0.0.1 : 4567 [Manad Common]		555
7.7.7.7 : 7777 [yuf,nmjk]		2
10.10.30.1 : 0 [PPPoETest]		2
127.0.0.1 : 5555 [DHCP test]		2

Тут? если нажать на кнопку добавить, то откроется список шлюзов и в нем можно выбрать шлюз, для которого выделяется VLAN для данного договора.



Тут выбираем нужный нам шлюз . Если нужна дополнительная фильтрация, то нажимаем на кнопку "Фильтр", откроется дополнительный фильтр, который в точности повторяет фильтр [настройки шлюзов](#) .

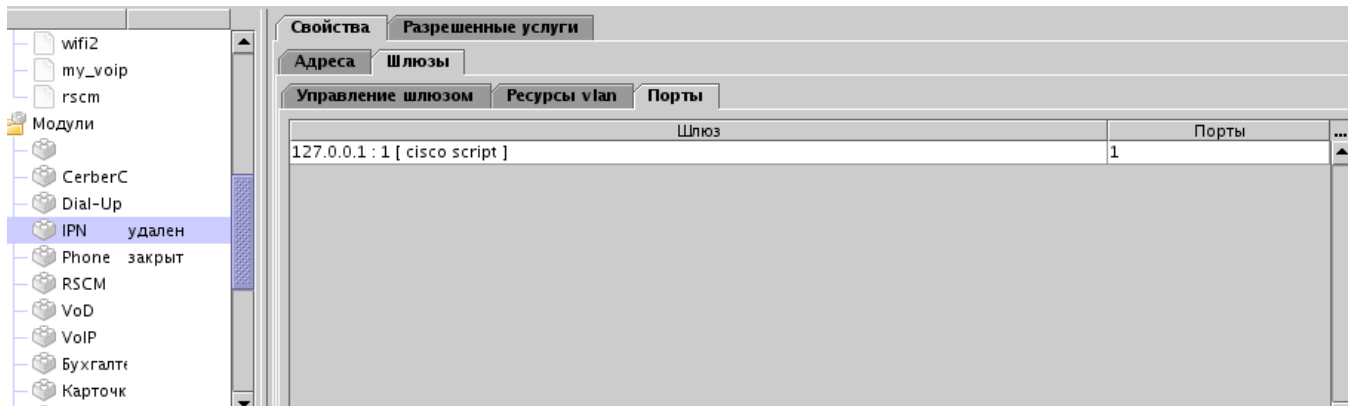
Можно задать VLAN вручную, а можно заполнить его автоматически первым свободным VLAN на данном шлюзе (галочка авто). Свободный VLAN выделяется по умолчанию из диапазона : 2-4096 . Для того, чтобы это изменить нужно в конфигурации шлюза поставить :

```
vlan.range=2-1000;1010;2000-4096
```

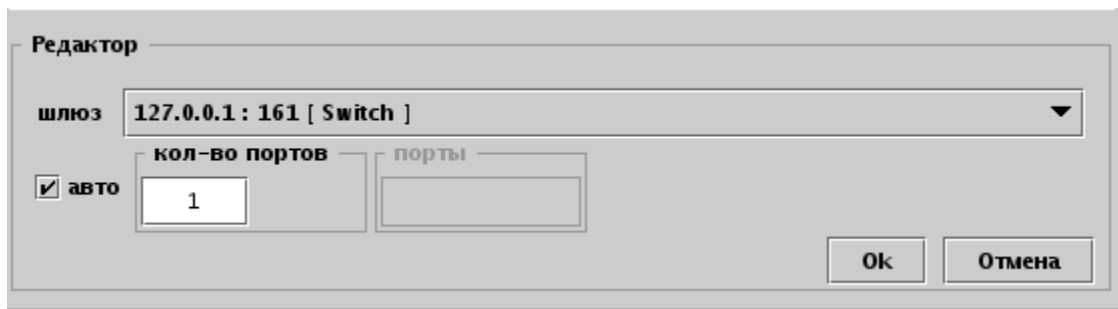
В этом примере выделяются диапазоны: со 2-го по 1000 и с 2000-го по 4096-ой и отдельно 1010.

Настройка портов шлюза.

После добавления в договор шлюза, можно указать на отдельной вкладке его порты.



При нажатии на кнопку добавить (или при редактировании уже добавленных позиций) появляется редактор:



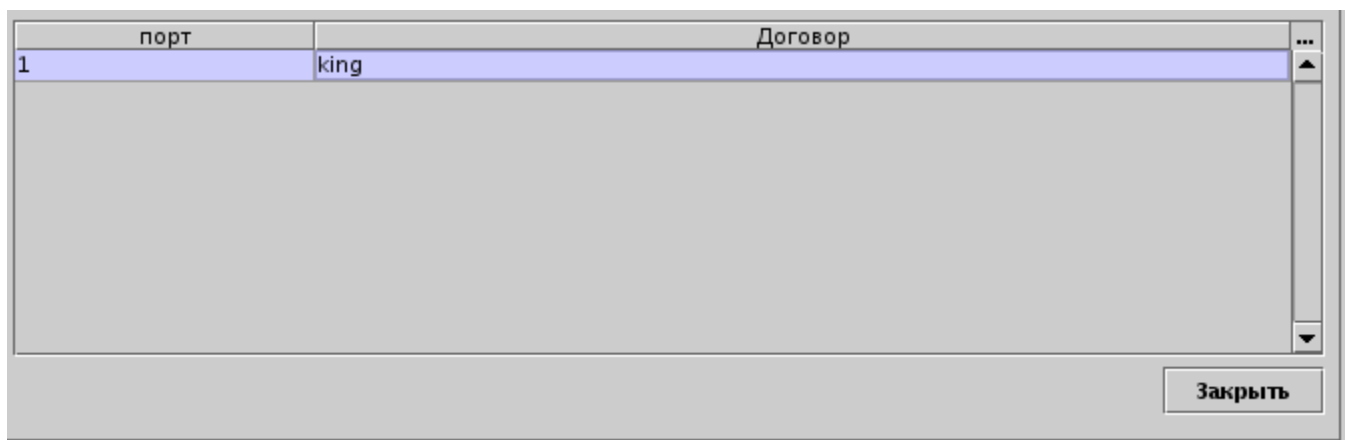
Тут есть 2 варианта: либо выделить порты автоматически (галочка авто), либо вручную перечислить через запятую в поле "порты". В случае автоматического выделения по умолчанию порты выделяются из диапазона 1-24. Этот диапазон можно поменять, указав в конфигурации шлюза опцию :

```
port.range=1-10;12;16-24
```

В данном примере порты могут быть с 1-го 10-ый , 12-ый, и с 16-го по 24-ый.

Данные порты не используются в стандартных типах шлюзов, и, если необходимы для управления, то их можно использовать только в шлюзах, реализованных на [BeanShell](#).

Также можно вести учет занятых портов на шлюзе. Для этого в редакторе шлюзов(Модули->IPN>Шлюзы) , нужно вызвать правой кнопкой контекстное меню на конкретном шлюзе и в нем выбрать "Мониторинг портов".



Настройка шлюзов типа Manad

Данный тип шлюза позволяет управлять файрволом на PC-роутерах. Мы предлагаем 2 варианта: для роутеров FreeBSD (файрвол ipfw) и Linux (файрвол iptables +iproute2).

На вкладке модуля IPN **Типы шлюзов** добавляется тип шлюза Manad со следующей конфигурацией:

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.ManadContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.ManadGateWorker
```

На роутере должен быть запущен manad - менеджер файрвола. Это программа написана на Perl. Её задача висеть на порту и слушать команды биллинга по закрытию или открытию договора.

Спецификация Manad

Если вы не хотите реализовывать Manad собственными силами, либо изменять один из предлагаемых вариантов, то вы можете пропустить этот раздел и перейти сразу к описанию того варианта (FreeBSD или Linux), который вас устраивает.

Manad принимает от BGBilling-сервера следующие строки:

1. add num rules;
2. remove num rules;
3. test.

В качестве разделителя используется символ табуляции. num - это id договора клиента, rules - это строка с набором команд, внутри которой команды разделяются символом "|". Подразумевается, что команда add добавляет правила, команда remove - удаляет. В ответ на команду test Manad шлёт список id договоров (разделённых символом пробела), открытых на данном шлюзе.

Обработка команд Manad.

На стороне сервера типы правил проходят следующую обработку : для всех тегов <LOOP> </LOOP> все их содержимое дублируется для каждого ip-адреса. При этом:

1. Макрос вида {A} заменяются на ip-адреса клиента из выбранного диапазона;
2. Подстановки вида {XN}, где X - латинская буква, а N - цифра также обрабатываются для каждого ip-адреса. Начинать нумеровать эти последовательности нужно обязательно с нуля. Для каждого ip-адреса по каждой букве увеличивается индекс. Т.е., например, подстановки {A0}, {A1}, {B0} для 1-го ip-адреса преобразуется в {A0}, {A1}, {B0} ; для второго ip-адреса в {A1}, {A2}, {B1} и т.п.

Т.е. например строки :

```
<LOOP>
pipe {P0} config bw 128000
pipe {P1} config bw 128000
add {N0} pipe {P0} ip from any to {A} out
add {N0} pipe {P1} ip from {A} to any in
</LOOP>
```

для ip-адресов 192.168.184.10, 192.168.184.11 и 192.168.184.12 преобразуются в :

```
pipe {P0} config bw 128000
pipe {P1} config bw 128000
add {N0} pipe {P0} ip from any to 192.168.184.10 out
add {N0} pipe {P1} ip from 192.168.184.10 to any in
pipe {P2} config bw 128000
pipe {P3} config bw 128000
add {N1} pipe {P2} ip from any to 192.168.184.11 out
add {N1} pipe {P3} ip from 192.168.184.11 to any in
pipe {P4} config bw 128000
pipe {P5} config bw 128000
add {N2} pipe {P4} ip from any to 192.168.184.12 out
add {N2} pipe {P5} ip from 192.168.184.12 to any in
```

3. Аналогичным образом происходит обработка тега <LOOP_NET > для сетей. Строки дублируются для каждой сети. Тут доступны следующие макросы: {IP} - первый адрес сети, {MASK} - маска сети вида xx.xx.xx.xx, {MASK_WILD} - инвертированная маска сети, {MASK_BIT} - битовая маска сети вида xx.

Например строки вида :

```
<LOOP_NET>
permit {IP}/{MASK_BIT}
</LOOP_NET>
```

Для сетей 192.169.185.0/24, 192.169.186.0/24 преобразуется в:

```
permit 192.169.185.0/24
permit 192.169.186.0/24
```

Настройка шлюзов типа Manad под FreeBSD

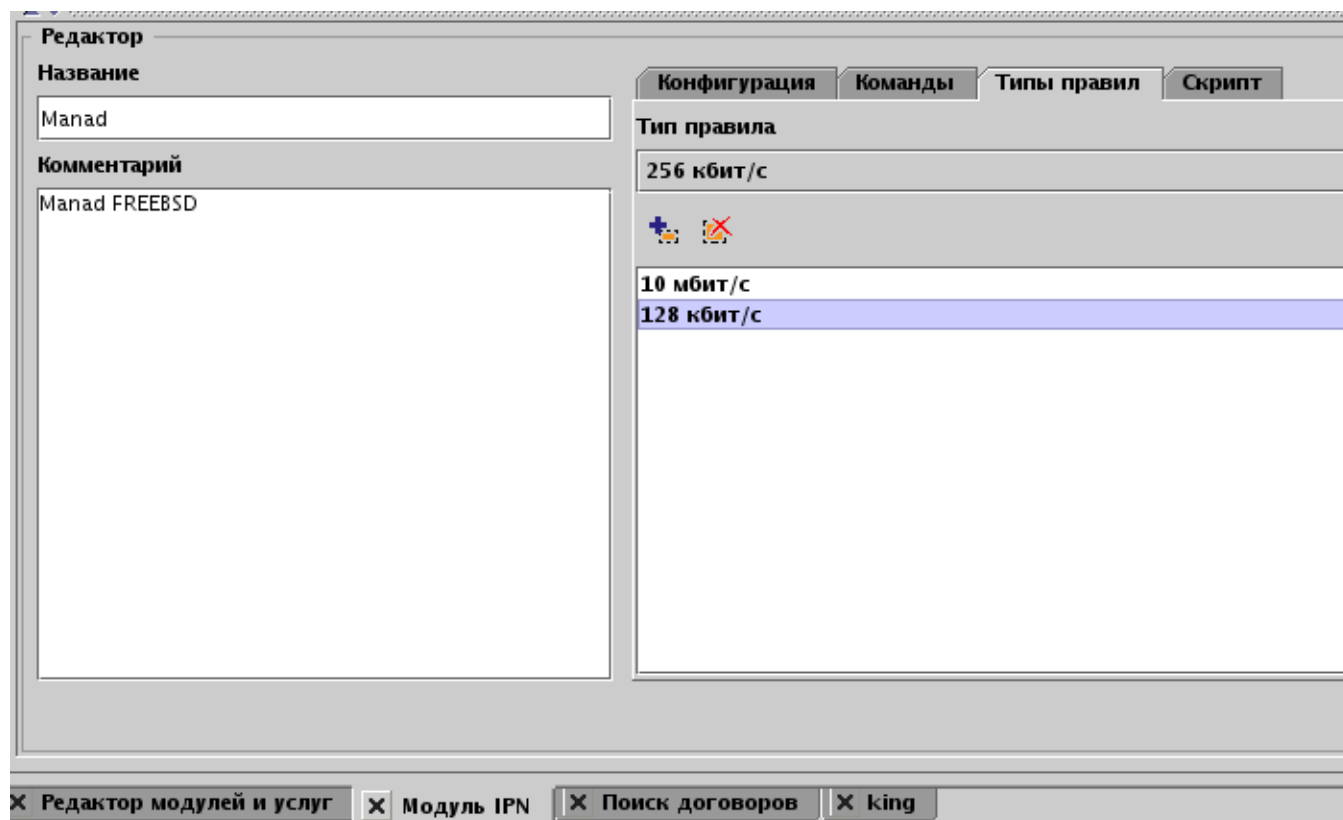
В настоящее время реализован manad для шлюзов под управлением FreeBSD. Скачать его можно [здесь](#). Необходимо установить его на вашем шлюзе (предварительно потребуется установить Perl) и добавить в автозапуск.

На вкладке **Типы правил** добавим два типа правила для данного типа шлюза. Допустим, мы предоставляем доступ двух типов: 10-мегабитный и 128-килобитный. Добавляем в типах правил две записи.

```
speed=10000000
```

и

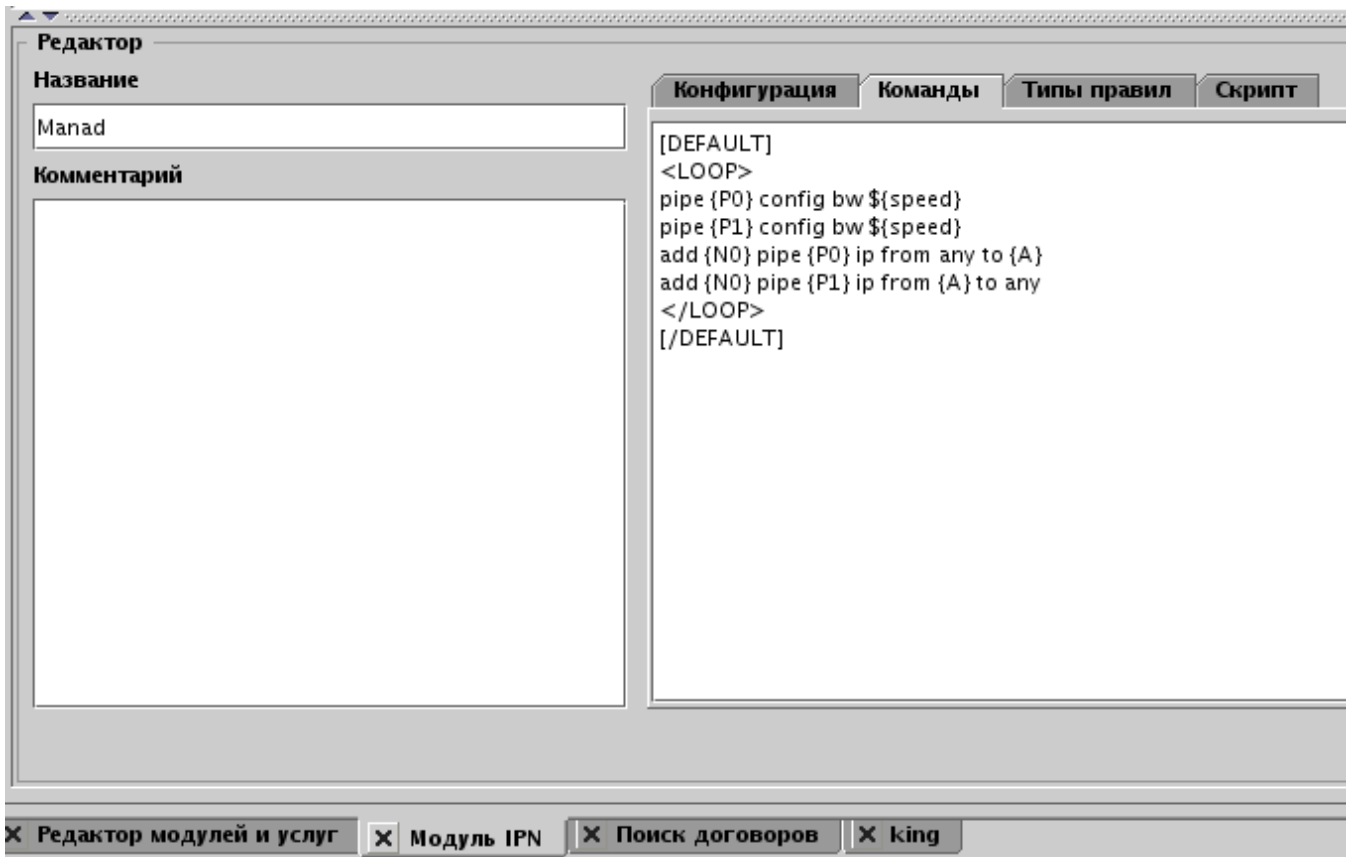
```
speed=128000
```



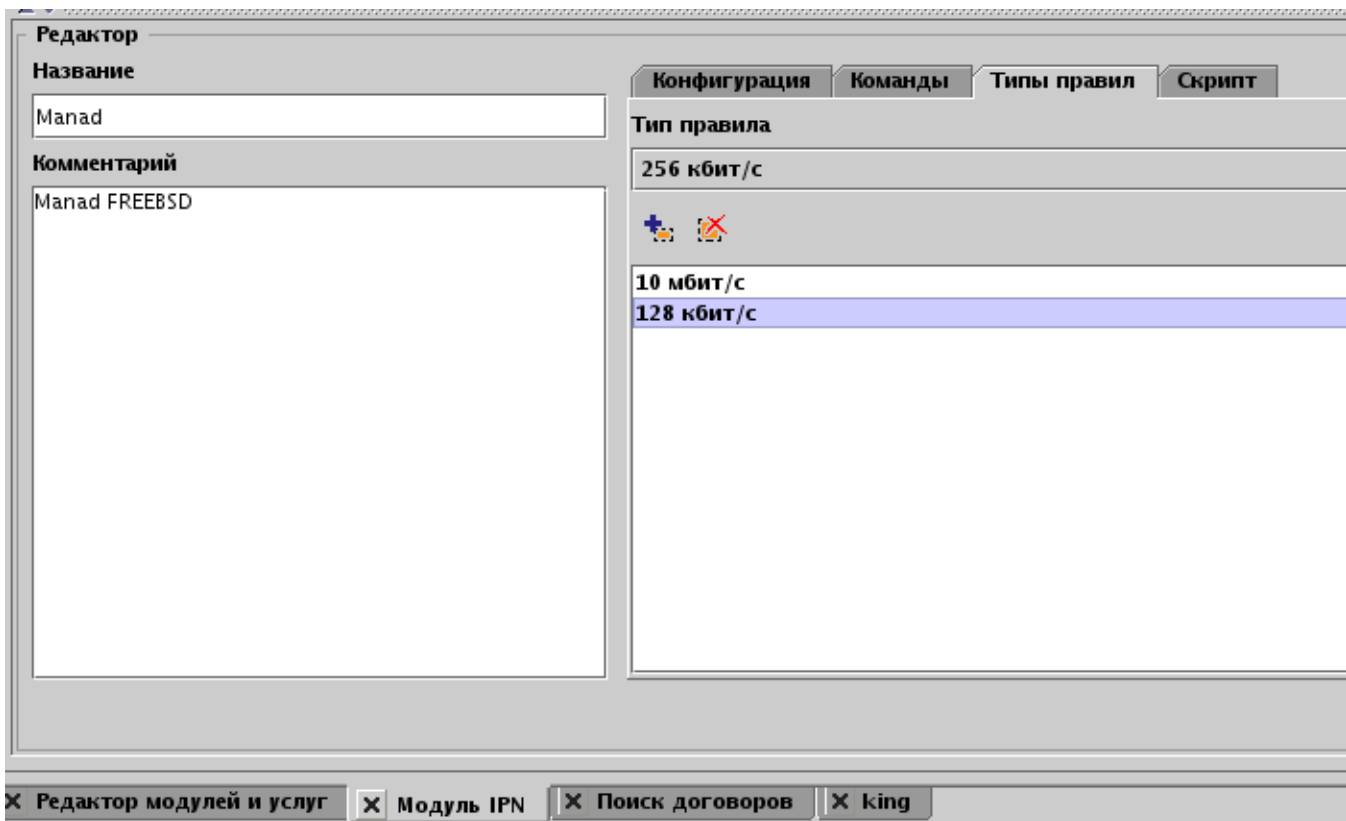
The screenshot shows a web-based configuration interface for Manad. The main window is titled "Редактор" (Editor) and contains a form for editing a rule named "Manad". The "Комментарий" (Comment) field contains "Manad FREEBSD". On the right side, there are four tabs: "Конфигурация" (Configuration), "Команды" (Commands), "Типы правил" (Rule Types), and "Скрипт" (Script). The "Типы правил" tab is active, showing a list of rule types with their respective speeds. The list includes "256 кбит/с", "10 мбит/с", and "128 кбит/с". The "128 кбит/с" entry is highlighted in blue. At the bottom of the window, there is a taskbar with several open windows: "Редактор модулей и услуг", "Модуль IPN", "Поиск договоров", and "king".

На вкладке **Команды** в типе шлюза пишем :

```
[DEFAULT]
<LOOP>
pipe {P0} config bw ${speed}
pipe {P1} config bw ${speed}
add {N0} pipe {P0} ip from any to {A} out
add {N0} pipe {P1} ip from {A} to any in
</LOOP>
[/DEFAULT]
```



На вкладке **Правила** в типе шлюза добавляем оба новых правила:



Как видно из примера команды Manad включает в себя подстановки адреса вида {A}, а также другие виды подстановок вида {XN}, где X - латинская буква, а N - цифра. Подстановка адреса осуществляется биллингом перед отсылкой команд, либо при сохранении правила для договора, в зависимости от используемого вида правила (пользовательское/типизированное). Подстановки вида {XN} отправляются на manad вместе с командами для определения номеров пайпов и правил.

Подстановки вида {NET} и {NET_MASK} заменяются на выбранные сети для тегов <LOOP_NET> и <LOOP_NET_MASK>. О том, как это происходит читайте [обработку правил Manad](#).

После создания типа шлюза Manad, типов правил и команд для него необходимо определить конкретные адреса шлюзов на вкладке **Шлюзы**.

127.0.0.1:1855	DHCP	DHCP
127.0.0.1:1868	DHCP	DHCP_TEST
192.168.184.39:4567	Manad	Manad Common

На вкладке **Конфигурация** шлюзов типа Manad ничего указывать не нужно.

Для добавления пользователя на шлюз откройте договор на вкладке модуля IPN - **Шлюзы**.

Адреса
Шлюзы

Состояние шлюза

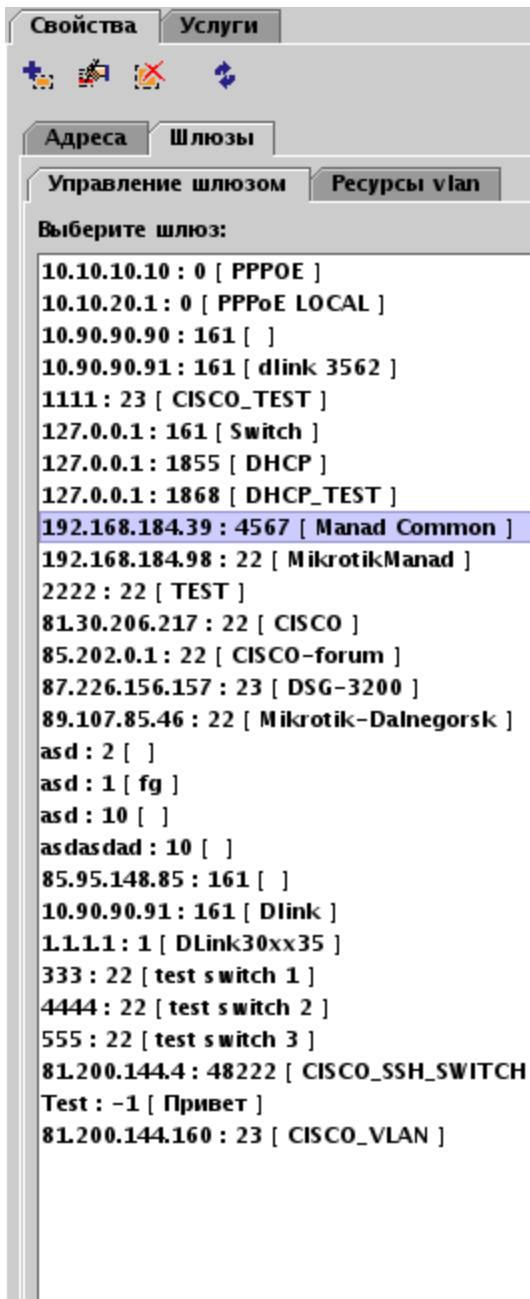
Открыт ▼

Шлюзы

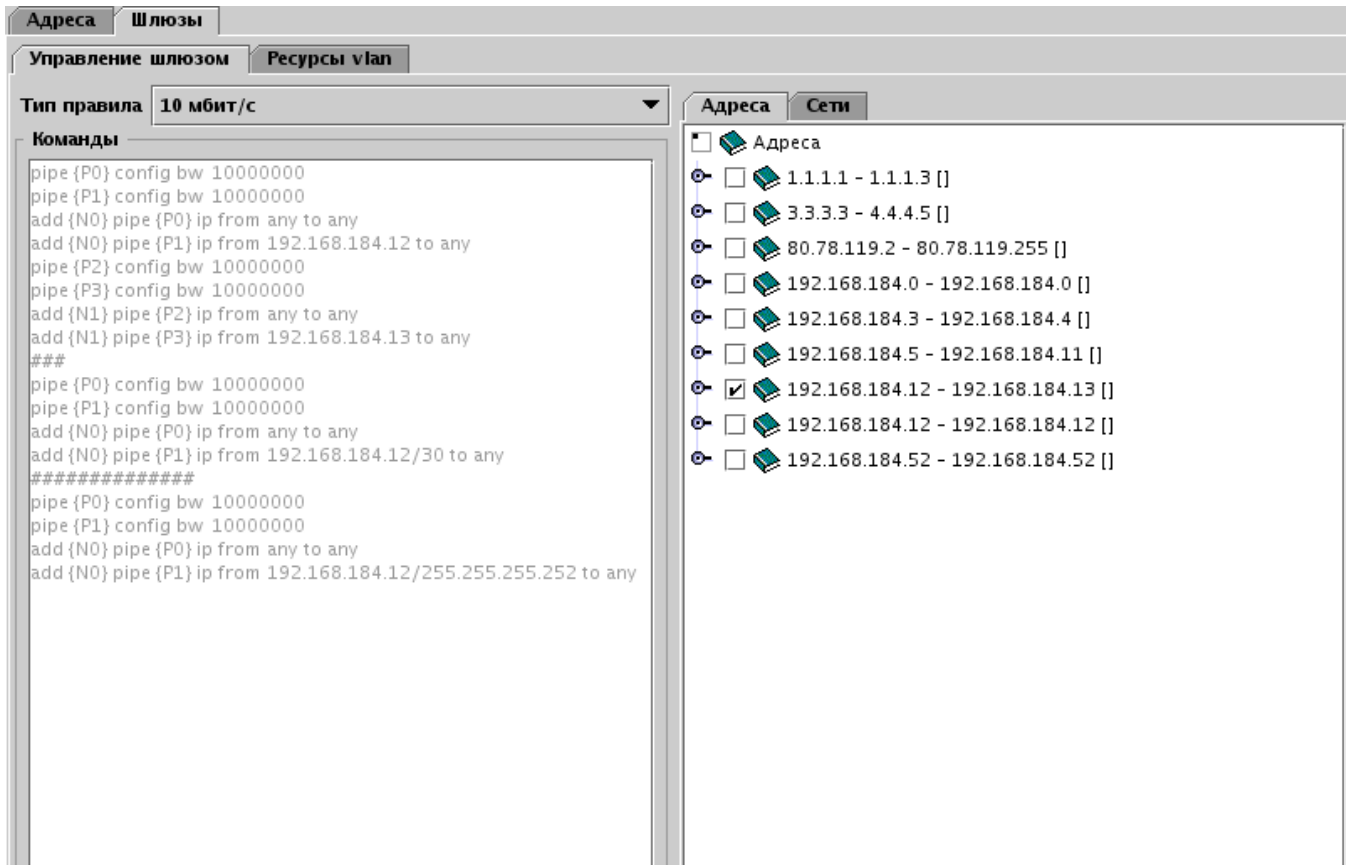
Статистика состояния шлюза

Дата и время	Действие	Исполнитель	
21.04.2006 21:44:16	открыт	Шамиль	Шлюз ▲
21.04.2006 21:44:15	открыт	Шамиль	Шлюз
21.04.2006 21:43:46	закрыт	Шамиль	Шлюз
21.04.2006 21:43:45	закрыт	Шамиль	Шлюз
21.04.2006 21:43:42	открыт	Шамиль	Шлюз
21.04.2006 21:43:41	открыт	Шамиль	Шлюз
21.04.2006 21:42:59	закрыт	Шамиль	Ошиб
21.04.2006 21:42:58	закрыт	Шамиль	Шлюз
21.04.2006 21:42:45	открыт	Шамиль	Шлюз
21.04.2006 21:42:44	открыт	Шамиль	Шлюз
21.04.2006 21:41:21	закрыт	Шамиль	Ошиб
21.04.2006 21:41:06	закрыт	Шамиль	Шлюз
21.04.2006 21:40:39	открыт	Шамиль	Шлюз
21.04.2006 21:40:38	открыт	Шамиль	Шлюз
21.04.2006 21:40:32	закрыт	Шамиль	Шлюз
21.04.2006 21:40:31	закрыт	Шамиль	Шлюз
21.04.2006 21:39:48	открыт	Шамиль	Шлюз
21.04.2006 21:39:45	открыт	Шамиль	Шлюз
21.04.2006 21:34:13	закрыт	Шамиль	Шлюз
21.04.2006 21:34:12	закрыт	Шамиль	Шлюз ▼

Затем нажмите кнопку **Добавить** и произведите выбор шлюза. При этом отобразятся только те шлюзы, на которые клиент ещё не был заведён.



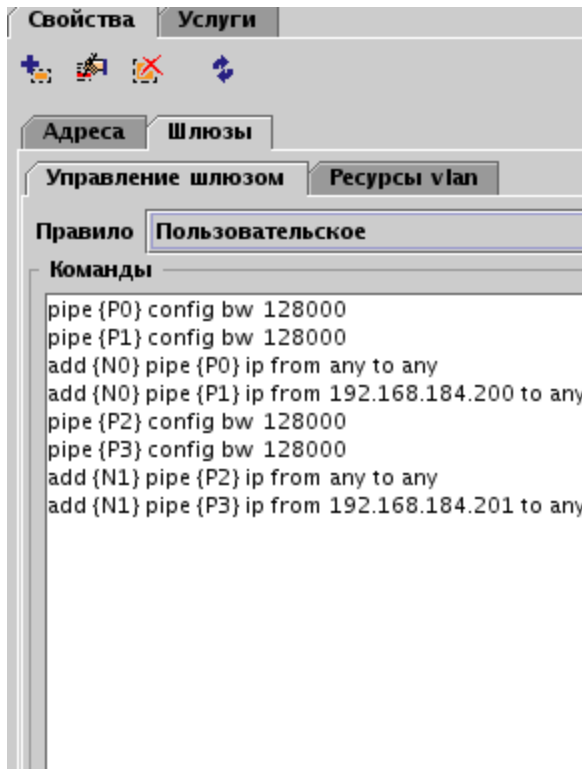
Если выбранный шлюз был типа Manad, то откроется редактор правил договора Manad.



Слева в выпадающем списке можно выбрать тип правила, в дереве справа - адреса и сети, на которые оно применяется. В области **Команды** отображается текст команд, которые будут отправлены на шлюз. При этом, если выбран тип "Пользовательский", то область **Команды** редактируемая и правило сохраняется в точно таком же виде в БД. Т.е. в этом случае вы можете установить клиенту любые команды, не завязываясь на типе.

Если же выбран тип правила и список адресов, то в базе сохраняется только это сочетание, а текст команд вычисляется динамически в момент отправки на шлюз. При этом выглядит оно так, как отображено в области **Команды**. Использование типов позволяет более просто менять команды для многих пользователей, просто поправив шаблон в типе.

Для создания пользовательского правила на основании шаблонного выбирается тип правила, адреса, после чего тип правила переключается в "Пользовательский". После чего текст правила может быть изменён.



После настройки биллинга необходимо произвести настройку самого ipfw и установку manad на шлюз.

Доступ клиентам разрешается путём добавления динамических правил файрвола с номером менее 20000.

Предварительно в файле **/etc/rc.firewall** добавьте правило, запрещающее всем пакетам проход через шлюз. Кроме того, в нашем случае нужно добавить адрес NAT-сервера для того, чтобы мы сами могли выходить в сеть.

Другим решением является изменение структуры сети с установкой шлюза между клиентами и магистралью провайдера.

Ниже приведена конфигурация, подходящая для нашей сети:

Код

```
fw="/sbin/ipfw -q"

# flush out the list before we begin.
${fw} -f flush

# localhost spoofprotect
${fw} add 10 pass all from any to any via lo0
${fw} add 10 deny all from any to 127.0.0.0/8
${fw} add 10 deny all from 127.0.0.0/8 to any

# deny netbios stuff
${fw} add 20 deny tcp from any to any 135-139,445,593
${fw} add 20 deny udp from any to any 135-139,445

#ssh,telnet, ftp NAT
${fw} add 30 pass tcp from 14.3.3.8 to me 20-23
${fw} add 30 pass tcp from me 20-23 to 14.3.3.8

# manad -
${fw} add 40 pass tcp from 14.3.3.8 to me 4567
${fw} add 40 pass tcp from me 4567 to 14.3.3.8

#####
# 14.3.3.0
#####
${fw} add 20000 deny all from any to 14.3.3.8/24
${fw} add 20000 deny all from 14.3.3.8/24 to any
```

```
## manad ,
```

```
#{fw} add 30000 pass all from any to any
```

Настройка шлюза типа Manad под Linux

Здесь предлагается пример manad'a для шлюзов под управлением Linux. Скачать его можно [здесь](#). Необходимо установить его на вашем шлюзе (предварительно потребуется установить Perl) и добавить в автозапуск. В данном примере происходит управление службой iptables и ограничение трафика с помощью iproute2. Архив состоит из 3 частей:

1. manad_linux.pl - сам manad (написан на perl);
2. init_shaping.sh - инициализация правил ограничения трафика в системе (описано ниже);
3. start_manad.sh - скрипт запуска (вызывает вначале init_shaping.sh, а потом manad_linux.pl);
4. stop_manad.sh - скрипт остановки.

На вкладке **Команды** в типе шлюза добавим команды для данного типа шлюза. Пусть мы хотим для всех ip-адресов клиента (т.е. все они поделят между собой канал) ограничивать входящий трафик значением в 256 кбит/с, а исходящий трафик значением 128 кбит/с. Для этого создадим 2 класса, отдельно для входящего и исходящего трафика .

```
Код

[DEFAULT]

[OPEN]

<LOOP>
iptables -A FORWARD -t filter -s {A} -j ACCEPT
</LOOP>

/sbin/tc class add dev eth0 parent 1:0 classid 1:[N1] htb rate 256kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:[N1] handle [N1]: sfq perturb 10 quantum 1500

/sbin/tc class add dev eth0 parent 1:0 classid 1:[N2] htb rate 128kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:[N2] handle [N2]: sfq perturb 10 quantum 1500

<LOOP>
/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio [N1] u32 match ip dst {A} flowid 1:[N1]
</LOOP>

<LOOP>
/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio [N2] u32 match ip src {A} flowid 1:[N2]
</LOOP>

[/OPEN]
[CLOSE]
<LOOP>
/sbin/iptables -D FORWARD -t filter -s {A} -j ACCEPT
</LOOP>
/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio [N1]
/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio [N2]

/sbin/tc class del dev eth0 parent 1:0 classid 1:[N1] htb rate 256kbit burst 4k prio 1
/sbin/tc class del dev eth0 parent 1:0 classid 1:[N2] htb rate 128kbit burst 4k prio 1

[/CLOSE]

[/DEFAULT]
```

Здесь теги [OPEN]/[/OPEN] и [CLOSE] [/CLOSE] - это теги, которые обрабатывает данный конкретный тип manad'a и означают они, соответственно, набор открывающих правил и набор закрывающих правил . Аналогично тег [NX] - тоже обрабатывается самим manad'ом , туда просто вставляется свободный индекс, который увеличивается с каждым новой командой на открытие manad'a. В данном случае отличие от [NX] в том, что это комбинации не обрабатывается га стороне сервера(сервер увеличивает N для каждого ip), а обрабатывается самим manad'ом Т.е. в данном случае на стороне сервера данный набор правил преобразится в:

```
Код

[OPEN]

iptables -A FORWARD -t filter -s 192.168.184.10 -j ACCEPT
```

```

iptables -A FORWARD -t filter -s 192.168.184.11 -j ACCEPT

/sbin/tc class add dev eth0 parent 1:0 classid 1:[N1] htb rate 256kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:[N1] handle [N1]: sfq perturb 10 quantum 1500

/sbin/tc class add dev eth0 parent 1:0 classid 1:[N2] htb rate 128kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:[N2] handle [N2]: sfq perturb 10 quantum 1500

/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio [N1] u32 match ip dst 192.168.184.10 flowid 1:[N1]
/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio [N1] u32 match ip dst 192.168.184.11 flowid 1:[N1]

/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio [N2] u32 match ip src 192.168.184.10 flowid 1:[N2]
/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio [N2] u32 match ip src 192.168.184.11 flowid 1:[N2]

[/OPEN]

[ CLOSE ]

/sbin/iptables -D FORWARD -t filter -s 192.168.184.10 -j ACCEPT
/sbin/iptables -D FORWARD -t filter -s 192.168.184.11 -j ACCEPT

/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio [N1]
/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio [N2]

/sbin/tc class del dev eth0 parent 1:0 classid 1:[N1] htb rate 256kbit burst 4k prio 1
/sbin/tc class del dev eth0 parent 1:0 classid 1:[N2] htb rate 128kbit burst 4k prio 1

[/CLOSE]

```

Это то, что получит сам manad. Как видим, сервер вырезал теги <LOOP> и {A} Аналогично, подстановки вида {NET} и {NET_MASK} заменяются на выбранные сети для тегов <LOOP_NET> и <LOOP_NET_MASK>. О том, как это происходит, читайте [обработку правил Manad](#).

Manad работает следующим образом. Он получает правила, заменяет в них [NX] на число (в данном случае свободный номер класса по ограничению трафика), далее делит правила на 2 части: открывающие и закрывающие. Открывающие применяются, а закрывающие запоминаются и применяются при получении команды на закрытие для данного договора.

Для инициализации правил iproute2 в системе используется скрипт init_shaping.sh :

```

INDEV="eth0"

/sbin/tc qdisc del dev $INDEV root 2> /dev/null

##### speed server->client(downstream)

/sbin/tc qdisc add dev $INDEV root handle 1: htb default ffff r2q 1

#default
/sbin/tc class add dev $INDEV parent 1:0 classid 1:ffff htb rate 100mbit burst 4k prio 3
/sbin/tc qdisc add dev $INDEV parent 1:ffff handle ffff: sfq perturb 10 quantum 1500

```

Инициализацию правил iptables вы можете проводить в зависимости от вашей настройки сети. В данном примере надо, как минимум, установить запрет на цепочку FORWARD таблицы filter, т.к. туда добавляются разрешающие правила. Это можно установкой политики по умолчанию:

```

/sbin/iptables -P FORWARD REJECT

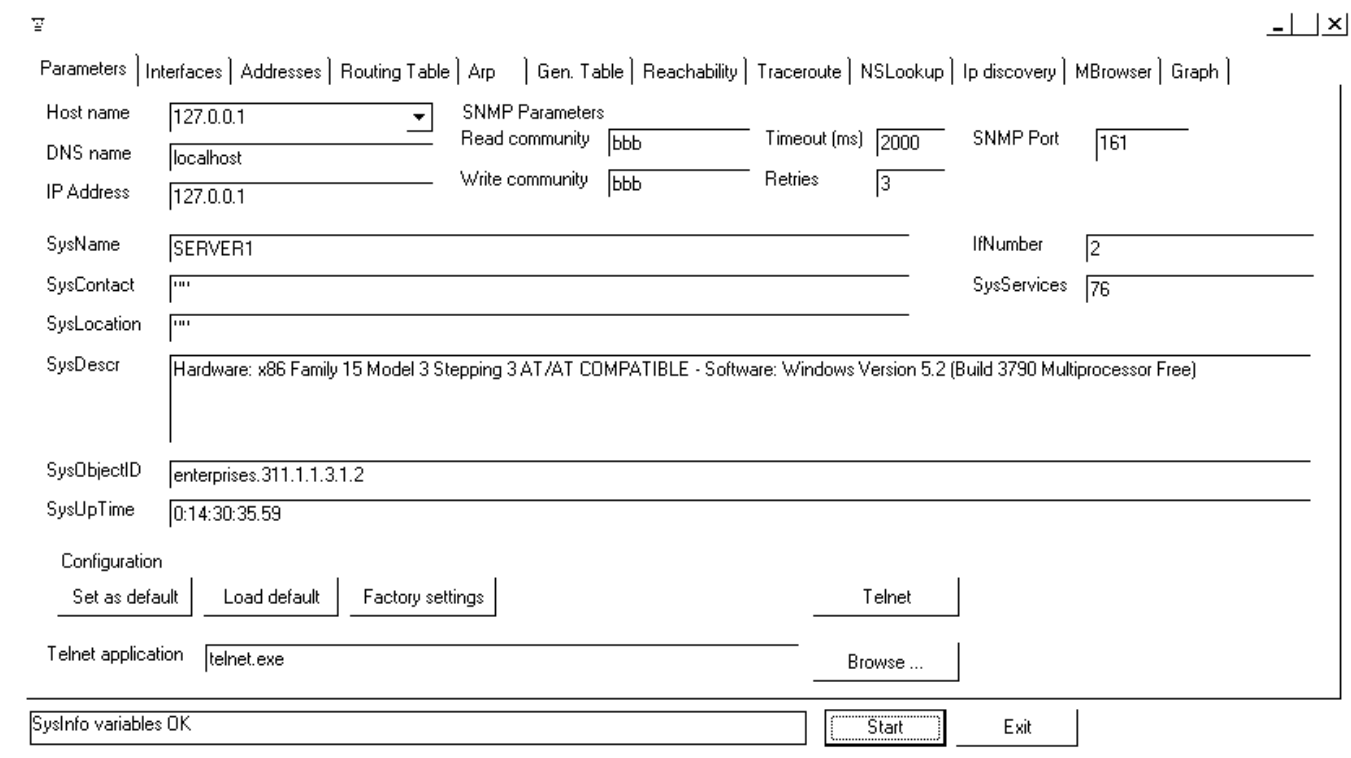
```

Настройка шлюзов типа Switch

Шлюзом типа Switch может выступать любой управляемый коммутатор с поддержкой SNMP и MIB-IFACES. Предполагается, что SNMP на коммутаторе уже настроен и заведено community с r-w правами, например billing.

Для начала необходимо установить номера интерфейсов, которые есть на вашем коммутаторе. Очень удобно для этого под Windows использовать утилиту GetIf, которую вы можете скачать по адресу: <http://www.wtcs.org/snmp4tpc/FILES/Tools/SNMP/getif/getif-2.3.1.zip>

Определения существующих интерфейсов установите в первой вкладке имя community и адрес хоста, соединитесь.



The screenshot shows the GetIf utility window with the following configuration:

- Host name: 127.0.0.1
- DNS name: localhost
- IP Address: 127.0.0.1
- SNMP Parameters:
 - Read community: bbb
 - Write community: bbb
 - Timeout (ms): 2000
 - Retries: 3
 - SNMP Port: 161
- SysName: SERVER1
- SysContact: ""
- SysLocation: ""
- SysDescr: Hardware: x86 Family 15 Model 3 Stepping 3 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Multiprocessor Free)
- SysObjectID: enterprises.311.1.1.3.1.2
- SysUpTime: 0:14:30:35.59

Configuration options:

- Set as default
- Load default
- Factory settings
- Telnet
- Telnet application: telnet.exe
- Browse ...

Buttons: Start, Exit

В данном случае соединение шло с SNMP-агентом MS WIN машины. Далее на вкладке MBrowser выберите ветку .1.3.6.1.2.1.2.2.1.2 и нажмите Start.

Parameters | Interfaces | Addresses | Routing Table | Arp | Gen. Table | Reachability | Traceroute | NSLookup | Ip discovery | MBrowser | Graph

.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifType

.1.3.6.1.2.1.2.2.1.3

+ rmon
+ system
- interfaces
 ifNumber
 - ifTable
 ifEntry
 ifIndex
 ifDescr
 ifMtu

Type: integer Enums: other (1)
Access: readonly Status: mandatory

The type of interface, distinguished according to the physical/link protocol(s) immediately 'below' the network layer in the protocol stack.

```

interfaces.ifNumber.0 : 2
interfaces.ifTable.ifEntry.ifIndex.1 : 1
interfaces.ifTable.ifEntry.ifIndex.65539 : 65539
interfaces.ifTable.ifEntry.ifDescr.1 : 405320544350204C6F6F706261636B20696E74655726661636500
interfaces.ifTable.ifEntry.ifDescr.65539 : 33436F6D2047696761626974204C4F4D202833433934302900
interfaces.ifTable.ifEntry.ifType.1 : softwareLoopback

interfaces.ifTable.ifEntry.ifMtu.1 : 1520
interfaces.ifTable.ifEntry.ifMtu.65539 : 1500
interfaces.ifTable.ifEntry.ifSpeed.1 : 10000000
interfaces.ifTable.ifEntry.ifSpeed.65539 : 100000000

```

.1.3.6.1.2.1.2.2.1.3.65539 | i (integer) | ethernet-csmacd 45 entry(s) Set Add to graph Add to Gen

Ready Start Exit

Далее необходимо составить карту-код интерфейса - его понятное администратору обозначение.

Например в нашем случае интерфейс 1 - это LO, 65539 - ETH1. К сожалению getif некорректно отображает значение поля ifDescr, поэтому приходится ориентироваться по другим полям, например ifType.

В случае, если вы предпочитаете работать с UNIX-системами, воспользуйтесь утилитой snmpwalk из пакета (bbb - это имя community):

```
snmpwalk -Os -c bbb -v 1 192.168.184.2 .1.3.6.1.2.1.2.2.1.2
```

Вывод будет примерно следующий:

```
[root@gate ~]# snmpwalk -Os -c bbb -v 1 192.168.184.2 .1.3.6.1.2.1.2.2.1.2
ifDescr.1 = STRING: MS TCP Loopback interface
ifDescr.65539 = STRING: 3Com Gigabit LOM (3C940)
```

Таким образом мы получаем карту интерфейсов.

Следующий шаг - создание типа шлюза. Допустим, мы хотим создать тип шлюза CISCO XXX с определённым набором фиксированных интерфейсов (портов), причём на каждом из шлюзов этого типа могут быть дополнительные интерфейсы типа VLAN.

Источники		Менеджер источников	Шлюзы	Привязка услуг	Поиск адресов	Начисление	Типы шлюзов	Типы правил
CISCO XXX		Manad					sdfsd	Комментз
<p>Редактор</p> <p>Название</p> <input type="text" value="CISCO XXX"/>		<p>Конфигурация</p> <pre>=bitel.billing.module.services.ipn.editor.SwitchRuleTypeEditor =bitel.billing.module.services.ipn.editor.SwitchContractRuleEditor =bitel.billing.server.ipn.SwitchGateWorker =ETH1</pre>						
<p>Комментарий</p> <div style="border: 1px solid black; height: 100px;"></div>								
							Ok	Отмена

В конфигурации должно быть указано следующее:

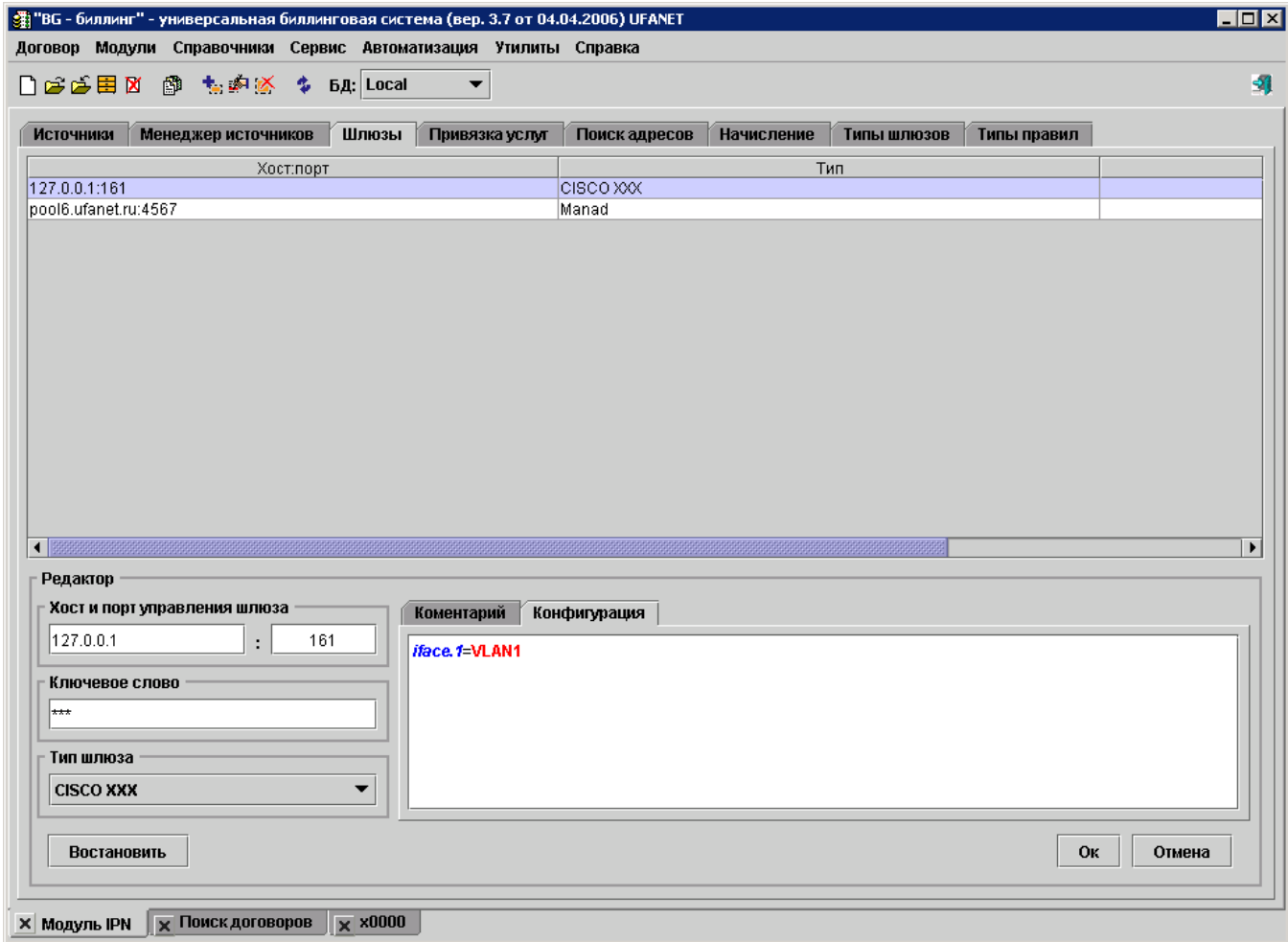
```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.SwitchContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.SwitchGateWorker
# snmp 1 2c
snmp.version=1
```

При указании **snmp.version=0** биллинг не будет выполнять действий по управлению устройством при смене статуса шлюза, в этом случае шлюз может быть использован только для учёта.

Далее указываются номера интерфейсов

```
iface.65539=ETH1
```

Типы правил для Switch-шлюзов создавать не нужно, можно сразу создавать шлюз. В шлюзе можно добавить специфичные для него интерфейсы.



Теперь при добавлении шлюза в договор после выбора шлюза 127.0.0.1 отобразится следующий редактор.

Программа отобразит только свободные на данном шлюзе порты. Выбрав порт вы укажете программе, что его нужно блокировать в случае перевода статуса договора в IPN модуле в закрыт или заблокировано.

Также в конфигурации шлюза можно поставить опцию

```
# snmp. 500,1000,2000,5000,5000 - .  
#retry.intervals=500,1000,2000,5000,5000
```

Настройка шлюза BGRadiusIPN

Данный тип шлюза позволяет предоставлять доступ к виртуальным частным сетям на оборудовании CISCO с авторизацией по логину/паролю. В свойствах шлюза могут быть указаны логин и пароль авторизации, выдаваемый фиктивному интерфейсу адрес и маршрутизируемые через него сети.

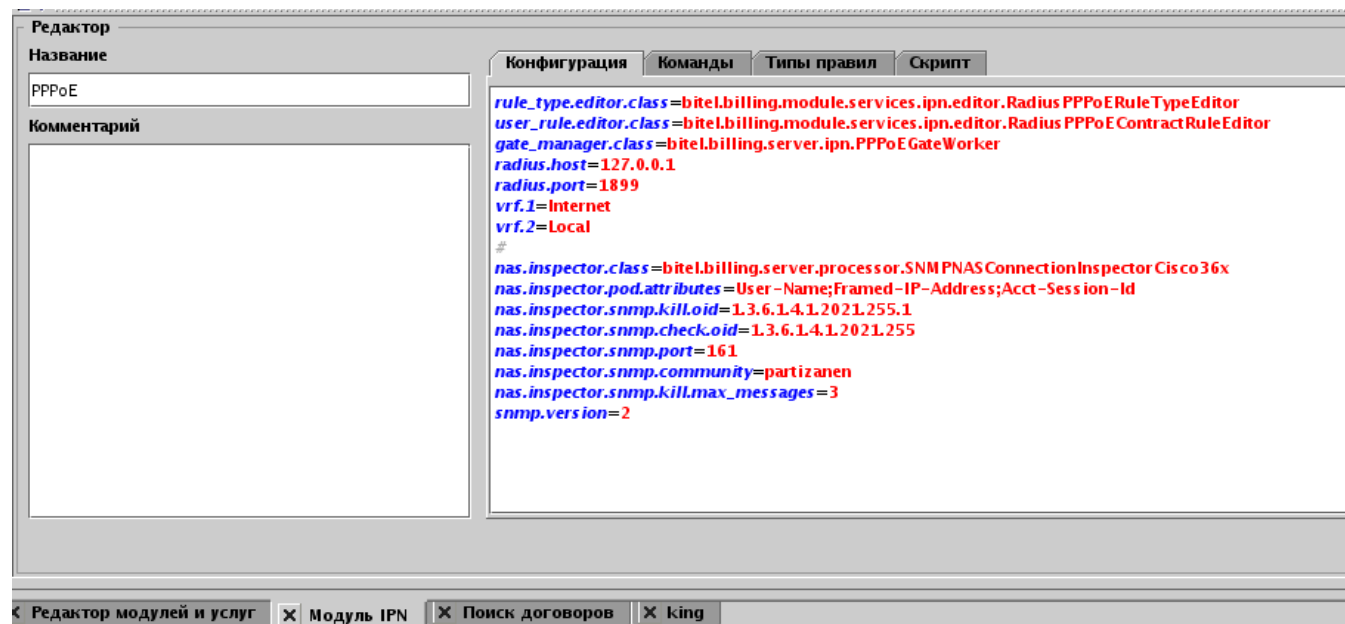
Установите и настройте BGRadiusIPN

```
processor.class=bitel.billing.server.processor.ipn.IPNProcessor
processor.mid=0
#
auth.port=
# ( )
admin.port=

auth.thread.count=10
netflow.thread.count=10
```

Создайте тип шлюза RadiusPPPoE с параметрами

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.RadiusPPPoEContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.PPPoEGateWorker
# / BGRadiusIPN
radius.host=
radius.port=
# VRF,
# ( cisco-avpair=lcp:interface-config=ip vrf forwarding ${vrf} )
vrf.1=Internet
```



Редактор

Название: PPPoE

Комментарий:

Конфигурация

```
rule_type.editor.class=bitel.billing.module.services.ipn.editor.RadiusPPPoERuleTypeEditor
user_rule.editor.class=bitel.billing.module.services.ipn.editor.RadiusPPPoEContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.PPPoEGateWorker
radius.host=127.0.0.1
radius.port=1899
vrf.1=Internet
vrf.2=Local
#
nas.inspector.class=bitel.billing.server.processor.SNMPNASConnectionInspectorCisco36x
nas.inspector.pod.attributes=User-Name;Framed-IP-Address;Acct-Session-Id
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.2021.255.1
nas.inspector.snmp.check.oid=1.3.6.1.4.1.2021.255
nas.inspector.snmp.port=161
nas.inspector.snmp.community=partizanen
nas.inspector.snmp.kill.max.messages=3
snmp.version=2
```

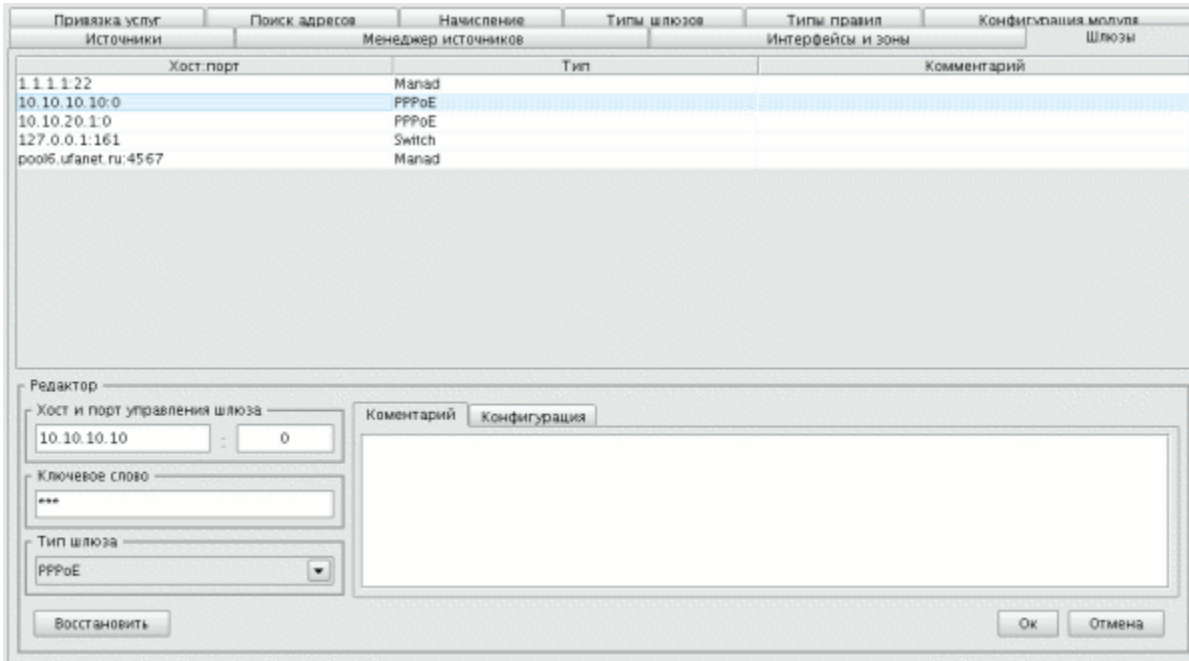
← Редактор модулей и услуг | X Модуль IPN | X Поиск договоров | X king

На вкладке **Типы правил** создайте типы правил. Тип правила для данного типа шлюза понимается как реалм. Вот пример текста типа правила:

```
realm=default
attributes=Service-Type=2;Framed-Protocol=1;Framed-MTU=1492;Port-Limit=1;Auth-Type=accept;Acct-Interim-Interval=60;Cisco-AVPair=multilink:max-links=6;Cisco-AVPair=multilink:min-links=1;Cisco-AVPair=lcp:interface-config=ip unnumbered Loopback135
```

Необходимо указать реалм (realm) и список атрибутов по умолчанию, которые будут присваиваться пользователю при аутентификации им через данный реалм (attributes) в формате атрибут=значение;атрибут=значение без использования синонимов для числовых типов значений (т.е. не Framed-Protocol=PPP, а Framed-Protocol=1). Создадим для примера 2 реалма - default и local. Сами типы правил пусть называются PPPoE-default и PPPoE-local. Оба правила нужно добавить для данного типа шлюза (при редактировании типа шлюза - вкладка **Типы правил**).

Создайте шлюз на вкладке **Шлюзы**, указав тип шлюза, NAS ip в качестве параметра хоста, 0 в качестве параметра порт и секрет наса - **Ключевое слово**.



С версии 4.4 возможен сброс существующих сессий при закрытии шлюза. Для этого в конфигурации шлюза необходимо указать класс инспектора сессий и его параметры, аналогично модулю DialUp (см. Настройка NASов).

Пример конфига:

```
snmp.version=2
#
nas.inspector.class=bitel.billing.server.processor.SNMPNASConnectionInspectorType3
nas.inspector.snmp.port=161
nas.inspector.snmp.community=XXXXXX
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.9.9.150.1.1.3.1.5
nas.inspector.snmp.check.oid=1.3.6.1.4.1.9.9.150.1.1.3.1.5
```

Далее для клиента в параметрах договора на вкладке модуля добавьте созданный шлюз, в настройках укажите используемые правила (реалмы), ip-адрес, маршрутизируемые сети, VRF.

IP-адреса берутся из одиночных диапазонов (81.30.224.1-81.30.224.1), список маршрутизируемых сетей ("Сети") - из обычных диапазонов (вкладка **Адреса**)

Адреса Шлюзы

Логин: 123 Пароль: ****

Реалм (правило)	VRF	Адрес	Сети
default	Internet	127.0.0.1	21.10.0.0 255.255.0.0, 217.115.84.84 255.255.255...

Редактирование

Реалм: default

IP: 127.0.0.1

VRF: Internet

Сети

- 21.10.0.0 255.255.0.0
- 217.115.84.84 255.255.255.252

Все Сброс Инверт.

Ok Отмена

Ok Отмена

После добавления/изменения параметров шлюза необходимо закрыть и снова открыть шлюз для обновления параметров на самом шлюзе.

Клиент привязывается к данному шлюзу и сможет авторизироваться только на нем. Шлюз определяется по NAS IP.

Настройка шлюза CISCO

Данный тип шлюза используется для управления ACL-записями на CISCO маршрутизаторах посредством SSH-соединения. При этом сама ACL-запись должна быть создана администратором вручную и в неё должны быть добавлены правила, разрешающие доступ к DNS и Биллинг-сервера в начале списка и правило запрета всего в конце списка. Биллинг работает только с extended ACL-списками.

Между ними биллинг помещает правила, открывающие конкретные адреса клиентов.

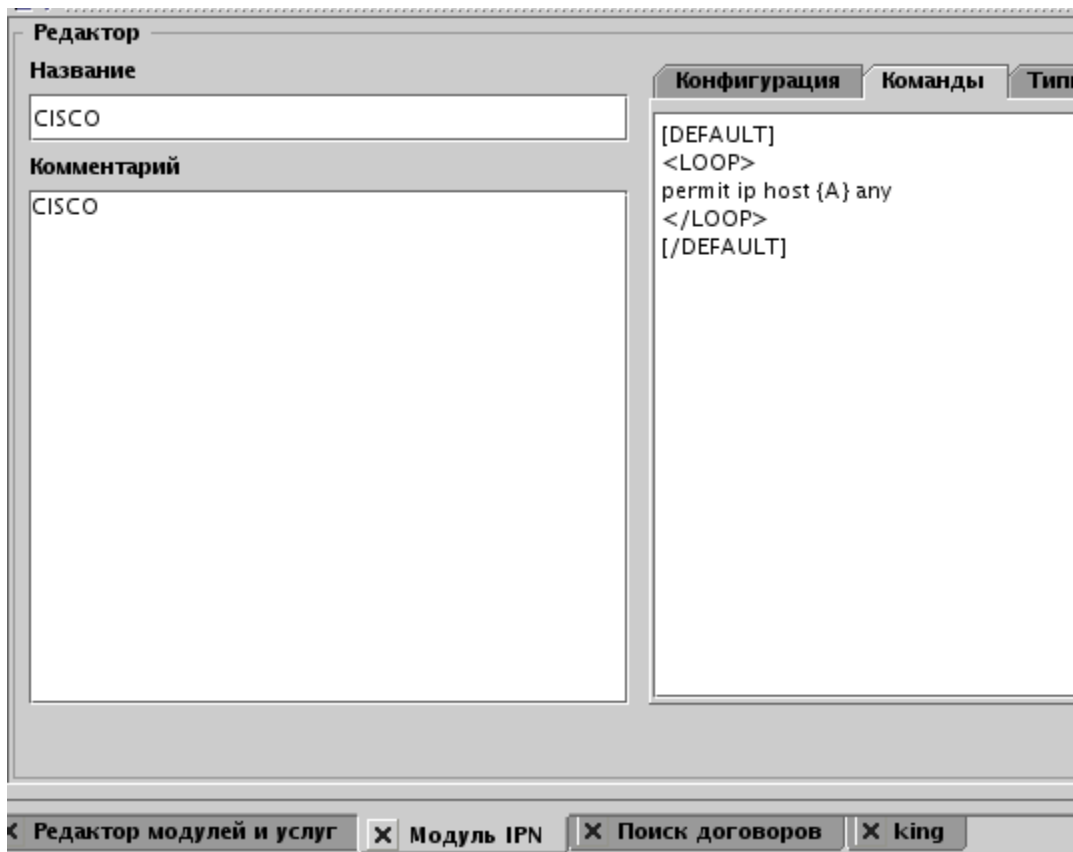
На вкладке **Типы правил** добавьте тип правила "Пустое правило" и оставьте текст этого правила пустым.

На вкладке **Типы шлюзов** создайте тип шлюза CISCO и установите в нем следующую конфигурацию:

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.CiscoContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.CiscoGateWorker
```

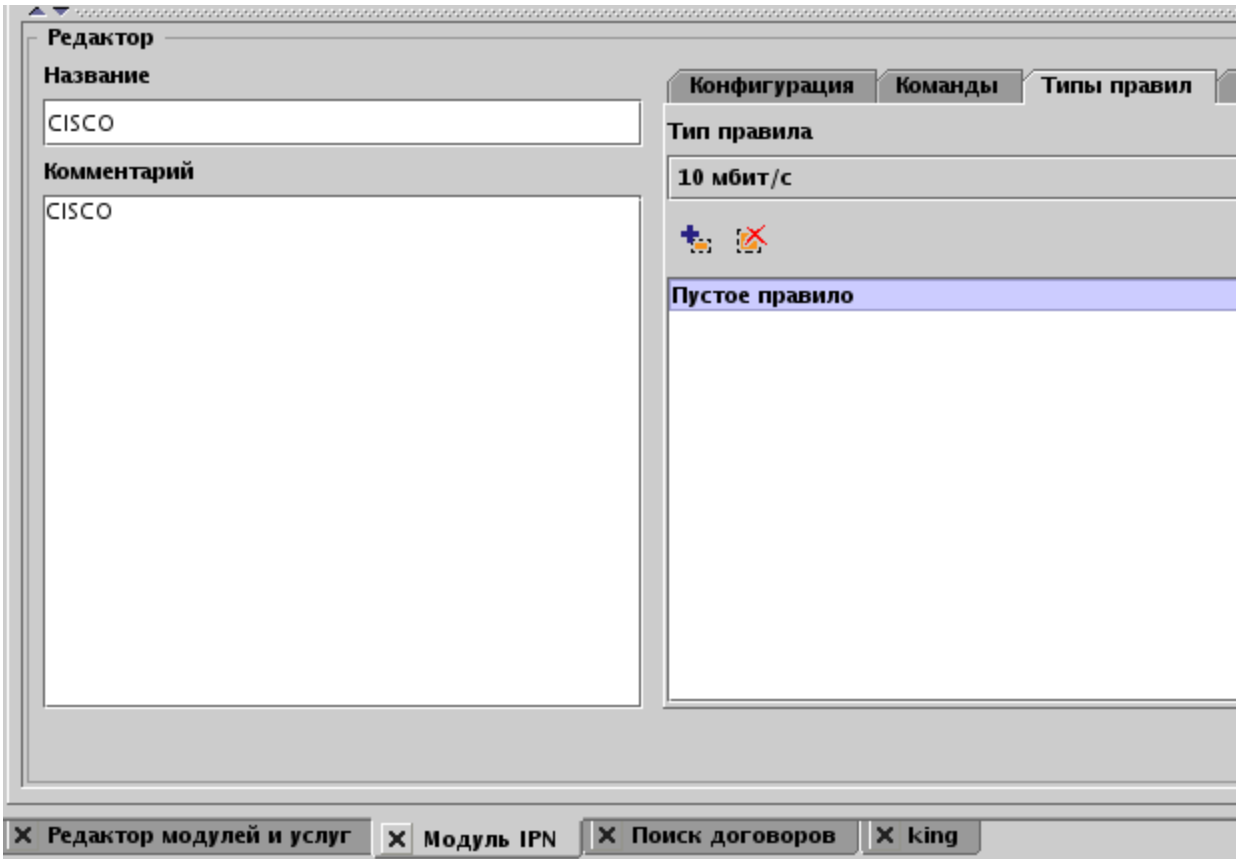
В командах этого типа шлюза установите, например:

```
[DEFAULT]
<LOOP>
permit ip host {A} any
</LOOP>
[/DEFAULT]
```

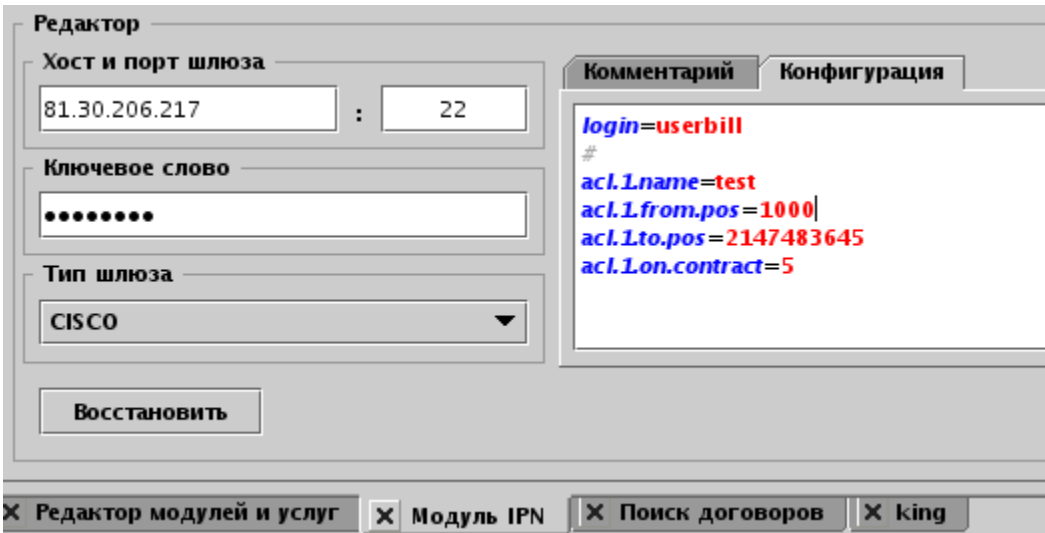


Шаблон может быть многострочным, вместо {A} будет подставлен адрес клиента. К каждому адресу будут применены все строки шаблона. Обработка команд cisco происходит аналогичным образом, как и [обработка команд Manad](#). Т.е макросы вида {A} заменяются на адрес из выбранных диапазонов для тегов <LOOP>, а макросы {NET} и {NET_MASK} заменяются на выбранные сети для тегов <LOOP_NET> и <LOOP_NET_MASK> соответственно.

На вкладке **Правила** данного типа шлюза добавьте "Пустое правило".



На вкладке **Шлюзы** заведите шлюз типа CISCO.



Необходимо указать хост и порт SSH-соединения, SSH-пароль указывается в поле **Ключевое слово**. В конфигурации шлюза указывается SSH-логин (userbill на снимке) и список ACL-записей.

```

acl.<id>.name=< ACL->
acl.<id>.from.pos=<   >
acl.<id>.to.pos=<     >
acl.<id>.on.contract=< >

```

Так же можно указать время ожидания ответа, по истечению которого, шлюз сбрасывает соединение и выдаёт ошибку :

```

timeout=2000

```


Параметр **id** - уникальный номер ACL-записи в биллинге должен быть уникальным и не меняться после создание ACL в конфигурации, если необходимо добавить ещё ACL, увеличьте его.

Пользователь на CISCO должен быть создан с привилегией 15 следующим образом:

```
aaa new-model
!
!
aaa authentication login default local enable
aaa authorization exec default local
!
username xxx password yyy
username xxx privilege 15
```

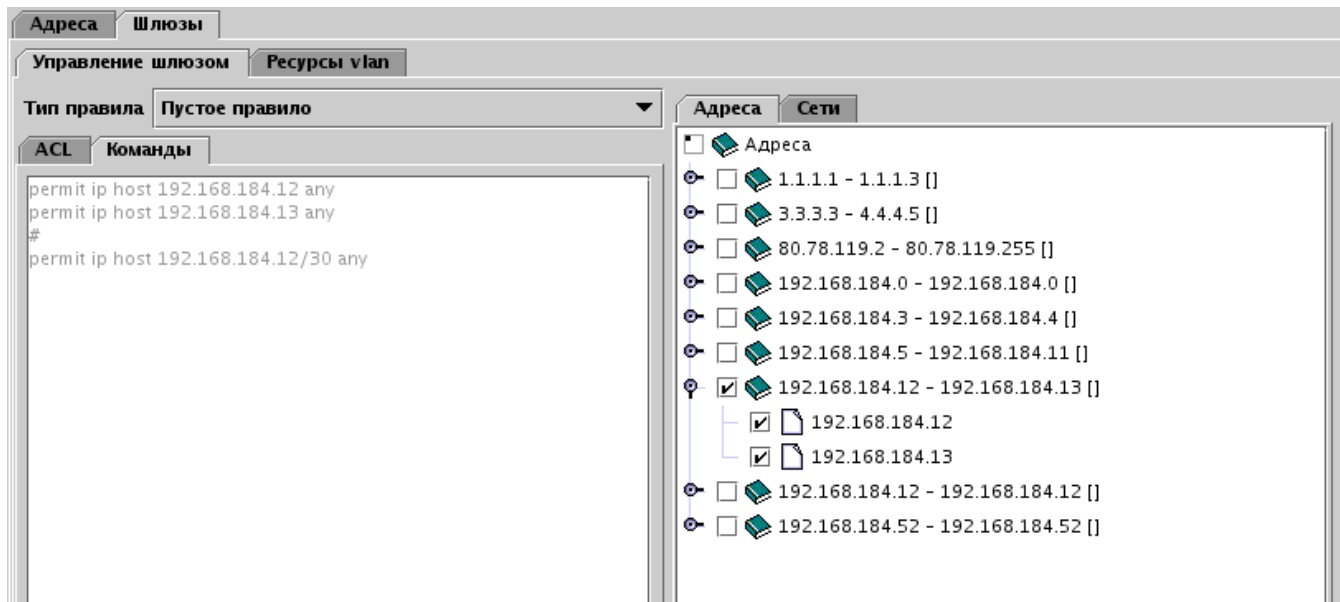
Для корректной работы шлюза на cisco символ приглашения должен быть - #.

Биллинг разделяет область правил от **from.pos** до **to.pos** на зоны, принадлежащие договорам размером **on.contract** (этот параметр желательно ставить с запасом, например 1000 так, чтобы договору точно хватило размера зоны). Далее при процедуре управления шлюзами правила либо добавляются в соответствующую договору зону, либо удаляются из неё.

При добавлении клиенту шлюза CISCO открывается редактор следующего вида. В верхнем поле необходимо выбрать правило, в списке снизу ACL, в котором открывается клиент, в дереве справа - адреса клиента на данном шлюзе.

The screenshot shows a web-based configuration interface for a Cisco gateway. The main window is titled 'Управление шлюзом' (Gateway Management) and has a sub-tab 'Ресурсы vlan' (VLAN Resources). The 'Тип правила' (Rule Type) is set to 'Пустое правило' (Empty rule). Under the 'ACL' tab, the 'С позиции:' (From position) field is set to '1000' and the 'ACL список:' (ACL list) field contains 'test'. The 'Команды' (Commands) tab is currently empty. On the right side, there is a tree view under 'Адреса' (Addresses) showing a list of IP address ranges with checkboxes. The range '192.168.184.12 - 192.168.184.13' is selected with a checked checkbox.

Поле **С позиции** пусто при добавлении шлюза клиенту, при редактировании в нем отразится с какой позиции выделена клиенту свободная зона правил ACL. На вкладке **Команды** можно видеть какие команды будут добавлены для открытия клиента.



После добавления шлюза клиенту переключение выпадающего списка управления в **Открыть** должно добавить правила на выбранный ACL.

Настройка шлюза DLINK 35xx, 38xx

Данный тип шлюза используется для управления ACL-записями данного коммутатора посредством SNMP-управления. Необходимо настроить коммутатор для приёма SNMP-запросов с сервера биллинга и создать RW community. Шлюз рассчитан на подключение одного IP-адреса на порт коммутатора с авторизацией DHCP OPTIONS 82 по порту коммутатора.

Создайте в биллинге тип шлюза DLINK со следующими настройками:

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.DlinkContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.DlinkGateWorker
# snmp 1 2c
snmp.version=1
#1 - 35xx, 2 - 38xx
model=1
```

На вкладке **Шлюзы** создайте шлюз типа **DLINK**.

Редактор

Хост и порт шлюза

85.95.148.85 : 161

Ключевое слово

Тип шлюза

Dlink

Восстановить

Комментарий

Конфигурация

```
=255.255.255.255
=85.95.148.88
=50
=49,50
=4
=8
=12
=16
=20
```

Ок

Отмена

В адрес и порт шлюза внесите адрес управления и SNMP-порт, ключевое слово - SNMP community. В конфигурации укажите следующее:

```
#
open.mask=255.255.255.255
#
open.address=85.95.148.88
#
ports=50
# uplink ( )
uplink=49,50
#
profile.4=4
profile.8=8
profile.12=12
profile.16=16
profile.20=20
profile.24=24
profile.28=28

# snmp. 500,1000,2000,5000,5000 - .
#retry.intervals=500,1000,2000,5000,5000
```

Открытые сервера - статистика, DNS и все, те которые должны быть открыты даже для заблокированного клиента. Открыты будут все адреса наложение указанной маски, на которые даст указанный адрес.

Правило

IP адрес	Порт коммутатора
77.233.171.129	43

OK

Отмена

В свойствах шлюза в договоре необходимо указать какой адрес занимает какой порт коммутатора.

Настройка в связке с DHCP шлюзом.

О том, как настроить DHCP - сервер/шлюз читайте [тут](#). Если DHCP будет работать в связке со шлюзами DLink, то необходимо добавить такие настройки для сервера DHCP.

```
processor.class=bitel.billing.server.ext.dhcp.DHCPRelayProcessor

# Option 82, VLAN ( 1)
dhcp.82.key.option.code=1
# ( ) , ( 0).
dhcp.82.key.position=5
```

После синхронизации клиента на шлюзе DLINK будет вызвана синхронизация клиента на BGDhcpIPN, а именно порт коммутатора: IP-адрес.

При подключении клиента, он отправит запрос на получение IP-адреса, коммутатор, при включённом и настроенном DHCP Relay, добавив данные RelayAgent Options, перенаправит запрос на DHCP-сервер BGDhcpIPN, который по ip шлюза и порту клиента, указанному в RelayAgent Options, выдаст IP-адрес.

Настройка сервера/шлюза DHCP

Данный тип шлюза используется для автоматической выдачи IP адресов клиентам, заведённым через шлюз DLINK и Cisco2. На коммутаторе включается режим и указывается DHCP relay server - сервер, на который коммутатор будет перенаправлять запросы DHCP. В данном случае сервер, где будет стоять BGDhcpIPN.

Распакуйте BGDhcpIPN и установите как службу. В dhcp.properties:

```
# /
admin.port=1855

# , BGDhcpIPN. ( ) - 67 .
#dhcp.port = 67
# , BGDhcpIPN. , .
#dhcp.host=

# - ,
dhcp.thread.count=10

# /:ip
dhcp.data.save.delay=3600
#
dhcp.data.save.period=3600
# ,
dhcp.data.save.path=<some_path>
# Stop, ,
#

# . ip- ,
dhcp.server.identifier=10.0.0.2
```

Специфические настройки для работы с DLink и Cisco2 читайте в главах, описывающих соответствующие шлюзы.

Запускать BGDhcpIPN необходимо от root, иначе сервер не сможет открыть DHCP-порт.

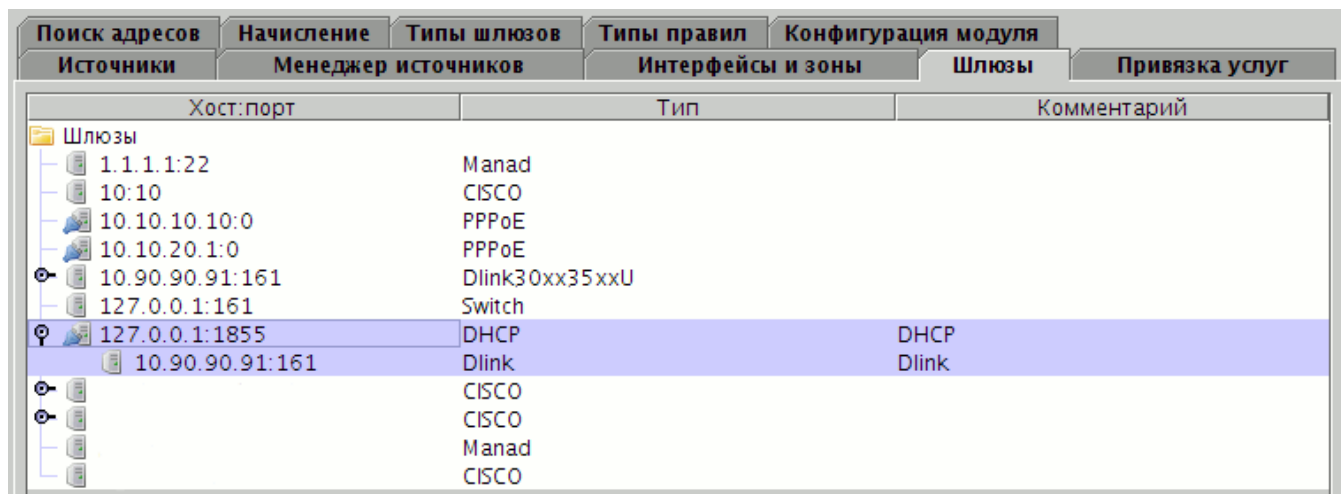
Добавьте тип шлюза DHCP с конфигурацией

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.EmptyContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.DHCPGateWorker
```

Добавьте шлюз DHCP, указав адрес и порт управления/синхронизации, а шлюзы (DLink или Cisco2), которые будут пересылать DHCP-запросы на BGDhcpIPN сделайте дочерними по отношению к этому шлюзу.

Таким образом, после синхронизации клиента на шлюзе DLINK или Cisco2 будет вызвана синхронизация клиента на BGDhcpIPN.

При подключении клиента он отправит запрос на получение IP-адреса, коммутатор, при включенном и настроенном DHCP Relay, добавив данные RelayAgent Options, перенаправит запрос на DHCP-сервер BGDhcpIPN, который по данным в RelayAgent Options, выдаст IP-адрес. По каким правилам выдавать адрес зависит от конкретного шлюза и описывается в описании этих шлюзов.



Хост:порт	Тип	Комментарий
Шлюзы		
1.1.1.1:22	Manad	
10:10	CISCO	
10.10.10.10:0	PPPoE	
10.10.20.1:0	PPPoE	
10.90.90.91:161	Dlink30xx35xxU	
127.0.0.1:161	Switch	
127.0.0.1:1855	DHCP	DHCP
10.90.90.91:161	Dlink	Dlink
	CISCO	
	CISCO	
	Manad	
	CISCO	

Параметры, выдающиеся клиенту кроме ip-адреса, могут задаться 2-мя способами.

Первый способ.

Параметры настраиваются в конфигурации шлюзов, причём родительские параметры наследуются потомками. Т.е если указать параметры только в конфигурации шлюза DHCP, то для всех шлюзов-потомков будут выдаваться такие параметры, если же в каком-либо из потомков параметр должен отличаться, можно указать только его.

```
# Time Offset
dhcp.timeOffset=-18000
# P(), -
dhcp.router=Cg
#
#dhcp.domain=
# DNS-(), -
dhcp.dns=
#
#dhcp.subnetMask=255.0.0.0
dhcp.subnetMask=

# ip-, , 43200
#dhcp.ipAddressLeaseTime
#
# dhcp, dhcp.option.x=ffffffff
# x - dhcp, ffffffff - 16
# , NTP 127.0.0.1, :
#dhcp.option.42=7F000001
# rfc2132
```

Второй способ.

Этот способ перетирает опции, установленные первым способом .

Параметры настраиваются в dhcp.properties. Они задаются с привязкой к сети, из которой будет выдаваться ip.

```
# ip 1
net.1.ip=1.1.33.0
# ( )
net.1.bits=24
# (), -
net.1.dhcp.router=10.10.10.10
#
net.1.dhcp.subnetMask=255.255.255.0

# ip 2
net.2.ip=1.1.34.0
# 2 ( )
net.2.bits=24
# (), -
net.2.dhcp.router=10.10.10.10
#
net.2.dhcp.subnetMask=255.255.255.0
```

Синтаксис задания параметров такой же, как для шлюзов, за исключением приставки net.X, где X - порядковый номер сети. Эти параметры перетирают параметры, установленные в конфигурации шлюзов.

Специфические настройки для работы с DLink и Cisco2 читайте в главах, описывающих соответствующие шлюзы.

Настройка шлюза Mikrotik RouterOS

Данный тип шлюза используется для управления маршрутизаторами Mikrotik посредством SSH или Telnet-соединения. Биллинг помещает правила, открывающие конкретные адреса клиентов.

На вкладке **Типы шлюзов** создайте тип шлюза Mikrotik и установите в нем следующую конфигурацию:

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.MikrotikContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.MikrotikTelnetGateWorker
```

Приведенная выше конфигурация для telnet-соединения. Для ssh-соединения поменяйте gate_manager.class=bitel.billing.server.ipn.MikrotikGateWorker. Но мы не рекомендуем его использовать для больших нагрузок.

На вкладке **Команды** при редактировании данного типа шлюза задайте:

```
[DEFAULT]
[OPEN]
<LOOP>
ip firewall address-list add address={A} list=ACCESS_LIST comment=!!{CID}!!
</LOOP>
[/OPEN]

[CLOSE]
<LOOP>
ip firewall address-list remove "!!{CID}!!"
</LOOP>
[/CLOSE]

[DELETE]
<LOOP>
ip firewall address-list remove "!!{CID}!!"
</LOOP>
[/DELETE]
[/DEFAULT]
```

Здесь команды разделяются на 3 блока: открывающие команды (между тегами [OPEN]/[OPEN]), закрывающие команды (между тегами [CLOSE]/[CLOSE]) и удаляющие команды (между тегами [DELETE]/[DELETE]). В данном примере закрывающие и удаляющие команды одинаковые. Сюда можно помещать любые команды, которые поддерживаются ssh-консолью Mikrotik. При этом команды проходят преобработку на стороне сервера Биллинга. Обработка команд происходит аналогичным образом, как и [обработка команд Manad](#). Т.е макросы вида {A} заменяются на адрес из выбранных диапазонов для тегов <LOOP>, а макросы {NET} и {NET_MASK} заменяются на выбранные сети для тегов <LOOP_NET> и <LOOP_NET_MASK> соответственно.

Ещё для шлюза Mikrotik производятся дополнительные преобразования: макрос {CID} - преобразуются в код договора, независимо от того, где встретится (внутри цикла LOOP или нет).

Т.е. например правила:

```
<LOOP>
ip firewall address-list add address={A} list=ACCESS_LIST comment=!!{CID}!!
</LOOP>
```

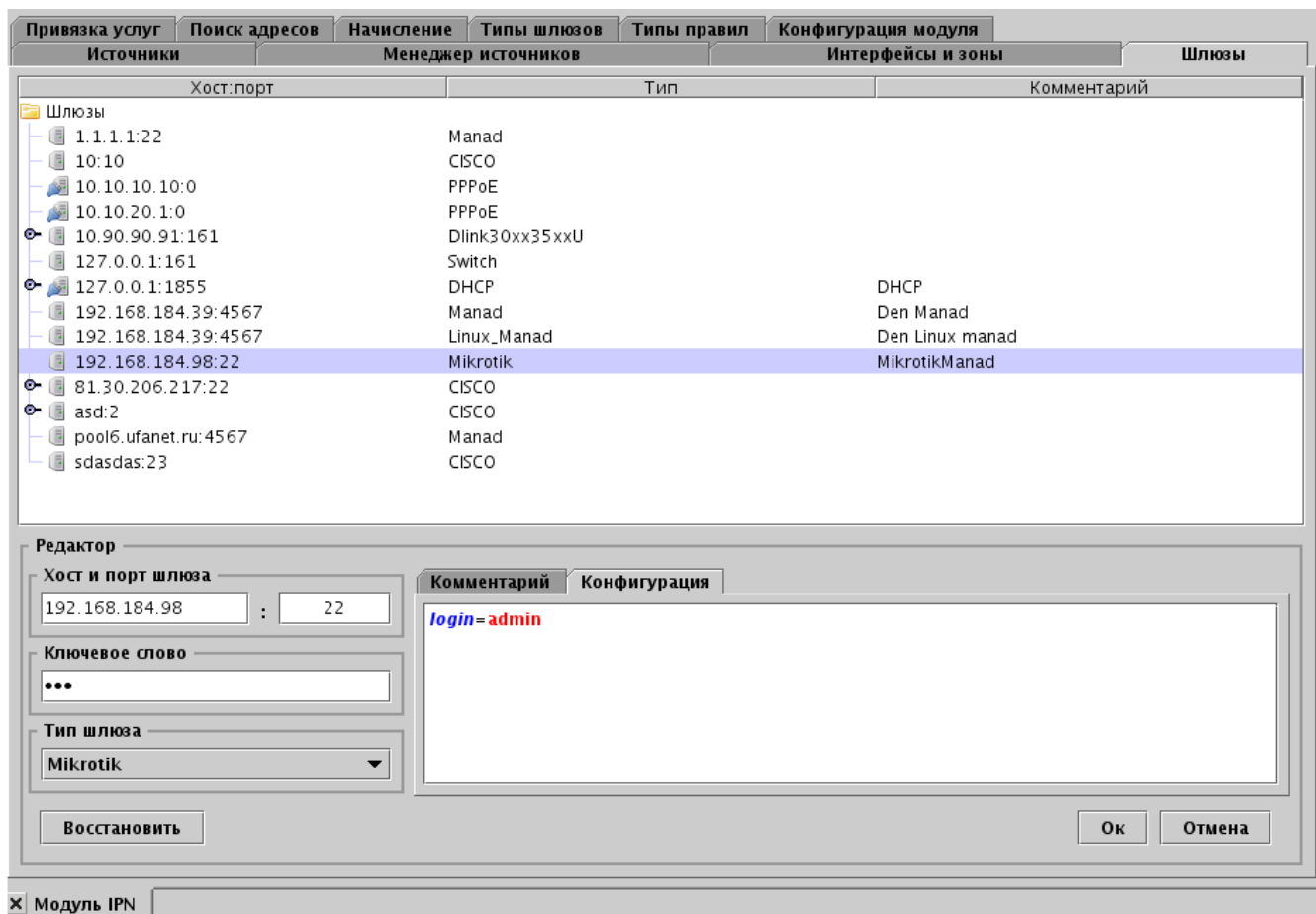
Преобразуется в :

```
ip firewall address-list add address={192.168.184.10} list=ACCESS_LIST comment=!!12345!!
ip firewall address-list add address={192.168.184.11} list=ACCESS_LIST comment=!!12345!!
```

если выбраны адреса 192.168.184.10 и 192.168.184.11 и код договора - 12345.

На вкладке **Типы правил** можете завести (если у вас ещё нет) - "Пустое правило" и добавить его для данного типа шлюза .

На вкладке **Шлюзы** создайте новый шлюз типа Mikrotik и установите в нем следующую конфигурацию:

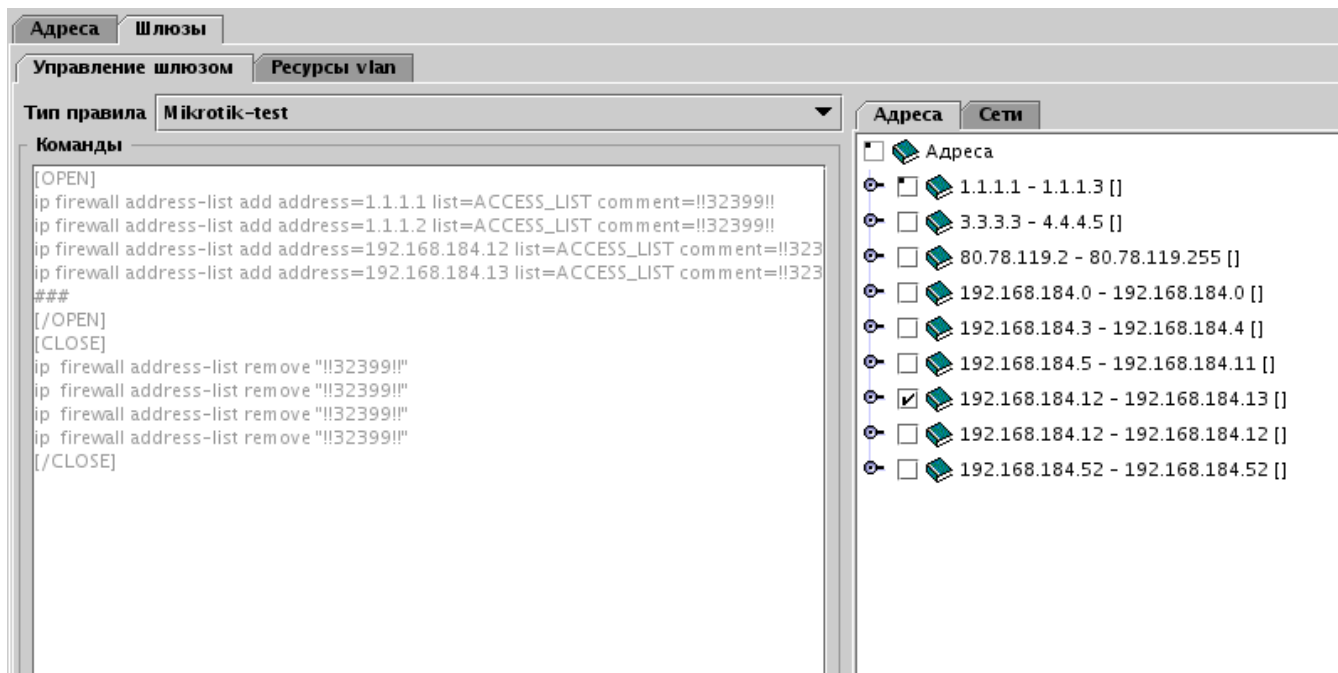


Здесь login - это логин, а ключевое слово - пароль для ssh-соединения Mikrotik.

Также можно указать время ожидания ответа, по истечению которого шлюз сбрасывает соединение и выдаёт ошибку :

```
timeout=2000
```

Далее в договоре выбираем модуль IPN, в нем на вкладке **Шлюзы** добавляем наш новый шлюз и для него указываем наш тип правил test:



Для инициализации правил в системе Mikrotik запускаем следующие команды:

```
[admin@MikroTik] > ip firewall filter add chain=forward action=accept dst-address-list=ACCESS_LIST
[admin@MikroTik] > ip firewall filter add chain=forward action=accept src-address-list=ACCESS_LIST
[admin@MikroTik] > ip firewall filter add chain=forward action=drop
```

В данном примере происходит управление с помощью добавления и удаления ip-адреса в ACCESS_LIST. Все правила добавляются с комментарием "!! код договора !!", чтобы можно было их удалять по этому комментарию. Вы можете менять правила для управления шлюзом Mikrotik произвольным образом (добавлять ширину каналов и т.п.), но есть ограничение: должно быть хотя бы одно правило, добавляющее ip-адрес в address-list, т.к. проверка при синхронизации шлюза ищет с помощью команды "ip firewall address-list print" строку с комментарием "!! код договора!!". Присутствие этой строки означает, что шлюз открыт для данного договора.

Настройка шлюза Cisco2 с коммутаторами

Данный шлюз предназначен для комплексного управления CISCO и коммутатором Zyxel(ES-2108-G/ES-2024A/GS-3012F и совместимые с ними).
На каждого клиента есть возможность выделения отдельного VLAN.

Настройка шлюза Cisco2

Шлюз Cisco является родительским шлюзом для коммутатора . Данный шлюз управляется по telnet.

Создайте тип шлюза со следующей конфигурацией:

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.vlan.CiscoVlanContactRuleEditor
gate_manager.class=bitel.billing.server.ipn.vlan.CiscoVlanGateWorker
```

В командах этого типа шлюза задайте:

```
[DEFAULT]

[REMOVE]
  no vlan {VID}
  no interface Vlan {VID}
[/REMOVE]

[OPEN]
  vlan {VID}
  interface Vlan {VID}
  no shutdown
  ip unnumbered Loopback1
  ip helper-address 172.17.0.7
  exit
[/OPEN]

[CLOSE]
  interface Vlan {VID}
  shutdown
  exit
[/CLOSE]
[/DEFAULT]
```

[OPEN]/[/OPEN] - это команды, которые посылаются на cisco при открытии шлюза. [CLOSE]/[/CLOSE] - команды, которые посылаются на cisco при закрытии шлюза. [REMOVE]/[/REMOVE] - команды, которые посылаются при удалении шлюза из договора.

Команды проходят предобработку на стороне сервера Биллинга. Обработка команд происходит аналогичным образом , как и [обработка команд Manad](#). Т.е макросы вида {A} заменяются на адрес из выбранных диапазонов для тегов <LOOP>, а макросы {NET} и {NET_MASK} заменяются на выбранные сети для тегов <LOOP_NET> и <LOOP_NET_MASK> соответственно.

Ещё для шлюза Cisco2 производятся дополнительные преобразования: макрос {VID} заменяется на номер VLAN, которой выделен на данный договор и шлюз (как задаётся выделение vlan описано [тут](#)), независимо от того, где встретится (внутри цикла LOOP или нет).

В типах правил для этого скрипта можете добавить пустое правило.

Создайте шлюз данного типа. В конфигурацию данного шлюза добавьте :

```
#      enable
cfg.pswd=12345
#      VLAN
range=2-4096
#      ,
timeout=2000
```

Задайте ip-адрес шлюза, в качестве ключевого слова - забейте пароль к cisco, порт - 23.

Данный шлюз можно использовать автономно и в связке с дочерним коммутатором (в стандартной поставке это zyxel, но его можно подменить любым другим , например DLink, с помощью [реализации собственного шлюза на языке beanShell](#). В договор этот шлюз добавляется только, если он используется автономно (**Это важно!!**). При редактировании в договоре этот шлюз выглядит так

Адреса Шлюзы

Управление шлюзом Ресурсы vlan

Тип CISCO_VLAN

Номер vlan 5

Команды

```
[ADD]
[ADD]
[REMOVE]
no vlan 5
no interface Vlan 5
[/REMOVE]
[OPEN]
vlan 5
interface Vlan 5
no shutdown
ip unnumbered Loopback1
ip helper-address 172.17.0.7
ip 192.168.184.12
ip 192.168.184.13
###
net 192.168.184.12/30
net 192.168.184.48/30
exit
[/OPEN]
[CLOSE]
interface Vlan 5
shutdown
exit
[/CLOSE]
```

Адреса Сети

- Адреса
- 1.1.1.1 - 1.1.1.3 []
- 3.3.3.3 - 4.4.4.5 []
- 80.78.119.2 - 80.78.119.255 []
- 192.168.184.0 - 192.168.184.0 []
- 192.168.184.3 - 192.168.184.4 []
- 192.168.184.5 - 192.168.184.11 []
- 192.168.184.12 - 192.168.184.13 []
- 192.168.184.12 - 192.168.184.12 []
- 192.168.184.52 - 192.168.184.52 []

Логика работы шлюза :

```
password:xxxxx
xxxx>terminal length 0
xxxx>terminal width 0
xxxxx>enable
password:xxxxx
xxxx#
xxxx#permit 1.1.1.1
xxxx#no permit 1.1.1.2
....
xxxx#exit
```

Имеет большое значение настройка и завершаемые символы приглашения. Вначале в конце выводимой cisco информации ожидается ":" (пользователь вводит пароль, который забит в ключевом слове шлюза), потом ожидается ">" и шлюза 2 команды размеры терминала, посылаются команды enable, затем ожидается ":"(пользователь вводит пароль), после этого уже уждается "#" и посылаются команды управления. Настройки вашей cisco должны соответствовать этому алгоритму. Если этого сделать не получается (например, вы хотите пропустить процедуру авторизации или у вас иной завершающий символ приглашения и не хотите его менять), то вы можете использовать аналог этого шлюза на BeanShell из документации Wiki. О том как создавать собственные шлюзы читайте [тут](#).

Настройка шлюза коммутатора Zyxel

Этот шлюз работает только в связке со шлюзом Cisco2, описанным в предыдущей главе. Если вы хотите его запустить автономно, то все равно должен быть родительский шлюз Cisco2, логику которого можно подменить [пустым скриптом BeanShell](#). В текущий момент поддерживаются Zyxel ES 2024A, Zyxel ES 2108G, Zyxel GS 3012F и совместимые с ними.

Для родительского типа шлюза Cisco2 в этом случае надо прописать вот такую конфигурацию :

```
gate_manager.class=bitel.billing.server.ipn.vlan.CiscoVlanParentGateWorker
```

Типовая схема - каждый клиент помещается в отдельный VLAN. Клиент может быть подключён к cisco через цепочку управляемых шлюзов, поддерживающих протокол GVRP. VLAN прописывается на шлюзе cisco и на конечном шлюзе, к порту которого подключён абонент. Остальные шлюзы могут получать информацию о VLAN по протоколу GVRP.

Данный шлюз управляется по ssh.

Создайте тип шлюза со следующей конфигурацией:

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.vlan.CiscoSSHSwitchRuleEditor
gate_manager.class=bitel.billing.server.ipn.vlan.CiscoSSHSwitchGateWorker
```

В командах данного типа шлюза пропишите :

```
[DEFAULT]
[OPEN]
  <LOOP_PORT>
    interface port-channel {PORT}
      pvid {VID}
      exit
  </LOOP_PORT>

  vlan {VID}
  name "abonent {VID}"
  normal ""
  <LOOP_PORT>
    fixed {PORT}
  </LOOP_PORT>
  untagged 1-8
  exit
[/OPEN]

[CLOSE]
no vlan {VID}
[/CLOSE]

[/DEFAULT]
```

Обработка команд происходит аналогичным образом, как и [обработка команд Manad](#). Т.е макросы вида {A} заменяются на адрес из выбранных диапазонов для тегов <LOOP>, а макросы {NET} и {NET_MASK} заменяются на выбранные сети для тегов <LOOP_NET> и <LOOP_NET_MASK> соответственно.

Ещё для шлюза Zyxel производятся дополнительные преобразования:

1. Макрос {VID} заменяется на номер VLAN, который выделен на данный договор и шлюз (как задаётся выделение VLAN описано [тут](#)), независимо от того, где встретится (внутри цикла LOOP или нет).
2. Для каждого тега <LOOP_PORT> вместо макроса {PORT} подставляются порты клиента.

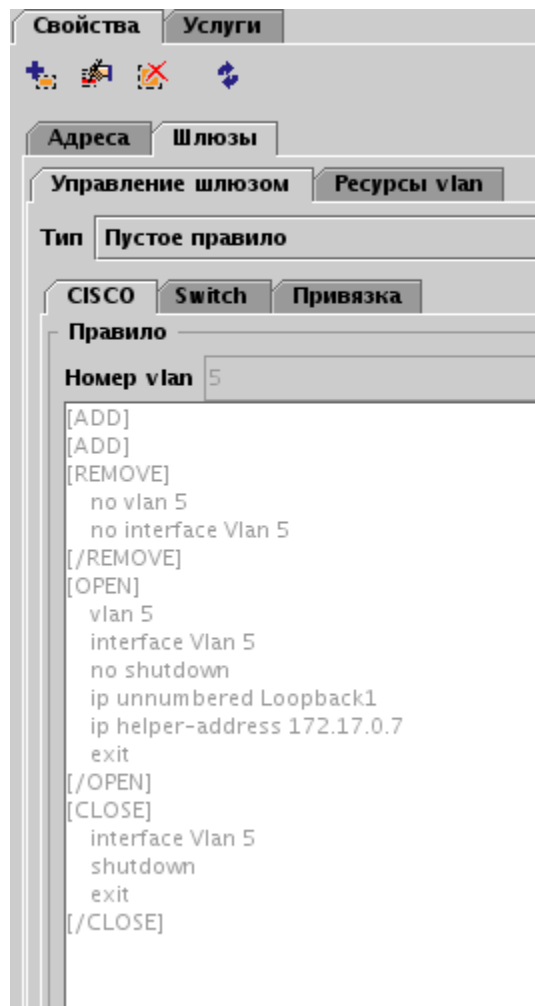
В правилах данного типа шлюза добавляется пустое правило.

Создайте шлюз данного типа как шлюз потомок для Cisco2. В конфигурацию данного шлюза добавьте :

```
#
login=admin
#
timeout=20000
```

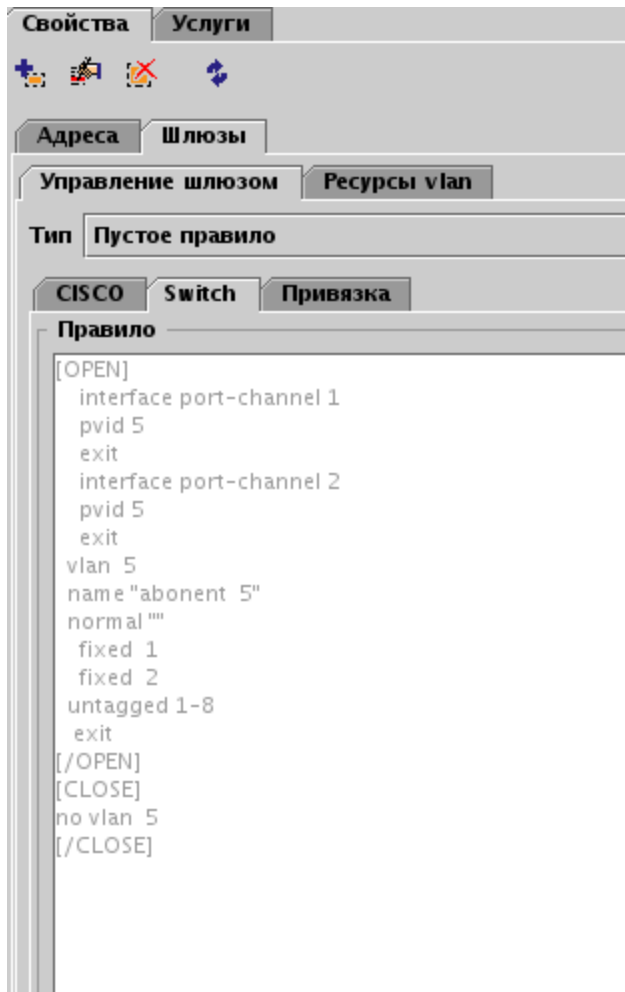
Задайте хост и порт (22) шлюза, в качестве ключевого слова забейте пароль.

Добавьте данный шлюз в договор пользователя. Вы увидите следующую картину :

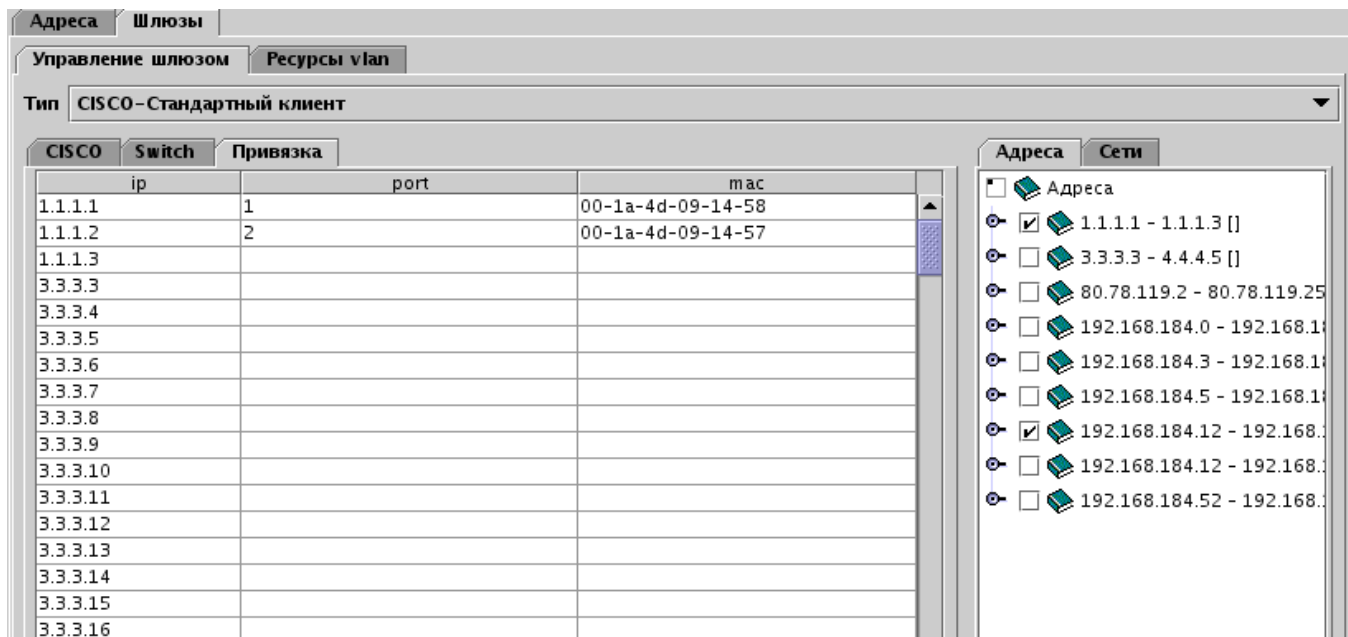


На вкладке Cisco показаны команды, которые будут вызываться на шлюзе Cisco.

На вкладке Switch - команды, которые будут вызываться на коммутаторе.



На вкладке привязка вы можете задавать привязку ip-адресов к портам и mac адресу .



Порты подставляются в команды коммутатора вместо макроса {PORT} . Также эта таблица используется шлюзом DHCP для выдачи ip по mac.

Выбор адресов и сетей влияет на команды (вкладки) Cisco и Switch.

Настройка шлюза DHCP в связке с Cisco2.

О том, как настроить DHCP-сервер/шлюз читайте [тут](#). Если DHCP будет работать в связке со шлюзом Cisco2, то необходимо добавить такие настройки для сервера DHCP. Причём эта схема работает только при наличии коммутатора, т.е должна быть связка 3-х шлюзов : DHCP, Cisco2 и Zyxel.

```
processor.class=bitel.billing.server.ext.dhcp.DHCPVlanRelayProcessor

# Option 82,      VLAN ( 1)
dhcp.82.key.option.code=1
# ( ) ,      VLAN ( 0).
dhcp.82.key.position=2
```

После синхронизации клиента на коммутаторе Zyxel, будет вызываться синхронизация с родительским шлюзом Cisco2, а потом будет вызываться синхронизация со шлюзом DHCP, который является родительским для Cisco2. Шлюз DHCP, в свою очередь, вызовет синхронизацию клиента на BGDhcpIPN. А именно порт коммутатора: IP-адрес:MAC-адрес.

При подключении клиента, он отправит запрос на получение IP-адреса, коммутатор при включенном и настроенном DHCP Relay, добавив данные RelayAgent Options, перенаправит запрос на DHCP-сервер BGDhcpIPN, который по данным в RelayAgent Options, выдаст IP-адрес. При этом возможны 2 варианта:

1. Выдать ip-адрес по VLAN;
2. Выдать ip-адрес VLAN и MAC-адресу клиента, указанному в привязке шлюза;

Для оптимизации работы скомпилированных шлюзов нужно поставить в конфигурацию модуля IPN:

```
gate.cache.script=1
```

Эта опция кешерования скомпилированных шлюзов, чтобы они каждый раз не компилировались при запуске синхронизации шлюза.

Реализация шлюза на языке BeanShell

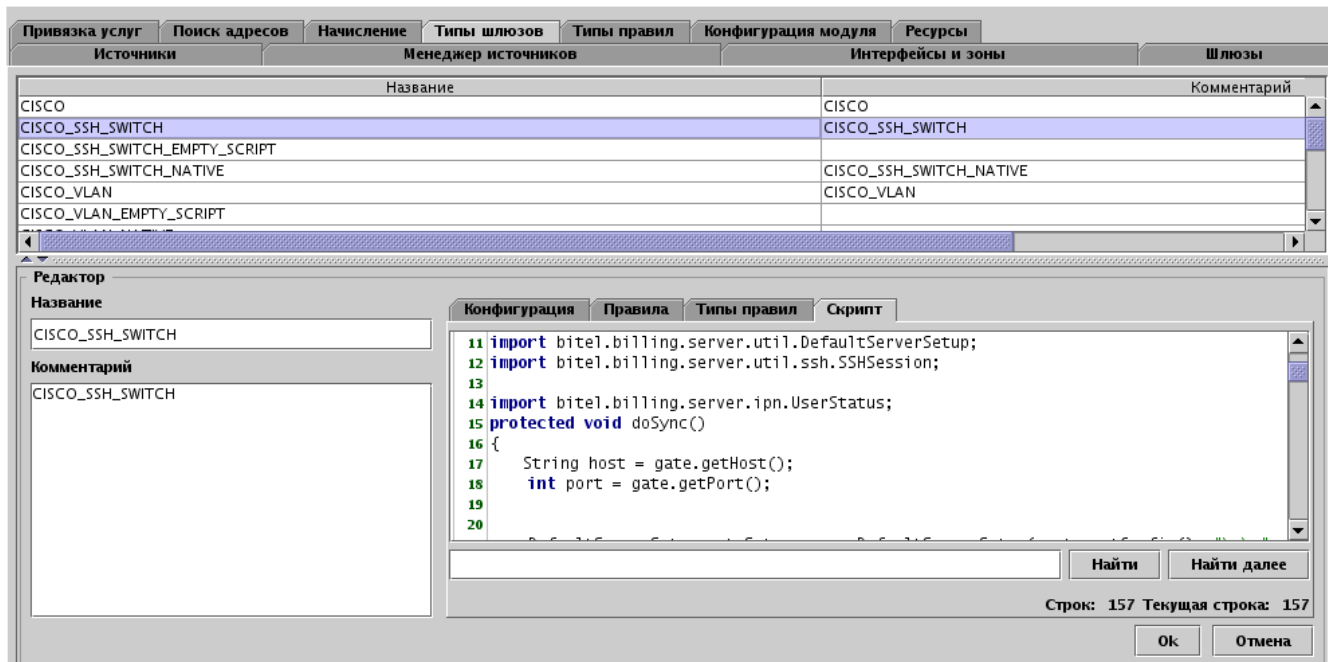
В версии 4.5 помимо стандартных шлюзов, возможно создание своего собственного шлюза с помощью встроенного языка BeanShell.

В этом случае в конфигурации типов шлюзов вы можете указывать например :

```
user_rule.editor.class=bitel.billing.module.services.ipn.editor.ManadContractRuleEditor
gate_manager.class=bitel.billing.server.ipn.ManadGateWorker
use.script=1
```

Здесь **user_rule.editor.class** - это класс, который отображает редактор шлюза в клиенте биллинга и сохраняет данные шлюза .Вы можете поставить один из стандартных классов (указаны в документации в соответствующих главах), либо поставить **user_rule.editor.class**. **EmptyContractRuleEditor**, если редактировать шлюз не нужно в договоре. **gate_manager.class** - это класс, осуществляющий взаимодействие со шлюзом на стороне сервера биллинга. Именно логику этого класса мы и подменяем своим скриптовым шлюзом. Вы можете указать **user_rule.editor.class=bitel.billing.server.ipn.EmptyGateWorker**. Но если вы подменяете логику работы одного из стандартных шлюзов, то желательно указывать класс этого шлюза, т.к. логика его работы будет подменена, но есть ещё другие правила, например, в случае сложных иерархических шлюзов производится синхронизация с родительским шлюзом и она пока не может быть реализована скриптовым шлюзом. **use.script=1** — это параметр, задающий, что вместо стандартной логики шлюза будет выполняться скрипт

Сам код шлюза заносится на вкладку "Скрипт" при редактировании типа шлюза:



Должна быть обязательно реализована функция doSync() - именно она и отвечает за взаимодействие со шлюзом. Вот минимальный код шлюза:

```
protected void doSync()
{
}
```

Также дополнительно может быть определена дополнительная функция синхронизации со шлюзом предком. Она определяется у шлюза предка и вызывается потомком перед вызовом doSync().

```
protected void parentSync( Gate child, GateWorker childWorker)
{
}
```

где child - объект типа bitel.billing.server.ipn.bean.Gate, представляющий шлюз, который вызывал этот метод.

childWorker - объект типа bitel.billing.server.ipn.GateWorker, представляющий шлюз, который вызывал этот метод.

В скрипт передаются такие переменные:

con — объект java.sql.Connection для соединения с БД;

gate — объект bitel.billing.server.ipn.bean.Gate;

ruleTypeMap — объект java.util.Map<Integer, bitel.billing.server.ipn.bean.RuleType>. В нем содержатся все типы правил из БД. Ключ — id правила;

gateTypeMap — объект java.util.Map<Integer, bitel.billing.server.ipn.bean.GateType> gateTypeMap. В нем содержатся все типы шлюзов из БД. Ключ — id типа шлюза;

Log — объект org.apache.log4j.Logger. Для вывода сообщений, ошибок шлюза;

StatusList — объект List<bitel.billing.server.ipn.UserStatus> statusList. В нем содержатся статусы пользователей, которые синхронизируются на данном шлюзе. Ключ — id статуса. ;

GateErrors — объект java.lang.StringBuilder, содержащий ошибки выполнения шлюза;

mid — int, код модуля;

worker - объект типа GateWorker, обработчик логики работы со шлюзом, который вызвал этот скрипт.

Примеры шлюзов и реализация большинства стандартных шлюзов есть на [Wiki](#).

Отчёты модуля, детализация

Модуль предоставляет единственный отчёт по трафику. При выборе года отображается отчёт по месяцам, при выборе года и месяца - по дням месяца. При выборе дня месяца - по часам.

Возможно задание режима отображения: байты, Кб, Мб, Гб. На вкладке **Печать** доступна печатная форма отчёта. В фильтре по адресам существует возможность выбрать интересные диапазоны для вывода их трафика в отчёта. Для выбора доступны только те адреса, период которых пересекается с выбранным периодом просмотра. При неустановленном фильтре по адресам отображается отчёт по всем адресам договора. Для вывода отчёта нажмите на кнопку **Вывести**.

Параметры Отчет Карточки HelpDesk CRM Документы

Ед. измерения
байт Кб Мб Гб

Период
2006 2007 2008 2009
01 02 03 04 05 06
07 08 09 10 11 12
Пн. Вт. Ср. Чт. Пт. Сб. Вс.
01
02 03 04 05 06 07 08
09 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28

Вывести

Услуги Адреса

Услуги
 Входящий POP3
 Исходящий внутренний
 Внешний вх.
 Локальный исх.
 Локальный вх.
 Максимальный IPN

Просмотр Печать

Всего: 22.8 Мбайт

01	0	11	0	21	0
02	0	12	0	22	0
03	0	13	0	23	0
04	0	14	0	24	0
05	0	15	0	25	0
06	0	16	0	26	0
07	0	17	0	27	0
08	0	18	0	28	0
09	22.8	19	0		
10	0	20	0		

Запросить детализацию

За час За период

Адреса: 192.168.184.35 [localhost: ANY] (06.02.2009-) Дата: 09.02.2009

E-Mail: test@bitel.ru Час:

Все Сброс Инверт. Выполнить запрос

Dial-Up IPN Phone VoIP Бухгалтерия

Для получения детализации по трафику используется область под отчётом **Запросить детализацию**.

Детализация трафика за час

Для получения отчёта введите в область под отчётом на вкладке **За час** диапазон адресов, дату, час и E-Mail для высылки отчёта. Задание на детализацию исполняет VGIPNetFlowCollector. Выбирая детализацию из бинарных файлов, он отправляет письмо с отчётом по указанному адресу.

Стандартная детализация за час представляет из себя одно или несколько писем с приложенным архивом detail.zip. Архив содержит CSV-файл с частью детализации. Размер максимального возможного detail.zip в одном письме задается переменной **ipn.collector.detail.max.attach.size** в файле netflow_ipn.properties. По умолчанию значение переменной составляет 4000000 байт. Каждое письмо сопровождается темой **IPN traffic detail [N]**, где N - увеличивающийся порядковый номер. Последнее письмо данной детализации снабжено темой **IPN traffic detail [LAST]** и содержит простейшую аналитику на какие и с каких адресов был максимальный трафик.

Можно заменить стандартную детализацию, для этого необходимо указать в netflow_ipn.properties

```
ipn.collector.detail.class=bitel.billing.server.netflow.ipn.detail.AnalyzedFlowDetailMaker
```

Данный класс создания детализации кроме стандартного csv добавляет в архив detail.zip html файл с графиком и анализом.



Анализ

Распределение трафика по ip адресам, протоколам и портам, данные представлены в килобайтах.

HTTP - трафик характерный при работе с www ресурсами

SMTP - отправка почты

POP3/IMAP - прием почты

FTP - трафик характерный при работе с ftp ресурсами

UDP named - трафик полученный в результате преобразования имен хостов в ip адрес или обратно, очень большой трафик по этому сервису характерен при заражении вирусом машин в локальной сети

Данные представлены в отсортированном по уменьшению порядке

IP	Всего		TCP/UDP/ICMP/Другое		TCP: HTTP		TCP: SMTP/POP3/IMAP/FTP		UDP named	
	Вх.	Исх.	Вх.	Исх.	Вх.	Исх.	Вх.	Исх.	Вх.	Исх.
124.40.41.23	157782.40	2959.82	157782.40	2959.82	157782.40	2959.82	0.00	0.00	0.00	0.00
89.208.157.162	67967.47	1333.74	67967.47	1333.74	67967.47	1333.74	0.00	0.00	0.00	0.00
94.100.179.39 content.video12.mail.ru	34408.11	714.05	34408.11	714.05	34408.11	714.05	0.00	0.00	0.00	0.00
88.212.196.109 host09.rax.ru	29448.65	600.36	29448.65	600.36	29448.65	600.36	0.00	0.00	0.00	0.00
93.158.133.213 clip06d.video.yandex.net	15092.48	284.01	15092.48	284.01	15092.48	284.01	0.00	0.00	0.00	0.00
195.208.8.201	14277.80	3079.97	14277.80	3079.97	0.00	0.00	53.00	2621.99	14224.80	457.98

Детализация в этом случае будет приходиться одним письмом, разбиения в данный момент не предусмотрено.

Детализация трафика за период

Детализация за период сохраняется в файл. Выберите диапазон адресов, период и имя файла. Каталог, куда будет выкладываться детализация на сервере прописывается в настройках коллектора в файле **netflow_ipn.properties**. Названия каталогов на разных коллекторах могут отличаться, но все коллекторы должны ссылаться на один каталог.

```
ipn.collector.detail.folder=/home/billing/detail
```

Запросить детализацию

За час За период

Адреса: ▼ Период: с по

Файл:

Web-интерфейс модуля

В меню **IP-статистика (IPN)** пользователь имеет возможность просмотра отчёта по трафику. Возможен просмотр отчёта за день, месяц и год, по одному или всем диапазонам адресов клиентов.

Меню

- Новости
- Просмотр баланса
- Смена пароля на доступ к статистике
- Подписка на рассылки
- Смена тарифных планов
- Управление лимитом
- Дополнительные действия
- IP-статистика (IPN)**
- Управление шлюзом (IPN)
- Счета
- Счета-фактуры
- Выход

IP-статистика

День: -- Месяц: ноябрь Год: 2007

Услуги

- (IPN) Wsus (входящий)
- (IPN) Wsus (исходящий)
- (IPN) Внешний трафик (входящий)
- (IPN) Внешний трафик (исходящий)
- (IPN) Локальный трафик (входящий)
- (IPN) Локальный трафик (исходящий)
- (IPN) Почтовый трафик с Уфанет (входящий)
- (IPN) Прокси IP1(входящий)
- (IPN) Радио (входящий)
- (IPN) Радио (исходящий)

Адреса

- 81.30.202.52 (26.03.2003 - 11.11.2004)
- 81.30.202.52 (12.11.2004 -)

Для выделения сразу нескольких услуг/адресов держите нажатой клавишу 'Ctrl'

Итого: 20 583 456 (20 Мбайт)

01	02	03	04	05	06
537 469	777 625	0	0	0	552 660
07	08	09	10	11	12
1 999 538	113 874	419 675	11 814	0	5 790 865
13	14	15	16	17	18
1 812 450	5 019 609	569 376	929 455	0	0
19	20	21	22	23	24
615 797	314 290	828 991	289 968	0	0
25	26	27	28	29	30
0	0	0	0	0	0

Для запроса детализации по трафику необходимо выбрать дневной режим просмотра, затем выбрать интересующий час, адрес и ввести E-Mail, на который будет отправлена детализация. Через Web-интерфейс возможен заказ только детализации за час.

Договор № x0000

Субдоговора: x0002

Меню

- Новости
- Просмотр баланса
- Смена пароля на доступ к статистике
- Подписка на рассылки
- Смена тарифных планов
- Дополнительные действия
- Управление ящиками
- Пересылки
- Смена пароля на ящики
- Платежные системы
- IP-статистика (IPN)
- Управление шлюзом (IPN)
- Пополнение счета с помощью Интернет-карт
- Активация TгауInfo
- Счета
- Счета-фактуры
- Просмотр сессий Voicelp
- Наработка по логинам Voicelp
- Смена пароля на логины Voicelp
- Наработка логина Voicelp по направлениям
- Кошелек
- Управление подпиской CerberCrypt
- Виртуальный кинотеатр CerberCrypt
- Просмотр сессий Телефонии
- Наработка по Телефонии
- Наработка по направлениям Телефонии
- Наработка по услугам Телефонии
- Детализация по Телефонии
- Просмотр сессий DialUp
- Наработка по логинам DialUp
- Смена пароля на логины DialUp
- Учетные периоды

IP-статистика

День: 21 Месяц: январь Год: 2008 Вывести

Услуги	Адреса
Внешний вх.	0.0.0.1 (10.01.2008 -)
Исходящий внутренний	81.30.0.0 - 81.30.255.255 (08.06.2006 -)
Локальный вх.	192.168.184.29 (02.01.2008 -)
Локальный исх.	192.168.184.97 (21.01.2008 -)
Максимальный IPN	192.168.184.110 - 192.168.184.120 (03.09.2007 -)
	192.168.184.254 (21.01.2008 -)
	192.168.228.8 (17.09.2007 -)
	217.115.84.0 - 217.115.84.255 (14.11.2006 -)

Для выделения сразу нескольких услуг/адресов держите нажатой клавишу 'Ctrl'



Итого: 372 626 400 (355 Мбайт)

00	01	02	03	04	05
0	0	0	0	0	0
06	07	08	09	10	11
0	0	0	0	0	0
12	13	14	15	16	17
0	0	0	92 222 240	280 404 160	0
18	19	20	21	22	23
0	0	0	0	0	0

Выслать детализацию для адресов: 0.0.0.1 (10.01.2008 -)
за час: 15 на email: test@bitel.ru Выслать

В меню **Управление шлюзом (IPN)** пользователь может временно закрыть свой шлюз, либо посмотреть текущее состояние, если он заблокирован.

Меню

- Новости
- Просмотр баланса
- Смена пароля на доступ к статистике
- Подписка на рассылки
- Смена тарифных планов
- Управление лимитом
- Дополнительные действия
- IP-статистика (IPN)
- Управление шлюзом (IPN)
- Счета
- Счета-фактуры
- Выход

Управление шлюзом

Текущее состояние шлюза: открыт Применить

Дата и время	Действие	Исполнитель	Комментарий
--------------	----------	-------------	-------------

В меню **Управление правилами шлюзов(IPN)** пользователь может менять типы правил на своих шлюзах.

Управление правилами шлюза

Шлюз	Тип правила
103 -127.0.0.1 : 11 [zyxel script]	10 мбит/с
61 -127.0.0.1 : 4567 [Manad Common]	30
Сменить	

Для того, чтобы пользователь мог менять правила на шлюзе в его конфигурации должно быть указано .

```
customer.control=1
```

Модуль DialUp

В настоящее время для данного модуля разработана более современная замена - модуль [Inet](#). Модуль DialUp оставлен для совместимости и постепенно будет удалён.

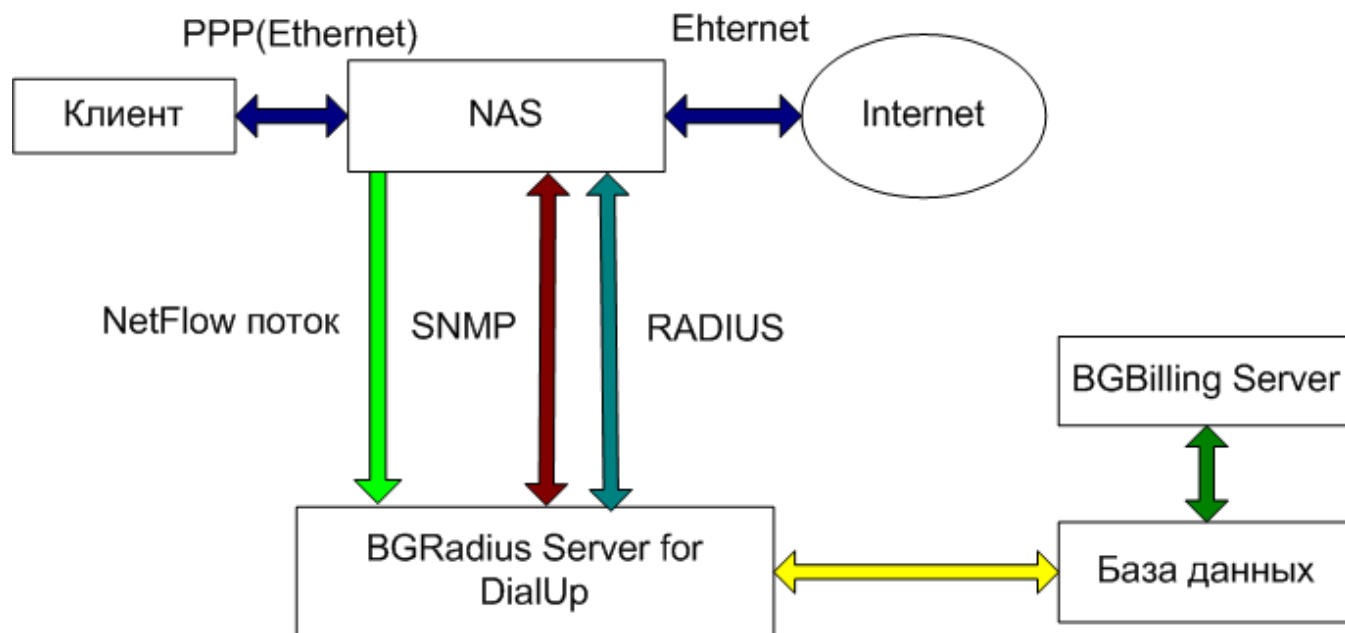
Назначение модуля

Модуль [DialUp](#) предназначен для подсчёта времени и трафика клиентов, работающих по выделенным телефонным линиям, либо через VPN-туннели. Обсчёт трафика возможен на основании данных из RADIUS пакетов (суммарный трафик), либо NetFlow потока (трафик может быть разбит по категориям). Модуль может производить лимитирование трафика и времени, ограничивать клиентов по телефонам доступа, времени доступа, динамически раздавать IP-адреса из пулов, либо устанавливать их жёстко. Поддерживается принудительный разрыв соединения по достижению нуля на счёту, либо наступлению какого-либо иного ограничения, подсчёт соединений производится в "горячем" режиме.

В общем случае модуль DialUp способен обчислять любое коммутируемое соединение, единственным условием является поддержка оборудованием протокола RADIUS - стандартного протокола аутентификации и аккаунта, поддерживаемого большинством производителей.

Базовые понятия и алгоритм работы модуля DialUp

Структура взаимодействия основных частей модуля DialUp изображена на рисунке.



NAS (Network Access Server) - сервер, через который происходит выход клиента в интернет.

В роли NAS может выступать как специализированное оборудование (Cisco, Huawei), так и обычные компьютеры с модемами, либо с программой поддержки VPN.

Основное назначение NAS - осуществлять доступ пользователей к внешней сети, при этом может происходить туннелирование трафика пользователя (VPN), преобразование протокола канального уровня (PPP -> Ethernet) и другие преобразования.

NAS посылает на RADIUS-сервер два типа запросов: авторизационные (запросы с просьбой установки соединения, содержащие логин и пароль) и запросы аккаунта (уведомительные, содержат информацию о начале - Start - или окончании - Stop - соединения, а также о протекании хода соединения - Update).

[Базовые сведения о протоколе RADIUS и конфигурировании атрибутов.](#)

Протокол RADIUS основан на UDP, представляет из себя пакет определённого типа с набором атрибутов. Рассмотрим ход типового соединения по логу RADIUS запросов (radius.log):

```
Код
INFO 17.01.2008 05:37:49 AUTH:
Type=AUTHENTICATION_REQUEST
Attributes:
  User-Name=1579
  NAS-Identifier=drs1.igs.ufanet.ru
  CHAP-Password=^i.Â.ÆÉ .îé°ª
«Ó^
  NAS-IP-Address=89.189.150.67
  NAS-Port=1
  Service-Type=2
  Framed-Protocol=1
  Calling-Station-Id=10.47.33.2
  NAS-Port-Type=5
  CHAP-Challenge=»^hãã&{{2Êi ñóÔ%âН°vÔÛc^bô..é`-´.o.+.^§

INFO 17.01.2008 05:37:49 RESPONSE:
Type=AUTHENTICATION_ACCEPT
Process time:113
Attributes:
  Service-Type=2
```

Framed-Protocol=1
Framed-IP-Address=89.189.151.3
mpd-limit=out#1=all shape 128000 pass
mpd-limit=in#1=all rate-limit 10000000 pass
mpd-limit=in#1=all rate-limit 10000000 pass

INFO 17.01.2008 05:37:49 ACCOUNT:
Type=ACCOUNTING_REQUEST
Attributes:

User-Name=1579
NAS-Identifier=drs1.igs.ufanet.ru
NAS-IP-Address=89.189.150.67
NAS-Port=1
Service-Type=2
Framed-Protocol=1
Framed-IP-Address=89.189.151.3
Acct-Status-Type=1
Acct-Session-Id=530269-p0001
Acct-Authentic=1
Acct-Link-Count=1
Acct-Multi-Session-Id=530269-p0001
NAS-Port-Type=5
Calling-Station-Id=10.47.33.2

INFO 17.01.2008 05:37:49 RESPONSE:
Type=ACCOUNTING_RESPONSE
Attributes:

INFO 17.01.2008 05:47:49 ACCOUNT:
Type=ACCOUNTING_REQUEST
Attributes:

User-Name=1579
NAS-Identifier=drs1.igs.ufanet.ru
NAS-IP-Address=89.189.150.67
NAS-Port=1
Service-Type=2
Framed-Protocol=1
Acct-Input-Octets=1250
Framed-IP-Address=89.189.151.3
Acct-Output-Octets=1327
Acct-Status-Type=3
Acct-Session-Time=600
Acct-Input-Packets=56
Acct-Session-Id=530269-p0001
Acct-Authentic=1
Acct-Link-Count=1
Acct-Multi-Session-Id=530269-p0001
Acct-Output-Packets=56
Acct-Output-Gigawords=0
Acct-Input-Gigawords=0
NAS-Port-Type=5
Calling-Station-Id=10.47.33.2

INFO 17.01.2008 05:47:49 RESPONSE:
Type=ACCOUNTING_RESPONSE
Attributes:

...
INFO 17.01.2008 18:22:16 ACCOUNT:
Type=ACCOUNTING_REQUEST
Attributes:

User-Name=1579
NAS-Identifier=drs1.igs.ufanet.ru
NAS-IP-Address=89.189.150.67
NAS-Port=1
Service-Type=2
Framed-Protocol=1
Acct-Input-Octets=59455380
Framed-IP-Address=89.189.151.194
Acct-Output-Octets=75469553
Acct-Status-Type=2

```
Acct-Session-Time=39570
Acct-Input-Packets=159904
Acct-Session-Id=536566-p0001
Acct-Authentic=1
Acct-Link-Count=1
Acct-Multi-Session-Id=536566-p0001
Acct-Terminate-Cause=6
Acct-Output-Packets=140465
Acct-Output-Gigawords=0
Acct-Input-Gigawords=0
NAS-Port-Type=5
Calling-Station-Id=10.47.33.2
```

```
INFO 17.01.2008 18:22:16 RESPONSE:
Type=ACCOUNTING_RESPONSE
Attributes:
```

Каждый пакет содержит информацию о NAsе (**NAS-Identifier** и/или **NAS-IP-Address**), на основании которой RADIUS-сервер сопоставляет пришедший пакет NASу в модуле. При сопоставлении сначала производится поиск NASа с названием, идентичным атрибуту **NAS-Identifier** пакета, затем, если результат отрицательный, идёт поиск NASа с IP-адресом, равным **NAS-IP-Address**. Если пришедшему пакету NAS не сопоставлен в **radius.log**, выводится ошибка **NAS not found for Packet!!!**.

Обмен сообщениями с каждым NASом шифруется определённым кодовым словом - секретом. Секрет должен совпадать для NASа в биллинге и для конфигурации самого NASа. При несовпадении секретов проверка пароля будет все время выдавать неверный результат, т.к. секрет используется при шифровании пароля.

Идентификатором соединения в пределах NASа для RADIUS-сервера выступает атрибут **NAS-Port**. Обратите внимание, что он идентичен для всех пакетов в пределах сессии. NAS должен контролировать, чтобы в один момент времени одинаковый **NAS-Port** не проставлялся в RADIUS пакетах, относящихся к разным сессиям.

Далее рассмотрим по пакетно обмен данными между NASом и RADIUS-сервером по ходу соединения.

1. AUTHENTICATION_REQUEST

Запрос авторизации отправляется NASом RADIUS-серверу и содержит помимо идентификационной информации соединения, указанной выше, информацию о логине и пароле пользователя. Логин передаётся в открытом виде, пароль шифруется. Поддерживаются протоколы шифрования PAP, CHAP, MS-CHAP v.2 с генерацией 128 битных MPPE ключей. Протокол авторизации определяется RADIUS-сервером автоматически на основании набора атрибутов.

Протокол PAP является самым ненадёжным, пароль шифруется обратимым способом с помощью секрета. RADIUS-сервер дешифрует его и сравнивает с паролем, указанным для логина в базе данных. Данный режим авторизации можно использовать для проверки секретов. При некорректном секрете пароль в PAP режиме не расшифровывается и отображается в **radius.log** и в мониторе ошибок не в том виде, в котором был введён пользователем.

Протоколы CHAP, MS-CHAP v.2 поддерживают необратимое шифрование, когда RADIUS-сервер сравнивает не открытый пароль, а результаты криптопреобразования открытого пароля, выполненного им самим и NASом.

MPPE ключи, передаваемые в **AUTHENTICATION_ACCEPT** пакете в режиме MS-CHAP v.2 авторизации, могут использоваться NASом для шифрования VPN-туннеля пользователя.

Порядок обработки авторизационного пакета следующий: запрос -> **скрипт предобработки** -> **антиспам** (блокировка) -> биллинг (проверка наличия логина/пароля, баланса и т.д.) -> **штатный Reject-To-Accept** -> обработка события "RADIUS-аутентификация" -> **антиспам** (сбор статистики) -> ответ.

2. AUTHENTICATION_REJECT

Отказ в авторизации, данный пакет не содержит атрибутов. Детальную причину отказа в авторизации пользователя можно посмотреть в **Мониторе модуля** в режиме ошибок.

3. AUTHENTICATION_ACCEPT

Пользователь авторизован. В данном пакете могут содержаться атрибуты, устанавливающие характеристики соединения пользователя (IP адрес, скорость, максимальную длину сессии, частоту UPDATE пактов и т.п.).

4. ACCOUNTING_REQUEST

Запросы аккаунтинга могут быть трёх типов: **Start, Stop, Update**. Различаются они атрибутом **Acct-Status-Type**, который равен 1, 2 или 3 соответственно. Данный тип запросов передаёт на RADIUS-сервер информацию о ходе соединения (соединение началось, завершилось или текущее состояние соединения).

5. ACCOUNTING_RESPONSE

Ответ RADIUS сервера о том, что он получил запрос аккаунтинга. Ответ не содержит никаких атрибутов. Исключение составляет ответ MPD серверу, который может содержать атрибут, информирующий NAS о необходимости разрыва соединения.

SNMP и NetFlow

Кроме одноимённого протокола RADIUS-сервером применяется протокол SNMP - для постоянной проверки активности соединения и его принудительного разрыва. Т.к. RADIUS-сервер обчитывает соединение в реальном режиме времени, нужна постоянная информация о статусе соединения, чтобы предотвратить обсчёт соединения, если клиент вышел, но по какой-либо причине Stop сигнал не пришёл. Кроме того по SNMP происходит посылка сигнала на окончание сессии, если клиент вышел за 0 своего баланса.

На основании пакетов Update или Stop может осуществляться подсчёт трафика, т.к. они могут содержать информацию об отправленных и принятых байтах. Вся конфигурация берётся RADIUS-сервером из базы данных, которая может правиться с помощью клиента.

Начиная с версии 3.5_r4 модуль содержит встроенный NetFlow-коллектор, позволяющий анализировать NetFlow потоки NASa и делить трафик соединения по типам.

Режимы работы RADIUS сервера

RADIUS-сервер для модуля DialUp может работать в двух режимах:

а) Режим активной проверки существования соединения (**CHECKER**). В этом режиме происходит постоянная посылка запросов на NAS с целью выяснения, активно ли соединение. В случае, если соединение активно, через каждые **update.time** (настраивается в конфигурации модуля) секунд происходит выделение мини аванса времени с начислением в баланс и поток обчёта засыпает до следующего обчёта.

Параметр **sleep.time** задаёт интервал в секундах, через который необходимо проверять нужен ли новый аванс времени. В случае, если аванс не может быть выделен, посылается сигнал завершения соединения.

Update-пакеты с информацией о трафике могут быть использованы, но они не обязательны и обчёт трафика также происходит по таймеру. Т.е. сначала начисляются деньги за скачанный трафик, после чего вычисляется время, которое клиент может проработать при условии, что он не будет потреблять трафик.

Преимущества: Точное время сброса клиента, если не идёт обчёт трафика.

Недостатки: Требуется поддержка на NASе запросов активности соединения, дополнительная вычислительная нагрузка.

б) Режим пересчёта по Update-пакетам (**UPDATE**). Пересчёт происходит по получению Update-пакета, если после этого баланс будет отрицателен, идёт сигнал на завершение соединения.

Преимущества: Уменьшенная нагрузка на сервер обчёта, обчёт происходит только по сигналам Update.

Недостатки:

- Необходима поддержка на NASе Update-пакетов;

- Невозможно довести клиента до полного нуля, он всегда его немного пройдёт, насколько - зависит от частоты Update-пакетов.

Предпочтительным режимом работы является Update, режим работы может быть установлен как глобально на BGRadiusDialup в конфигурации модуля, так и персонально для каждого NASa.

Статус соединения

Каждое соединение в ходе своей жизни проходит несколько статусов: **wait**, **sleep**, **active**. После авторизации статус соединения становится **wait** до прихода Start-пакета. Таймаут максимального ожидания старта задаётся переменной **max.wait.timeout** конфигурации модуля. Если в течении указанного времени Start-пакет по Nas-Port не будет получен, соединение удаляется.

После прихода старта статус становится **sleep** до момента подтверждения активности по SNMP (CHECKER режим) или посредством Update-пакета (UPDATE режим). При положительном результате проверки статус переходит в **active**, соединение учитывается при подсчёте активных соединений логина, идёт обсчёт его времени и трафика.

Если в течении некоторого времени не приходят UPDATE-пакеты (для режима UPDATE, таймаут задаётся переменной **max.update.timeout** конфигурации модуля в секундах), либо результат проверки по SNMP показывает, что соединение неактивно - оно вновь переходит в статус **sleep**.

Соединение, слишком долго пребывающее в статусе **sleep**, может быть закрыто [автоматически](#). В ином случае оно будет закрыто только, если придёт авторизация на такой же **NAS-Port**, что и у данного соединения.

Работа с соединением

По ходу работы соединения RADIUS-сервер постоянно производит его тарификацию, информация о [лимите](#) договора и его [статусе](#) перечитывается постоянно, позволяя прервать текущее соединение, либо наоборот, оставить его работающим при временном понижении лимита.

В момент авторизации для соединения устанавливается текущий тариф и "будущие" тарифы, если они есть. В начале каждого суток производится проверка тарифов, соединение автоматически разрывается при обнаружении изменения в установленных в договоре тарифов. Данная функция необходима, т.к. тариф может задавать опции сервиса (например, скорость), которые могут быть заданы только в момент авторизации. Если опции сервиса меняются по ходу соединения, используются [зоны](#) в тарифном плане.

Все соединения RADIUS-сервера должны быть сброшены после наступления нового месяца. Для этого можно устанавливать опцию в конфигурации NASa **month.break=1**, либо производить корректное завершение сессий на NASax иными средствами. Требование разрыва на границе месяца связано с особенностью тарификации биллинга.

Порядок тарификации

Понимание порядка тарификации очень важно для построения тарифных планов и понимания механизма их работы.

Рассмотрим в каком порядке происходит тарификация трафика и времени в разных режимах обчёта. В каждом соединении ведутся счетчики байт, посчитанных и потреблённых. При этом NetFlow-коллектор по мере получения потоков увеличивает счётчики потреблённых байт в памяти, также они могут изменяться по приходу UPDATE, либо STOP-пакетов (зависит от настройки трафиков в NASE). Ещё одна характеристика соединения - время его **окончания** - это то время, до которого было обчислено соединение.

1) При режиме тарификации **UPDATE**, после получения UPDATE-пакета:

- Увеличиваются счётчики потреблённых байт, если настроено извлечение трафика из RADIUS-пакетов.
- Тарифицируется потреблённый трафик как разница между счётчиками потреблённых и посчитанных байт. Весь трафик считается потреблённым единообразно в текущее время окончания соединения (время предыдущего UPDATE-пакета).
- Тарифицируется потреблённое время. Считается, что в момент текущего времени окончания, было потреблено количество секунд до текущего времени. Если при этом был переход часа, то тарификация времени осуществляется в два этапа. Считается, что в момент текущего времени окончания было потреблено количество секунд до границы часа, а на границе часа было потреблено оставшееся количество секунд.
- Изменяется время окончания соединения на текущее время. При необходимости (исчерпан баланс, смена зоны в тарифе) может быть послан сигнал сброса соединения, либо CoA пакет.

Обратите внимание, что каждая услуга считается потреблённой в какой-то определённый момент времени. Это время передаётся в тарифном запросе и используется узлами тарифных планов **Фильтр по типу времени, Период** и т.п.

2) При режиме тарификации **CHECKER** (устаревший режим). Тарификация осуществляется по таймеру, периодически просыпающимся потоком. При каждом обчёте:

- Тарифицируется потреблённый трафик, как разница между счётчиками потреблённых и посчитанных байт. Весь трафик считается потреблённым единообразно в текущее время окончания соединения (время предыдущего обчёта).
- От текущего времени окончания соединения отсчитывается аванс времени в **update.time** секунд. Обсчёт идет в будущее, при переходе часов осуществляется разбивка секунд аналогично режиму UPDATE.
- Изменяется время окончания соединения (время может стать в будущем). При необходимости (исчерпан баланс, смена зоны в тарифе) может быть послан сигнал сброса соединения, либо CoA пакет.
- Поток засыпает, просыпаясь каждые **sleep.time** секунд для проверки прохождения времени окончания соединения и, следовательно, необходимости нового обчёта.
- При получении STOP-пакета возможен обсчёт времени в обратном направлении, т.к. соединение может завершиться раньше уже просчитанного времени окончания.

RADIUS атрибуты

Посредством RADIUS-атрибутов производится весь обмен информацией между NASом и RADIUS-сервером. Набор поддерживаемых RADIUS атрибутов меняется для каждого NASa, ниже рассмотрены общие для всех атрибуты.

Таблица. Таблица атрибутов

Атрибут	Типы пакетов	Назначение
Framed-IP-Address	AUTHENTICATION_ACCEPT, ACCOUNTING_REQUEST	IP-адрес, выдаваемый клиенту RADIUS-сервером, либо NASом. Для выдачи сервером адрес может быть указан в свойствах логина на вкладке IP-адрес , либо Атрибуты RADIUS . Также адрес может быть выдан из пула адресов. Если адрес не выдаётся сервером в ответе авторизации - он может быть выдан NASом и передан в Start-пакете. Адрес используется RADIUS-сервером для регистрации его на встроенном NetFlow-коллекторе и привязке потоков к сессии клиента.
Framed-Pool	AUTHENTICATION_ACCEPT	Именованный пул адресов, из которого NAS должен выдать адрес клиенту. RADIUS-сервер может обрабатывать этот атрибут самостоятельно, выдавая при наличии данного атрибута в ответе авторизации свободный IP-адрес из указанного пула посредством Framed-Ip-Address атрибута.
Framed-Route	AUTHENTICATION_ACCEPT	Маршрутизируемая через тоннель сеть, например: 10.80.0.0 255.255.255.0 10.60.0.1 100 (IP-адрес сети) (маска) (куда маршрутизировать) (вес маршрута) Либо можно вместо маски указать длину префикса: 10.80.0.0/24 10.60.0.1 100 - то же самое, что и в предыдущем примере. Данный атрибут поддерживается не всеми VPN-шлюзами. RADIUS-сервер анализирует передачу информации о сетях в ответе авторизации и использует её при разнесении NetFlow потока на соединения.
Acct-Interim-Interval	AUTHENTICATION_ACCEPT	Период в секундах между Update-пакетами, не может быть меньше, чем 60 секунд.
Calling-Station-Id	ACCOUNTING_REQUEST, AUTHENTICATION_REQUEST	Номер звонящего для DialUP-соединения, MAC-адрес клиента для PPPoE-соединения, IP-адрес клиента для PPTP-соединения.
Called-Station-Id	ACCOUNTING_REQUEST, AUTHENTICATION_REQUEST	Вызываемый номер для DialUP-соединения, MAC-адрес сервера для PPPoE-соединения, IP-адрес сервера для PPTP-соединения.
Acct-Session-Time	ACCOUNTING_REQUEST	Длительность соединения в секундах; информация используется при завершении соединения для точного обчёта времени и корректировки длительности соединения в БД. При установке в конфигурации модуля опции ignore.acct.session.time=1 RADIUS-сервер игнорирует этот атрибут, сам вычисляя длительность, как разницу во времени между пакетами Start и Stop .
Acct-Input-Octets	ACCOUNTING_REQUEST	Исходящий для пользователя трафик в байтах.
Acct-Input-Gigawords	ACCOUNTING_REQUEST	Исходящий для пользователя трафик в 2 ³² байтах (gigaword), атрибут используется как старший разряд Acct-Input-Octets в случае, если количество отправленных байт более 2 ³² .
Acct-Output-Octets	ACCOUNTING_REQUEST	Входящий для пользователя трафик в байтах.
Acct-Output-Gigawords	ACCOUNTING_REQUEST	Входящий для пользователя трафик в 2 ³² байтах (gigaword), атрибут используется, как старший разряд Acct-Output-Octets в случае, если количество принятых байт более 2 ³² .
User-Name	ACCOUNTING_REQUEST, AUTHENTICATION_REQUEST	Введённый пользователем логин.

Описание атрибутов, специфичных для NASa, необходимо смотреть в документации к NASy.

Настройка модуля DialUp

Установите модуль на сервер, создайте экземпляр. Определите в **Редакторе модулей и услуг** услуги, обчисляемые этим модулем. Например: **Входящий трафик**, **Исходящий трафик**, **Время**. Услуга, соответствующая времени соединений, обязательна, хотя может и считаться по нулевой цене. Названия и количество услуг графиков могут быть расширены в дальнейшем, исходя из предоставляемых модулем сервисов.

В конфигурации модуля установите:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

#
password.length.min=5
password.length.max=10
#
password.length.auto=6
#
password.chars=1234567890

# Web -
web.menuItem1= DialUp
web.menuItem2= DialUp
web.menuItem3= DialUp
web.menuItem4=
web.menuItem5=
# web
showyears=5
# Web- - -
show.sessions.on.page=25
#XSL
xslt.1=dialup_login_sessions.xsl
xslt.1.csv=dialup_login_sessions_csv.xsl
reportTitle.1= DialUp
#XSL
xslt.2=dialup_login_amount.xsl
xslt.2.csv=dialup_login_amount_csv.xsl
reportTitle.2= DialUp

# -
vendors=9=Cisco;2011=Huawei;2021=Unix PPP;529=Lucent;6618=Quintum;529=Ascend;311=Microsoft;12341=MPD;
14988=Mikrotik

# , , ,
#service.recalc.ignore=

# RADIUS,
radius.attributes=Service-Type;Framed-Protocol;Framed-IP-Address;Framed-IP-Netmask;Framed-Routing;Filter-Id;
Framed-MTU;Framed-Compression;Login-IP-Host;Login-Service;Login-TCP-Port;Old-Password;Reply-Message;Callback-
Number;Callback-Id;Expiration;Framed-Route;Framed-IPX-Network;State;Class;Session-Timeout;Idle-Timeout;
Termination-Action;NAS-Identifier;Proxy-State;Framed-Pool

#-----
#
#
#closed.date.disabled.ActionWrapLogin=1
#
#closed.date.disabled.ActionMaxRecalculate=1
#
#closed.date.disabled.ActionPeriodDelete=1
#
#closed.date.disabled.ActionPeriodUpdate=1
#
#closed.date.disabled.ActionRecalculateSessions=1
#
#closed.date.disabled.ActionUpdateLoginInfo=1
```

```

#-----
#
# {dd1}{MM1}{yyyy1} - , , , {dd2}.{MM2}.{yyyy2}-, , , {dd2}.{MM2}.{yyyy2} , , .
#file.detail.template={cid}_{dd1}.{MM1}.{yyyy1}_{dd2}.{MM2}.{yyyy2}_{dd_now}.{MM_now}.{yyyy_now}

##### RADIUS- #####
#
top.nocard.login=10000
# " ", 0 - " "
card.module.id=0
# ,
#card.login.session.count=1

#1 - , " "
check.service=0

#
#1 - UPDATE - UPDATE-
#2 - CHECKER - , UPDATE-
dialup.workmode=1
# UPDATE - UPDATE-,
#( )
max.update.timeout=120
# wait Start-
max.wait.timeout=120
# , CHECKER
update.time=60
# , CHECKER
run.sleep=3
# Acct-Session-Time- NAS,
#ignore.acct.session.time=1

# , NetFlow
# " ", .. ,
#delay.stop=5

# Calling-Station-Id, " "
#( , NAS )
#check.duplicate.session=1
# , .
# CoA .
#no.session.break.on.tariff.change=1

#
#pools.global=

# 1 - Service-Type=2;Framed-Protocol=1 ( , . realm.default)
add.service.type.and.framed.protocol=0
# 1 - Auth Accept MPPE-128- MS_mppe_encryption_types ( 128 ) MS_mppe_encryption_policy=1 ( )
# http://rfclibrary.hosting.com/rfc/rfc2548/rfc2548-25.asp
add.mppe.enc.types.and.policy=1
# Accept- Framed-Pool , RADIUS-
#drop.framed.pool.attr=1
# User-Name \ ( ), Windows-, , , 0 -
remove.user.name.before.backslash=1

# REALM default
realmgr.default=default
#, AUTH_ACCEPT- default
# Update-, , PPP, , : RFC2865
realm.default=Acct-Interim-Interval=60;Service-Type=2;Framed-Protocol=1
# User-Name
trim.user.name=0
#
report.round.before.sum=1

```

Параметры **xslt.1** и **xslt.2** указывают название XSLT-шаблонов преобразования для печати сессий клиента, либо отправки их на E-mail.
 Параметры **reportTitle.1** и **reportTitle.2** задают название отчёта, отображаемое сверху и тему письма, содержащего этот отчёт (1 - по сессиям, 2 - по наработке). Это сделано для возможности редактирования оформления отчётов.

Пулы адресов определяются следующим образом:

```
pools.pool_1=10.0.0.1-10.0.0.3;34.4.4.1
```

После ключевого слова **pools**, указывается имя пула (pool_1). После символа равенства - один или несколько одиночных адресов, либо диапазонов с разделителем точка с запятой. Для каждого NASa пул адресов может быть переопределён, как это сделать читайте в разделе о настройке NASов.

С версии 4.5 появилась возможность добавлять параметры для логинов DialUp. Параметры применяются для действий, связанных с логинами (см "Динамическое управление ДНС сервером"). Параметры прописываются в конфигурации модуля в следующем виде:

```
#  
login.parameter.1.name=dialup.dns # ,  
login.parameter.1.title= #,  
login.parameter.1.type=5 # ( 1 - , 2 - , 3 - , 4 - , 5 - )  
login.parameter.1.default=0 # ( 5 , 0)  
login.parameter.1.listValue=; ; # 5 -
```

REALMы

REALMы - это зоны, в которых может работать ваш клиент, используя один и тот же логин. Фактически, REALM - это суффикс, добавляемый пользователем после своего логина, например login@local - пользователь хочет попасть во внутреннюю сеть, login@www - во внешнюю.

RADIUS-сервер в зависимости от REALMа, указанного клиентом, выдаёт ему определённый набор RADIUS-атрибутов, например, Framed-IP-Address - IP-адрес, который нужно выдать клиенту. Все REALMы должны быть определены в конфигурации модуля DialUp следующим образом:

```
realm.local=Framed-IP-Address=192.168.168.2;< >
realm.www=Framed-IP-Address=172.33.3.3;Service-Type=1
```

REALMы объединяются в группы следующим образом:

```
realmgr.dialup_user=local;www
realmgr.vpn_user=local
```

Теперь логины, имеющие группу REALMов **dialup_user** (группа REALMов устанавливается в свойствах логина), могут попасть как во внешнюю сеть, так и во внутреннюю, а логины с группой **vpn_user** - только во внутреннюю.

Если пользователь после своего логина не указал REALM, считается, что он хочет попасть в зону default. Также, если в свойствах логина не указана группа REALMов, значит он входит в группу **default**.

Таким образом, даже если вы не используете REALMы, у вас в конфигурации DialUp должна присутствовать строка.

```
realmgr.default=default
```

Это позволит логинам с не указанной группой REALMов использовать REALM default, т.е. входить без указания суффикса после логина. Если же этой строки не будет, то пользователь не сможет авторизоваться, т.к. для группы default не разрешено использование никаких REALMов.

REALMы удобно использовать для разделения услуг доступа к различным сетям, для задания перечня стандартных атрибутов RADIUS, которые нужно высылать всем пользователям, например:

```
realm.default=Framed-Pool=pool1;Acct-Interim-Interval=120
```

Атрибуты, передаваемые при логине с определённым REALMом, можно переопределить в конфигурации NASa.

Наборы атрибутов

Наборы атрибутов позволяют просто назначать логину, либо тарифному плану определённые RADIUS-атрибуты, которые будут переданы в AUTH ACCEPT-пакете. Наборы определяются в конфигурации модуля следующим образом.

```
attrset.<id>.title=< >  
attrset.<id>.attributes=< ;>
```

например (значения атрибутов приведены случайно, для примера)

```
attrset.1.title= 1  
attrset.1.attributes=Cisco-AVPair=multilink:min-links=1  
attrset.2.title= 2  
attrset.2.attributes=cisco-Fax-MDN-Address=1
```

Наборы атрибутов можно использовать для:

1. быстрого назначения определённого набора опций логину, не перечисляя каждый раз все атрибуты (например, ACL-список, либо скорость соединения) (см. далее настройку клиентов);
2. задания опций работы для тарифного плана (см. [далее](#) настройку тарифов).

Набор атрибутов может быть переопределён в конфигурации NASa.

Выдача атрибутов соединения и выделение IP адресов

При успешной авторизации в пакете **AUTHENTICATION_ACCEPT** выдаются атрибуты RADIUS, атрибуты выбираются в следующем порядке:

1. Атрибуты в ответ авторизации может установить [скрипт предобработки запроса](#), данные атрибуты добавляются как при успешной, так и при неуспешной авторизации;
2. Выбираются [наборы атрибутов](#) сначала логина (в свойствах логина на вкладке **Атрибуты RADIUS - Наборы атрибутов**), а затем из тарифного плана. Каждый набор может быть отправлен только один раз;
3. Выбираются атрибуты, установленные в свойствах логина на вкладке **Атрибуты RADIUS - Атрибуты**;
4. Если на вкладке **Атрибуты RADIUS** установлена опция присваивания атрибутов **Глобальные+локальные**, то выдаются атрибуты для текущего REALMa соединения, указанные в конфигурации модуля и NASA;
5. Атрибуты ответа могут быть изменены скриптом поведения договора при обработке события **"RADIUS-аутентификация"**.

IP адрес соединению присваивается по следующему алгоритму. Если на каком-либо шаге адрес установлен - последующие пропускаются.

1. Просматриваются установленные в **AUTHENTICATION_ACCEPT** атрибуты и если среди них присутствует **Framed-IP-Address**, считается что у соединения установлен указанный адрес.
2. Просматривается список адресов указанных в свойствах логина на вкладке **IP адрес**. Если адресов несколько - выдаётся любой свободный из них, устанавливается атрибут **Framed-IP-Address**.
3. Просматриваются установленные в **AUTHENTICATION_ACCEPT** атрибуты и если среди них присутствует **Framed-Pool** атрибут - делается попытка поиска свободного адреса в указанном пуле, описание самого пула последовательно ищется в конфигурации NASA и модуля.
4. Делается попытка выдачи адреса из глобального пула, определённого в конфигурации модуля как **pools.global**.
5. Если адрес не был выдан RADIUS сервером, он берётся из **Start** пакета для регистрации на встроенном коллекторе.

При успешной выдаче адреса и установленной опции конфигурации модуля **drop.framed.pool.attr=1** атрибуты Framed-Pool удаляются из пакета.

При исчерпании пула адресов система шлет специализированный аларм. Дополнительно в конфигурации модуля можно определить пороговый процент для любого из пулов. При проценте использовании пула больше или равном указанному будет высылаться предварительный аларм. Данная мера позволяет заблаговременно увеличивать размер пулов. Процент указывается следующим образом:

```
pool.alarm.fullness.< >=< >
```

Например (высылка аларма по глобальному пулу после его расходования более чем на 20 целых и 4 десятых процента):

```
pool.alarm.fullness.global=20.4
```

Настройка NASов

Основная информация, которую должна содержать конфигурация NASa - это его идентификатор, IP-адрес и секрет (необходим для шифрования пароля пользователя, должен совпадать на RADIUS-клиенте и сервере).

Кроме того, к нему должны быть привязаны услуги, определены параметры инспектора (осуществляет управление NASom с целью проверки активности соединения, либо его завершения).

gate	192.168.184.254	-01.01.2008	gate
huaw1	81.30.210.254	-01.09.2003	900111(Huawei Expert)
oracle	192.168.184.94	25.01.2006-01.01.2008	
pool-1	213.24.120.34	-01.08.2003	Unix
pool3	81.30.199.58	-01.09.2003	900111(Cisco53x)
pool7	213.59.17.253	-01.12.2005	797588(Cisco36x)
pool9	213.24.120.254	-01.09.2003	797588(Cisco53x)
stat	213.24.120.68	-01.01.2008	Unix
ufps	81.30.199.235	-	Unix (основной для тестов)
vg1	81.30.199.34	-01.09.2003	900111(Cisco53xx)
wfi	192.168.1.1	-01.01.2008	server.office.bitel.ru

Редактор NAS

Идентификатор: pool.bitel.ru

Адрес (X.X.XX): 1.1.1.1

Период: с 25.01.2008 по

Комментарий

Восстановить

Идентификатор: pool.bitel.ru

Вендор: Cisco

Секрет:

Конфигурация

Скрипт предобработки

Дата	Название	Пользователь
------	----------	--------------

Создать Открыть Удалить Сохранить Отменить Установить

Ок Отмена

Модуль Dial-Up

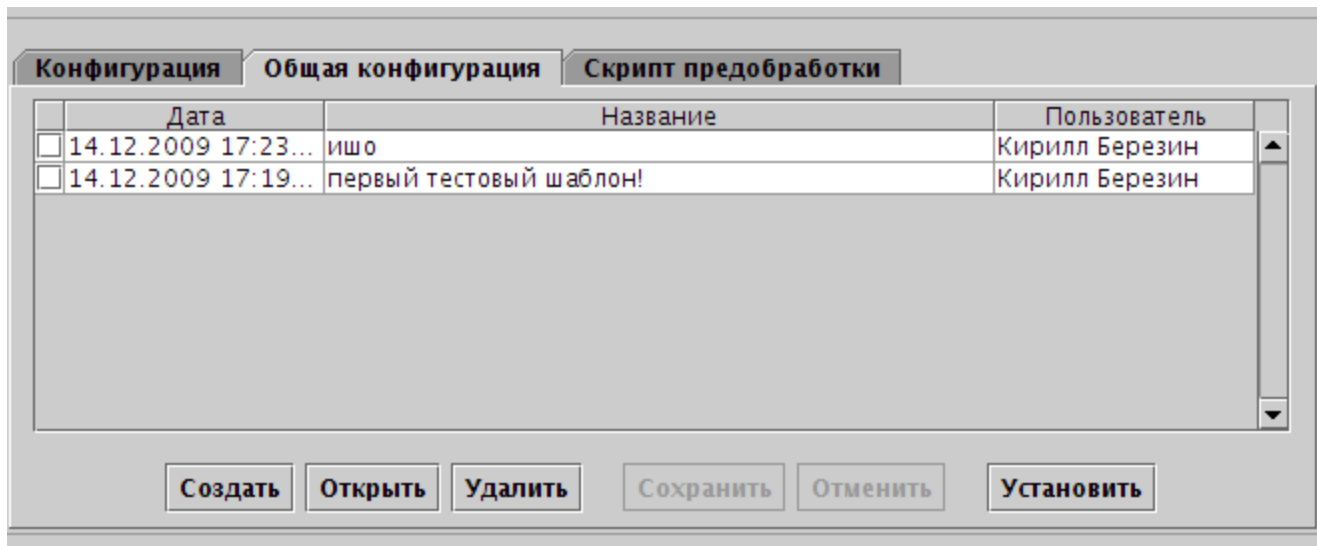
Настройка сервера доступа происходит в два шага.

Сначала кнопкой **Новый элемент** создаётся новый NAS, ему прописывается идентификатор (что будет приходить от него в атрибуте **NAS-Identifier**), IP-адрес (что будет приходить от него в атрибуте **NAS-IP-Address**), RADIUS-секрет, вендор и комментарий. Список вендоров задаётся в конфигурации модуля. После нажатия кнопки **Ок** должна появиться новая строка в таблице.

Обязательно устанавливайте правильного вендора для NASa.

Двойным кликом мыши откройте её для редактирования и кнопкой **Создать** добавьте текстовую конфигурацию. Название - произвольное.

Возможна также настройка общих конфигураций для NASов. Создание общих конфигураций происходит аналогично описанному далее, за исключением того, что они могут использоваться в нескольких NASax. При этом локальные настройки имеют больший приоритет (т.е. если оба флага присутствуют и во включенной общей, и в выбранной локальной конфигурациях, то будет использован локальный).



Содержимое конфигурации может меняться в зависимости от типа NASa, но обязательно содержит следующие данные:

```
# , * - ( , Calling-Station-Id)
#nas.port_phone.*=900111
#
nas.port_time.default.*=XXX
nas.port_traffic.default.*=XXX
# NAS
card.activate.service=0
# Callback (1-)
callback.support=0
#
month.break=1
# , NAS
month.break.period=3600
#
nas.inspector.sleep_time=60
#
nas.inspector.kill.max_messages=5
```

Режим работы **UPDATE**, либо **CHECKER** может быть задан независимо для каждого NASa опцией **dialup.workmode** аналогично конфигурации модуля DialUP. Если режим не установлен - используется значение режима из конфигурации модуля.

Наиболее важной настройкой каждого из NASов является конфигурация услуг. Она определяет, с какими услугами модуля будут сопоставлены трафики соединения и его время. Конфигурация услуг должна быть определена для каждого сочетания REALM+NAS-Port, доступного на данном NASe, в противном случае при авторизации будет выведена ошибка **Not found service id**.

Разделение по REALМу позволяет выделять отдельные услуги на разные сервисы, потребляемые пользователем. Например, вход просто с логином являет из себя услугу **VPN-время**, для которой определена цена 0 в тарифном плане и для данного типа входа обсчитывается трафик. А вход под REALМом **time** (login@time) определяет услугу время как **VPN-повременный**, для которой в том же тарифе определена ненулевая цена, а трафик при данном входе не учитывается (пустая конфигурация трафиков).

Разделение по порту предоставляет возможность разделения услуг для многоканальных и обычных телефонов доступа в пределах одного NASa.

Строка привязки услуги типа "время" к REALМу и порту выглядит следующим образом:

```
nas.port_time.<realm>.<port>=<sid>
```

где:

- **<realm>** - REALM под которым авторизуется пользователь (если после логина ничего не указано - REALM считается **default**);
- **<port>** - **NAS-Port** на который авторизуется пользователь, возможность привязки отдельной услуги типа время на выделенные порты полезна, например, для выделения многоканальных телефонов, * - любой порт;
- **<sid>** - числовой код услуги типа "время".

Привязка услуг типа "трафик" к REALМу и порту выглядит следующим образом:

```
nas.port_traffic.<realm>.<port>=<sid>
```

```
nas.port_traffic.<realm>.<port>=<service1>;<service2>;...<serviceN>
```

где значения **<realm>** и **<port>** идентичны таким же параметрам в указании услуги типа "время", а записи **<serviceX>** представляют из себя строку следующего вида:

```
<sid>:<keyword>
```

где **<sid>** определяет код услуги типа "трафик", а **<keyword>** может принимать следующие значения:

- **RADIN** - объем услуги в байтах, получается на основании RADIUS-пакета (атрибуты **Acct-Output-Octets**, **Acct-Output-Gigawords**);
- **RADOUT** - объем услуги в байтах, получается на основании RADIUS-пакета (атрибуты **Acct-Input-Octets**, **Acct-Input-Gigawords**);
- **RAD**(<vcode>,<atrcode>,<prefix>) - объем услуги в байтах, получается на основании RADIUS-пакета, строкового атрибута с кодом <atrvcode> вендора с кодом <vcode>, трафик указывается после префикса <prefix> и одного символьного разделителя. Например: 7:RAD(12341,10,local) - получение трафика по седьмой услуге из атрибута MPD mpd-output-octets после префикса "local:";
- **COLLECTOR** - объем услуги в байтах, получается по информации NetFlow-коллектора (коллектор определяет тип услуги на основании привязок услуг из конфигурации модуля, см. настройку встроенного коллектора);
- **MAX**(<sid2>,<sid3>) - услуга <sid> вычисляется как максимум из услуг <sid2>, <sid3> на каждый из моментов обчёта;
- **SUM**(<sid2>,<sid3>,<sid4>,<sid5>) - услуга <sid> вычисляется как сумма услуг <sid2>..<sid5> на каждый из моментов обчёта, количество параметров функции может быть от двух до четырёх

При получении трафика посредством NetFlow, либо sFlow-протокола необходима [соответствующая настройка встроенного коллектора и конфигурации NASa](#).

В случае, если трафик не нужно тарифицировать и учитывать, вы можете оставить в конфигурации только:

```
nas.port_traffic.<realm>.*=
```

Услуга времени должна быть определена всегда.

Рассмотрим несколько примеров записи конфигураций услуг для NASa:

Ставится задача обчёта суммарного входящего трафика клиента. При этом в вашем модуле DialUP заведены 3 услуги - Время (1), Входящий трафик (2), Исходящий трафик (3). А в тарифном плане клиента указаны цены за эти услуги (включая нулевые за время и исходящий трафик). В этом случае конфигурация времени и трафиков будет выглядеть следующим образом:

```
nas.port_time.default.*=1
nas.port_traffic.default.*=2:RADIN;3:RADOUT
```

Ставится задача обчёта времени работы и максимального из трафиков клиента. При этом в модуле DialUP заведены 4 услуги - Время (1), Входящий трафик (2), Исходящий трафик (3), Максимальный трафик (4). В тарифном плане клиента должны быть указаны 4 цены (2 из них нулевые - за входящий и исходящий трафик). Конфигурация будет выглядеть следующим образом:

```
nas.port_time.default.*=1
nas.port_traffic.default.*=2:RADIN;3:RADOUT;4:MAX(2,3)
```

Начиная с 4.2 версии в конфигурации трафика возможно использование функции **SUM** - суммарный трафик. Её использование аналогично **MAX**, но параметрами могут выступать до 4х кодов услуг.

Ставится задача обчёта времени работы и суммарного из трафиков клиента. При этом в модуле DialUP заведены 4 услуги - Время (1), Входящий трафик (2), Исходящий трафик (3), Максимальный трафик (4). В тарифном плане клиента должны быть указаны 4 цены (две из них с нулевой ценой - за входящий и исходящий трафик). Конфигурация будет выглядеть следующим образом:

```
nas.port_time.default.*=1
nas.port_traffic.default.*=2:RADIN;3:RADOUT;4:SUM(2,3)
```

Ставится задача обчислять по разным ценам локальный и глобальный входящий трафик клиентов. В модуле DialUP заводится 3 услуги: Время (1), Локальный входящий (2), Глобальный входящий (3). В тарифном плане определены все 3 стоимости.

```
nas.port_time.default.*=1
nas.port_traffic.default.*=2:COLLECTOR;3:COLLECTOR
```

Ключевое слово COLLECTOR в последнем случае обозначает, что трафик данного типа будет предоставляться со встроенного коллектора. О установке и настройке встроенного NetFlow-коллектора для BGRadiusDialup читайте [далее](#).

Ставится задача предоставления доступа по трафику и двух видов повременного доступа на скоростях 64 кбит/с и 256 кбит/с с различной оплатой часа. Для повременного соединения пользователь вводит имя пользователя login@64k, login@256k. Для данных REALMов в конфигурации модуля должны быть определены атрибуты RADIUS, устанавливающие параметры скорости.

В модуле DialUP заводятся 5 услуг: Время обычное (1), Входящий трафик (2), Исходящий трафик (3), Время 64к (4), Время 256к (5). В тарифном плане определены цены: 1 - нулевая стоимость, 2 и 3 - стоимость трафиков обычного соединения, 4 - стоимость соединения на скорости 64кбит/с за единицу времени, 5 - стоимость соединения на скорости 256кбит/с за единицу времени.

```
nas.port_time.default.*=1
nas.port_traffic.default.*=2:RADIN;3:RADOUT
nas.port_time.64k.*=4
nas.port_traffic.64k.*=
nas.port_time.256k.*=5
nas.port_traffic.256k.*=
```

Обратите внимание, что для повременного доступа конфигурации услуг трафиков пустые, т.е. трафики не учитываются вовсе. Ставится задача тарификации DialUP-доступа по времени, при этом определённые порты NASa (1 и 5) являются многоканальными, стоимость часа на них выше, чем на остальных. В модуле DialUP заводятся 3 услуги: Время простое (1), Время многоканальный (2). В тарифном плане клиента определяются стоимости часа по каждой из услуг.

```
nas.port_time.default.1=2
nas.port_time.default.5=2
nas.port_time.default.*=1
nas.port_traffic.default.*=
```

Чтобы прописать пул IP-адресов, добавьте в конфигурацию NASa строку (пример пула из 3х адресов):

```
nas.pools.myPool=192.168.169.3-192.168.169.4;192.168.169.33
```

После ключевого слова **nas.pools**, указывается имя пула (myPool). После знака равенства определяются один или несколько диапазонов, либо единичных адресов, разделённых точкой с запятой. Пулы, определённые в конфигурации NASов, более приоритетные, чем определённые в конфигурации модуля. Т.е. если пользователю атрибутами сопоставлен некий пул, и пул с таким именем определён как в конфигурации NASa, так и модуля, будет выбран пул из NASa.

Обратите внимание на параметр **card.activate.service** - это ограничение по типам Интернет-карточек, которые можно активизировать на данном NASe.

0 означает, что на нем можно активизировать любую карточку, в противном случае через точку с запятой необходимо перечислить коды услуг активации карт, разрешённых к активации на данном NASe.

Опция **month.break** означает, что RADIUS-сервер будет принудительно разрывать соединения на границе месяца для данного NASa. Сделано это для того, чтобы в ходе работы в следующем месяце у клиента не изменялся баланс за предыдущий, т.к. изменяется баланс того месяца, где началось соединение.

Остальная часть конфигурации NASa различается для серверов различного типа и отвечает за настройку инспектора - управляющим ходом соединений подсистемы. Задача инспектора - проверка активности соединения (для режима CHECKER) и сброс соединения по событию биллинга.

Все инспекторы, кроме универсального PoD, работают по протоколу SNMP. SNMP-порт и **community** устанавливаются следующим образом (дополнительно можно скорректировать размеры входящего/исходящего буферов сокета):

```
#SNMP ( PoD )
nas.inspector.snmp.port=161
nas.inspector.snmp.community=XXXXX
#
nas.inspector.snmp.buffer.in=4
#
nas.inspector.snmp.buffer.out=4
```

Далее, в зависимости от типа NASa, устанавливаются опции.

1. Huawei Expert (vendor=2011).

```
snmp.version=1
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.2011.2.3.4.3.5.2.1.3.0
nas.inspector.snmp.check.oid=1.3.6.1.4.1.2011.2.3.4.3.5.2.1.5.0
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorHuaweiExpert
```

2. Unix PPP (vendor=2021).

```
snmp.version=2
# 2.4.2 2.4.3, 2.4.4 2.4.3
pppd.version=2.4.2
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorPPPD
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.2021.255.1
nas.inspector.snmp.check.oid=1.3.6.1.4.1.2021.255
```

3. FreeBSD MPD 4.x, 5.x (vendor=12341).

```
dialup.workmode=1
snmp.version=2
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorMPD
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.2021.255.1
nas.inspector.snmp.check.oid=1.3.6.1.4.1.2021.255
```

4. Cisco 53x (vendor=9), либо другие модели Cisco см. [здесь](#).

```
snmp.version=2
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorCisco
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.9.9.150.1.1.3.1.5
nas.inspector.snmp.check.oid=1.3.6.1.4.1.9.9.150.1.1.3.1.5
# , Acct-Session-Id , ""
# Acct-Session-Id 4 SNMP- , 4
#session.mode=hex4
```

5. Cisco 36x (vendor=9).

```
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorCisco36x
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.9.2.9.10.0
nas.inspector.snmp.check.oid=1.3.6.1.4.1.9.2.9.2.1.18
```

6. Lucent Ascend (vendor=529).

```
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorCisco
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.529.12.3.1.3
nas.inspector.snmp.check.oid=1.3.6.1.4.1.529.12.3.1.4
#, (Acct-Session-Id)
#session.mode=dec
```

7. US Robotics NetServer (vendor=429).

```
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.
SNMPNasConnectionInspectorUSRoboticsNetServer
nas.inspector.snmp.check.oid=1.3.6.1.4.1.429.4.10.1.1.18
nas.inspector.snmp.kill.oid=1.3.6.1.4.1.429.4.10.13
```

8. MS RRAS для W2K (vendor=311).

```
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.SNMPNasConnectionInspectorMsRRAS
```

9. Универсальный инспектор, вызывающий для проверки и разрыва соединения внешний скрипт с набором параметров командной строки.

```
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.ScriptNasConnectionInspector
#      ,      1 - ,      -
#      ${NAS_IP} ${NAS_ID} ${USER_LOGIN} ${USER_SESSION} ${USER_PORT}
# , /usr/local/check.pl ${NAS_IP} ${USER_PORT}
nas.inspector.check.command=
#
#      ${NAS_IP} ${NAS_ID} ${USER_LOGIN} ${USER_SESSION} ${USER_PORT}
# , /usr/local/kill.pl ${NAS_IP} ${USER_PORT}
nas.inspector.kill.command=
```

10. WiFi-портал модуля Dialup.

```
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.WiFiConnectionInspector
nas.inspector.wifi.host=<IP , wifi->
nas.inspector.wifi.port=<, wifi- >
```

11. Универсальный инспектор для всех NASов с поддержкой PoD. Использование возможно только в режиме обчёта UPDATE. Дополнительно данный инспектор обладает возможностью отправки CoA-запросов согласно настройкам в тарифном плане для изменения параметров сессии без разрыва соединения.

Код

```
dialup.workmode=1
#
nas.inspector.class=ru.bitel.bgbilling.kernel.network.radius.inspectors.PodNasConnectionInspector
#      PoD/CoA
nas.inspector.radius.port=1812
#      ( - IP- NAS)
#nas.inspector.radius.host=
#      ( - NAS)
#nas.inspector.radius.secret=
#      PoD/CoA- ,      -
nas.inspector.radius.attributes=User-Name;Framed-IP-Address;Acct-Session-Id
# ,      CoA-
#      oA
nas.inspector.coa.timeout=5
#      CoA
nas.inspector.coa.retries=2
#      CoA-
nas.inspector.coa.threads=4
#      CoA- ,
#nas.inspector.coa.send.all.attributes=1
#      RADIUS- , CoA-
#coa.log=1
#      CoA-
#nas.inspector.coa.fixed.attributes=
#      PoD-
#nas.inspector.pod.fixed.attributes=
#      PoD- CoA
#nas.inspector.send.pod.instead.coa=1
```

Подробные инструкции по интеграции BGRadiusDialup с различными NASами доступны в [Wiki](#).

Для использования дополнительной возможности некоторых NASов самостоятельно ограничивать трафик, либо время соединения RADIUS способен высылать атрибуты с максимальными значениями трафика, либо времени с учётом текущего баланса пользователя.

Чтобы RADIUS отправлял атрибут с ограничением, он должен быть прописан в текущем REALMe пользователя, либо в дополнительных атрибутах логина.

Далее необходимо указать, что данный атрибут является ограничительным по какой-либо услуге, это производится в конфигурации NASA подобным образом:

```
service.limit.attribute=< >:< >
```


В модуле DialUP есть услуга "Входящий трафик" с кодом 23 и мы хотим, используя простейшие линейные тарифы, устанавливать ограничение на трафик соединения. Для этого в конфигурации модуля DialUP указывается.

```
realm.default=...;WISPr-Bandwidth-Max-Down=64000;ChilliSpot-Max-Output-Octets=limit
```

А в конфигурации NASa:

```
service.limit.attribute=ChilliSpot-Max-Output-Octets:23
```

Теперь данный атрибут с ограничением трафика будет высылаться всем подключившимся пользователям. Расчёт атрибута производится исходя из тарифного плана клиента.

Для того, чтобы отправлять дубликаты аккаунтинг запросов на другой хост, необходимо добавить в конфиг NASa такие параметры:

```
# , RADIUS-
radius.forward.host=
radius.forward.port=
# , , NASa
#radius.forward.secret=
# , User-Name
#radius.forward.user.name.prefix=
# , User-Name
#radius.forward.user.name.suffix=
```

Скрипт предобработки запроса

В вкладке **Скрипт предобработки** редактора NASa может быть написан [BGBS скрипт предобработки](#), предобрабатывающий все запросы, приходящие на данный NAS.

Начиная с 4.2 версии биллинга, код услуги типа время может быть установлен скриптом предобработки RADIUS-запроса в опцию **service_time**. Услуга, например, может быть выбрана на основании АОН звонящего. Услуга типа время, установленная скриптом, обладает БОльшим приоритетом по сравнению с установленной в конфигурации, примеры можно посмотреть в [Wiki](#).

Скрипты обработки RADIUS-запросов кэшируются RADIUS-сервером при первом исполнении. Для сброса кэша необходимо перезапустить RADIUS-сервер, либо выполнить команду в каталоге RADIUS-сервера.

Для Linux:

```
./radius.sh flush_script_cache
```

Для MS:

```
radius.bat flush_script_cache
```

Логи работы скриптов вы можете посмотреть в файле **BGBillingServer/log/script.log**

Пересылка RADIUS Accounting в DialUP

Для дублирования полученных RADIUS Accounting-запросов на сторонний сервер в конфигурации NASa следует указать.

```
packet.forward.host=<host>
packet.forward.port=<port>
packet.forward.secret=<secret>
packet.forward.user.name.prefix=<prefix>
packet.forward.user.name.suffix=<suffix>
```

Где:

- **<host>** - хост, на который будут ретранслироваться пакеты, единственный обязательный параметр для включения функционала;
- **<port>** - порт, на который будут ретранслироваться пакеты, по умолчанию равен 1813;
- **<secret>** - RADIUS-секрет, которым будет подписан пакет, по умолчанию берётся секрет NASa;
- **<prefix>** - префикс к значению атрибута User-Name из аккаунтинг пакета, если не указан - префикс не добавляется;
- **<suffix>** - суффикс к значению атрибута User-Name из аккаунтинг пакет, если не указан - суффикс не добавляется.

Суффикс и префикс могут быть полезны при необходимости ретранслирования пакетов с нескольких BGRadiusDialup с пересекающимися логинами на единый сервер.

Например:

```
packet.forward.host=bitel.ru
packet.forward.port=4444
packet.forward.secret=4343
packet.forward.user.name.prefix=test_
packet.forward.user.name.suffix=_forv
```

Настройка RADIUS-сервера для DialUp

BGRadiusDialup обновляется как обычное [серверное приложение биллинга](#). Необходимо обновить приложение перед первым запуском.

Установка BGRadiusDialup на Linux-платформу

1) Извлеките **BGRadiusDialup** из архива и скопируйте в каталог **/usr/local**;

2) Перейдите в каталог **/usr/local/BGRadiusDialup**;

3) Удалите все **.ini**, **.bat** и **.exe** файлы:

```
rm -f /*.bat & rm -f /*.exe & rm -f /*.ini
```

4) Откройте для редактирования файл **radius.sh** и пропишите в нем путь к Java машине, например так:

```
...
cd ${0%${0##*/}}.

JAVA_HOME=/opt/java/jdk16

if [ -z "$JAVA_HOME" ]; then
    echo "The JAVA_HOME environment variable is not defined"
    echo "This environment variable is needed to run this program"
    exit 1
fi
...
```

5) Проверьте **.sh** файлы на наличие символов **^M**, если символы присутствуют их можно удалить вручную, либо воспользоваться утилитой:

```
dos2unix *.sh
```

6) Установите права запуска для всех ***.sh** файлов:

```
chmod 744 *.sh
```

7) Возьмите из каталога **BGRadiusDialup/script** скрипт запуска **bgradius_dialup** и скопируйте его в каталог **/etc/init.d**, установите права на исполнение (см. выше). Если вы изменили каталог установки или переименовывали **BGRadiusDialup**, скорректируйте скрипт;

8) Выясните текущий уровень запуска системы командой:

```
[root@gate init.d]# runlevel
N 3
```

9) Создайте линк для автоматического запуска RADIUS-сервера:

```
ln -s /etc/init.d/bgradius_dialup /etc/rc5.d/S99bgradius_dialup
```

10) Произведите настройку **radius.properties** и запустите RADIUS-сервер (см. далее).

При необходимости установки нескольких **BGRadiusDialup**-серверов на одной машине конечный каталог может быть переименован, например, в **BGRadiusVPN**. Также требуется переименование и корректировка скрипта запуска, разнесение RADIUS-портов в **radius.properties**.

Установка BGRadiusDialup на Windows платформу

Для установки BGRadiusDialup на Windows платформу на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить RADIUS стоит Java машина. Если её нет, установите версию не меньше 1.6.0. Загрузить можете с нашего сайта.
- 2) Загрузите с сервера RADIUS-сервер для DialUp.
- 3) Распакуйте архив на диск C:
- 4) Установите переменную окружения **BGRAD_HOME_DIALUP=C:\BGRadiusDialup**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга.
- 5) Установите службу BGRadiusDialup, для чего запустите файл **radius_install.bat**.
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем можете удалить эту службу, используя **radius_uninstall.bat**.

Настройка radius.properties

Откройте файл **radius.properties** и произведите настройку:

```
#-
processor.class=ru.bitel.bgbilling.modules.dialup.radius.DialUpRadiusProcessor
# dialup
processor.mid=XXX

#
auth.port=1812
acct.port=1813
admin.port=1955

#
db.driver=com.mysql.jdbc.Driver
db.url=jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false
db.user=bill
db.pswd=bgbilling

# MQ
mq.url=failover:(nio://127.0.0.1:61616)
mq.user=bill
mq.pswd=bgbilling
```

Установите переменную **processor.mid**=<числовой код экземпляра модуля DialUp>. При необходимости скорректируйте параметры доступа к базе данных и к **MQ**-серверу. Параметры **auth.port** и **acct.port** - порты, на которых сервер будет слушать авторизацию и аккаунт. Установлены значения по умолчанию. Параметр **admin.port** определяет порт, на котором работающий процесс BGRadiusDialup открывает сокет управления, данный порт не должен быть доступен с других машин. Если на этой же машине установлены другие сервера, использующие такие же порты, их следует изменить.

Изменения описанных в данном разделе параметров в radius.properties применяются только после перезапуска BGRadiusDialup. Все иные настройки могут быть изменены в ходе работы приложения и применяются при редактировании NASов и их конфигураций, редактировании конфигурации ядра (система алармов), конфигурации модуля.

Администрирование BGRadiusDialup

Управление BGRadiusDialup

Для запуска и останова сервера RADIUS для DialUp используйте:

- 1) для Windows: консоль запуска и управления службами, служба BGRadiusDialup;
- 2) для UNIX: скрипты **radius_start.sh** и **radius_stop.sh**.

После запуска посмотрите логи в папке **BGRadiusDialup/log**.

- **radius.log** - распечатка пакетов запросов и ответов;
- **radius.out** - выходной поток, критичные ошибки;
- **connection.log** - лог хода соединения, обсчётов.

Если запуск прошёл успешно, в логе **connection.log** должен вывестись список загруженных NASов, указанных вами в модуле DialUp.

В **radius.log** должно быть сообщение вида:

```
INFO 18.05.2004 13:04:41 Starting radius auth_port:1812 acct_port:1813 admin_port:1899
INFO 18.05.2004 13:04:41 Init processor
class: bitel.billing.server.processor.voiceip.DialUpProcessor
mid: 6

INFO 18.05.2004 13:04:42 Starting PortListener port=1812|type=AUTH_LISTENER
INFO 18.05.2004 13:04:42 Starting PortListener port=1813|type=ACCOUNT_LISTENER
INFO 18.05.2004 13:04:42 Starting PortListener port=1899|type=ADMIN_LISTENER
```

Это свидетельствует о том, что сервер запущен и ожидает пакеты.

Если сервер не запустился, ищите причину в файле **radius.out**. В него пишутся все критичные ошибки.

С работающего RADIUS-сервера возможно получение с сервера списка соединений и статуса. Это достигается запуском скрипта **radius.bat(.sh)** с параметрами. Список параметров можно получить простым запуском **radius.bat(.sh)**. Ниже приведена выводимая при этом справка.

```
Usage: [start|stop|help|status|ps|kill|flush_script_cache]
Params:
  help|? - show this help
  start - starting RADIUS server
  stop - stopping RADIUS server
  status - current connections status
  flush_script_cache - flush BGS script cache
##### Only for DialUp RADIUS #####
  ps - active connections list
  kill [-port <#port>] [-nas ] [-login <#login>]
      - kill connections by filter
  kill doesn't work with empty params list

Example: radius.sh start
Example: radius.sh kill -nas supernas.bayan.com -login 11
```

radius.sh(.bat) ps - вывод списка текущих соединений в следующем формате

```
bill@bill-reg BGRadiusDialup]$ ./radius.sh ps
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
|  NAS_ID  |  NAS_IP  |          Session          |  Start  |  Login  |  IP  |
|  FromNum |  Contract |  Status  |          |          |      |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
|  drs1    | 89.189.150.67 | 3419823-p0159 | 19.02.2008 16:17:03 | 2443 | 77.79.162.9 |
| 10.47.8.140 | 294NK007193-08 | active | | | |
|  drs1    | 89.189.150.67 | 3419689-p0158 | 19.02.2008 16:14:49 | 2230@local | 172.27.35.248 |
| 10.47.55.194 | FREEDOM_0/294NK005198-07 | active | | | |
|  drs1    | 89.189.150.67 | 3413354-p0157 | 19.02.2008 14:29:16 | 2499 | 89.189.151.103 |
| 10.47.1.130 | FREEDOM_0/294NK005420-07 | active | | | |
```

radius.sh(.bat) status - краткий статус сервера

```
version 4.4 build 93 from 05.03.2009 17:14:01
12.03.2009 16:31:25      31035  30998  37    0
Request accounts per minute start: 264; stop: 248; update: 2931
Request auths per minute accept: 264; reject: 231
Netfow packets per minute: 144334
Ignore per minute auth: 0; update: 122
Antispam ban count: 20; used per minute: 143
Started: 06.03.2009 05:54:26    Uptime: 6 d 10:36:59
Memory total: 5 234 163 712; max: 5 234 163 712; free: 1 973 243 208
Trees in cache: 39
Connections pool to Master status Idle: 50; Active: 2; maxActive: 100; maxIdle: 50
```

Построчно статус:

1. Версия, номер и дата билда BGRadiusDialup;
2. Текущее время, общее число соединений на сервере, число в **статусе active, sleep, wait**;
3. Количество запросов аккаунтинга за последнюю минуту с разделением по типам;
4. Количество запросов авторизации за последнюю минуту с разделением по успешным и не успешным авторизациям;
5. Количество NetFlow-дейтаграмм, полученных за последнюю минуту;
6. "Проглочено" за последнюю минуту авторизаций и аккаунтинг Update-пакетов;
7. Количество записей в спам-базе и количество использований спам-базы за последнюю минуту;
8. Время старта и uptime BGRadiusDialup;
9. Статус по потребляемой памяти, количеству деревьев в кэше соединений и пулу соединений к БД, более подробно объяснения по данным параметрам [здесь](#);
10. Статус по количеству деревьев в кэше соединений, более подробно объяснения по данным параметрам [здесь](#);
11. Статус по пулу соединений к БД, более подробно объяснения по данным параметрам [здесь](#).

radius.sh(.bat) kill <фильтры> - послать команду сброса для соединений.

Если фильтр не установлен, сигнал будет послан для всех соединений на RADIUS-сервере.

radius.sh(.bat) flush_script_cache - сброс кэша [скриптов](#) преобработки RADIUS-запросов.

Расширение словаря RADIUS

Все указываемые в конфигурациях модуля атрибуты RADIUS должны быть описаны в файле dictionary.xml. Недостающие вендоры и их атрибуты могут быть добавлены администратором системы самостоятельно. Имя любого атрибута должно быть уникально в пределах dictionary.xml.

Антиспам

Спам-запросы на авторизацию порождаются, обычно, забытым оборудованием, самостоятельно предпринимающем попытки авторизации на RADIUS. Постоянные обращения в базу данных при таких запросах существенно повышают загрузку. Признаком спам-запроса являются несколько Reject-ответов на одинаковые запросы авторизации в течение определённого интервала времени. В этом случае подобные запросы попадают в спам-базу на определённое время и по ним автоматически выдаётся Reject-ответ без полных проверок, производимых при авторизации. Для включения антиспам-системы в конфигурации модуля добавьте:

```
antispam.key.attributes=<key_attributes>
antispam.reject.count=<reject_count>
antispam.reject.per.time=<per_time>
antispam.ban.time=<ban_time>
```

Где:

- **<key_attributes>** - RADIUS-атрибуты, идентифицирующие запрос, через запятую;
- **<reject_count>** - количество Reject, возвращённых на запрос с одинаковыми идентифицирующими атрибутами;
- **<per_time>** - за какое количество секунд выдано указанное в <reject_count> количество Reject-ответов;
- **<ban_time>** - время в секундах, на которое данный запрос попадает в спам-базу.

Например, запрос идентифицируется атрибутами User-Name и Calling-Station-Id. При десяти Reject-ответах в течении минуты запрос попадает в спам-базу на полчаса.

```
antispam.key.attributes=User-Name,Calling-Station-Id
antispam.reject.count=10
antispam.reject.per.time=60
antispam.ban.time=900
```

Поведение BGRadiusDialup при критических нагрузках

Вы можете пропустить этот раздел при первичной настройке системы.

На порты авторизационных и аккаунтинговых запросов в BGRadiusDialup определён пул потоков-обработчиков размерами, задаваемыми переменными в `radius.properties` **`auth.thread.count`** и **`acct.thread.count`** соответственно. Каждый пришедший запрос обрабатывается в одном из свободных потоков пула.

По мере роста числа запросов авторизации, если число запросов в очереди превысит значение, определённое переменной в `radius.properties` **`auth.thread.queue`**, RADIUS-сервер начинает "проглатывать" запросы авторизации без обработки, предотвращая, тем самым, накопление устаревших запросов в буфере приёма сетевой подсистемы и обрабатывая только самые свежие запросы.

По мере роста числа запросов аккаунтинга, если число запросов в очереди превысит значение, определённое переменной в `radius.properties` **`acct.thread.queue`**, RADIUS-сервер начинает "проглатывать" Update-запросы, оставив оставшиеся потоки для обработки запросов старта и стопа, т.к. они критичнее.

Изменяя параметры количества потоков, следует также изменять количество максимальных активных соединений с базой данных, т.к. ожидание свободного соединения с базой данных блокирует потоки обработки запросов.

Оптимизация работы с базой данных

Вы можете пропустить этот раздел при первичной настройке системы.

При обчёте сессий в реальном режиме времени происходит значительная нагрузка на базу данных. Постоянно обновляются таблицы `log_session`, `session_detail`, `session_account`, `contract_balance`, `contract_account`.

В таблицу `session_detail` производится запись объёма потреблённых по ходу сессии услуг с разбивкой по часам. Возможно снизить число обновлений таблицы, отменив запись по некоторым тарифицируемым по нулевой стоимости и не нужных услуг (например, время при тарификации трафик). Также возможно сгруппировать записи, например, по суткам, если тарифные планы не предполагают различную стоимость услуг по времени суток. Правила "свёртки" таблицы `session_detail` определяются в конфигурации модуля следующим образом:

```
detail.compress.<sid>.<hour_range>=<rule>
```

Где:

- **<sid>** - числовой код услуги, запись о которой вносится в `session_detail`;
- **<hour_range>** - диапазон часов, к которым относится правило;
- **<rule>** - ключевое слово SKIP (пропуск обновления), либо час, в который преобразуется реальный час вносимой записи.

Например, для отброса записи в `session_detail` информации об услуге Время с кодом 2 и свёртке по суткам записей по наработке по услугам трафиков 23 и 24 определяются следующие правила:

```
detail.compress.2.0-23=SKIP  
detail.compress.23.0-23=0  
detail.compress.24.0-23=0
```

В случае, если стоимость трафика изменяется от времени суток, возможно преобразование реального часа потребления услуги в час начала диапазона.

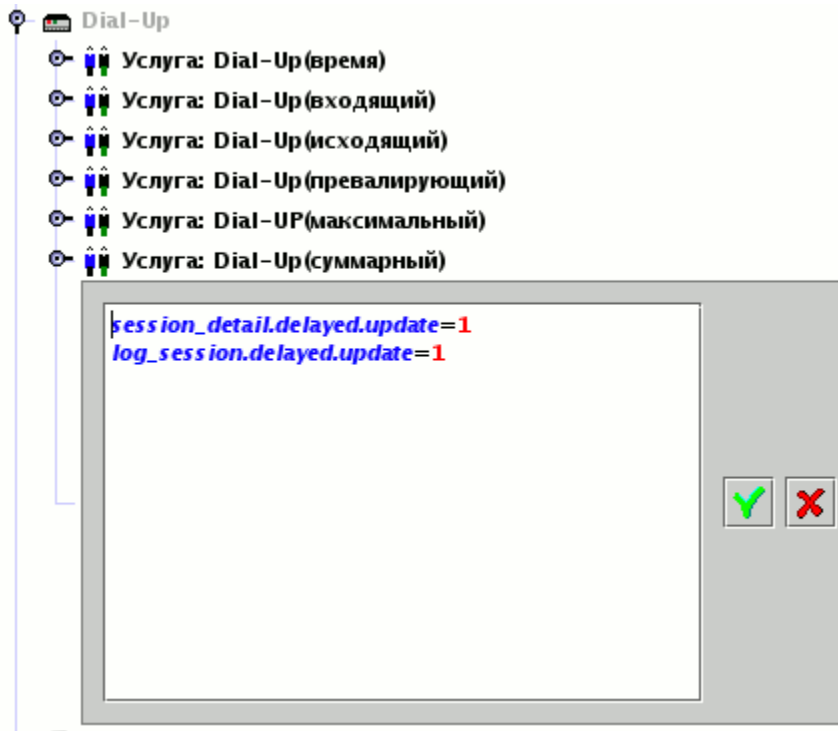
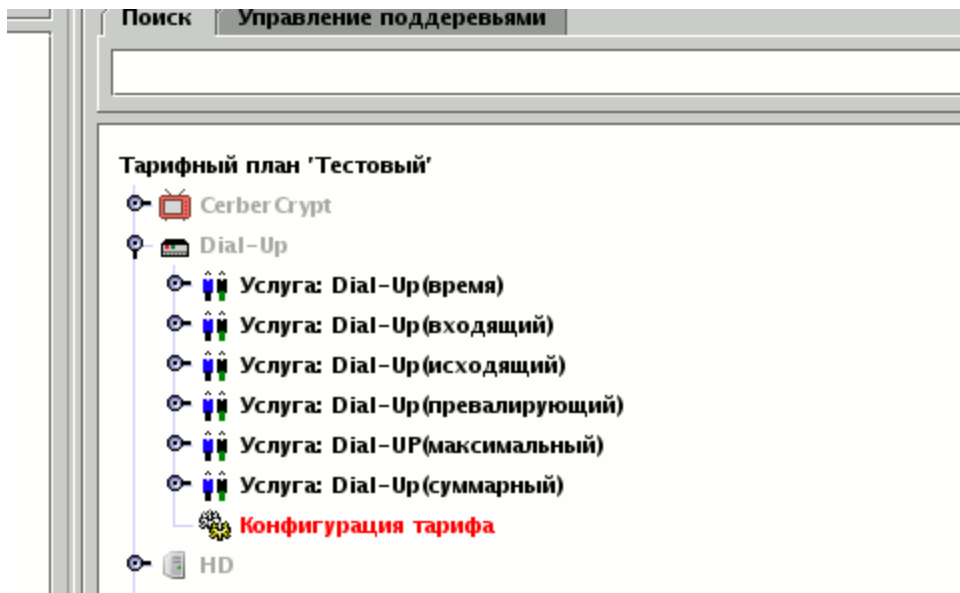
При безлимитных тарифах жёсткое фиксирование посчитанного по нулевой цене трафика в `session_detail` не обязательно, информация может быть сохранена в конце часа, либо сессии. Так же не требуется постоянное обновление таблицы `log_session` (стоимость сессии и время её окончания). Для включения отложенного обновления таблиц `session_detail` и `log_session` возможно определение в узле **Конфигурация** тарифного плана следующих правил:

```
session_detail.delayed.update=1  
log_session.delayed.update=1
```

При необходимости можно добавить только одну из строк.

Установка опции в значение 0, равно как и в любое другое, не отключает ее! Для отключения необходимо удалить, либо закомментировать строку символом # в начале строки.

На снимке экрана ниже изображён узел в тарифном плане.



Отложенное обновление таблицы log_session может происходить только при нулевом приращении стоимости сессии. Отложенное обновление таблицы session_detail при ненулевых тарифных планах также возможно, однако следует учитывать следующие нюансы:

- При обчёте трафика по RADIUS-атрибутам при перезагрузке BGRadiusDialup произойдёт потеря трафика, при восстановлении сессии средства за этот трафик будут начислены повторно;
- При тарификации трафика по NetFlow-данным в случае перезагрузки BGRadiusDialup произойдёт просто потеря накопленного счётчика в памяти трафика;
- При использовании тарифов с узлом **Диапазон наработки** инициализация тарифа при старте сессии происходит по таблице session_detail, при нескольких параллельных сессиях одного договора возможна несколько некорректная тарификация. Тариф будет инициализироваться недостаточной наработкой. Проблема может быть решена переобсчётом в конце месяца всех сессий.

Reject-To-Accept

Вы можете пропустить этот раздел при первичной настройке системы.

Иногда в случае неудачной авторизации нужно все равно разрешить установку сессии, но с определенными атрибутами. Например, если у клиента нет денег на счету, но соединение все равно нужно установить, чтобы пустить его к серверу статистики. Для настройки режима Reject-To-Accept в конфигурацию модуля добавьте:

```
reject_to_accept.< >=< >
reject_to_accept.nas=< nas->
reject_to_accept.db.write=0
```

Коды ошибок авторизации доступны [здесь](#). В случае возникновения ошибок авторизации с кодами, указанными в этой конфигурации, происходит выдача accept - пакета. Клиенту выдается адрес из пула адресов стандартным образом (в списке передаваемых атрибутов должен быть Framed-Pool=<имя пула>). Так же можно указать коды NASов, для которых производится данная подмена ответа RADIUS. Если это параметр не указан, то обработка производится для всех NASов.

Пример:

```
reject_to_accept.2,3=Session-Timeout=100;cisco-Fax-MDN-Address=1;Framed-Pool=fake_pool
reject_to_accept.4,11=Session-Timeout=200;Framed-Pool=fake_pool_2
reject_to_accept.nas=10,15
```

При указании параметра **reject_to_accept.db.write=1** ошибки заносятся в таблицу БД **dialup_reject_to_accept_<mid>** для возможности перенаправления пользователя на страницу с описанием ошибки. Страница доступна по адресу: **http://<ip>:<port>/bgbilling/pubexecuter?module=dialup&action=RejectToAccept&mid=<mid>**, где:

- <ip> - адрес сервера биллинга;
- <port> - порт сервера биллинга;
- <mid> - код экземпляра модуля.

При получении запроса на данную страницу пользователю генерируется HTML страница с использованием шаблона **dialup_reject_to_accept_<mid>.xsl**, где <mid> - код экземпляра модуля. В качестве образца поставка модуля включает шаблон **dialup_reject_to_accept.xsl**, который должен быть скорректирован соответствующим образом и переименован к нужному виду. Идентификация пользователя производится по IP-адресу, с которого был произведен запрос. IP-адрес может быть передан в HTTP-заголовке, указанном в параметре **header.name.remote.addr** конфигурации сервера биллинга, например X-Real-IP.

Обычная схема перенаправления пользователя на страницу с ошибкой выглядит следующим образом:

1. пользователю выдается IP-адрес из определенной сети для Reject-To-Accept соединений;
2. маршрутизация настроена таким образом, что все обращения из данной сети перенаправляются на проксирующий HTTP-сервер, например nginx;
3. проксирующий HTTP-сервер получает все запросы на определенном порту и проксирует их на указанный выше URL, подставляя адрес, с которого было обращение, в заголовок HTTP-запроса;
4. сервер биллинга получает запрос на данный URL и по IP-адресу определяет пользователя, генерируя ему персональную страницу с ошибкой.

При разработке собственного XSLT-шаблона использовать следующую методику:

1. настроить режим reject-to-accept с записью ошибок в БД, добиться создания таблицы **dialup_reject_to_accept_<mid>** и внесения в нее записей;
2. изменить в таблице **dialup_reject_to_accept_<mid>** какую-либо из записей с тем, чтобы поле ip было равно IP-адресу вашей машины;
3. обратиться по URL: **http://<ip>:<port>/bgbilling/pubexecuter?module=dialup&action=RejectToAccept&mid=<mid>&contentType=xml** при этом страница собирается в браузере клиента, XSLT-шаблон запрашивается браузером по пути, указанному в параметре **web.xslt** конфигурации сервера биллинга;
4. посмотреть исходный код страницы для просмотра XML дерева, доступного из XSLT шаблона.

Таблица **dialup_reject_to_accept_<mid>** может храниться в "мусорной" базе. Для ускорения работы для данной таблицы используется тип таблиц **Method**, хранимых в памяти. Для корректной работы режима перенаправления на страницу с ошибкой в списке RADIUS-атрибутов, передаваемых в accept пакете, должны обязательно быть Framed-Ip-Address и Session-Timeout, в противном случае запись в таблицу произведена не будет.

Настройка встроенного коллектора

Встроенный NetFlow-коллектор поддерживает NetFlow версий 1, 5 и 7 (9ая версия не поддерживается), sFlow версии 5.

Сконфигурируйте **radius.properties**, указав порты для приема потоков:

```
#
#collector.capture.flow.buffer.capacity=4194304
#
#collector.capture.flow.socket.buffer.capacity=524288
#
#collector.capture.flow.thread.count=10

# netflow
#
collector.capture.flow.port.1=2001
# - netflow
collector.capture.flow.port.1.type=netflow
#
#collector.capture.flow.port.1.thread.count=8

# sflow
#
collector.capture.flow.port.2=2002
# - sflow
collector.capture.flow.port.2.type=sflow
```

Коллектор может принимать данные на несколько портов, при этом на один порт может принимать данные только одного типа (netflow или sflow). Для обратной совместимости в версии 4.6 порт типа netflow может также принимать данные sflow. Для каждого порта прописывается тип слушателя, также можно прописать количество потоков.

Опишите правила деления трафика по услугам в **Конфигурации модуля**.

```
netflow.service.link.1=23 IN 81.30.199.0 110-112
netflow.service.link.2=24 IN 81.30.199.12
netflow.service.link.3=25 OUT 0.0.0.0-255.255.255.255
netflow.service.link.4=27 IN 0.0.0.0-255.255.255.255
```

Читается так:Правило с приоритетом 1 к услуге с кодом 23 привязывается трафик с адреса 81.30.199.0 и с портов 110-112. Правило с приоритетом 2 к услуге с кодом 24 привязывается входящий для клиентов трафик с адреса 81.30.199.12....

Конфигурация состоит из строк следующего формата (разделители-пробелы):

```
netflow.service.link.<pos>=<sid> <direction> <address> <ports>
```

где:

- **<pos>** - позиция правила, просмотр идёт в порядке позиций;
- **<sid>** - код услуги, к которой привязано правило;
- **<direction>** - ключевое слово IN или OUT, определяет тип трафика по отношению к клиенту, которое описывает правило;
- **<address>** - один диапазон адресов, включая концы, либо одиночный адрес;
- **<ports>** - диапазон портов, либо одиночный порт, параметр может быть пропущен.

В каждом правиле можно указать только один диапазон адресов и портов. Для добавления нескольких диапазонов последовательно располагаются несколько правил. В последних позиции должны стоять правила, в которые попадают все входящие и все исходящие пакеты (см. пример), это гарантирует вам что трафик не "потеряется" совсем.

Для того чтобы ассоциировать NAS с коллектором в биллинге добавьте в конфигурации NASa строку:

```
collector.agent.address=<IP NAS>
```

После добавления этих строк все адреса клиентов, авторизовавшихся на NASE, будут добавлены на коллекторе и по ним будет сниматься периодическая статистика по количеству потреблённых услуг.

При регистрации адреса на коллектор кроме него самого передаётся адрес агента NetFlow, с которого будет приниматься данные для этого адреса. Это сделано с целью недопущения повторного обчёта трафика в случае, если трафик идёт от одного клиента к другому.

Для уточнения портов, на которые приходят потоки для NASA, добавьте в конфигурации NASA:

```
collector.agent.ports=< , NAS>
```

Для применения изменённых настроек коллектора достаточно перезапустить BGRadiusDialup, сохранить конфигурацию, модуля либо изменить что-то в списке NASов. После изменения настроек коллектора проверьте лог **collector.log** на предмет ошибок разбора конфигурации привязки услуг.

Если вы используете NetFlow и тарифицируете трафики с делением по типам, то для их отображения в отчёте добавьте строку в конфигурации модуля:

```
traffics=<sid1>/<sid2>/..<sidN>;<title1>/<title2>/..<titleN>
```

Например:

```
traffics=1/2;./.
```

В этом случае в отчётах по сессиям клиентов через дробь будут выводиться наработки по трафикам с кодами услуг 1 и 2 через дробь. Надпись "Внешн./Лок." будут добавлены в заголовок таблицы отчёта по сессиям клиента в Web. Если такая строка не указана в трафиках сессии выводятся суммарный входящий и исходящий трафики по данным RADIUS.

Для возможности предоставления детализации по сессиям необходимо установить IPN модуль, создать в нем источник, соответствующий NASy DialUP-модуля и направить на него дублирующий поток. В конфигурации NASA прописать опции:

```
#
ipn.module.id=< IPN>
ipn.source.id=< , NAS>
```

В мониторе модуля DialUp отображается всегда трафик по данным RADIUS, использовать его для сверки цены сессий некорректно при обчёте по данным NetFlow.

Переобработка NetFlow трафиков

При переобработке трафиков трафики, определенные как MAX в конфигурации NASa, вычисляются как максимум записей по трафикам, из которых вычисляется максимум. В результате значения этих трафиков могут измениться после переобработки, т.к. в ходе тарификации максимум высчитывается по состоянию на каждый Update-пакет. Настоятельно рекомендуем вам сделать резервную копию таблицы session_detail_<mid>_ууууММ перед переобработкой трафиков!

Переобработку трафиков осуществляет планировщик заданий BGScheduler, переобработка производится на основании логов формата коллектора IPN, т.е. необходимо организовать параллельный сбор логов в этом модуле, либо сконвертировать существующие логи в формат IPN-коллектора.

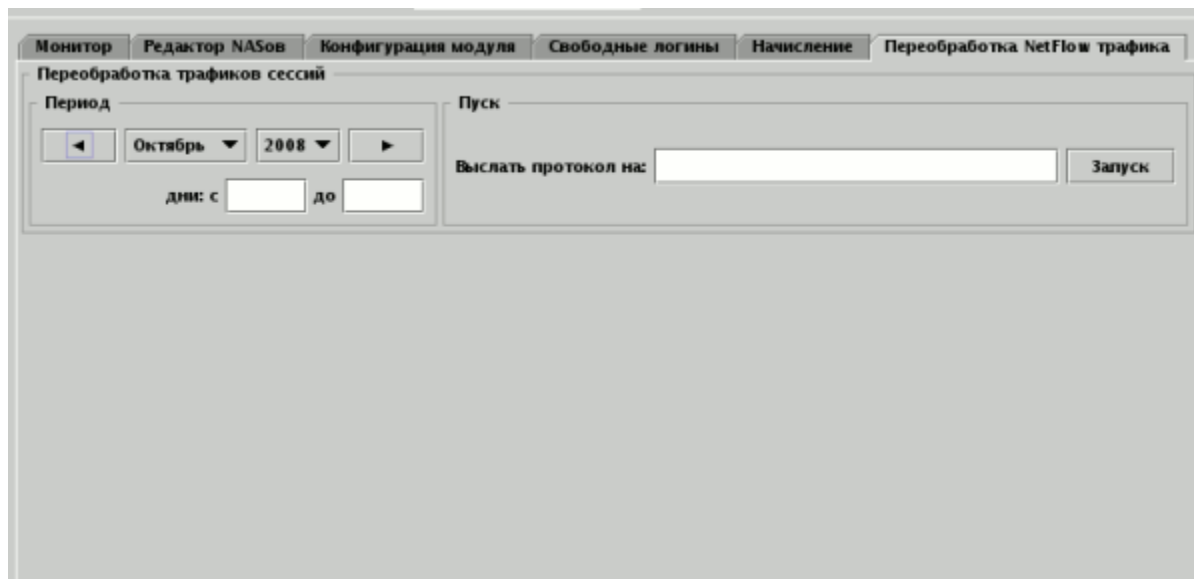
Путь к логам прописывается в конфигурации NASов следующим образом:

```
netflow.log.path=< source- IPN->
```

Например:

```
netflow.log.path=/storage/drs/drs18.ufanet.ru
```

Путь ищется на машине, где запущен планировщик заданий. Для запуска переобработки используется вкладка **Переобработка NetFlow трафика** в модуле.



Для того, чтобы переобработать трафик только по некоторым NASам в данный момент можно только комментировать строки с путём к первичным логам в конфигурации NASa. Письмо о завершении переобработки приходит на указанный E-Mail.

Настройка абонентов DialUp

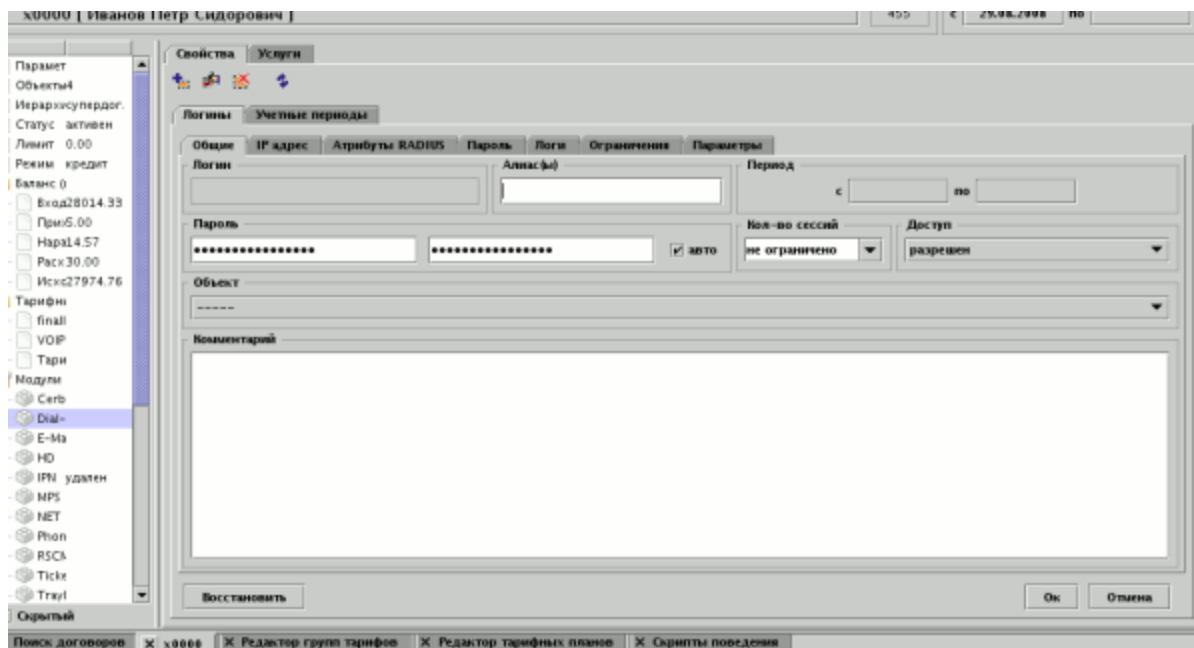
Для того, чтобы добавить клиенту возможность использования модуля DialUp, откройте его договор, добавьте ему экземпляр модуля, выбрав узел **Модули** дерева договора и нажав кнопку **Новый элемент** стандартной панели инструментов.

Добавление разрешённых услуг модуля в договор необходимо:

- При установке опции **check.service=1** в конфигурации модуля. Клиент не сможет авторизоваться, если в договоре не будет хотя бы одной из прописанных на NAsе услуг;
- При использовании тарифных планов с узлом **Диапазон** в режиме **пропорционально периоду разрешённой услуги**. Объем пропорциональной квоты определяется долей месяца, в течении которой была разрешена услуга. Отсутствие разрешённой услуги означает полный объем;
- При начислении максимальных трафиков, см. [далее](#).

Откройте модуль и в список логинов добавьте новый логин. Для этого воспользуйтесь панелью инструментов, расположенной слева от списка логинов. При добавлении логина сначала нажмите **Новый элемент**, заполните настройки на вкладке **Общие**, другие вкладки по необходимости и сохраните изменения. Числовой логин выдаётся автоматически, изменить можно только алиас (замена числовому логину при входе). **Числовые алиасы запрещены**. Можно задать несколько алиасов, для этого перечислите их, используя пробел, как разделитель.

В основных свойствах установите период действия логина, алиас (по необходимости), ограничение по числу сессий (обязательно при выдаче статических адресов), пароль (галочка **авто** позволяет сгенерировать пароль автоматически длиной, заданной переменной в конфигурации модуля **password.length.auto** и набором символов из переменной **password.chars**).



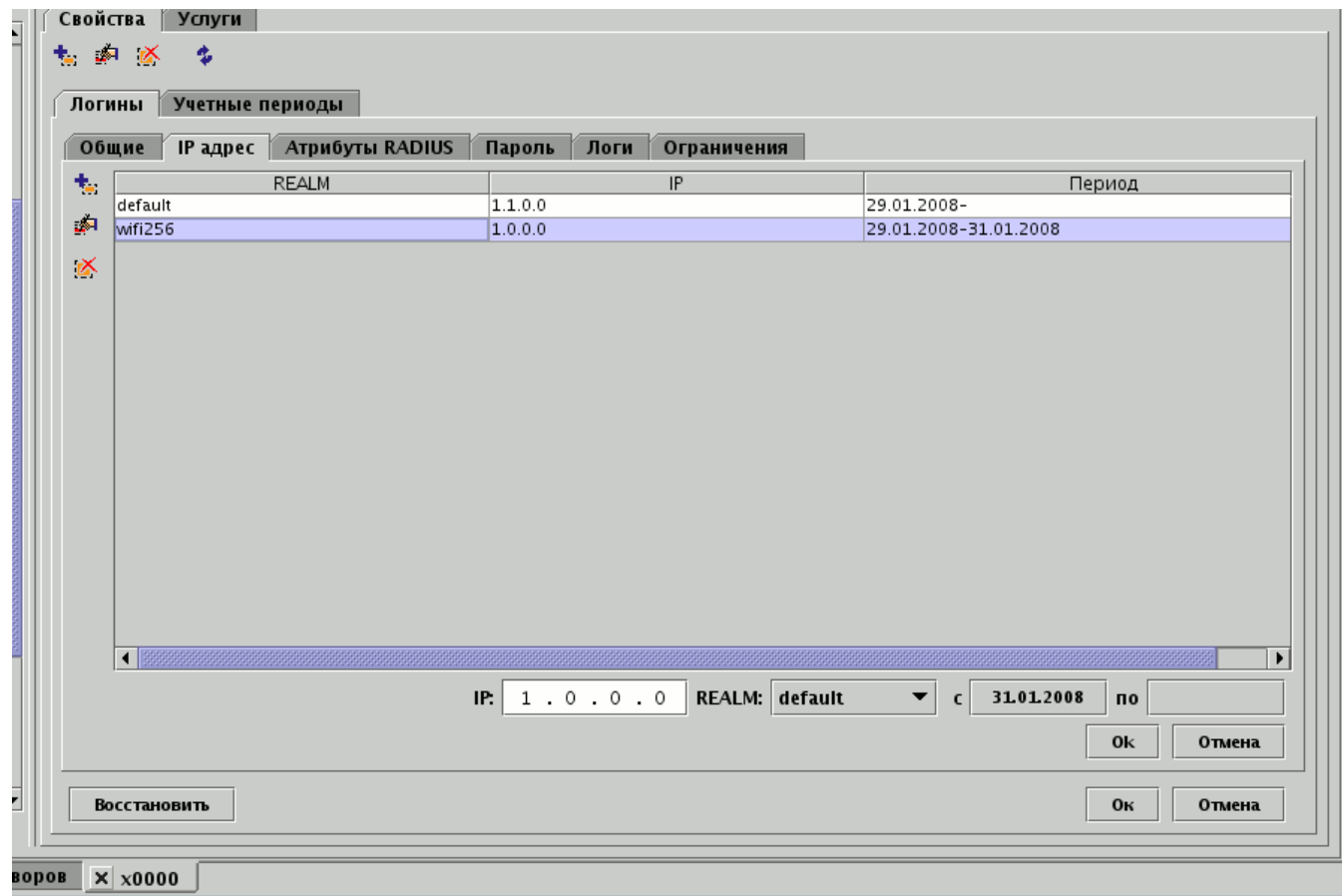
Опции, устанавливаемые на прочих вкладках свойств логина и допустимые манипуляции с созданными логинами (переносы), описаны ниже.

Вкладка "IP-адрес"

На вкладке **IP-Адрес** задаются адреса, присваиваемые клиенту в момент авторизации. Адреса задаются с привязкой к реалму. Если на один реалм назначено несколько адресов, то будет не занятый адрес на каждую последующую сессию логина.

В отличие от назначения адресов указанием атрибута **Framed-IP-Address** на данной вкладке осуществляется контроль совпадения адресов у разных логинов.

При назначении клиенту одного или нескольких фиксированных адресов обязательна установка лимита на количество одновременных соединений. Если на реалм установлено несколько адресов - они будут выдаваться для соединений данного логина как из пула.

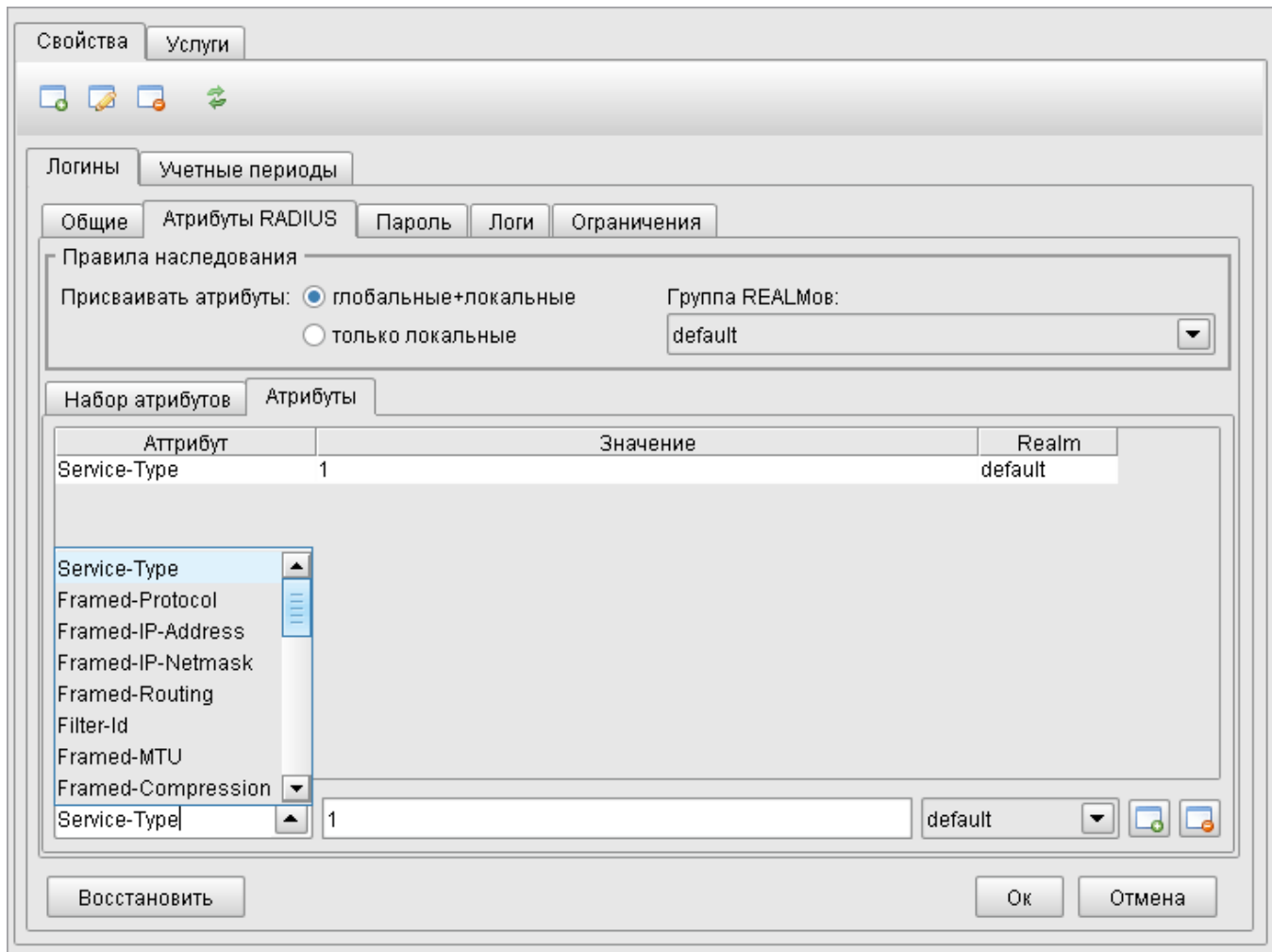


Вкладка "Атрибуты RADIUS"

Вкладка **Атрибуты RADIUS** позволяет задавать атрибуты, такие как **Framed-Ip-Address** - IP-адрес, выдаваемый пользователю и **Framed-Pool** - пул адресов, установленный для пользователя. Атрибуты можно задавать как наборами, так и отдельными атрибутами. Все атрибуты (наборы) задаются с привязкой к REALMу.

The screenshot shows a software configuration window with the following elements:

- Top tabs: **Свойства** (selected), **Услуги**
- Sub-tabs: **Логины**, **Учетные периоды**
- Internal tabs: **Общие**, **Атрибуты RADIUS** (selected), **Пароль**, **Логи**, **Ограничения**
- Section: **Правила наследования**
- Options: **Присваивать атрибуты:** глобальные+локальные, только локальные
- Field: **Группа REALMов:** dropdown menu with options: default, default, test
- Section: **Набор атрибутов**
- Sub-tab: **Атрибуты**
- Table with columns: **Набор**, **Realm**
- Table content: Канал 128, default
- Bottom controls: **Канал 128** dropdown, **default** dropdown, and icons for save, delete, and refresh.
- Buttons: **Восстановить**, **Ок**, **Отмена**



В поле **Группа REALMов** укажите группу, в которую входит данный логин.

Также можно задать режим выдачи RADIUS-атрибутов логину. По умолчанию логину выдаются атрибуты, прописанные для REALMa, под которым он зашёл в конфигурации модуля DialUp + прописанные непосредственно в свойствах логина для данного REALMa. При выборе флага **Только локальные** глобальные атрибуты RADIUS для данного REALMa будут игнорироваться.

Вкладка "Ограничения"

Вкладка **Ограничения** содержит список возможных ограничений логина. Возможные ограничения: по времени доступа и работы, по телефонам доступа, по наработке денег и по наработке услуги.

Видам ограничений соответствуют следующие числовые коды:

1. ограничение по телефону доступа (либо IP-адресу NASa, т.е. по содержимому атрибута **Called-Station-Id**);
2. по времени доступа/работы (без разрыва уже установленного соединения или с разрывом);
3. по наработке в единицах услуги;
4. по наработке в деньгах;
5. по телефону клиента (либо MAC-адресу, т.е. по содержимому атрибута **Calling-Station-Id**).

3 и 4 типы ограничений привязаны к услугам, то есть наработку можно ограничить только по каким-либо услугам.

Следует быть внимательным, программа совершенно корректно воспримет, если вы выберете в списке услуг трафик и время и ограничите их, например, 3600. То есть, сумма полученных байт и проработанных секунд будет ограничена 3600.

Для редактирования списка ограничений используется собственная панель инструментов. Каждое ограничение имеет временные маски, ограничивающие время работы ограничения. С их помощью можно ограничить различными значениями, например, трафик в обычные и в выходные дни. Значения в масках указываются через запятую. Часы - начиная с 0, все остальное - с 1. Для ограничений по времени входа временные маски задают, когда это ограничение пускает или нет. Если временные маски не указаны, ограничение работает всегда.

При авторизации ограничения сортируются по типам, просматриваются в порядке их указания, каждое последующее перекрывает предыдущее только, если оно одного типа. То есть, если вы решите пускать пользователя только на телефон 11111, то добавление ограничения Разрешить вход на 11111 ничего не даст, т.к. вход на него и так разрешён по умолчанию. Следует сделать запрет на все телефоны доступа, указав в качестве телефона *, а затем добавить строку с разрешением входа на 11111. Группировка по типам при авторизации делается для того, чтобы ограничения одного типа не перекрывали другие. То есть, если человек не прошёл по телефону доступа, его не должно пустить только потому, что доступ в это время разрешён.

Дополнительные сведения по ограничениям:

- 1. * означает любое значение в телефонах доступа;
- 2. дополнительно существуют ограничения по количеству одновременных сессий и вообще разрешение или запрет входа, они вынесены на вкладку **Общие**;
- 3. ограничения по наработке услуги или времени должны быть привязаны к списку ограничиваемых услуг;
- 4. любое ограничение действует только в указанный для него период действия.

Для создания ограничения нажмите кнопку **Новый элемент**, выберите его тип и настройте параметры, затем примените, нажав кнопку **Ок**. Для редактирования - выберите ограничение и нажмите кнопку **Редактировать**. Перемещая ограничения вверх или вниз с помощью кнопок **Вверх на одну строку** и **Вниз на одну строку**, можно изменять порядок следования ограничений, тем самым, меняя порядок их просмотра.

Значения в счётчике ограничений начинают накапливаться только после установки ограничения на логине. Т.е. если вы поставите ограничение по наработке 600Мб на логин в середине месяца, считаться эти 600Мб начнут не с начала месяца, а с момента установки ограничения

Несколько примеров использования ограничений.

Общие IP адрес Атрибуты RADIUS Пароль Логи Ограничения Параметры

Тип: По телефону клиента

Ограничение: разрешить телефон 454444

Период: с 28.06.2010 по

Услуги: Dial-UP(максимальный)

Временные критерии Комментарий

Восстановить

Ok Отмена

Восстановить

Ok Отмена

Логин поведения Библиотека скриптов Поиск договоров x0000

Логин Учетные периоды

Общие IP адрес Атрибуты RADIUS Пароль Логи Ограничения Параметры

Ограничения

Тип	Ограничение	Период
3	deny*	28.06.2010 -
5	allow454444	28.06.2010 -

Восстановить

Ok Отмена

Логин поведения Библиотека скриптов Поиск договоров x0000

Логин Учетные периоды

Общие IP адрес Атрибуты RADIUS Пароль Логи Ограничения Параметры

Тип: По времени доступа

Ограничение: запретить действие работа

Период: с 28.06.2010 по

Услуги: Dial-UP(максимальный)

Временные критерии Комментарий

час => (0 - 7);

Восстановить

Ok Отмена

Восстановить

Ok Отмена

Логин поведения Библиотека скриптов Поиск договоров x0000

Общие IP адрес Атрибуты RADIUS Пароль Логин Ограничения Параметры

Ограничения

Тип	Ограничение	Период
2	deny работа	28.06.2010 -

Восстановить Ок Отмена

Типы поведения X Библиотеки скриптов X Поиск договоров X x0000

Договоры Чужие периоды
Новый элемент

Общие IP адрес Атрибуты RADIUS Пароль Логин Ограничения Параметры

Тип
По работе (д. услуг)

Ограничение
д. услуг 1000000 за за месяц

Период
с по

Услуги
 Ды1-ИР(исключаемый)
 Ды1-Ур(исключаемый)

Применить критерии Комментарий

Все Сброс Инверт

Восстановить Ок Отмена

Восстановить Ок Отмена

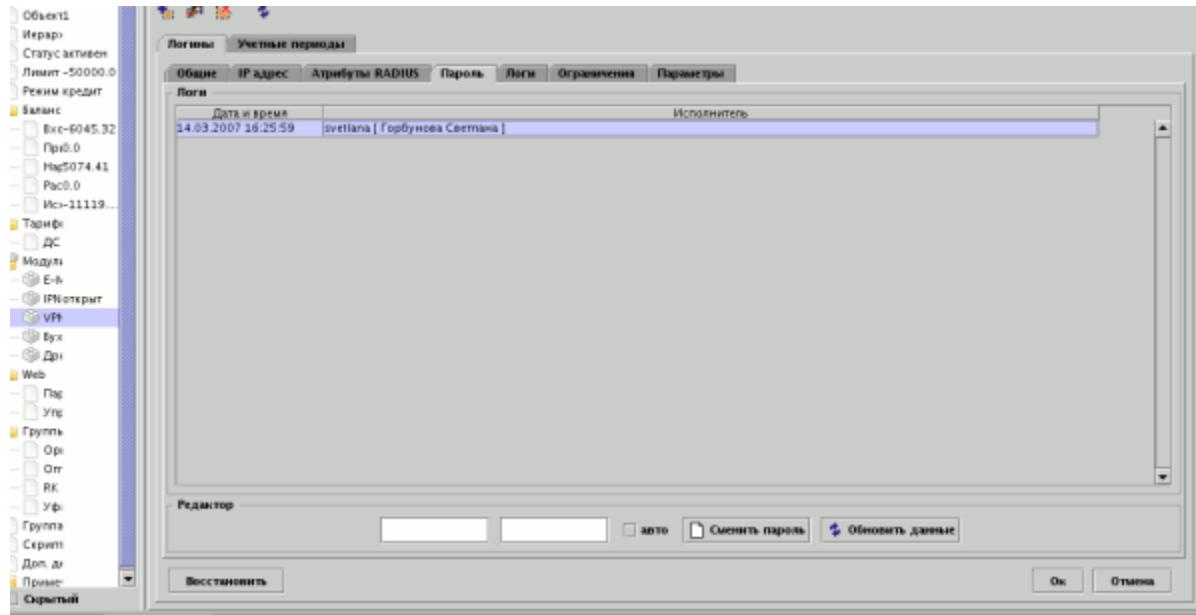
X Поиск договоров X x0000

Вкладка "Логи"

Вкладка **Логи** позволяет просмотреть последние сессии клиента и ошибки авторизации.

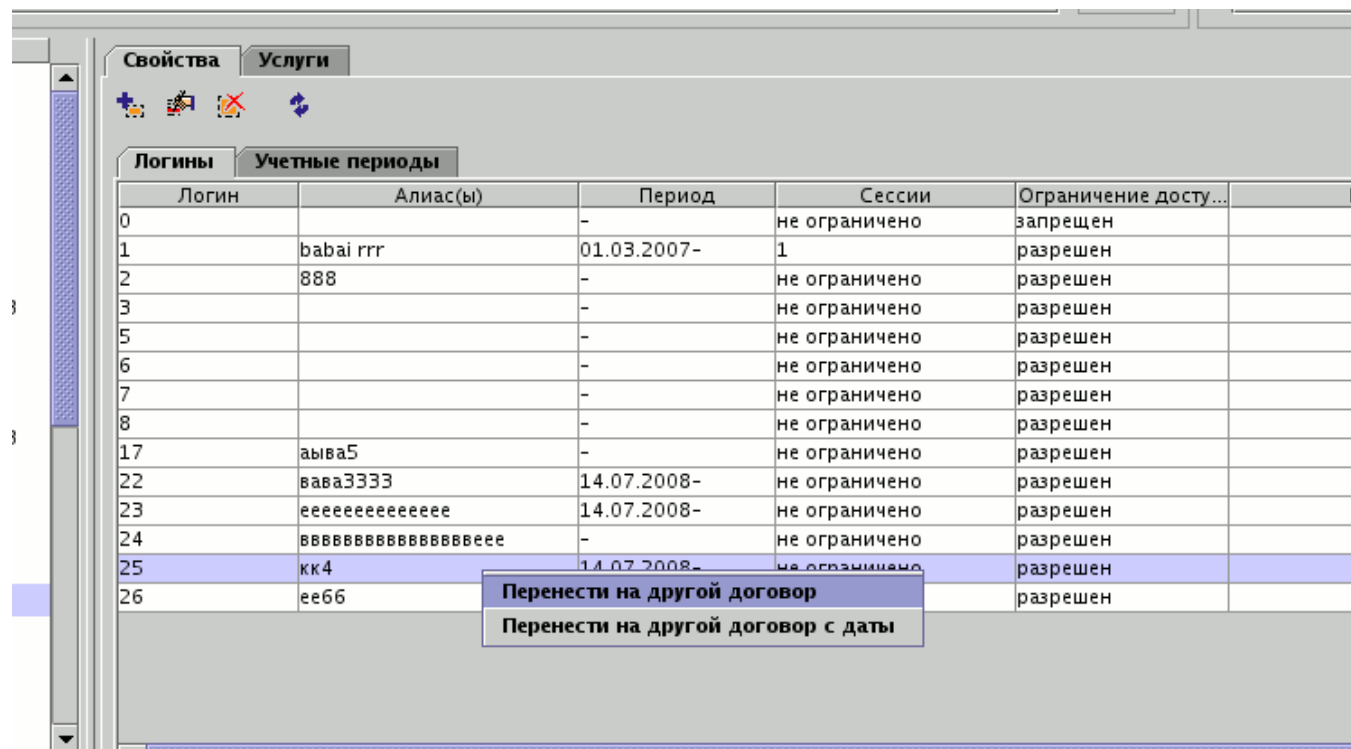
Вкладка "Пароль"

Вкладка **Пароль** позволяет изменить пароль доступа и посмотреть логи последних изменений.

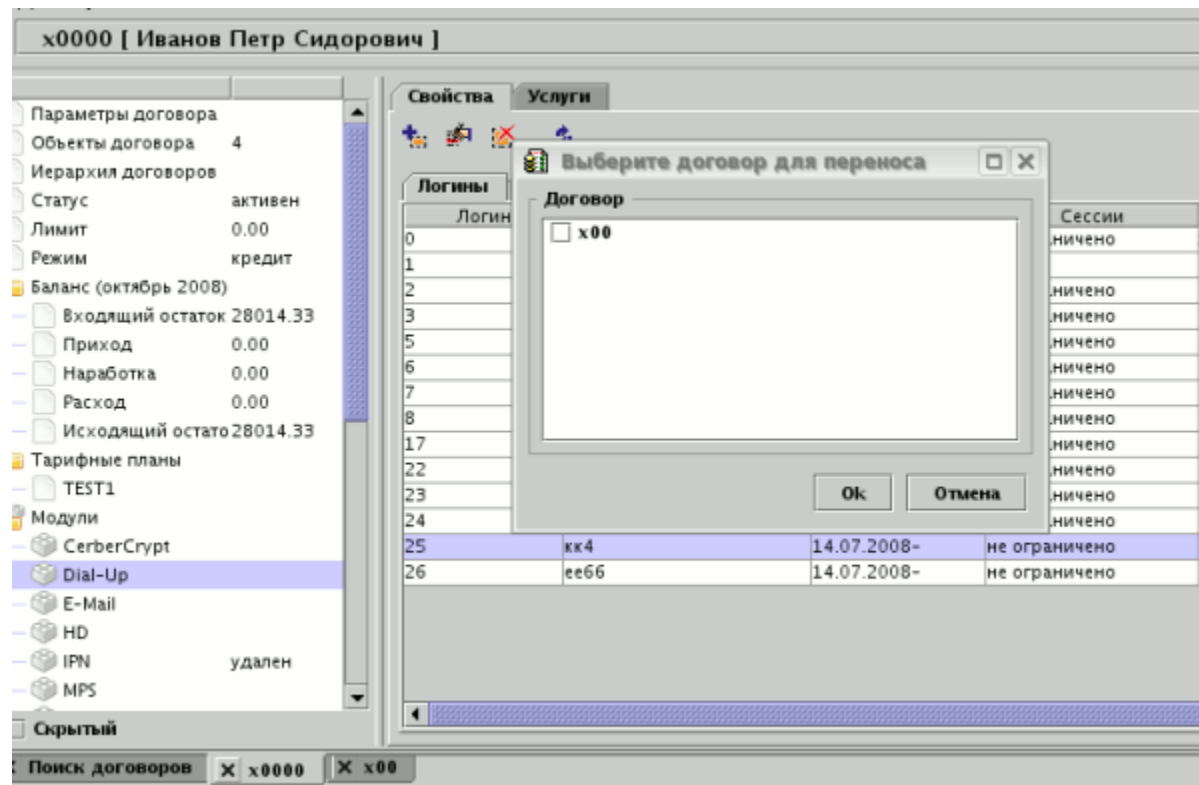


Перенос логинов

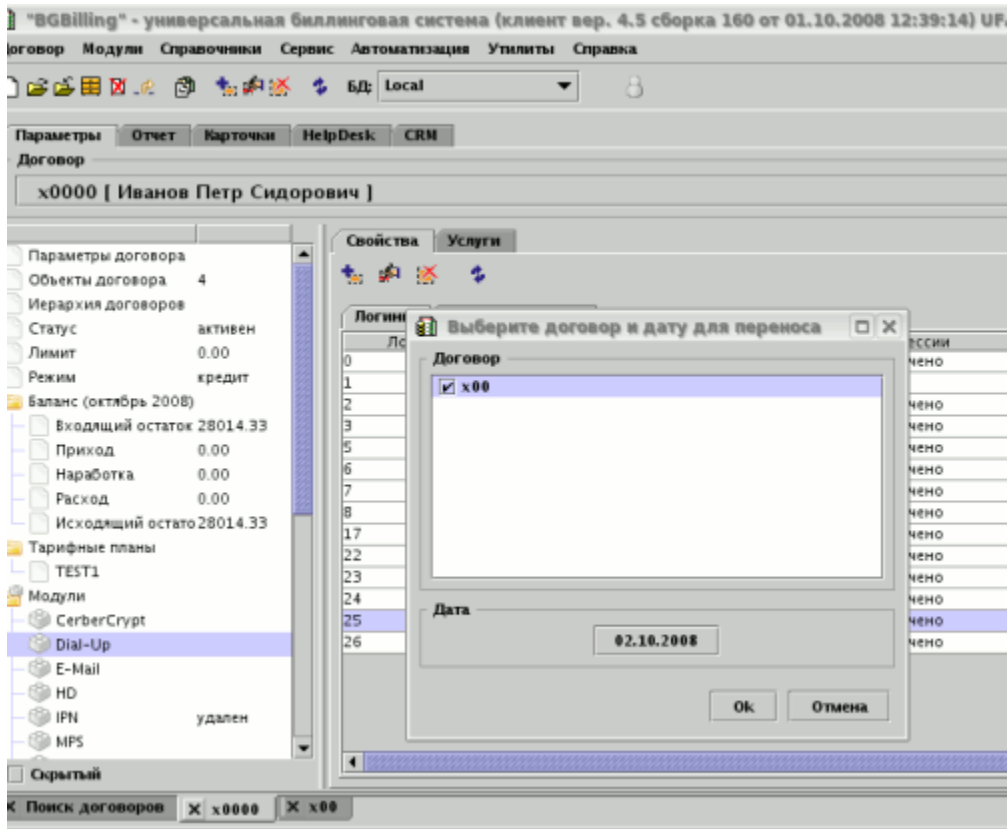
При выборе строки в таблице логинов и нажатии правой кнопкой мыши отображается всплывающее меню.



Пункт **Перенести на другой договор** переносит логин из одного договора в другой (карточка должна быть открыта в текущий момент), при этом перемещается и вся наработка по данному логину. При переносе переносится наработка по логину только на период действия логина.



Пункт **Перенести на другой договор с даты** закрывает логин на существующем договоре и открывает аналогичный логин на целевом договоре с другим периодом. Все алиасы и свойства логина копируются. Возможен перенос только логина с неустановленной датой закрытия.



В обоих режимах перераспределение наработки между договорами происходит автоматически, но для этого должен быть запущен планировщик заданий.

Настройка тарифных планов DialUp

В один момент времени на договоре может действовать только один тарифный план, включающий в себя поддерево экземпляра модуля DialUp. Если у вас не было тарифного плана, создайте его, создайте для него поддерево, либо расширьте от другого тарифа. Как это сделать можете прочесть [здесь](#). Там же описана логика работы тарифных деревьев и поведения стандартных узлов тарифных деревьев, общих для всех модулей. Логика поиска тарифа соответствует [Алгоритму 1](#).

После того как вы откроете дерево, в нем должен отобразиться узел со значком модема и названием экземпляра модуля DialUp.

В тарифном запросе модуля DialUp передаются следующие параметры:

- код потребляемой услуги;
- время момента потребления.

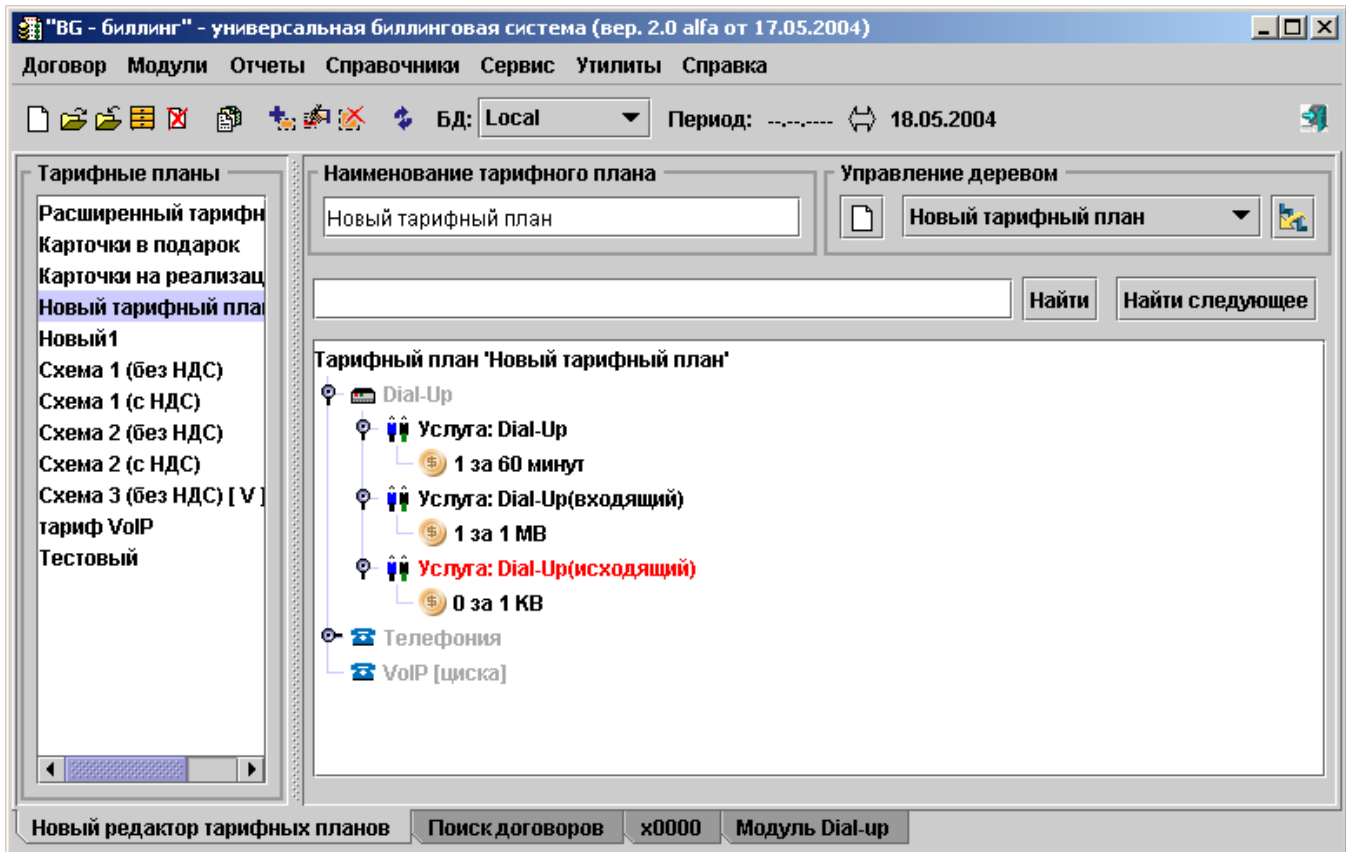
В ответе возвращаются:

- стоимость услуги (обязательно);
- признак акцепта (устанавливается узлом **Стоимость услуги**);
- параметры соединения (опционально);
- действия, которые нужно выполнить с соединением (опционально).

Дополнительные примеры тарифных планов модуля DialUp доступны на нашем [Wiki](#).

Простейший тариф

Взимается 1 рубль за 60 минут нахождения в сети, 1 рубль за входящий мегабайт, исходящий трафик - бесплатный. Добавьте в узел поддрева модуля DialUP узлы типа **Услуга**. В эти узлы добавьте узлы типа **Стоимость услуги**, в которых определите сколько рублей за сколько единиц и чего вы хотите взимать.



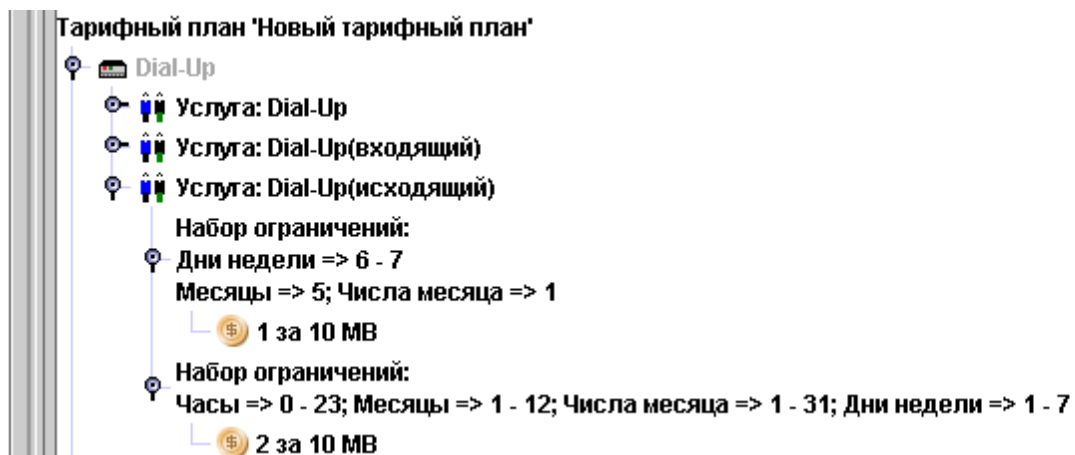
Будьте внимательны при определении цен. Программа никак не отреагирует, если вы добавите в услугу, соответствующую входящему трафику, цену 1 рубль за 1 час. Однако это будет означать, что за каждые, потреблённые клиентом, 3600 (60*60 секунд) байт будет снято 1 рубль, что вас вряд ли устроит. Для программы килобайты, мегабайты и минуты - всего лишь множители.

Узел **Стоимость услуги** при получении запроса проставляет в ответную часть параметры **cost** - стоимость, **divisor** - делитель объёма и, опционально, **costType** - тип услуги. Делитель определяет за какой объём услуги установлена цена (МБ, КБ, часы). Тип услуги используется в детализациях по тарифу, см. далее.

Должны быть определены цены всех услуг, описанных в конфигурации NASA для данного соединения. В данном случае, если не указать стоимость услуги **Dial-Up(исходящий)**, которая указана в конфигурации NASA, то при авторизации будет выходить 11 ошибка **Цена не найдена**.

Разделение стоимости по времени суток

Если вам необходимо варьировать цену по времени, можете воспользоваться узлами **Временной фильтр** и **Фильтр по типу времени**. Ниже приведён пример, как можно задать другую цену в выходные дни и 1 мая.



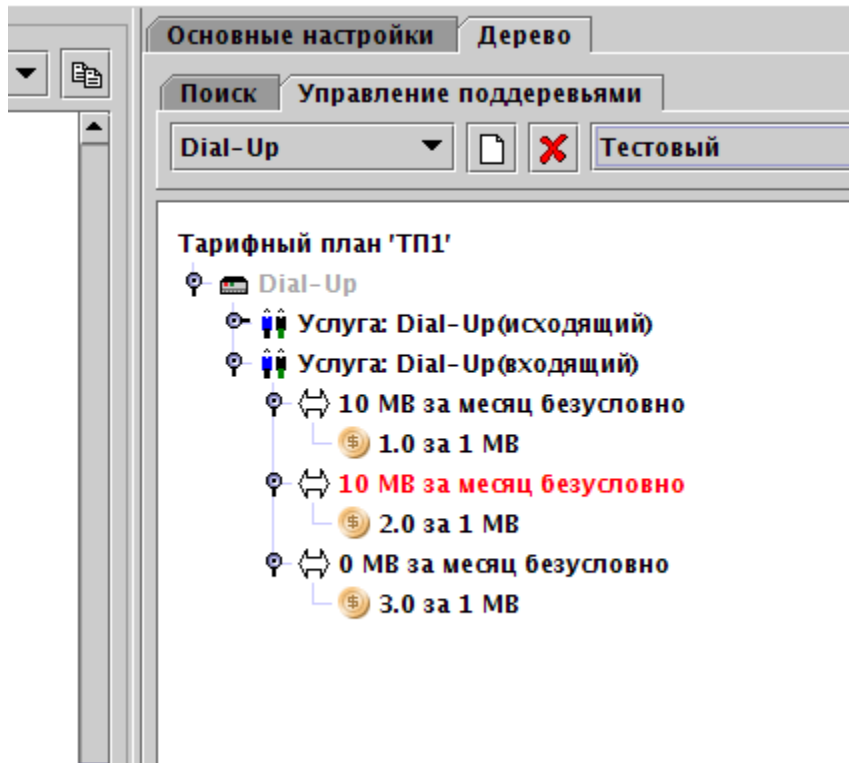
Учётные периоды

Функционал учётных периодов позволяет предоставлять услугу периодами, отвязанными от календарных месяцев. Например, безлимитный доступ на 30 дней от первого входа с однократным списанием средств. Учётный период активируется скриптом при входе абонента, если на текущий момент у него нет активированного периода. Для активации периода генерируется событие BGBS-скрипта **Запрос учётного периода**. Если скрипт договора не обрабатывает это событие, то предполагается работа без учётных периодов и аутентификация происходит успешно. Если скрипт есть и обрабатывает событие, то он либо возвращает новый учётный период, производя необходимые списания и т.п., либо возвращает ошибку активации периода. Примеры подобных скриптов вы можете найти на [Wiki](#).

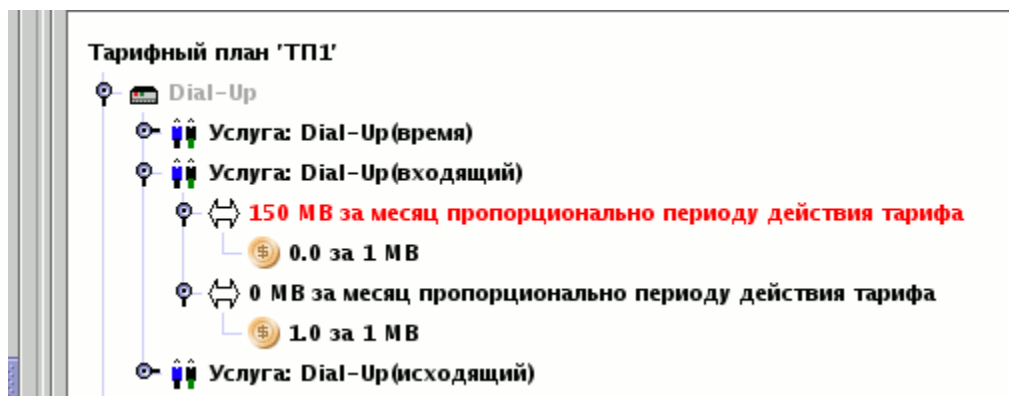
Для использования учётных периодов необходима реализация BGBS-скрипта его активации. Договор может работать либо в обычном режиме, либо в режиме учётных периодов.

Зависимость стоимости от объема

Рассмотрим пример, когда первые 10 МБ входящего трафика будут идти по 1 руб., следующие 10 МБ - по 2 руб., а оставшиеся - по 3 руб.



Совершенно аналогично можно создать тарифный план с предоплаченным трафиком. В первом узле типа **Диапазон** указывается нулевая стоимость. Для начисления за предоплаченный трафик в модуле абонплат устанавливается абонентская плата.



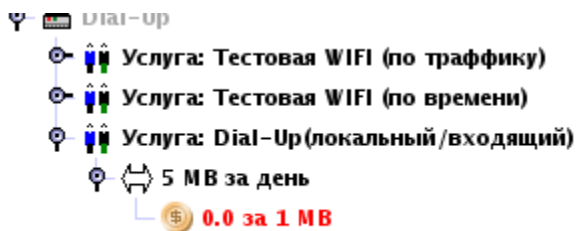
Узел **Диапазон** может быть **за день**, **за месяц**, либо **за учётный период** и работать в режимах: **безусловно**, **пропорционально периоду разрешённой услуги**, **пропорционально периоду действия тарифа**. Разрешённые сочетания параметров и нюансы поведения узла приведены в таблице ниже. **Квота базовая** - это указанное в узле количество услуги. **Квота** - это используемое при тарификации для сравнения количество услуги.

Таблица. Логика работы узла

За период	Режим	Оцениваемый объём услуги	Квота
за день	безусловно	Объём услуги за сутки.	Квота = Квота базовая.
за месяц	безусловно	Объём услуги за период действия тарифа в месяце.	Квота = Квота базовая.
за месяц	пропорционально периоду разрешённой услуги	Объём услуги за период действия тарифа в месяце.	Квота = Квота базовая * (Количество дней с разрешённой услугой в месяце / Количество дней в месяце).
за месяц	пропорционально периоду действия тарифа	Объём услуги за период действия тарифа в месяце.	Квота = Квота базовая * (Количество дней действия тарифа в месяце / Количество дней в месяце).

за месяц	пропорционально периоду действия тарифа (с учётом приостановленных статусов)	Объём услуги за период действия тарифа в месяце.	Квота = Квота базовая * (Количество дней действия тарифа в месяце с не приостановленными статусами / Количество дней в месяце).
за учётный период	безусловно	Объём услуги за период действия тарифа в учётном периоде.	Квота = Квота базовая.
за учётный период	пропорционально периоду разрешённой услуги	Объём услуги за период действия тарифа в учётном периоде.	Квота = Квота базовая * (Количество дней с разрешённой услугой в учётном периоде / Количество дней в учётном периоде).
за учётный период	пропорционально периоду действия тарифа	Объём услуги за период действия тарифа в учётном периоде.	Квота = Квота базовая * (Количество дней действия тарифа в учётном периоде / Количество дней в учётном периоде).
за учётный период	пропорционально периоду действия тарифа (с учётом приостановленных статусов)	Объём услуги за период действия тарифа в учётном периоде.	Квота = Квота базовая * (Количество дней действия тарифа в учётном периоде с не приостановленными статусами / Количество дней в учётном периоде).

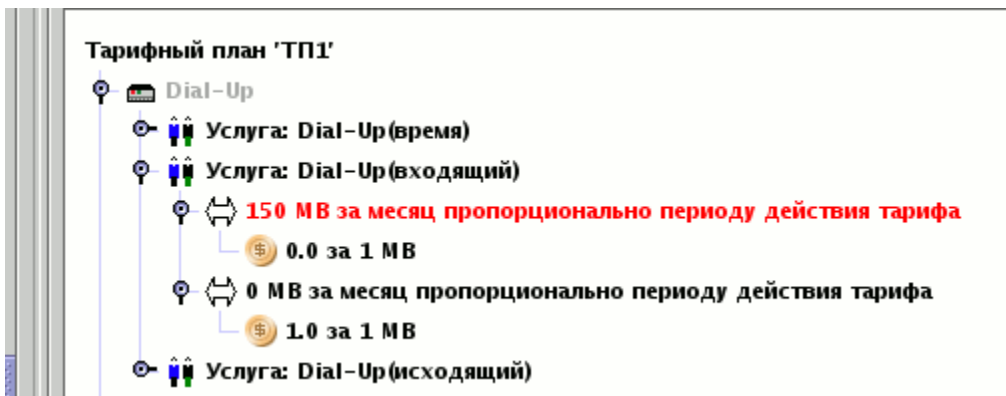
Если клиент исчерпает разрешённые в тарифном плане объёмы, тарификация прекратится и услуга более ему не будет предоставляться. Таким образом, можно создавать ограничивающие тарифные планы. Ниже приведён пример плана, разрешающего потреблять клиенту 5МБ в течение суток.



Логика работы узла **Диапазон** следующая:
















- В запросе узел получает количество услуги **amount**, которое необходимо протарифицировать.
- Оценивается текущее значение счётчика услуги в узле для данного договора и квота, в данном узле может быть протарифицирован объём **MIN (amount, КВОТА - ТЕК. ЗНАЧЕНИЕ)**. Если в данном узле значение счётчика ещё не достигло квоты, то запрос посылается внутрь узла, откуда должна возвратиться стоимость единицы услуги узлом **Стоимость услуги** и быть установлен флаг акцепта. Возможный объём тарифицируется, значение **amount** в запросе уменьшается, увеличивается значение параметра ответа **costAmount**.
- Текущее значение счётчика для договора в узле увеличивается отдельным запросом после запроса получения цены. Передаётся тарифный **init** запрос с протарифицированным объёмом услуги **amount** и узлы последовательно "разбирают" объём наработки, увеличивая счётчики.

По итогам обработки тарифного запроса RADIUS либо процесс переобсчёта реагируют либо на параметр **costAmount** в тарифном ответе, либо на **cost** и **divisor** (устанавливает узел **Стоимость услуги**). Поэтому при оценке услуги диапазонами недопустимо размещать цену вне диапазона. Например, вместо такого тарифа.



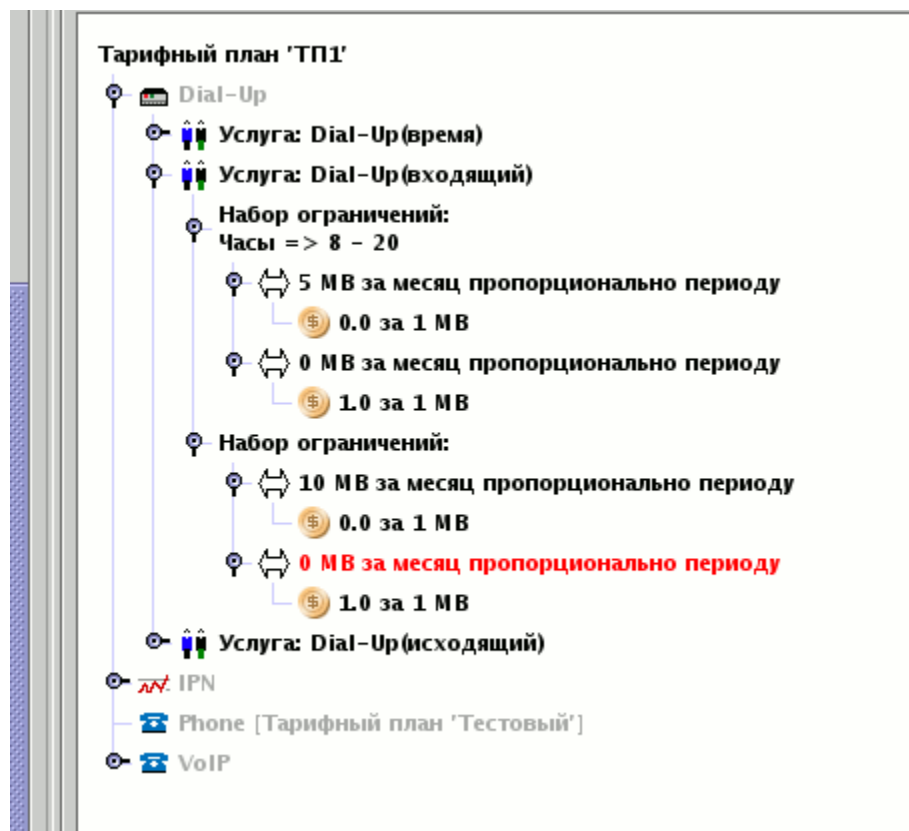
Недопустимо определять цену по умолчанию следующим образом.

Тарифный план 'ТП1'

-  Dial-Up
 -    **Услуга: Dial-Up(время)**
 -    **Услуга: Dial-Up(входящий)**
 -    **150 MB за месяц пропорционально периоду действия тарифа**
 -  **0.0 за 1 MB**
 -  **1.0 за 1 MB**
 -    **Услуга: Dial-Up(исходящий)**

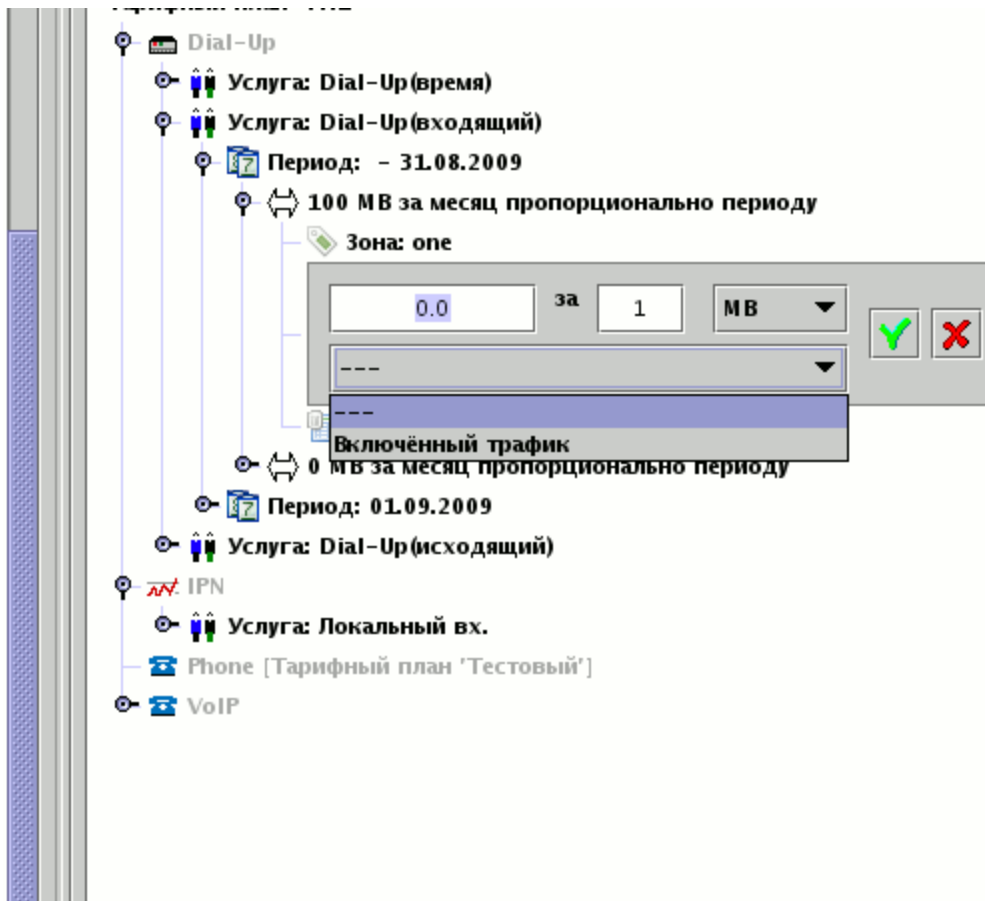
Комбинированные зависимости

Возможно совместное использование различных типов узлов. Например, тарификация по диапазонам трафика дневного и ночного. Первые 5Мб дневного трафика и 10Мб ночного идут бесплатно, далее - по 1 руб.



Детализация по тарифу DialUp

Детализация по тарифу позволяет в счетах модуля бухгалтерии разделить наработку по какой-либо из услуг, посчитанных по различным ценам или условиям. Например, можно вынести в отдельные строки счета бесплатный входящий трафик и платный трафик. Суть метода состоит в определении категорий услуги в конфигурации модуля и указании их в выпадающем списке узла **Стоимость услуги**. В результате каждая протарифицированная единица услуги относится к той или иной категории.



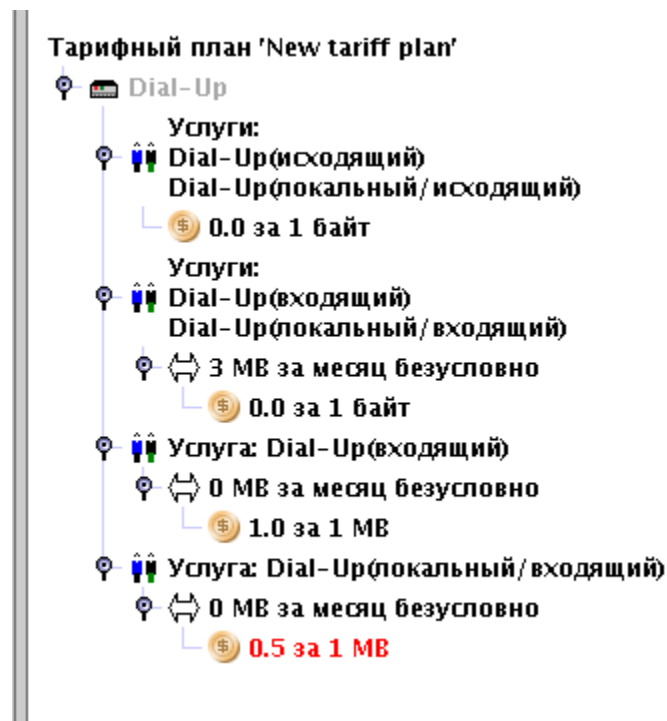
О детализации по тарифу написано подробно [в документации по настройке позиций модуля бухгалтерии](#).

Использование узла "Мультиуслуга"

Вместо нескольких узлов **Услуга** возможно создание узлов типа **Мультиуслуга**. Данный узел аналогичен по функциям узлу **Услуга**, но пропускает в себя запросы цены для нескольких указанных в нем услуг. Использование данного узла целесообразно для:

- указания в одном узле цен для нескольких услуг с одинаковой стоимостью
- создания тарифов с квотами трафика, в которых начальная квота общая для нескольких услуг

Рассмотрим оба этих случая в едином примере:

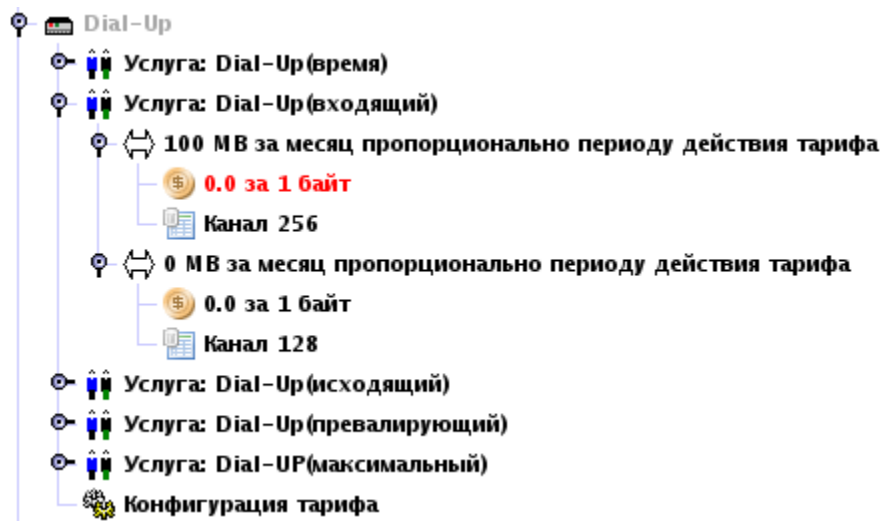


В данном примере стоимость услуг исходящих трафиков равна 0. На услуги входящего локального и внешнего трафиков выделена общая бесплатная квота 3МБ, при превышении которой входящий внешний трафик тарифицируется по 1 руб/Мб, а локальный по 0.5 руб/Мб.

Указание в тарифе свойств соединения

В тарифных планах возможна установка наборов RADIUS-атрибутов, передаваемых в AUTH ACCEPT-пакете. Например, это может использоваться для заужения канала клиента в зависимости от потреблённого им трафика в течении месяца. Пример подобного тарифа приведён ниже. Наборы передаваемых RADIUS-атрибутов должны быть предварительно [настроены](#) в конфигурации модуля.

При получении тарифного запроса узел передаёт в ответную часть набор атрибутов, который должен быть установлен. С использованием флага **перекрыть остальные** (см. далее) узел может удалять из ответной части установленные ранее наборы атрибутов.



Недостаток приведенного примера в том, что атрибуты передаются только при авторизации. При прохождении порога 100Мб по ходу соединения ничего не произойдет, до следующей авторизации клиент будет работать на большой скорости.

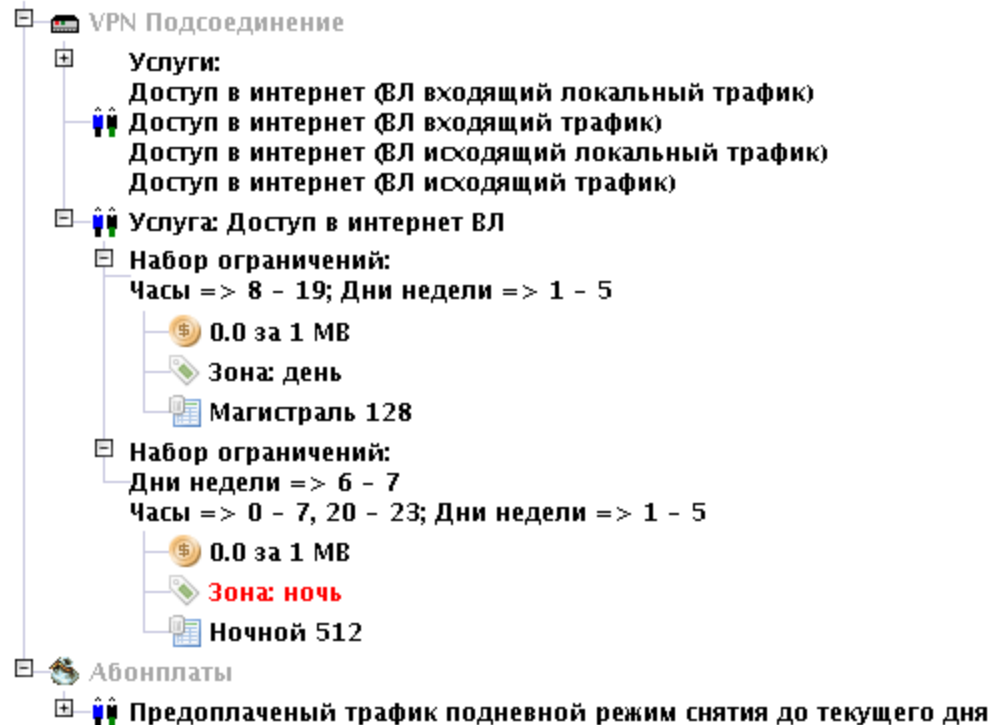
Для выполнения каких-либо действий при необходимости изменения свойств соединения используются зоны. Нижеприведенный пример разделяет весь потребленный объем на две зоны, при этом переход из одной зоны в другую вызывает разрыв соединения и, как следствие, повторную авторизацию. Разрыв необходимо настроить в обеих зонах, т.к. в начале месяца происходит переход в первую зону.

Помимо разрыва соединения при смене зон возможна отправка CoA запроса, в том случае, если в качестве инспектора соединений для NASa указан [PoD инспектор](#), а NAS поддерживает PoD и CoA-запросы. В этом случае нужно выбрать действие **Отправить CoA**.

При установке в поле **Событие BGBS** любой строки при переходе на данную зону генерируется событие **Тарифная зона изменилась**, которое можно обработать скриптом поведения. При этом строка передаётся в событии.

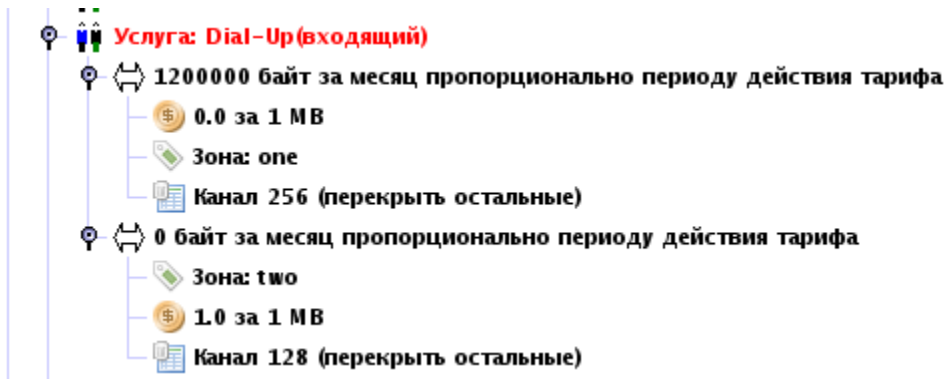
При прохождении тарифного запроса узел **Зона** предоставляет свой идентификатор в ответную часть. RADIUS отслеживает смену зоны по ходу соединения и выполняет те действия, которые указаны в новой зоне. Также данный узел можно использовать для разрыва соединений при переходе между разными временами суток с различной скоростью канала. Сброс должен быть установлен в обеих зонах.

Тариф 'дневной/ночной анлим' для договора 'ИР/ВЛ-23790'



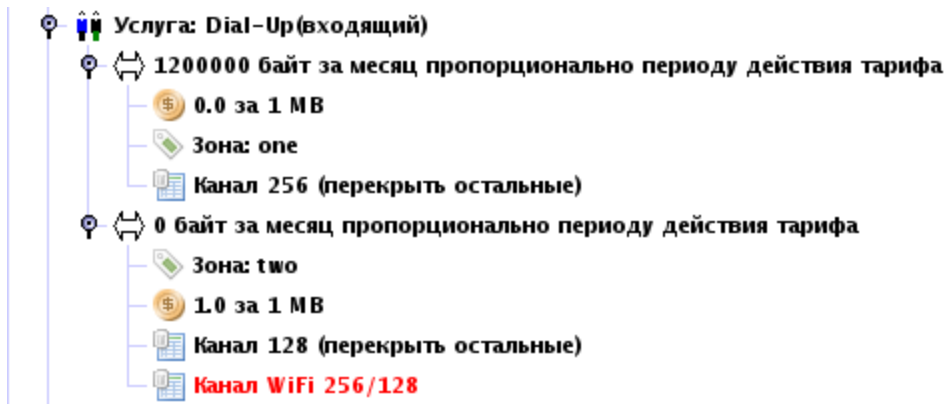
Важно понимать, что смена зоны происходит только при очередном обчёте, который, в свою очередь, при Update режиме тарификации происходит по Accounting Update пакету. В случае, если зоны изменяются по времени, они должны быть установлены в узле услуги типа "Время". В случае, если они меняются от объема потребленного трафика - в узле услуги типа "Трафик". Это связано с алгоритмом тарификации, а именно с тем, что трафик всегда считается потреблённым единой "порцией" в момент предыдущего Update-пакета. Время же при переходе часа обчитывается двумя "порциями": в момент предыдущего Update-пакета считается потреблённым время до границы часа, и на начало часа учитывается потребление остатка времени. Поэтому, если вы установите зоны, которые должны изменяться по времени в услугу по трафику, то реально действие произойдёт при втором Update-пакете после наступления нужного часа.

Обратите внимание на флаг **Перекрыть остальные** в узле **Набор RADIUS атрибутов**. Необходимость данного флага можно рассмотреть на следующем примере тарифа. Пусть при смене зон отправляется CoA-пакет.



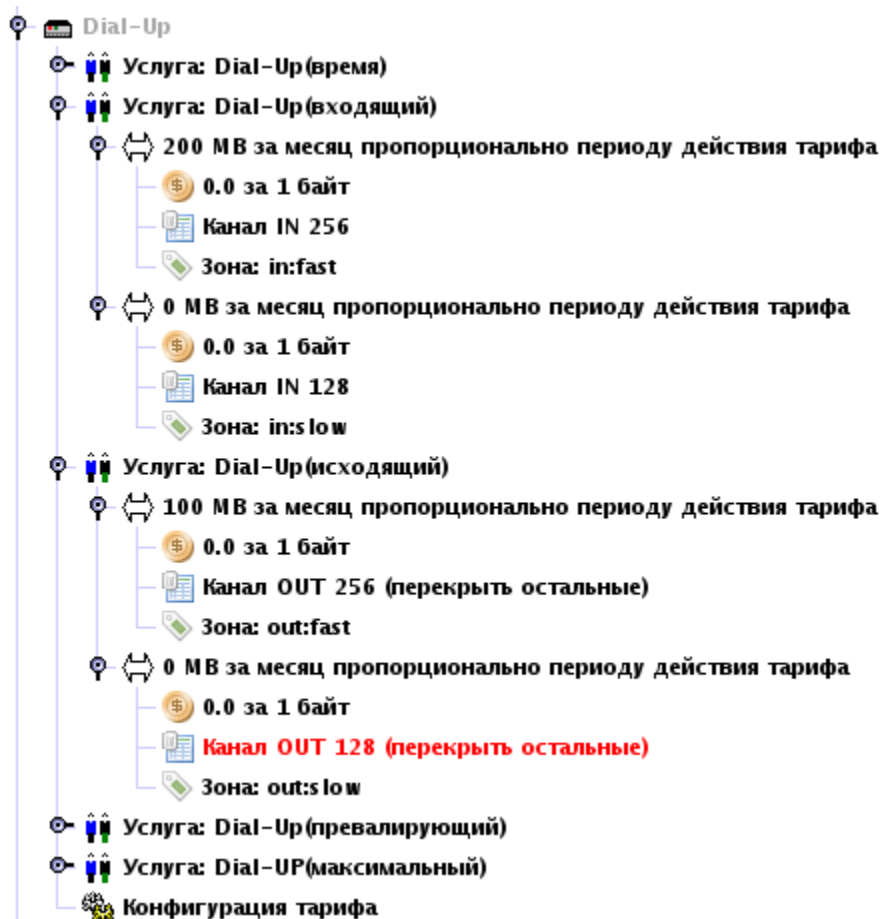
Без установки этого флага набор атрибутов просто добавляется в тарифный запрос. В результате по приведённому выше тарифу, но без установки этого флага при очередном обчёте трафика по Update-пакету в ответе тарифного запроса вернутся два набора атрибутов (часть трафика будет обчитана по 0 руб. а остаток по 1 руб.), которые и будут отправлены в CoA-пакете. При установке же данного флага, последний набор атрибутов удалит информацию о предыдущем в результате будет отправлен только набор **Канал 128**. При изменении зоны по времени наличие данного флага не принципиально по описанным чуть ранее причинам, связанным с различными алгоритмами обчёта времени и трафика.

Вот ещё один пример тарифа, когда после смены зоны отправляются уже два набора атрибутов **Канал 128** и **Канал WiFi 256/128**.



При отправке CoA используются стандартно набор(ы) атрибутов, полученные из тарифа при очередном обчёте. Однако в реальности зачастую необходима также повторная отправка атрибутов логина и релама. В этом случае в конфигурации NASa при настройке PoD-инспектора необходима установка опции `nas.inspector.coa.send.all.attributes=1`. При установке данного флага в CoA-пакете будут отправлены атрибуты, аналогичные отправляемым в Auth Ассерт-пакете: из свойств логина, реалма. Также будут добавлены все наборы атрибутов, полученные из тарифного плана при последнем обчёте.

Изменение зон может быть отслежено только в пределах тарификации одной услуги. Однако, зачастую могут быть ситуации, когда требуется изменять параметры доступа в зависимости от нескольких услуг. Например, в приведённом ниже примере скорость входящего и исходящего канала понижаются независимо. Обратите внимание на префикс с двоеточием перед названием зоны. Это группа зон. RADIUS отслеживает смену зон в каждой из групп и при этом выполняет действия, указанные в новой зоне. Если бы вместо "группа:зона" в данном тарифе использовались просто зоны, то при превышении наработки 200 Мб входящего и исходящем меньше 100 Мб возникало бы противоречие в зонах.



Всвязи с наличием возможности изменения в тарифном плане свойств соединения возникает необходимость обработки ситуации смены тарифного плана по ходу работы сессии. Штатно логика обработки события изменения тарифных планов договора во время активной сессии описана ниже.

При смене тарифа разрывается соединение только, если в результате смены:

1. изменился текущий действующий тариф, его нужно в общем случае переинициализировать, если есть узлы **Диапазон наработки**;
2. правился сам тарифный план в справочнике текущего действующего тарифа, при этом идёт загрузка тарифного дерева заново и, опять таки, нужно переинициализировать узлы **Диапазон наработки**;
3. не стало тарифа на текущую дату.

Также производится разрыв соединения, если в ходе обчёта начал использоваться тариф, отличный от предыдущего (начались новые сутки). Однако этот сброс можно отключить опцией **no.session.break.on.tariff.change=1** в конфигурации экземпляра модуля, котролируя разрыв/CoA, когда нужно зонами в тарифах, т.к. смена зон отслеживается и при переходе на другой тариф.

Уровни

Зачастую необходима возможность управления глобально свойствами текущих соединений. Например, уменьшение пропускной способности каналов активных пользователей, но только в моменты пиковой загрузки внешнего канала. Для этого в BGRadiusDialup реализована возможность сопоставления договорам неких абстрактных чисел - уровней. Если договору не сопоставлен уровень, то он считается нулевым. Сопоставление осуществляется командой **set_levels** на сокет управления RADIUS-сервера с указанием перечня договоров и уровней.

Формат команды: **set_levels**\n<levels>, где <levels> - набор записей <код договора>\t<уровень>\n

Например, передача извне файла с уровнями levels, содержащего уровни для договоров с кодами 4 и 5:

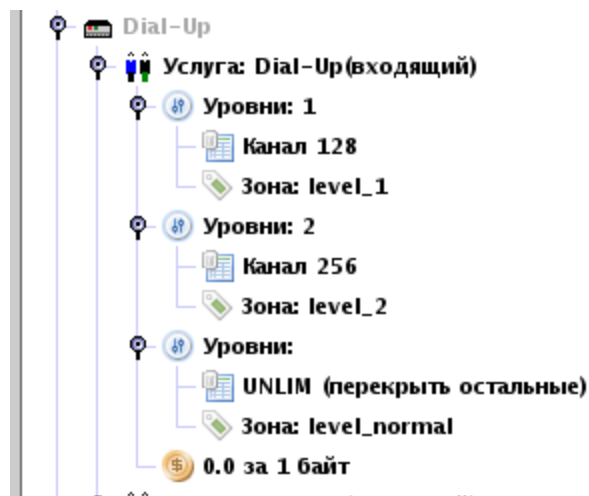
```
4\t1
5\t2
```

Может быть осуществлена следующим образом (BGRadiusDialup установлен на локальной машине и его admin.port=1955):

```
echo ./levels | nc 127.0.0.1 1955
```

После выполнения команды **set_levels** для договоров, уровень которых не был передан, он становится нулевым. Переданные уровни сохраняются в файле levels в каталоге BGRadiusDialup, для возможности восстановления после перезагрузки BGRadiusDialup. Логика формирования уровней полностью отдана стороннему приложению. Пример подобного управляющего скрипта вы можете посмотреть в [Wiki](#).

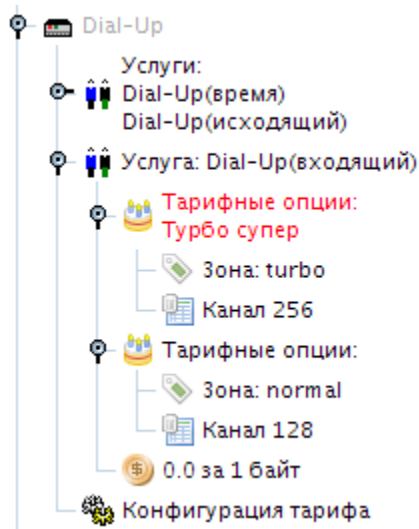
В биллинге уровень передаётся в тарифный запрос при каждом обчёте и воспринимается узлом **Фильтр по уровню**. В соответствии с уровнем могут изменяться зоны и текущие параметры соединения. Ниже приведён пример подобного тарифного плана.



Тарифные опции в DialUp

В зависимости от активированных опций можно менять логику тарифа. Ветка **Тарифные опции** обрабатывает, если в момент обсчета одна из указанных в ветке опций активна. В случае срабатывания одной из веток, последующие этого же уровня не отработают, даже, если указаны те же опции. Отсутствие указанных опций означает, что ветка отработает в том случае, если не было срабатывания ветки **Тарифная опция** выше.

Для того, чтобы сделать повышение скорости на период, можно создать такой тариф:



При отсутствии активированных опций запрос попадет во вторую ветку и отработает зона "normal". При активации опции Турбо супер, запрос попадет в первую ветку и зона сменится на "turbo". Логика работы с зонами можно изучить выше.

Следует учитывать, что в модуле DialUp наработка хранится по часам, поэтому, если в зависимости от тарифной опции меняется цена, то активация тарифной опции не должна быть в режиме "часов (с текущего момента)" (как исключение, можно установить разрыв соединения), также не должна быть в режиме "с текущего часа/дня/недели/месяца", чтобы цена прошедшего времени не изменялась. Если же в зависимости от тарифной опции выполняется только посылка CoA, то режим "часов (с текущего момента)" может быть использован.

Тарифы с переоценкой всего потребленного трафика

Занесение бонусов при смене зон актуально для создания тарифов, при которых трафик переоценивается за весь объем.

The screenshot shows a configuration tree for 'DialUP-VPN'. The tree includes the following items:

- Услуга: время DialUP
- Услуга: Вход интернет трафик DL
 - 200 MB за месяц безусловно
 - 2.0 за 1 MB
 - Зона: one
 - 500 MB за месяц безусловно
 - 1.86 за 1 MB
 - 0 байт за месяц безусловно
 - 1.5 за 1 MB
 - Зона: 3
- Услуга: Вход Локальный Трафик DL
- Услуга: Вход Трафик ОмскАрена DL

A dialog box is open over the '500 MB' item, with the following fields and controls:

- Зона: two
- Разорвать соединение
- Занести приход: Бонус (dropdown menu)
- 28.00 (input field)
- Ok (button)
- Отмена (button)

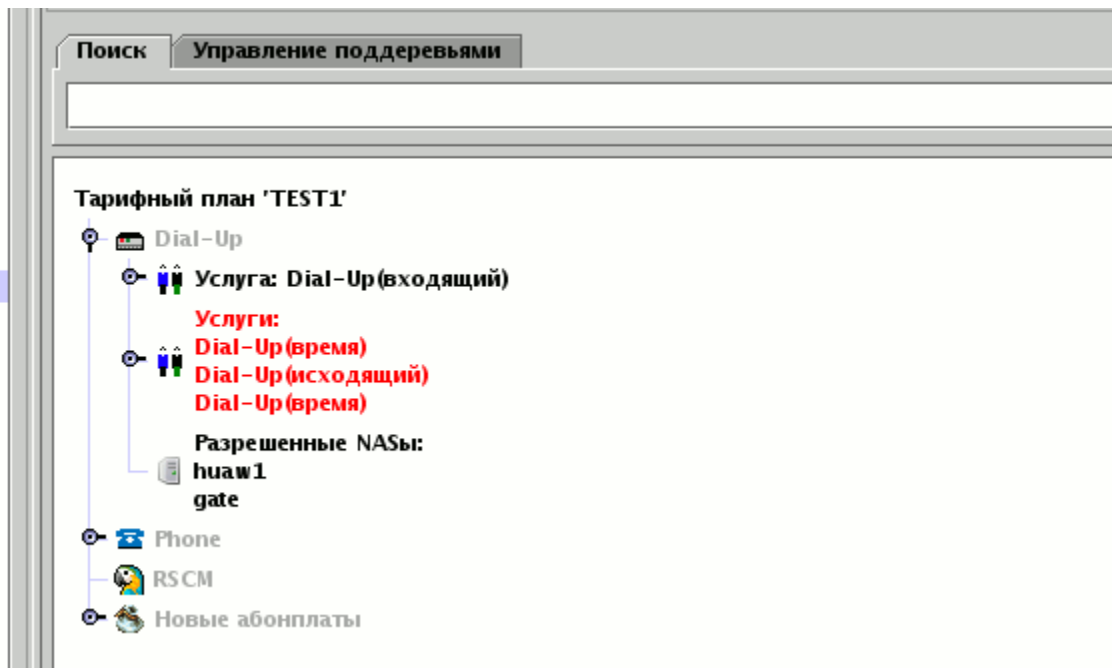
Получив 200 мегабайт по 2 рубля, клиент тратит 400 рублей. После этого тарификация переходит в другую зону, с ценой по 1.86 за мегабайт. Бонус 28 рублей компенсирует траты клиента по 14 копеек на 200 мегабайт, так что теперь можно считать, что весь трафик он полностью получал по 1.86 руб.

И в тарифе для клиента можно указывать, что при наработке свыше 200 мегабайт трафик считается по 1.86. Аналогично происходит переход на тариф по 1.5 руб за мегабайт.

При переобсчете сессий с подобными тарифами платежи не проводятся повторно и не корректируются на правильные. Занесение платежей происходит только непосредственно в момент тарификации.

Ограничение по NASam

В тарифном плане можно устанавливать ограничение по NASam, на которых может работать пользователь. Ограничение производится узлом **Фильтр по NASam**.



Узел "Конфигурация тарифа"

Редактор узла представляет собой простое текстовое поле для введения стандартных конфигураций вида "ключ=значение".

The screenshot shows a configuration editor for 'Dial-Up'. On the left, there is a tree view with the following items:

- Услуга: Dial-Up (время)
- Услуга: Dial-Up (входящий)
- Услуга: Dial-Up (исходящий)
- Услуга: Dial-Up (превалирующий)
- Услуга: Dial-Up (максимальный)
- Услуга: Dial-Up (суммарный)

The main configuration area contains a text field with the following content:

```
#session_detail.delayed.update=1  
#log_session.delayed.update=1
```

Below the text field, there are two buttons: a green checkmark button and a red 'X' button.

Позволяет передавать RADIUS-серверу различные параметры, [влияющие на работу с базой данных](#).

Блокировка отправки атрибутов REALMa

Производится флагом конфигурации **not.send.realm.attributes=1**. Это аналогично установке в свойстве логина отправки только локальных атрибутов.

Фильтр по RADIUS-атрибутам пакета авторизации

Также возможна установка ключей-фильтров по RADIUS-атрибутам пакета авторизации, что позволяет ограничивать авторизацию с [REGEXP](#) фильтром по RADIUS-атрибутам. Фильтры указываются следующим образом:

```
radius.auth.attr.filter.<filter_number>=<attr_name>=<regex>
```

Где:

- <filter_number> - порядковый номер фильтра, просмотр фильтров идёт по номерам;
- <attr_name> - название RADIUS-атрибута, совпадающее с указанным в файле dictionary.xml;
- <regex> - [REGEXP](#) фильтр по атрибуту.

Так, для добавления фильтра по атрибуту Calling-Station-Id из шести цифр необходимо добавить:

```
radius.auth.attr.filter.1=Callind-Station-Id=\d{6}
```

Все фильтры соединяются условием И, т.е. должно выполняться выполнение всех указанных фильтров.

Переобсчёт соединений

Для переобсчёта сессий задним числом при ошибке в тарифных планах, либо неверно установленных планах для клиентов воспользуйтесь вкладкой **Начисление**. Задачу переначисления выполняет планировщик заданий, он должен быть запущен.

В простейшем случае для переобсчёта сессий достаточно выбрать месяц, ввести E-Mail для оповещения о завершении обсчёта и нажать **Запуск**. Левая область вкладки предназначена для переобсчёта сессий, правая - для переначисления за [максимальные трафики](#).

Дополнительно возможна установка фильтра по договору, группам договоров и периоду в днях, когда начинались сессии. По завершению переначисления на указанный E-Mail будет выслано письмо с отчётом и темой письма **DialUp session recalculate**.

Монитор соединений

Вкладка **Монитор** предназначена для мониторинга работы DialUP модуля. Монитор работает в 3х режимах:

- **Ошибки** - отображает ошибки авторизации пользователей;
- **Логи** - логи соединений;
- **Текущие** - текущие соединения по данным из БД.

Для переключения режимов вывода используются соответствующие кнопки.

Режимы **Логи** и **Ошибки** поддерживают фильтр по времени, ряд кнопок в верхней области позволяет смотреть ошибки и логи за 0-1 часа назад, 1-2 часа, либо произвольные сутки.

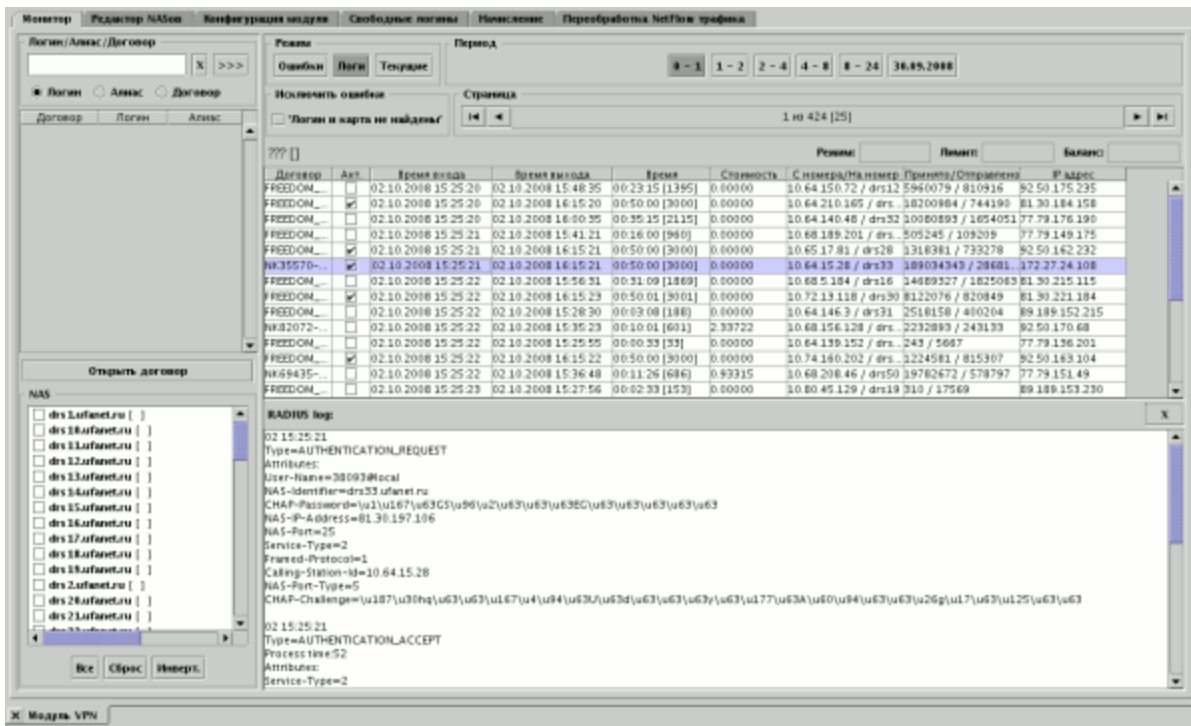
В таблице логов и текущих соединений отображаются время начала и окончания сессий, номер звонящего и вызываемый номер (атрибуты Calling-Station-Id, Called-Station-Id RADIUS-авторизации). Информация о принятых и отправленных байтах отображается на основании данных из RADIUS Update-пакетов.

The screenshot shows the 'Монитор' application interface. At the top, there are tabs for 'Монитор', 'Редактор NASов', 'Конфигурация модуля', 'Свободные логины', 'Платежи', and 'Переработка NetFlow трафика'. Below the tabs, there are sections for 'Логин/Алиас / Договор' and 'Режим' (with buttons for 'Ошибки', 'Логи', and 'Текущие'). A 'Период' section shows a date range from 0-1 to 30.03.2008. There are also buttons for 'Исключить ошибки' and 'Страница' (14 of 743 [25]).

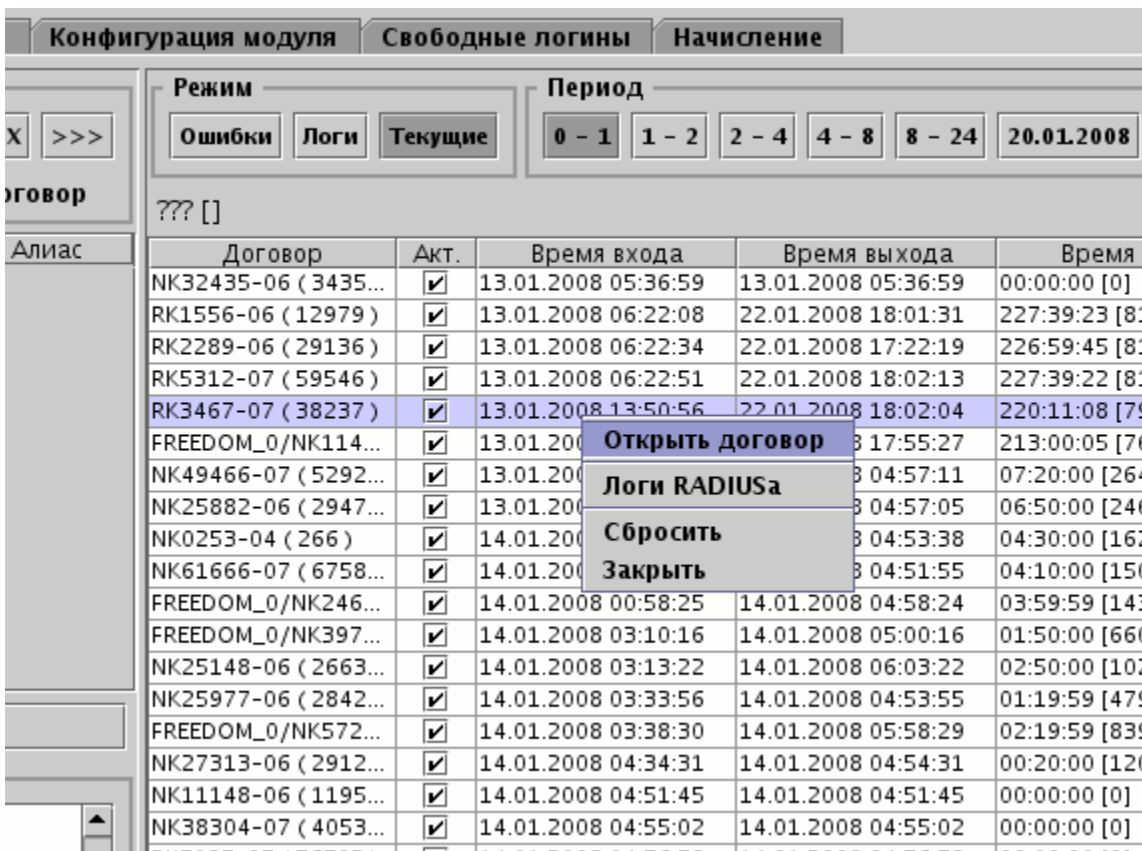
Договор	Статус	Время входа	Время выхода	Время	Стоимость	С. номер/№ номер	Принято/Отправлено	IP адрес	
FREEDOM_	✓	01.10.2008 05:00:00	02.10.2008 16:20:18	35:20:18	0.00000	10.66.114.11 / drs12	4817176148 / 4388	77.79.139.166	
RK6707-0	✓	01.10.2008 05:00:00	02.10.2008 16:20:42	35:20:42	12724	17.77391	10.66.89.6 / drs14	11413665 / 2027449	77.79.176.53
RK28563-	✓	01.10.2008 05:00:01	02.10.2008 16:20:07	35:20:06	12720	0.00000	10.64.207.144 / drs	14772253519 / 4165	172.27.11.138
RK2116-0	✓	01.10.2008 05:00:01	02.10.2008 16:20:01	35:20:00	12720	436.78052	10.67.29.8 / drs5	489865216 / 41869	77.79.158.60
MA0182-	✓	01.10.2008 05:00:06	02.10.2008 16:20:03	35:19:57	12719	31.41061	00187883456 / rdr	20042481 / 3409879	89.189.145.84
RK3844-0	✓	01.10.2008 05:00:10	02.10.2008 16:20:24	35:20:14	12721	45.87314	10.66.157.12 / drs44	30383424 / 9608012	89.189.135.234
FREEDOM_	✓	01.10.2008 05:00:11	02.10.2008 16:20:23	35:20:12	12721	0.00000	10.68.158.124 / drs	116882421 / 15963	92.50.169.175
RK6949-0	✓	01.10.2008 05:00:11	02.10.2008 16:21:15	35:21:04	12726	218.23859	10.81.110.2 / drs31	337679676 / 78431	77.79.171.2
FREEDOM_	✓	01.10.2008 05:00:14	02.10.2008 16:20:52	35:20:18	12721	0.00000	10.64.182.214 / drs	929170776 / 37014	77.79.170.198
FREEDOM_	✓	01.10.2008 05:00:16	02.10.2008 16:20:18	35:20:02	12720	0.00000	10.64.164.21 / drs39	6078938984 / 1408	92.50.180.26
RK2504-0	✓	01.10.2008 05:00:16	02.10.2008 16:20:24	35:20:08	12720	995.79547	10.66.157.4 / drs44	621199826 / 29354	89.189.155.113
RK54808-	✓	01.10.2008 05:00:16	02.10.2008 16:20:12	35:19:56	12719	0.15393	10.64.120.49 / drs23	175804 / 162131	89.189.132.98
FREEDOM_	✓	01.10.2008 05:00:16	02.10.2008 16:20:17	35:20:01	12720	0.00000	10.64.56.27 / drs15	10269517033 / 280	81.30.187.191
FREEDOM_	✓	01.10.2008 05:00:16	02.10.2008 16:20:13	35:19:57	12719	0.00000	10.224.116.157 / dr	2303618679 / 3477	92.50.182.253
FREEDOM_	✓	01.10.2008 05:00:16	02.10.2008 16:20:12	35:19:56	12719	0.00000	10.64.120.6 / drs23	2865928530 / 3608	89.189.158.72
FREEDOM_	✓	01.10.2008 05:00:17	02.10.2008 16:20:27	35:20:10	12721	0.00000	10.64.34.219 / drs37	1313965645 / 3169	92.50.174.109
RK3102-0	✓	01.10.2008 05:00:20	02.10.2008 16:20:42	35:20:22	12722	1.40516	10.66.68.14 / drs14	1810011 / 311091	89.189.145.177
FREEDOM_	✓	01.10.2008 05:00:21	02.10.2008 16:20:07	35:19:46	12718	0.00000	10.225.116.130 / dr	277891752 / 53724	81.30.217.64
FREEDOM_	✓	01.10.2008 05:00:21	02.10.2008 16:20:07	35:19:46	12718	0.00000	10.73.160.31 / drs47	5988566880 / 7193	89.189.145.127
FREEDOM_	✓	01.10.2008 05:00:21	02.10.2008 16:20:16	35:19:55	12719	0.00000	10.64.175.165 / drs	5736527412 / 1265	89.189.145.224
FREEDOM_	✓	01.10.2008 05:00:22	02.10.2008 16:20:07	35:19:45	12718	0.00000	10.68.165.13 / drs38	2232975612 / 1590	89.189.144.48
RK2058-0	✓	01.10.2008 05:00:22	02.10.2008 16:20:20	35:19:58	12719	32.41997	10.65.155.46 / drs43	203333013 / 2989681	81.30.217.216
RK6119-0	✓	01.10.2008 05:00:22	02.10.2008 16:21:18	35:20:56	12725	193.56847	10.66.89.3 / drs14	225913401 / 89774	89.189.142.206
FREEDOM_	✓	01.10.2008 05:00:24	02.10.2008 16:20:30	35:20:06	12720	0.00000	10.64.182.165 / drs	18203762 / 1823755	77.79.119.1

Сортировка соединений производится по дате начала для режимов **Логи** и **Текущие** и в обратном порядке для режима **Ошибки**. Поддерживается постраничный просмотр, при переключении режима отображается первая страница. Для постоянного мониторинга какой-либо из страниц нажимайте кнопку **Обновить** стандартной панели инструментов.

При двойном клике по строке таблицы показывается RADIUS-лог соединения.



При нажатии правой кнопкой мыши по выбранной строке таблицы появляется контекстное меню, позволяющее как открыть договор, которому соответствует строка лога, либо просмотреть логи RADIUS-соединения.

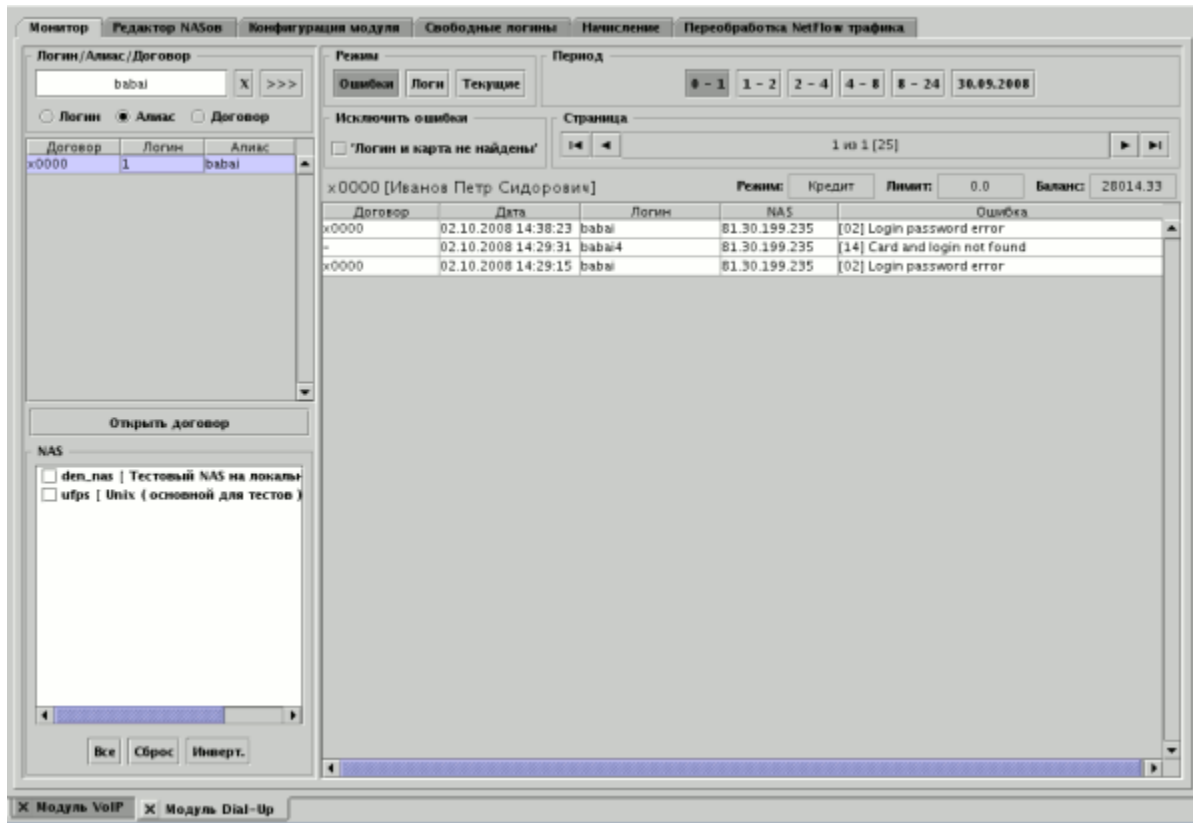


В режиме **Логи** и **Текущие** возможно послать сигнал сброса сессии через контекстное меню. При этом возможно использование команд **Сбросить** (послать сигнал на сброс сессии и ждать STOP-пакета) и **Закрыть** (принудительное закрытие соединения).

Команда **Сбросить** означает попытку принудительного разрыва соединения пользователя и, далее, закрытие его по приходу Stop-пакета с NASa. Команда **Закрыть** предназначена для удаления "Висящих соединений", которые точно не присутствуют на NASe и не тарифицируются RADIUS-сервером. Рекомендуется всегда сначала пытаться сбросить соединение и ждать корректного Stop-пакета, а удаление применять только для "зависших" соединений.

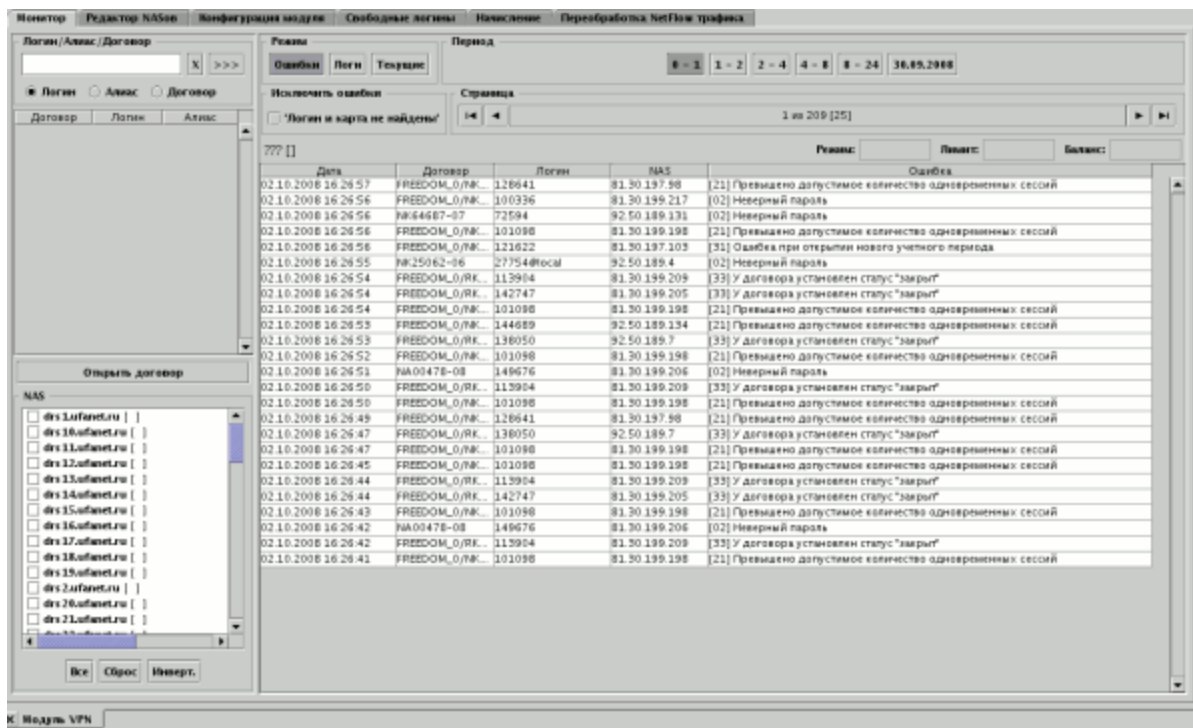
При вызове команд управления сервер биллинга обращается к RADIUS-серверу через порт управления, для чего в конфигурации модуля DialUP должна быть установлена опция **radius.manage=<IP>:<PORT>** где <IP> - адрес RADIUS-сервера, <PORT> - его порт управления. (см. конфигурацию модуля по-умолчанию).

В левой верхней области монитора расположено окно поиска, позволяющее найти клиента по номеру договора, логину, либо алиасу. При выборе строки в таблице с результатами поиска в мониторе будут отображаться логи и текущие соединения только выбранного логина и ошибки выбранного логина, либо неопознанного логина (ошибка **Логин и карта не найдены**). По найденному договору отображается краткая информация над таблицей (название, текущий режим и баланс).



Для всех режимов доступен фильтр по NASам в левом нижнем углу, при не выбранных NASax отображается сводная информация по всем шлюзам доступа.

В режиме просмотра ошибок монитор отображает договор (если он был найден), дату и время, логин (атрибут **User-Name**), IP-адрес NASa и описание ошибки. Для исключения ошибок **Логин и карта не найдены** предусмотрена галочка в фильтре.



Перечень типов ошибок в мониторе и их расшифровка:

Таблица. Таблица кодов ошибок

Код ошибки	Название	Описание
1	Неверный пин-код карты	Карта найдена, но введённый пароль не соответствует её пин-коду.
2	Неверный пароль логина	Логин найден, но введённый пароль не соответствует его паролю.
3	Тарифные планы не найдены	У договора не установлен ни один тарифный план для данного модуля на день авторизации.
4	Ошибка баланса	Остаток баланса договора меньше лимита договора.
6	NAS не найден	RADIUS-пакету не сопоставлен NAS в модуле.
7	Не найден код услуги	В конфигурации NASa не определены коды услуг.
8	Карта просрочена	Истёк период годности карты.
9	Карта заблокирована	Карта не передана дилеру.
10	Карта активирована на баланс	Карта уже активирована для пополнения баланса.
11	Цена не найдена	На одну или несколько услуг, заявленных в конфигурации NASa, не найдена цена в тарифном плане.
12	Ошибка сохранения соединения	В RADIUS-пакете отсутствует атрибут NAS-Port, что помешало сохранить объект соединения.
14	Логин и карта не найдены	Не найдены ни логин, ни карта по данным пакета.
17	Невозможно активировать карту на этом NASe	карта не может быть активирована на данном NASe, её услуга активации не прописана в параметре card.activate.service конфигурации NASa.
18	REALM запрещён	Использование данного REALM запрещено для группы логинов логина.
19	Услуга запрещена	В конфигурации модуля установлена опция check.service=1 , в услугах договора нет одной или нескольких услуг, определённых в конфигурации NASa для данной авторизации.
21	Превышен лимит сессий	У логина установлено ограничение на число одновременных сессий, оно уже исчерпано.
22	Запрещён вход на данный телефон	У логина установлено ограничение на телефон доступа (Called-Station-Id).
23	Запрещён вход в это время	У логина установлено ограничение по времени входа, либо работы.
24	Превышен лимит услуги	У логина установлено ограничение по объёму услуги.

25	Превышен лимит наработки	У логина установлено ограничение по денежной наработке.
26	Доступ заблокирован	Доступ логина установлен в Запрещён .
27	Запрещён вход с данного телефона	У логина установлено ограничение на телефон звонящего (Calling-Station-Id).
31	Ошибка установки расчётного периода	Ошибка получения нового учётного периода из скрипта.
33	Договор не активен	Статус договора не позволяет авторизовать логин
34	NAS запрещён	Запрещена авторизация на данном NASе узлом тарифного дерева Разрешённые NASы .
35	Истек срок жизни карточного договора	У договора, для создания которого активирована карта, закончился период действия.

Возможно изменение текста отображаемой ошибки, для этого в конфигурации модуля указывается:

```
error.message.code.<code>=<>
```

Где **<code>** - код ошибки, **<название>** - отображаемое в таблице название.

Интеграция с модулем "Карточки"

Для того, чтобы ваш модуль DialUp смог работать с карточками необходимо в конфигурации модуля в параметре **card.module.id** указать числовой код модуля Карточки.

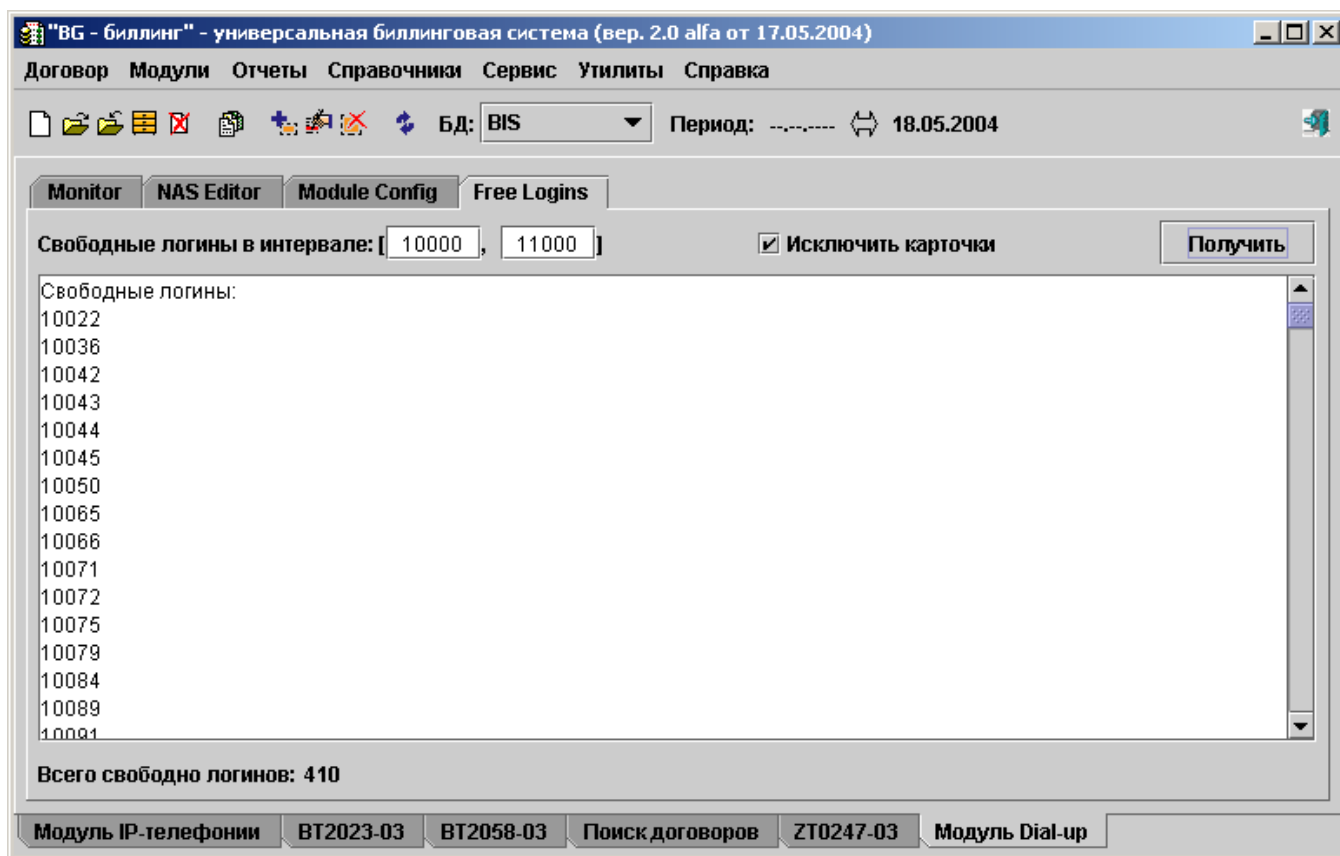
Каждая карточка имеет три параметра: серийный номер (уникальный, не должен повторяться), логин и пароль. При первом звонке клиента-карточника для него создаётся договор. Шаблон имени договора может быть определён в [шаблоне договора](#). Если шаблон имени не указан, то договор именуется К{логин}-YY, где YY - год создания. На этот договор заносится логин и пароль с карты.

Если карта уже активирована другим модулем, например Voip, то в уже существующий договор будет добавлен логин и пароль DialUP модуля. Данный подход позволяет выпускать универсальные карты, по которым можно одновременно получать несколько видов сервиса.

Далее подключение идёт по обычной схеме. Следует уделить особое внимание неперекрыванию диапазонов логинов обычных и карточных. Дело в том, что при обычном создании логины создаются последовательно, карточные же просто переносятся из карты.

Поэтому над диапазоном обычных логинов должен быть запас, т.е. карточные логины должны начинаться с 10000 или выше, чтобы не произошло пересечения.

Чтобы узнать незанятые логины в диапазоне используйте вкладку **Свободные логины** в модуле DialUp.



В верхнем окошке выбирается диапазон, в котором вы хотите выбрать свободные логины, галочка **Исключить карточки** позволяет исключить загруженные карты из диапазона.

Выбрав свободные логины, сгенерируйте для них пароли и серийные номера (они не должны повторяться), после чего можете загрузить в модуль "Карточки".

Возможно разграничение типов карт по NASам, можно разрешить активировать на каждом NASе только карты с определёнными кодами услуг активации, перечислив их через точку с запятой в переменной **card.activate.service** конфигурации NASa (например: 1;4;5). Значение 0 означает, что на данном NASе можно активировать карты любого типа. Если переменная не указана для NASa, то активация карт на NASе запрещена.

Отчёты

Для просмотра отчётов, связанных с пользователем, откройте договор и выберите вкладку **Отчёты**. В нижнем ряде вкладок выберите интересующий экземпляр модуля DialUp.

Трафики, выводимые в отчётах, берутся либо по данным RADIUS, либо из переменной **traffics** конфигурации модуля (см. настройку встроенного NetFlow коллектора). Имеется возможность распечатки и сохранения сессий (в HTML или в CSV-формате) и наработки, а также отправки на E-mail. Чтобы иметь эту возможность, в конфигурации модуля должны быть верно настроены URL-адреса шаблонов преобразования XSL, отвечающих за оформление информации для печати или отправки на Email. Шаблон **xslt.1** - оформление сессий, **xslt.2** - наработок. При необходимости можно модернизировать эти файлы, меняя оформление. Для отправки сессий на E-mail также должны быть настроены опции почты в файле конфигурации сервера биллинга.

Отчёт по сессиям, детализация

Для просмотра сессий выберите интересующий логин (ы), единицы измерения, месяц и интервал дней. Для выбора нескольких логинов используйте кнопки Ctrl и Shift. Доступные логины отображаются в зависимости от указанного периода - отображаются только те логины, период которых пересекается с указанным интересующим периодом. Затем нажмите кнопку с галочкой. При выборе месяца можно проматывать месяцы назад и вперёд с помощью кнопок со стрелками.

The screenshot shows the 'BGBilling' application window. The title bar reads: "BGBilling" - универсальная биллинговая система (клиент вер. сборка от) UFANET. The menu bar includes: Договор, Модули, Справочники, Сервис, Автоматизация, Утилиты, Справка. The toolbar contains icons for file operations and a dropdown menu for the database (БД: local_). The main window has tabs: Параметры, Отчет, Карточки, HelpDesk, CRM. The 'Отчет' tab is active, showing a report for 'babai' in March 2008. The report includes a table of sessions and summary statistics.

Логин	Время входа	Время выхода	Длительность	Стоимость	С номера/На но...	Вх.	
babai	18.03.2008 14:20:47	18.03.2008 14:21:47	00:01:00 [60]	0.00000	543228 /	0	0
babai	18.03.2008 14:25:34	18.03.2008 14:26:23	00:00:49 [49]	0.00000	543228 /	0	0
babai	18.03.2008 14:30:40	18.03.2008 14:32:46	00:02:06 [126]	0.00000	543228 /	0	0
babai	18.03.2008 14:34:54	18.03.2008 14:35:44	00:00:50 [50]	0.00000	543228 /	50000	20000
babai	24.03.2008 16:57:45	24.03.2008 16:57:55	00:00:10 [10]	0.00000	543228 /	80000	50000
babai	24.03.2008 17:09:01	24.03.2008 17:09:11	00:00:10 [10]	0.00000	543228 /	80000	50000
babai	24.03.2008 17:12:25	24.03.2008 17:12:35	00:00:10 [10]	0.00000	543228 /	80000	50000
babai	24.03.2008 17:23:34	24.03.2008 17:23:43	00:00:09 [9]	0.00000	543228 /	80000	50000
babai	24.03.2008 17:28:29	24.03.2008 17:28:39	00:00:10 [10]	0.00000	543228 /	80000	50000

Summary statistics:

Кол-во сессий:	9	Время [сек.]:	00:05:34 [334]	Сумма:	0.00
Трафики (Вх./Исх./Прев./Сум.):	450000 / 270000 / 0 / 0	Трафики (МБ) (Вх./Исх./Прев./Сум.):	0 / 0 / 0 / 0		

Правой кнопкой мыши вызывается контекстное меню выбранной сессии. Возможно просмотреть RADIUS-лог сессии, либо получить детализацию по трафику.

The screenshot shows the 'BGBilling' application window. The title bar reads: "BGBilling" - универсальная биллинговая система (клиент вер. сборка от) UFANET. The menu bar includes: Договор, Модули, Справочники, Сервис, Автоматизация, Утилиты, Справка. The toolbar contains icons for file operations and a dropdown menu for the database (БД: Local). The main window has tabs: Параметры, Отчет, Карточки, HelpDesk, CRM, Документы. The 'Отчет' tab is active, showing a report for 'babai' in February 2008. The report includes a table of sessions and summary statistics. A context menu is visible over the selected session, with options: RADIUS лог, Получить детализацию.

Логин	Время входа	Время выхода	Длительность	Стоимость	
babai	07.02.2008 10:58:15	07.02.2008 10:58:15	00:00:00 [0]	0.00000	543228
babai	07.02.2008 11:00:59	07.02.2008 11:01:09	00:00:10 [10]	0.00000	543228
babai	07.02.2008 11:07:55	07.02.2008 11:08:04	00:00:09 [9]	0.00000	543228
babai	07.02.2008 11:11:03	07.02.2008 11:11:13	00:00:10 [10]	0.00000	543228

RADIUS-лог сессии также можно просмотреть двойным кликом по строке сессии.

Справка

Линия	Время входа	Время выхода	Длительность	Скорость	
hhba	07.02.2008 11:05:15	07.02.2008 11:05:15	00:00:00	000000	>492287
hhba	07.02.2008 11:05:15	07.02.2008 11:05:19	00:00:04	000000	>492287
hhba	07.02.2008 11:05:19	07.02.2008 11:05:24	00:00:05	000000	>492287
hhba	07.02.2008 11:05:24	07.02.2008 11:05:28	00:00:04	000000	>492287

RADIUS log

```
rt:19.133  
User-Name= hhba  
NAS-IP-Address= 81.30.199.235  
NAS-Port= 2  
acct-From-Type=1  
acct-From-Code=2146819 pcc3025  
  
07/11/07/15  
Type=ACCOUNTING_REQUEST  
rt:19.133  
User-Name= hhba  
NAS-IP-Address= 81.30.199.235  
NAS-Port= 2  
acct-From-Code=30000  
acct-Output-Code=80000  
Auth-Status= yes 1  
acct-Session= 11111111
```

IP: 81.30.199.235

07.02.2008 11:05:28 N: x0000

Детализация трафика за сессию

Для получения детализации должны быть произведены [настройки](#). Детализацию предоставляет коллектор модуля IPN на основании бинарных логов трафика. При выборе пункта контекстного меню **Получить детализацию** необходимо указать E-Mail адрес на который будет отправлена детализация.

Стандартная детализация за сессию представляет из себя одно или несколько писем с приложенным архивом detail.zip. Архив содержит CSV-файл с частью детализации. Размер максимального возможного detail.zip в одном письме задается переменной `ipn.collector.detail.max.attach.size` в файле `netflow_ipn.properties`. По умолчанию значение переменной составляет 4000000 байт. Каждое письмо сопровождается темой **Session traffic detail [N]**, где N - увеличивающийся порядковый номер. Последнее письмо данной детализации снабжено темой **Session traffic detail [LAST]** и содержит простейшую аналитику: на какие и с каких адресов был максимальный трафик.

Можно заменить стандартную детализацию, для этого необходимо указать в `netflow_ipn.properties`

```
ipn.collector.detail.class=bitel.billing.server.netflow.ipn.detail.AnalyzedFlowDetailMaker
```

Данный класс создания детализации кроме стандартного csv добавляет в архив detail.zip html-файл с графиком и анализом.



Анализ

Распределение трафика по IP адресам, протоколам и портам, данные представлены в килобайтах.

HTTP - трафик характерный при работе с www ресурсами

SMTP - отправка почты

POP3/IMAP - прием почты

FTP - трафик характерный при работе с ftp ресурсами

UDP named - трафик полученный в результате преобразования имен хостов в IP адрес или наоборот, очень большой трафик по этому сервису характерен при заражении вирусом машин в локальной сети

Данные представлены в отсортированном по уменьшению порядке

IP	Всего		TCP/UDP/ICMP/Другое		TCP: HTTP		TCP: SMTP/POP3/IMAP/FTP		UDP named	
	Вх.	Исх.	Вх.	Исх.	Вх.	Исх.	Вх.	Исх.	Вх.	Исх.
124.40.41.23	157782.40	2959.82	157782.40	2959.82	157782.40	2959.82	0.00	0.00	0.00	0.00
89.208.157.162	67967.47	1333.74	67967.47	1333.74	67967.47	1333.74	0.00	0.00	0.00	0.00
94.100.179.39 content.video12.mail.ru	34408.11	714.05	34408.11	714.05	34408.11	714.05	0.00	0.00	0.00	0.00
88.212.196.109 host09.rax.ru	29448.65	600.36	29448.65	600.36	29448.65	600.36	0.00	0.00	0.00	0.00
93.158.133.213 clip06d.video.yandex.net	15092.48	284.01	15092.48	284.01	15092.48	284.01	0.00	0.00	0.00	0.00
195.208.8.201	14277.80	3079.97	14277.80	3079.97	0.00	0.00	53.00	2621.99	14224.80	457.98

Детализация в этом случае будет приходиться одним письмом, разбиения в данный момент не предусмотрено.

Детализация трафика за месяц

Также можно получить детализацию за месяц по логинам. Получение детализации по логинам возможна только при условии настройки детализации по сессиям.

Ед. измерения

байт Кб Мб Гб

Период

Март 2009

дни: с до

Логин

3633[babai]

Логи

Сессии Нарработка

Логин	Время входа	Время
babai	03.03.2009 14:57:02	03.03.2009
babai	03.03.2009 15:04:42	03.03.2009
babai	03.03.2009 15:10:43	03.03.2009
babai	03.03.2009 15:56:41	03.03.2009
babai	03.03.2009 15:56:59	03.03.2009
babai	03.03.2009 16:07:48	03.03.2009
babai	03.03.2009 17:01:41	03.03.2009
babai	26.03.2009 17:26:17	26.03.2009

Получить детализацию за месяц

Детализация сохраняется в файл. Каталог, куда будет выкладываться детализация на сервере, прописывается в настройках коллектора в файле **netflow_ipn.properties**. Названия каталогов на разных коллекторах могут отличаться, но все коллекторы должны ссылаться на один каталог.

```
ipn.collector.detail.folder=/home/billing/detail
```


Отчёт по наработке логинов

Для просмотра наработки по логинам достаточно выбрать месяц и период.

Параметры Отчет Карточки CRM

Период: Январь 2008
дни с: до

Логин: 711[], 54643[dinka]

Логин: 711[], 54643[dinka]

Логги: Сессии **Наработка** 1 из 1 [25]

Логин	Сессий	Время	Наработка	Трафики
711[]	47	59:29:04 [214144]	49.75	126853146 / 10508923731 / 0 / ...
54643[dinka]	76	234:49:41 [845381]	3540.25	4109408307 / 14240687623 / 0 / ...

Кол-во сессий: 123 Время [сек.]: 294:18:45 [1059525] Сумма: 3590.01

Трафики (Внеш./Внутр./Беспл./VOIP/Радио/WSUS/SaO): 4236261453 / 24749611354 / 0 / 0 / 0 / 0 / 0 Трафики (МБ) (Внеш./Внутр./Беспл./VOIP/Радио/WSUS/SaO): 4040 / 23603 / 0 / 0 / 0 / 0 / 0

IPN VPN Бухгалтерия

Модуль VPN Поиск договоров ZK0007-04

Web-интерфейс DialUp

Через Web-интерфейс модуля DialUp имеется возможность просмотра сессий клиента по логинам, наработки по логинам и учётных периодов логинов. Имеется возможность доступа к Web-статистике по логину и паролю DialUp. Пример подобной настройки смотрите .

Чтобы отредактировать названия пунктов меню в Web-интерфейсе модуля используйте параметры **web.menuItem1 - web.menuItem4** в конфигурации модуля.

По умолчанию все пункты меню названы применительно к DialUp, что может не соответствовать истине, если модуль используется для предоставления VPN доступа.

Просмотр сессий

Логин: Ед. измерения Месяц: Год: Числа месяца: с по

Отчет в виде: [CSV](#), [HTML](#), [CSV/ZIP](#), [HTML/ZIP](#) (После изменения фильтра сначала нажмите "Вывести")

Страница

Логин	Время входа	Время выхода	Длительность сеанса, мин.:сек	Сумма, руб.	С номера/На номер	Вх.	Исх.	Прев.	Сум.	
babai	14.05.2009 17:49:07	14.05.2009 17:49:07	00:00:00 [0]	0.00000	UNDEF / 73511234567	0	0	0	0	<input type="radio"/>
babai	19.05.2009 11:11:08	19.05.2009 11:11:29	00:00:21 [21]	0.00000	UNDEF / 73511234567	0	0	0	0	<input type="radio"/>
Итого сессий: 2			00:00:21 [21]	0.00		0	0	0	0	

Выслать детализацию для выбранной сессии на email:

В отчёте по сессиям также есть возможность получения детализации на e-mail. Для предоставления пользователю такой возможности должны быть установлены [настройки NASов](#).

Начисление наработки за максимальные трафики

Начисление наработки за максимальные трафики осуществляется дополнительно к основной тарификации модуля периодически в ручном режиме, либо, используя планировщик. В конфигурации модуля должны быть определены зависимости услуг, представляющих собой максимальные трафики, например:

```
max.traffic.74=39,40
```

В данном случае услуга с кодом 74 представляет собой максимум между услугами с кодами 39 и 40.

Начисление осуществляется по следующему алгоритму:

1. выбираются все договоры с разрешённой услугой типа "Максимальный трафик" за обчисляемый месяц;
2. выбираются действующие у клиента тарифные планы в период действия услуги, получая наборы: договор - услуга - тариф - период;
3. для каждого пункта набора осуществляется тарификация, причём дата в тарифном запросе передаётся равной последнему дню набора.

Предположим что у нас есть договор X, у которого с 3 сентября по 20 сентября разрешена услуга **Макс. трафик 1**. Предположим также, что в течении сентября у договора был **Тариф1** с 1 по 10 сентября и **Тариф2** с 11 по 30 сентября.

В данном случае будут обсчитаны две позиции:

1. услуга **Макс. трафик 1** вычисленная на период с 1 по 10 по тарифу **Тариф1**
2. услуга **Макс. трафик 1** вычисленная на период с 11 по 30 по тарифу **Тариф2**

Для начисления максимальных трафиков можно использовать ручной режим, используя вкладку **Начисление** модуля.

Монитор Редактор NASов Конфигурация модуля Свободные логины **Начисление**

Обсчет сессий

Группы договоров

B
 C
 E
 L
 M
 S
 V
 VIP-клиенты

Период: Май 2006
дни: с до

Договора: >>> X

Пуск
Выслать протокол на: Запуск

Максимальные трафики
Месяц: Май 2006
выслать протокол на: Запуск

Модуль Отчеты Модуль Счета Модуль Dial-Up Модуль IPN

Для автоматического начисления необходимо настроить задачу **Максимальные трафики DialUP** в планировщике задач. В конфигурации задачи должно быть установлено:

```
mid=< >
```

Как и обсчёт логов IPN, задача берет месяц, отнимая час от текущего времени. Это позволит вам обчислить все трафики по окончании месяца, если запуск задачи будет установлен на 0 часов 55 минут последующего месяца. При этом необходимо настроить сброс сессий пользователей на границе месяца.

Динамическое управление DNS-сервером

Применение динамического DNS решает проблему серверов пользователей с динамическим адресом (например, игровых). По старту сессии BGRadiusDialup может регистрировать имя хоста с адресом, назначенным для сессии клиента, разрегистрация производится по стопу сессии.

Есть три режима работы динамического ДНС:

- 1) выключен;
- 2) по логину входа (имя хоста получается из логина, под которым зашёл пользователь, заменой @ на -);
- 3) по цифровому логину (в качестве имени хоста используется цифровой логин, при входе под реалмом к цифровому логину добавляется тире и реалм).

Под динамически управляемую зону провайдер выделил **vpn.prov.com**. Пользователь с цифровым логином **777** входит в сеть под логином **vasya@local**, ему присваивается имя хоста **vasya-local** и его компьютер будет доступен из сети под именем **vasya-local.vpn.prov.com**. В случае использования режима **по цифровому логину** имя будет присвоено **777-local.vpn.prov.com**, даже если он зашел под алиасом **vasya**.

Имя хоста всегда приводится к нижнему регистру. В случае нескольких сессий под одним логином каждая последующая сессия "перехватывает" хост на себя, разрегистрация хоста происходит по стопу последней "перехватившей" сессии.

Пример настройки DNS сервера

Рассмотрим по шагам настройку динамической зоны совместно с DNS-сервером **bind**. Для примера взята зона **dyn.ufanet.ru**, NS сервер - **ns2.ufanet.ru**, **81.30.199.70** - IP-адрес RADIUS-сервера.

1) Запустить утилиту **rndc-confgen** для генерации случайного ключа.

```
[root@shamil ~]# rndc-confgen
# Start of rndc.conf
key "rndckey" {
    algorithm hmac-md5;
    secret "T92pAc/L6HM4tQUPi/vu+Q==";
};

options {
    default-key "rndckey";
    default-server 127.0.0.1;
    default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndckey" {
#     algorithm hmac-md5;
#     secret "T92pAc/L6HM4tQUPi/vu+Q==";
# };
#
# controls {
#     inet 127.0.0.1 port 953
#         allow { 127.0.0.1; } keys { "rndckey"; };
# };
# End of named.conf
```

2) Добавить в **named.conf** записи о разрешении управлением DNS с адреса RADIUS-сервера и описание самой зоны. Ключ скопирован из вывода **rndc-confgen**.

```
key "dyndns" {
    algorithm hmac-md5;
    secret "T92pAc/L6HM4tQUPi/vu+Q==";
};

controls {
    inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "dyndns"; };
    inet 81.30.199.67 port 953
        allow { 81.30.199.70; } keys { "dyndns"; };
};

zone "dyn.ufanet.ru" {
    type master;
    file "dynamic/dyn.ufanet.ru";
    allow-update { key "dyndns"; };
};
```

3) В файле **dynamic/dyn.ufanet.ru** (путь относительно каталога в котором расположен **named.conf**) описать саму зону.

```
$ORIGIN .
$TTL 600      ; 10 minutes
dyn.ufanet.ru      IN SOA  ns2.ufanet.ru. root.ufanet.ru. (
                    20345026 ; serial
                    28800    ; refresh (8 hours)
                    7200     ; retry (2 hours)
                    604800   ; expire (1 week)
                    600      ; minimum (10 minutes)
                    )
                    NS      ns2.ufanet.ru.
$ORIGIN dyn.ufanet.ru.
```


Пример настройки модуля

Для включения динамического DNS, согласно предшествующему примеру настройки DNS-сервера, необходимо внести в конфигурацию модуля:

```
# DNS
dyn.dns.zone=dyn.ufanet.ru.
dyn.dns.resolver=ns2.ufanet.ru
dyn.dns.shared.key.name=dyndns
dyn.dns.shared.key.value=T92pAc/L6HM4tQUPi/vu+Q==
dyn.dns.ttl=600

#
login.parameter.1.name=dialup.dns
login.parameter.1.title=
login.parameter.1.type=5
login.parameter.1.default=0
login.parameter.1.listValue=; ;
```

Опции **login.parameter.*** определяют в модуле параметр спискового типа с именем **dialup.dns**. В БД параметр сохраняется как число от 0 до 2, которым соответствуют текстовые обозначения от **Выключен** до **Логин входа** в примере. Текстовые обозначения могут быть изменены. Опция **login.parameter.1.default** определяет, какое значение имеет параметр по умолчанию. Соответственно, можно включить по умолчанию динамический DNS всем логинам, установив опцию в 1 или 2.

Имя параметра **dialup.dns** является ключевым для BGRadiusDialup, который получает из значения данного параметра режим работы динамического DNS для логина. После настройки модуля необходим перезапуск BGRadiusDialup.

Пример настройки логина

Во вкладке **Параметры** логина должен появиться параметр, созданный в конфигурации модуля. Его переключение изменяет режим динамического DNS для логина.

Клиент имеет также возможность самостоятельно управлять режимом динамического DNS через Web-интерфейс, пункт меню **Управление динамическим DNS** (либо иное название, прописанное в конфигурации модуля).

Настройка автозакрытия соединений

Признаком "висящего" соединения для RADIUSa является его неактивность при наличии активных соединений на этом же NASe. Автозакрытие следует использовать в случае, если закрытие по приходу аутентификации на порт невозможно (например, если в **NAS-Port** идёт код сессии).

Для автозакрытия соединений, "висящих" более часа при наличии активных соединений на этом же NASe, добавьте в конфигурацию NASa строку:

```
drop.sleep.timeout=3600
```

Используйте автозакрытие очень осторожно, т.к. неверно удалённое соединение из RADIUS не обсчитывается сервером. Не рекомендуется ставить таймаут автозакрытия менее 3600 секунд.

Поддержка CallBack

Механизм CallBack нужен в ситуации, когда используется повременная оплата телефонной связи. При соединении CallBack после авторизации клиента на NAsе происходит обратный звонок клиенту.

В конфигурации NAsа опция **callback.support** должна быть установлена в 1.

В этом случае за звонок платит провайдер, а оплата звонков для юр.лиц, в большинстве своём, фиксированная. Чтобы сообщить NAsy о необходимости перезвонить, нужно установить для клиента набор атрибутов RADIUS.

Атрибуты могут меняться в зависимости от оборудования, ниже приведён пример для Ascend Max6060.

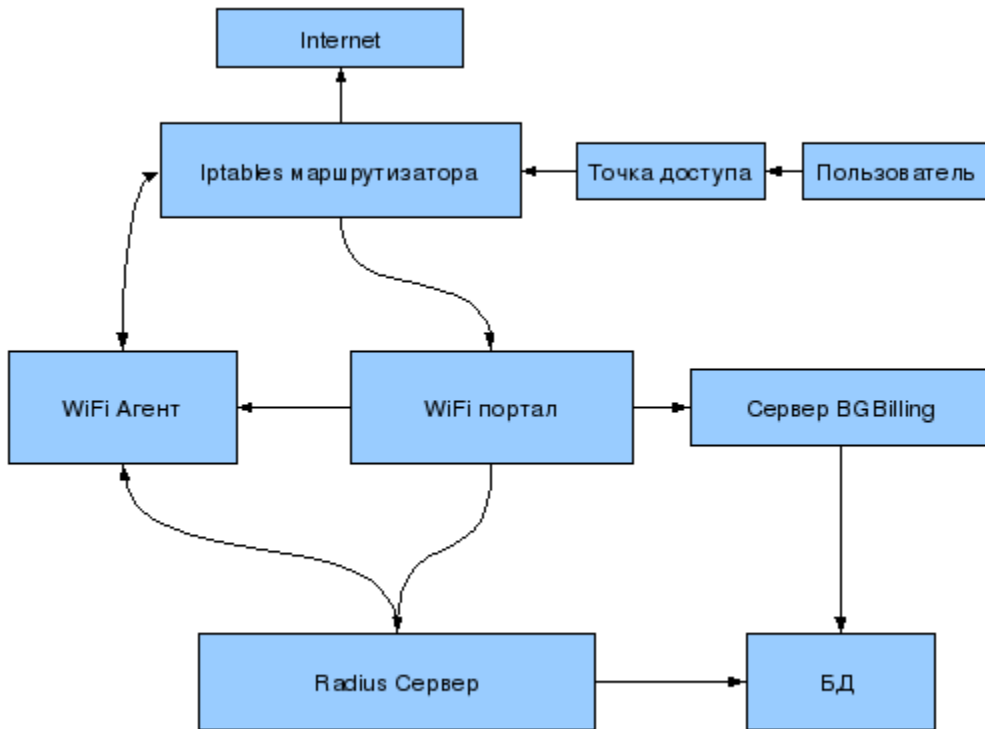
```
Ascend-Assign-IP-Pool = 1
Ascend-CBCP-Enable = 0 (CBCP-Enabled) (, Unix- =1, Microsoft=0)
Ascend-CBCP-Mode = 7 (CBCP-Any-Or-No)
Ascend-CBCP-Trunk-Group = 9
Ascend-Client-Primary-DNS = 212.110.226.2
Ascend-Data-Svc = 42 (Switched-modem)
Ascend-Send-Auth = 0 (Send-Auth-None)
Service-Type = 2 (Framed-User)
```

Для определения набора атрибутов для другого оборудования обратитесь к документации производителя.

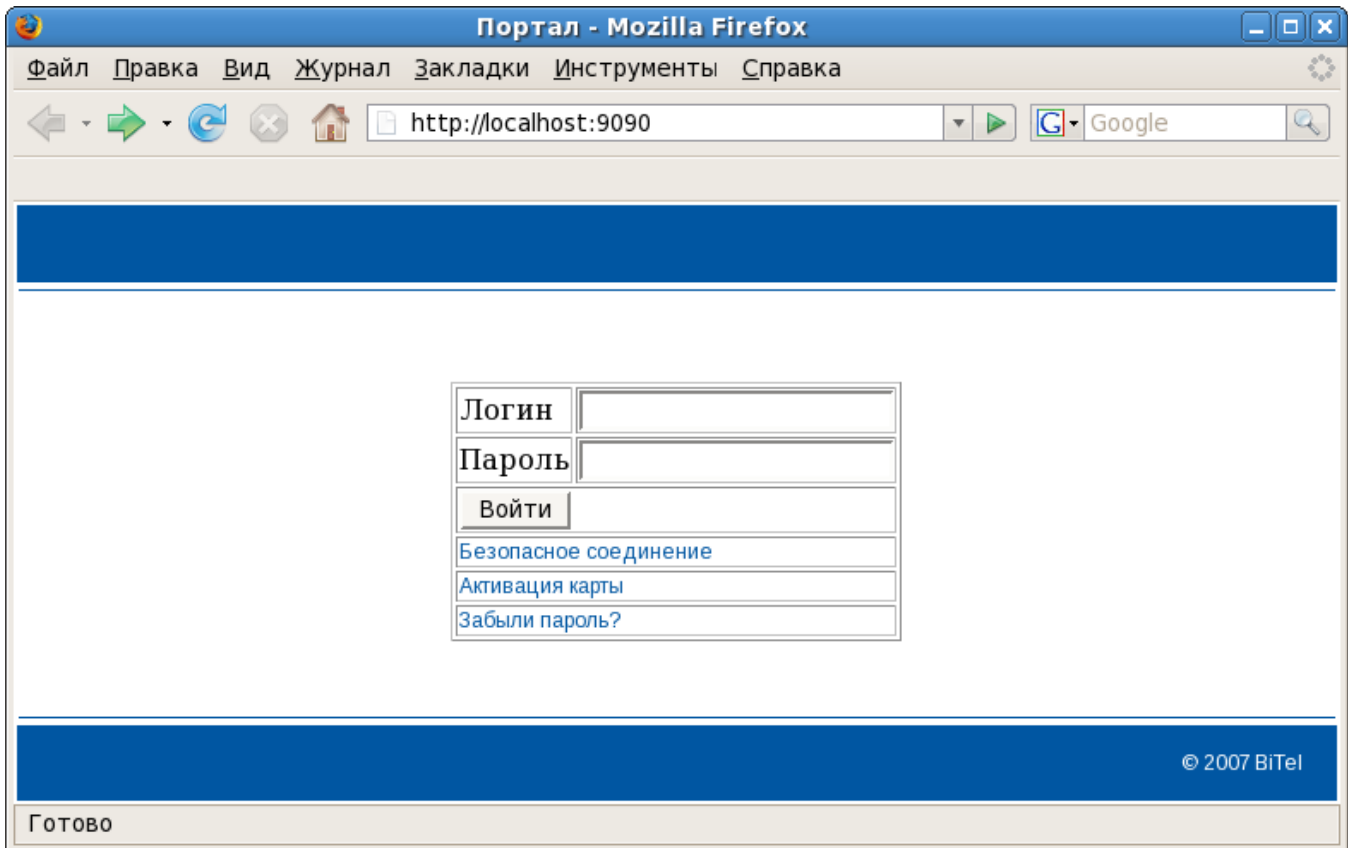
Настройка WiFi-агента для работы с модулем Dialup

Описание WiFi-агента (DialUp)

WiFi-агент предназначен для манипулирования клиентами в WiFi-сетях. Он взаимодействует с сервером BGBilling и RADIUS-сервером для передачи информации о появлении клиента, его уходе и актуальном времени сессии. Также он управляет маршрутизатором. Общая схема взаимодействия может быть представлена на следующей диаграмме :

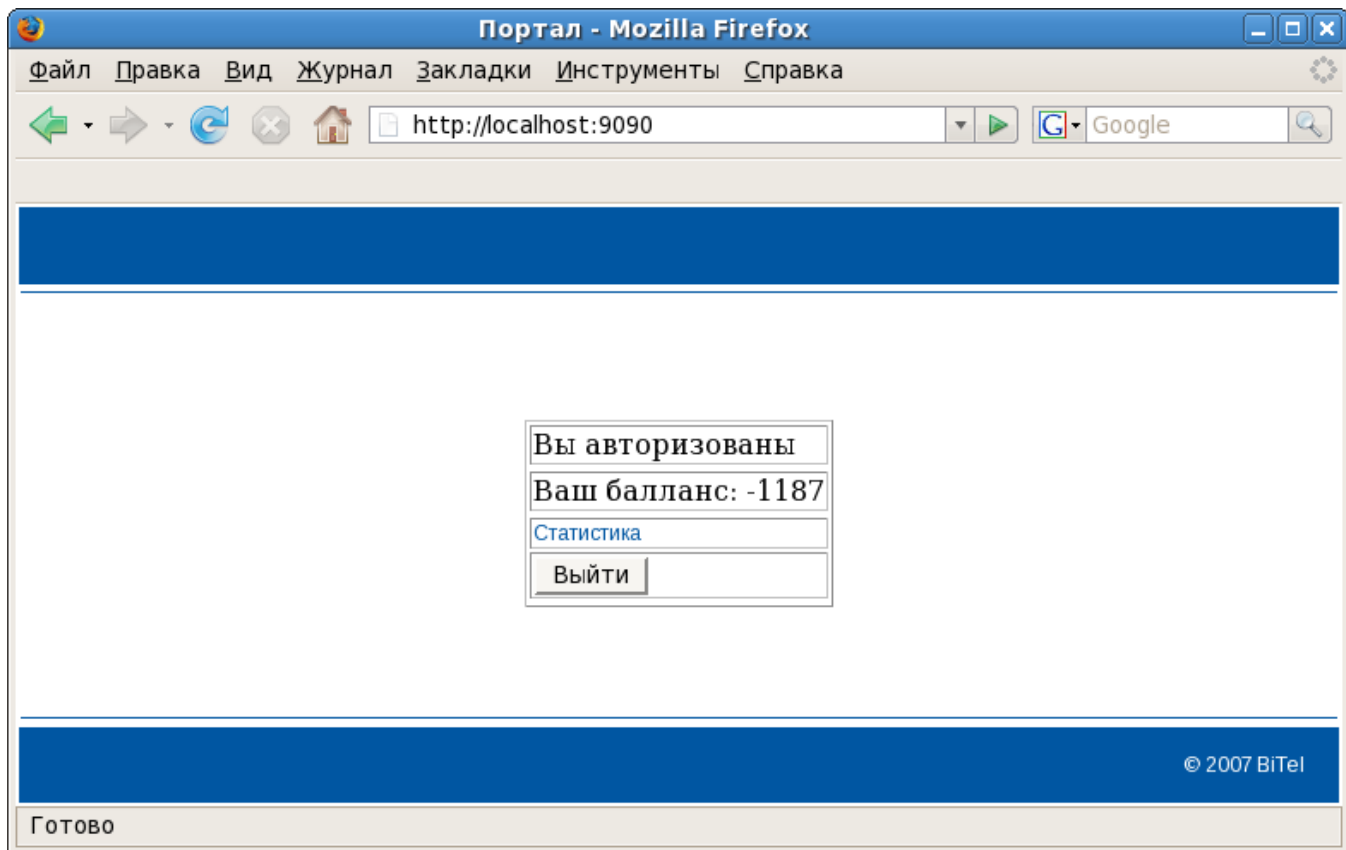


В данный момент WiFi-агент и WiFi-портал реализованы одним приложением и поднимаются на машине маршрутизатора. Это решение работает только для ОС семейства Linux. Пользователь выходит в WiFi-сеть, по dhcp получает ip-адрес с DHCP-сервера, установленного в этой сети, и через точку доступа попадает на маршрутизатор, на котором стоит ОС Linux со службой iptables. Iptables настраивается таким образом, что по умолчанию для клиента закрыты все порты кроме порта, на котором висит WiFi-портал. Все запросы на 80-ый порт перебрасываются на страницу портала, на которой пользователь вводит логин и пароль для доступа к WiFi-сетям. Страница авторизации имеет такой вид :



Пользователь вводит логин/пароль.

Далее Портал обращается к RADIUS-серверу и шлёт на него авторизационный radius-пакет. Если авторизация проходит успешно (в ответ получен radius-пакет подтверждения авторизации), то портал шлёт об этом запрос WiFi-агенту. WiFi-агент, в свою очередь, шлёт стартовый radius-пакет на RADIUS-сервер и меняет правила iptables, разрешая пользователю выйти в интернет. После успешной авторизации пользователя перекидывают на сервисную страницу :



На эту страницу пользователь может попасть всегда (даже после авторизации), зайдя на <http://192.168.184.39:9090> (путь зависит от настроек). Здесь он видит свой баланс, ссылку на первоначальный ресурс, который он набирал до того, как его перенаправили на страницу авторизации. Также есть альтернативная ссылка на эту страницу через <https>.

Сессия клиента может быть завершена следующими способами :

1. Клиент зашёл на страницу портала и нажал кнопку "выйти";
2. Клиент исчерпал свой баланс и RADIUS-сервер послал сообщение WiFi-агенту о завершении работы клиента с данным ip;
3. WiFi-агент, периодически проверяющий (в текущий момент через каждые 60 сек) состояние счётчиков iptables, определит, что клиент был неактивен в течение некоторого времени;

В случае завершения работы (не важно каким из выше перечисленных способов) WiFi-агент пошлёт Stop-пакет Radius-серверу и удалит разрешающие правила iptables для нужного ip-адреса.

С точки зрения модуля DialUp WiFi-агент выступает как NAS. В текущий момент он поддерживает следующие возможности :

1. Режим работы NAS в режимах UPDATE и CHECKER (dialup.workmode);
2. Тарификация с помощью любых тарифов (по времени, по трафику), которые можно завести в системе BGBilling и которые поддерживаются модулем DialUp. Для учёта трафика на маршрутизаторе нужно поднять [NetFlow-агент](#) (описание настройки Radius-сервера [тут](#)). Также клиент имеет возможность выбора [REALM](#)ов сразу на странице авторизации;
4. Доступ к статистике клиента и все возможности, которые предоставляет модуль DialUp для Web-интерфейса и клиента BGBilling (мониторинг сессий и т.п.);
4. Возможно в зависимости от атрибутов, переданных RADIUS-сервером;
5. Вызов внешних скриптов при входе и выходе пользователя в сеть, что позволяет сжимать каналы в зависимости от тарифа или сбрасывать установленные соединения, или выполнять любые другие действия;
6. [Защита WiFi-сети от ARP-спуффинга](#);
7. Функция восстановления пароля через почтовый ящик клиента;
8. Возможность [взаимодействия с модулем Карточка](#), т.е. возможность активации карты с последующей авторизацией по этой карточке в сети WiFi;

9. Возможность клиента работать с https-версией портала.

Установка, настройка и запуск (DialUp)

В текущий момент WiFi-агент и WiFi-портал реализованы одним приложением, которое и называется WiFi-агент. Для его установки его нужно скачать с сайта и распаковать в какую-нибудь папку. Например, в папку /user/local. Далее для простоты будем считать, что мы установили в эту папку.

При установке рекомендуется идти от простого к сложному. Т.е. вначале заставить работать более простую конфигурацию, а потом уже пытаться добавлять дополнительные возможности.

Далее идут шаги, необходимые, для установки.

- 1) Для работы WiFi-агента нужна [Java-машина](#).
- 2) Все настройки агента хранятся в файле dialup_wifi_agent.properties. Вот пример этого файла :

Конфигурация

```
#wifi agent class
wifi.agent.class=ru.bitel.bgbilling.modules.dialup.wifi.DialupWiFiAgent

#mq options
mq.url=failover:(nio://127.0.0.1:61616?socketBufferSize=1000000)
mq.user=bill
mq.pswd=bgbilling

#radius options
radius.auth.host=127.0.0.1
radius.auth.port=1812

radius.account.host=127.0.0.1
radius.account.port=1813

radius.nasId=den_nas
radius.secret=hello

# update- Accounting- (1 )
radius.update.send=1

#billing server options
billing.server.login=admin
billing.server.passwd=admin
billing.server.http.url=http://localhost:8080/bgbilling
#billing.server.https.url=https://localhost:8443/bgbilling
billing.server.moduleId=XXX

billing.server.show.statistics=0
billing.server.password.remind=0

#portal options
portal.http.port=9090
#portal.https.port=9091
#portal.https.keystore.password=bgbilling
#portal.card.link=http://127.0.0.1:8080/bgbilling/pubexecuter?
action=CreateContract&module=card&mid=5&activateType=1

#tariff options
#portal.use.realm=1
#portal.tarif.1.realm=wifi_128_128
#portal.tarif.1.title=128 kb/s
#portal.tarif.2.realm=wifi_256_128
#portal.tarif.2.title=256 kb/s

portal.http.url=http://localhost:9090
#portal.https.url=https://localhost:9091

#wifi agent options
wifi.agent.port=5555
```

```
wifi.agent.port.admin=5556

wifi.agent.radius.live.time=60000
wifi.agent.client.live.time=24000000

wifi.agent.arp.command=/sbin/arp
#wifi.agent.server.https=1

#radius attributes
#wifi.agent.radius.atrubute.1.vendor.code=1111
#wifi.agent.radius.atrubute.1.attr.code=1
#wifi.agent.radius.atrubute.1.type=integer
#wifi.agent.radius.atrubute.2.vendor.code=1111
#wifi.agent.radius.atrubute.2.attr.code=2
#wifi.agent.radius.atrubute.2.type=integer

#dhcp options
#dhcp=1
#dhcp.server.host=192.168.184.254
#dhcp.server.port=67
#dhcp.agent.host=192.168.184.39
#dhcp.minThreadCount=10
#dhcp.maxThreadCount=10

#
#portal.allow.linking=1
# ssi ( shtml )
#portal.use.ssi=1
```

Установите переменную `billing.server.moduleId=<числовой код экземпляра модуля DialUp>`. При необходимости скорректируйте параметры доступа к БД и к [MQ](#)-серверу.

А теперь остановимся более подробно на каждой настройке.

radius.auth.host, radius.auth.port - хост и порт, на котором поднят Radius-сервер авторизации;

radius.auth.port, radius.account.port - хост и порт, на котором поднят Radius-сервер аккаунтинга (возможно у вас поднят один Radius-сервер на одном хосте для авторизации и аккаунтинга, но порты будут отличаться);

radius.nasId - Идентификатор NASa;

radius.secret - секретный ключ для NASa (как настроить NAS для WiFi-агента будет описано ниже);

radius.update.send - Слать update-пакеты на Accounting-сервер. По умолчанию включено;

billing.server.login, billing.server.passwd - логин и пароль доступа к серверу (те же самые, что используются в клиенте биллинга). Нужны для получения баланса пользователя. Данный пользователь должен быть заведён в системе и обладать правами на действия из группы : "Dial-Up->WiFi". О том как администрировать пользователей читайте [здесь](#);

billing.server.http.url, billing.server.https.url - URL сервера биллинга (тот же самый, по которому обращается клиент биллинга). Соответственно для http и https. Если https-соединение не нужно, то не указывайте **billing.server.https.url**;

billing.server.moduleId - код модуля DialUp на BGBilling-сервере, с которым будет работать этот агент;

billing.server.show.statistics - показывать ли ссылку на статистику сервера (1- показывать; 0 - не показывать, стоит по умолчанию). При этом, если в текущий момент клиент работает с порталом по http, то для него эта будет ссылкой на http-версию статистики, а если он работает по https с порталом, то это будет ссылка на https версию статистики;

billing.server.password.remind - показывать ли ссылку на страницу напоминания пароля (1- показывать; 0 - не показывать, стоит по умолчанию). Если этот флаг установлен в 1, то в конфигурации модуля DialUp должен быть указан параметр **mail.contract.param.code** - код параметра договора, в котором задаётся e-mail адрес клиента;

portal.http.port, portal.https.port - порты портала для обычного и безопасного соединения. Если https-соединение не нужно, то не указывайте `portal.https.port`;

portal.https.keystore.password - пароль для https-соединения. Для работы https-соединения нужен файл `.keystore` (основанный на этом пароле), который необходимо положить в папку агента. Как получить этот файл описано [здесь](#). Если https-соединение не нужно, то не указывайте этот параметр;

portal.http.url, portal.https.url - URL портала для http и https-соединений. Это URL, по которому будут обращаться клиенты WiFi-сети, чтобы попасть на сервисную страницу. Если https-соединение не используется, то параметр **portal.https.url** не указывается;

wifi.agent.port - это порт, на котором будет подниматься wifi_agent и RADIUS-сервер будет обращаться к этому порту;

wifi.agent.port.admin - порт для управления WiFi-агентом;

wifi.agent.radius.live.time - время жизни (в миллисекундах) клиента с точки зрения Radius-сервера. Т.е. это время неактивности клиента, через которое Radius-сервер считает, что сессия не активна и останавливает подсчёт по ней. По умолчанию стоит одна минута. Активность клиента проверяется по изменению счётчиков iptables в цепочке WIFI (о том, как настроить iptables читайте ниже);

wifi.agent.client.live.time - время жизни (в миллисекундах) клиента с точки зрения WiFi-агента. Т.е. это время неактивности клиента, через которое WiFi-агент считает, что клиента больше нет, сбрасывает сессию клиента (отсылает stop-пакет Radius-серверу, очищает iptables, вызывает внешние скрипты). По умолчанию стоит 40 минут. Активность клиента проверяется по изменению счётчиков iptables в цепочке WIFI (о том как настроить iptables читайте ниже);

wifi.agent.arp.command - путь к команде arp в ОС Linux. Она нужна для получения mac-адреса клиента и манипулирования arp-таблицами (в случае настройки [защиты от ARP-спуффинга](#)). По умолчанию обычно /sbin/arp;

wifi.agent.server.https - использовать или нет при взаимодействии между WiFi-агентом и сервером протокол https (1 - https; 0-http, стоит по умолчанию);

3) Для работы внешних скриптов нужно настроить файл conf.sh (часть этих настроек дублируется в dialup_wifi_agent.properties). Пусть у нас имеется Linux-маршрутизатор с двумя сетевыми интерфейсами: eth0 - локальный (сеть 172.16.1.0/24, через него выходят клиенты WiFi), eth1 - внешний интерфейс для выхода в интернет (имеет внешний ip - 81.30.199.220).

```
# Java-,
JAVA_HOME=/opt/java/jre
#
PORTAL_HTTP_PORT=9090
PORTAL_HTTPS_PORT=9091
# ? WiFi- RADIUS-
WIFI_AGENT_PORT=5555
# iptables,
WIFI_CHAIN_NAME=WIFI
#, WiFi-
WIFI_INTERFACE=eth0
#
EXTERNAL_INTERFACE=eth1
# IP
EXTERNAL_IP=81.30.199.220
# WIFI
WIFI_NET=172.16.1.0/24
```

4) Необходимо сделать запускаемыми все скрипты в папке агента. Для этого надо перейти в эту папку и выполнить команду :

```
chmod 755 *.sh *.pl
```

5) Настроить скрипты входа/выхода абонента.

login.sh - сюда добавляются команды открытия доступа для авторизовавшегося клиента. В этот скрипт в качестве параметра передаётся ip пользователя и атрибуты RADIUS, полученные в асерт-пакете (настройка атрибутов описана ниже).

Скрипт имеет такой вид в стандартной поставке:

```
#!/bin/sh
cd ${0%${0##*/}}.
./conf.sh

IP=$1
DOWNSTREAM_SPEED=$2
UPSTREAM_SPEED=$3
```

```

date >> ./log/manad.out
echo `sbin/iptables -I $WIFI_CHAIN_NAME 1 -t nat -j ACCEPT -s $IP` >> ./log/manad.out 2>&1
echo `sbin/iptables -I $WIFI_CHAIN_NAME 1 -t nat -j ACCEPT -d $IP` >> ./log/manad.out 2>&1

if [ $USE_MANAD -eq 1 ]; then
./tell_manad.pl "add $IP $DOWNSTREAM_SPEED $UPSTREAM_SPEED" $MANAD_PORT
fi

#use it for shaping
#./shape.sh $IP $PARAM1 $PARAM2

```

logout.sh - сюда добавляются команды закрытия доступа для клиента . В этот скрипт в качестве параметра передается ip пользователя. Скрипт имеет такой вид в стандартной поставке:

```

#!/bin/sh
cd ${0%${0##*/}}.
. ./conf.sh

IP=$1

date >> ./log/manad.out

echo `sbin/iptables -D $WIFI_CHAIN_NAME -t nat -j ACCEPT -s $IP` >> ./log/manad.out 2>&1
echo `sbin/iptables -D $WIFI_CHAIN_NAME -t nat -j ACCEPT -d $IP` >> ./log/manad.out 2>&1

if [ $USE_MANAD -eq 1 ]; then
./tell_manad.pl "remove $IP" $MANAD_PORT
fi

```

6) Настроить скрипт проверки активности клиента - ip_counts.pl. Скрипт поставляется в стандартной поставке и рассчитан на парсинг цепочки с разрешающими правилами . На выходе скрипт выдает данные вот в таком формате

```

192.168.185.10 13423 6878
192.168.185.20 133423 6878

```

Тут для каждого ip, найденного в цепочке, выводится информация о входящих (первый столбец) и исходящих байтах (второй столбец) на этот адрес . Столбцы разделены символом табуляции . Стандартный скрипт рассчитан на работу со стандартным файлом login.sh, считывает счетчики iptables из цепочки WIFI. Скрипт можно менять, главное, чтобы выходной формат оставался такой же, как описан выше.

7) Необходимо настроить iptables. По умолчанию WiFi-агент при старте системы инициализирует правила в системе с помощью скрипта iptables.sh. Рекомендуется все ваши настройки iptables также помещать в этот скрипт. Этот скрипт также может использовать администратор для очистки правил и сбрасывания всех текущих клиентов. Вот пример этого скрипта:

Конфигурация

```

#!/bin/sh
cd ${0%${0##*/}}.
. ./conf.sh

```

```

/sbin/iptables -F -t nat
/sbin/iptables -F -t filter

/sbin/iptables -P PREROUTING DROP -t nat

#external interface #####
#ssh
/sbin/iptables -A PREROUTING -s ! $WIFI_NET -t nat -p tcp --dport 22 -d $EXTERNAL_IP -j ACCEPT
#drop others from external interface
/sbin/iptables -A PREROUTING -s ! $WIFI_NET -t nat -j DROP
#end of external interface #####

#internal interface
#####

#before wifi chain we must add redirects for authorized users
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 80 -d $EXTERNAL_IP -j REDIRECT --to-ports $PORTAL_HTTP_PORT
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 443 -d $EXTERNAL_IP -j REDIRECT --to-ports
$PORTAL_HTTPS_PORT

#chain for WiFi (accept rules for authorized users)

/sbin/iptables --delete-chain $WIFI_CHAIN_NAME -t nat
/sbin/iptables -N $WIFI_CHAIN_NAME -t nat
/sbin/iptables -A PREROUTING -j $WIFI_CHAIN_NAME -t nat

#below is rules for internal not authorized users :

#dns
/sbin/iptables -A PREROUTING -t nat -p udp --dport 53 -j ACCEPT

#http
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 80 -j REDIRECT --to-ports $PORTAL_HTTP_PORT
/sbin/iptables -A PREROUTING -t nat -p tcp --dport $PORTAL_HTTP_PORT -j ACCEPT

#https
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 443 -j REDIRECT --to-ports $PORTAL_HTTPS_PORT
/sbin/iptables -A PREROUTING -t nat -p tcp --dport $PORTAL_HTTPS_PORT -j ACCEPT

#statistics
/sbin/iptables -A PREROUTING -t nat -p tcp --dport 8080 -d $EXTERNAL_IP -j ACCEPT

# NAT
iptables -A POSTROUTING -o $EXTERNAL_INTERFACE -s $WIFI_NET -j SNAT -t nat --to-source $EXTERNAL_IP

#RST packets for dropping established connections
/sbin/iptables -A FORWARD -p tcp -j REJECT --reject-with tcp-reset

```

Этот скрипт нужно поправить под ваш случай. В процессе работы правила будут иметь вид :

Конфигурация

```

# iptables -L -t nat -n
Chain PREROUTING (policy DROP)
target      prot opt source                destination              tcp dpt:22
ACCEPT     tcp  -- !172.16.1.0/24         81.30.199.220
DROP       all  -- !172.16.1.0/24         0.0.0.0/0
REDIRECT   tcp  -- 0.0.0.0/0              81.30.199.220           tcp dpt:80 redir ports 9090
REDIRECT   tcp  -- 0.0.0.0/0              81.30.199.220           tcp dpt:443 redir ports 9091

```

```

WIFI      all  --  0.0.0.0/0          0.0.0.0/0
ACCEPT   udp  --  0.0.0.0/0          0.0.0.0/0          udp dpt:53
ACCEPT   tcp  --  0.0.0.0/0          0.0.0.0/0          tcp dpt:9090
ACCEPT   tcp  --  0.0.0.0/0          0.0.0.0/0          tcp dpt:9091
ACCEPT   tcp  --  0.0.0.0/0          81.30.199.220      tcp dpt:8080

Chain POSTROUTING (policy ACCEPT)
target    prot opt source                destination
SNAT      all  --  172.16.1.0/24        0.0.0.0/0          to:81.30.199.220

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination

Chain WIFI (1 references)
target    prot opt source                destination
ACCEPT   all  --  172.16.1.105         0.0.0.0/0
ACCEPT   all  --  172.16.1.57          0.0.0.0/0
ACCEPT   all  --  172.16.1.94          0.0.0.0/0

# iptables -L -t filter -n
Chain INPUT (policy ACCEPT)
target    prot opt source                destination

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination
REJECT    tcp  --  0.0.0.0/0            0.0.0.0/0          reject-with tcp-reset

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination

Chain RH-Firewall-1-INPUT (0 references)
target    prot opt source                destination

```

Очень не рекомендуется заносить правила в цепочку WIFI, т.к. она редактируется автоматически и могут возникнуть проблемы. Свои дополнительные правила вы можете заносить в любые другие цепочки и таблицы (учитывая логику работы iptables и WiFi-агента).

Например, если Radius-сервер находится на другой машине, то ему надо разрешить 5555-ый (в данном случае) порт для обращения к WiFi-агенту. Аналогично можно разрешить порты для ssh и т.п., если это необходимо. На шлюзовой машине, где устанавливается агент, скорее всего, будет, как минимум, один внешний интерфейс и один внутренний интерфейс, через который будут работать клиенты WiFi. В этом случае, например, если по ssh будут обращаться только через внешний интерфейс, можно повесить разрешающие правила на внешний интерфейс. Вариантов много, но главное чтобы выход во внешнюю сеть был закрыт для клиентов локальной сети по умолчанию. Одним из вариантов организации сети может быть набор виртуальных (vlan) интерфейсов, которые будут заведены на данной шлюзовой машине (на интерфейсе eth0) и агент будет добавлять правила для всех клиентов этих виртуальных сетей. В этом случае агент может управлять сразу многими локальными сетями, которые могут быть физически разнесены далеко друг от друга.

Отметим, что для правильной работы сети кроме правила NAT, добавленного выше, в случае ОС Linux необходимо ещё включить ipforwarding. Для дистрибутива Fedora необходимо поставить `net.ipv4.ip_forward = 1` в `/etc/sysctl.conf` и выполнить команду `echo 1 >> /proc/sys/net/ipv4/ip_forward` (чтобы изменения немедленно применились). Рекомендуется вначале проверить работу точки доступа и выход в интернет через неё, прежде чем применять запрещающие правила iptables, описанные выше.

8) Установить скрипт `service/bgwifiaagent_dialup` как службу. Для этого нужно скопировать файл `bgwifiaagent_dialup` в `/etc/rc.d/init.d` и потом вызвать следующие команды

```

chmod 755 /etc/rc.d/init.d/bgwifiaagent_dialup
chkconfig --add bgwifiaagent_dialup
chkconfig --level {lev} bgwifiaagent_dialup on

```

Где {lev}- это уровень запуска в вашей системе. Узнать его можно так:

```

[root@king ~]# runlevel
N 5

```

Т.к. агент изменяет правила iptables, то в системе он должен запускаться с правами root-a.

9) В файле setenv.sh нужно прописать JAVA_HOME;

10) Перед запуском агента нужно запустить скрипт

```
./update.sh
```

в папке агента. Это обновит все библиотеки на нем (скачает с сервера).

11) Для запуска агента можно выполнить команду:

```
service bgwifiagent_dialup start
```

Агент не запустится, если не будет связи с сервером BGBilling или там не будет установлены лицензии на модуль Dialup или сам портал.

12) Подсоединиться клиентом к серверу биллинга, настроить [NAS](#) для работы с нашим агентом. В конфигурацию NAS нужно добавить:

```
#
nas.inspector.sleep_time=60
#
nas.port_time.default.*=109
nas.port_traffic.default.*=111:COLLECTOR

nas.inspector.class=bitel.billing.server.processor.WiFiConnectionInspector
nas.inspector.wifi.host=127.0.0.1
nas.inspector.wifi.port=5555
```

Здесь вы должны указать хост и порт, на котором работает агент. RADIUS-сервер будет периодически обращаться к нему, проверять активность клиента и слать сигнал о завершении работы в случае завершения баланса клиента. При этом вы должны сделать идентификатор NASa равным параметру **radius.nasId**, а секретный ключ NAS-a равный **radius.secret**.

13) Установить [RADIUS-сервер для модуля DialUp](#). Настройки радиуса должны совпадать с параметрами **radius.auth.host**, **radius.auth.port**, **radius.account.host**, **radius.account.port** в файле dialup_wifi_agent.properties.

Связь WiFi-агента с модулем "Карточки" (DialUp)

Если у вас установлен модуль "Карточки", то портал может отображать ссылку на активацию карты. Для этого в конфигурационном файле `dialup_wifi_agent.properties` вы должны прописать строку

```
portal.card.link=http://127.0.0.1:8080/bgbilling/pubexecuter?action=CreateContract&module=card&mid=5
```

Формат этой ссылки описан в документации по модулю [Карточки](#).

Защита WiFi-сети от ARP-спуффинга (DialUp)

ARP-spoofing (ARP-poisoning) — техника сетевой атаки, применяемая преимущественно в Ethernet, но возможная и в других, использующих протокол ARP, сетях, основанная на использовании недостатков протокола ARP и позволяющая перехватывать трафик между узлами, которые расположены в пределах одного широковещательного домена. Суть её состоит в том, что любой желающий может посылать ARP-запросы, подменять таким образом ARP-таблицы на других компьютерах и сопоставлять свой mac-адрес с чужим ip-адресом.

Одним из методов защиты является использование статических ARP-таблиц, т.е. запрет на изменение ARP-таблицы. Именно это решение мы предлагаем для использования в нашем WiFi-агенте. Для этого внутри WiFi-агента реализуется ещё одно приложение - DHCP relay-агент. По описанию протокола DHCP (RFC 2131) сервер DHCP может отвечать не только на запросы клиентов на получение ip-адреса, но и на запрос relay-агентов, которые пробрасывают запросы клиентов на сервер (проставляя свой адрес, чтобы сервер мог им ответить), а ответ отправляют клиенту. Таким образом обычно реализуется возможность получения ip-адресов, если DHCP сервер находится в другой сети и часто роль relay-агентов выполняют аппаратные шлюзы. Мы используем программный relay-агент для того, чтобы при получении ip-адреса от DHCP-сервера редактировать arp-таблицу на шлюзе (где установлен WiFi-агент) и для удаления записи из arp-таблицы по истечению срока аренды ip-адреса (который задаётся DHCP-сервером и проставляется клиенту, а клиент, в свою очередь, обязан до истечения срока аренды послать запрос на продление ip-адреса). Динамическое обновление arp-таблицы отключается и никто другой не может править arp-таблицу, кроме DHCP relay-агента.

Для запуска DHCP-агента нужно в файл `dialup_wifi_agent.properties` добавить следующие настройки:

```
#dhcp options
dhcp=1

dhcp.servers.1.host=192.168.184.254
dhcp.servers.1.port=67

dhcp.servers.2.host=192.168.184.253
dhcp.servers.2.port=67

dhcp.agent.host=192.168.154.39
dhcp.minThreadCount=10
dhcp.maxThreadCount=10

dhcp.servers.1.host=192.168.184.254
dhcp.servers.1.port=67
```

Здесь

dhcp=1 - это флаг, говорящий о том, что нужно запускать DHCP агент.

Далее идут настройки dhcp-серверов. Поддерживается несколько серверов (<id> - 1,2,3 и т.п.):

dhcp.server.<id>.host - ip-адрес сервера DHCP. Он нужен relay-агенту.

dhcp.server.<id>.port - порт, на котором запускается сервер DHCP. Вообще, стандартным портом для сервера DHCP является 67-ой. Но в данном случае, возможно, вам понадобится изменить этот порт. Например, если DHCP-агент и DHCP-сервер находятся в одной локальной сети. В этом случае они оба будут получать широковещательные dhcp-запросы клиентов на получение ip-адреса и сервер будет отправлять ответы клиентам напрямую. Нам же нужно, чтобы запрос приходил только к DHCP-агенту, а агент его уже пробрасывал на сервер. Этого можно добиться с помощью организации сети или ещё какими-либо другими способами, но один из вариантов - это поднять сервер на другом порту, чтобы он не получал запросы клиентов на 67-ом. Стандартный сервер dhcpd позволяет это сделать. В этом случае вам понадобится изменить этот параметр.

dhcp.agent.host - ip-адрес DHCP-агента, на который потом будет отвечать DHCP-сервер. Это адрес проставляется в dhcp-запрос, который проходит через relay-агент.

dhcp.minThreadCount, dhcp.maxThreadCount - это минимальное и максимальное количество потоков в пуле при обработке запросов от клиентов и серверов DHCP. Эти параметры можно оставить по умолчанию.

dhcp.arp.command - путь к команде arp в ОС Linux. Она нужна для манипулирования arp-таблицами. По умолчанию обычно `/sbin/arp`.

Для отключения динамического обновления arp-таблицы нужно выполнить команду

```
ifconfig eth0 -arp
```

При этом нужно посмотреть какие адреса уже попали в эту таблицу и, при необходимости, её отредактировать (например, добавить адреса каких-либо серверов в локальной сети, которые имеют статический ip).

Настройка ограничения скорости (шейпинг) для трафика WiFi-сети (DialUp)

Шейпинг осуществляется с помощью iproute2. Его реализация происходит через внешние скрипты и настраиваемые атрибуты Radius-сервера, получаемые в ассерт-пакете. Вы можете реализовать свой вариант или изменить наш под ваши нужды. Общий принцип такой :

1. Скрипт init.sh(вызывается при старте системы) - в нем можно проводить инициализацию правил шейпинга;
- 2.login.sh - сюда добавляются правила шейпинга для нового клиента, появившегося в сети. В это скрипт в качестве параметра передаётся ip пользователя и атрибуты RADIUS, полученные в ассерт-пакете (настройка атрибутов описана ниже);
- 3.logout.sh - здесь удаляются правила шейпинга при выходе клиента из сети. В этот скрипт в качестве параметра передаётся ip пользователя.

Мы вам предлагаем свой вариант реализации этих скриптов. Для корректной работы этого варианта в системе должен быть установлен perl. Для его конфигурации надо добавить в файл conf.sh следующие строчки:

```
USE_MANAD=1
MANAD_INTERFACE=eth0
MANAD_PORT=4567
```

Здесь USE_MANAD=1 обозначает, что будет использоваться шейпинг, MANAD_INTERFACE - это интерфейс, на котором будет контролироваться трафик клиента и MANAD_PORT - порт, на котором будет слушать perl-скрипт wifi_manad.pl, управляющий шейпингом . Этот скрипт слушает на определённом порту команды на удаление и добавление нового клиента . При добавлении клиента, например, с ip 192.168.184.33, скоростью входящего трафика (downstream) - 256 кбит/сек, скоростью исходящего трафика (upstream) -128 кбит/сек, он добавляет для него следующие правила :

```
/sbin/tc class add dev eth0 parent 1:0 classid 1:3 htb rate 256kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:3 handle 3: sfq perturb 10 quantum 1500

/sbin/tc class add dev eth0 parent 1:0 classid 1:4 htb rate 128kbit burst 4k prio 1
/sbin/tc qdisc add dev eth0 parent 1:4 handle 4: sfq perturb 10 quantum 1500

/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio 3 u32 match ip dst 192.168.184.33 flowid 1:3

/sbin/tc filter add dev eth0 parent 1:0 protocol ip prio 4 u32 match ip src 192.168.184.33 flowid 1:4
```

А при удалении клиента 192.168.184.33 скрипт выполняет следующие команды:

```
/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio 3
/sbin/tc filter del dev eth0 parent 1:0 protocol ip prio 4

/sbin/tc class del dev eth0 parent 1:0 classid 1:3 htb rate 256kbit burst 4k prio 1
/sbin/tc class del dev eth0 parent 1:0 classid 1:4 htb rate 128kbit burst 4k prio 1
```

Для правильной работы данного скрипта нужно настроить RADIUS-атрибуты, которые мы хотим получить из ассерт-пакета от RADIUS-сервера . В данном случае нас интересует два атрибута: ограничение входящей скорости (downstream) и значение исходящей скорости (upstream). Для этих атрибутов в файл в dialup_wifi_agent нужно добавить следующие настройки:

```
wifi.agent.radius.attribute.1.vendor.code=1111
wifi.agent.radius.attribute.1.attr.code=1
wifi.agent.radius.attribute.1.type=integer
wifi.agent.radius.attribute.2.vendor.code=1111
wifi.agent.radius.attribute.2.attr.code=2
wifi.agent.radius.attribute.2.type=integer
```

Формат добавления атрибутов следующий:

agent.radius.attribute.X - общий вид.

X - это код атрибута. Нумерация должна идти по порядку - 1, 2 и т.д. vendor.code - код производителя, attr.code - код атрибута, type - тип атрибута. **Атрибуты** настраиваются в файле dictionary.xml Radius-сервера и WiFi-портала (там тоже есть такой файл), и для данного примера можно добавить, например, такие атрибуты:

```
<vendor code="1111" name="linuxWiFi">
  <attribute add="no" name="WiFi-Downstream-Speed-Limit" type="integer" code="1"/>
  <attribute add="no" name="WiFi-Upstream-Speed-Limit" type="integer" code="2"/>
</vendor>
```

В конфигурация модуля DialUp нужно добавить группы атрибутов :

```
attrset.4.title= WiFi 256/128
attrset.4.attributes=WiFi-Downstream-Speed-Limit=256;WiFi-Upstream-Speed-Limit=128
```

Сами группы атрибутов можно задать:

1. для каждого **логина DialUp**;
2. для **услуги в тарифе**;
3. для **REALM**а (можно использовать, например, realm.default для всех пользователей).

Значения атрибутов, описанных в файле dialup_wifi_agent.properties в точно в таком же порядке (после ip-адреса), подаются в качестве параметров в скрипт login.sh, вызываемый после установки пользователем соединения.

Настройка REALM'ов (DialUp)

Клиент при авторизации может использовать [REALМы](#). Список доступных REALМов может отображаться клиенту в виде выпадающего списка :

Логин	den
Пароль	•••••
Услуга	128 kb/s ▾
<input type="button" value="Отправить"/>	
Безопасное соединение	

Пусть, мы хотим использовать 2 REALМа: входящий трафик 128 кбит/с и 256 кбит/с. Исходящий же трафик в обоих случаях пусть будет 128 кбит/сек. Для добавления этой возможности нужно добавить следующие настройки в файл `dialup_wifi_agent.properties`:

```
#tarif options
portal.use.realm=1
portal.tarif.1.realm=wifi_128_128
portal.tarif.1.title=128 kb/s
portal.tarif.2.realm=wifi_256_128
portal.tarif.2.title=256 kb/s
```

В конфигурацию модуля для этого случая нужно добавить :

```
realm.wifi_128_128=WiFi-Downstream-Speed-Limit=128;WiFi-Upstream-Speed-Limit=128
realm.wifi_256_128=WiFi-Downstream-Speed-Limit=256;WiFi-Upstream-Speed-Limit=128
realmgr.default=default;wifi_128_128;wifi_256_128
```

В конфигурацию NASa нужно добавить

```
nas.port_time.wifi_128_128.*=112
nas.port_traffic.wifi_128_128.*=

nas.port_time.wifi_256_128.*=113
nas.port_traffic.wifi_256_128.*=
```

Где 112 и 113 - коды услуг обчёта данных REALМов (в данном примере обчёт только по времени). Эти услуги должны быть добавлены в тариф (читайте [настройку тарифов](#)).

Web-интерфейс WiFi-портала (DialUp)

Web-интерфейс WiFi-портала сделан с помощью [jsp](#) и располагается в каталоге portal. Также используются технологии [jstl](#) и [struts](#).

Модули Телефонии

Модуль Phone

Назначение модуля

Модуль предназначен для учёта телефонных соединений обычной телефонии на основании текстовых логов, приведённых к определённому формату, и позволяет вести учёт в соответствии с последними требованиями законодательства РФ.

В модуле реализован учёт и расчёт услуг, предоставляемых клиентам и учёт операторских взаиморасчётов.

Настройка модуля

После установки модуля, создайте его экземпляры и в **Редакторе модулей и услуг** создайте услуги, предоставляемые абонентам и операторам.

Услуги абонентов - это предоставление местной, зонавой, либо МГМН связи для абонента в режиме преселект или хотчойс через какого-либо оператора. Соответственно, количество услуг для абонента будет представлять из себя произведение количества операторов на количество типов связи и режимов, например:

- Местная связь
- Зонавая связь Оператор 1 преселект
- Зонавая связь Оператор 1 хотчойс
- МГМН связь Оператор 2 преселект
- МГМН связь Оператор 2 хотчойс
- МГМН связь Оператор 3 преселект
- МГМН связь Оператор 4 хотчойс

Услуги операторов - это предоставление услуг завершения вызовов, либо инициации. Перечень услуг операторов можно ограничить названием предоставляемых услуг, например **Инициация, Завершение**.

На вкладке **Конфигурация** создайте и установите конфигурацию модуля.

Конфигурация

```
#
contract.status.active.codes=0
#
#-----
#
#-----
#
#closed.date.disabled.ActionClientItemUpdate=1
#
#closed.date.disabled.ActionSetBalance=1
#
#-----
#   CDR
#-----
#   " " (cost) log_session
#   log_session
session.call.cost.scale=2
#   " " (oper_cost) log_session
#   log_session
session.oper.cost.scale=5
#
#store.incoming.calls=1
#   , CDR-   , 1 -
error.on.session.rule.notfound=0
#   , CDR-   , 1 -
error.on.session.rule.many=0
#
#-----
#
#-----
# 1 -   ,
point.resource.strict.check=0
#
phone.resource.reserve=1
#
```

```
#-----  
# Web-  
#-----  
#  
web.menuItem1=  
web.menuItem2=  
web.menuItem3=  
#     Web-  
show.on.page.web=50  
# . - Web ( )  
show.on.page.web.max=1000  
#  
show.on.page.session=50  
  
#  
#-----  
# - . 1- .  
#-----  
web.menuItem.blackwhitelist.enable=0
```

Подготовка логов

Исходными данными для работы модуля являются CDR-записи, разделённые по АТС и предоставленные в следующем формате.

```
-      (dd.MM.yyyy HH:mm:ss)
-      (
-      # A
-      # A (E.164), #A164
-      # B
-      # B (E.164), #B164
-      port_from
-      port_to
-
-      (
-
-
-
```

Поле **Категория** в данный момент не используется биллингом, вместо него могут быть подставлены 0.

Время соединения используется для учёта межоператорских взаимозачётов правилами и всегда больше или равно **длительности звонка**. Это полное время от начала установки соединения с АТС оператора до окончания соединения. Поинты тарифицируются на основании **длительности звонка** - времени разговора абонента. Т.е. для тарификации сессии по правилу будет использована длительность соединения а для тарификации абонента - длительность звонка.

Стоимость вызова может задавать как цену всего звонка для абонента, так и минуты звонка, использование данного поля задаётся в тарифном плане.

Поля **Время соединения**, **Стоимость** и **Стоимость вызова для оператора** могут отсутствовать.

CDR-запись представляет собой строку в текстовом файле кодировке ASCII, в качестве разделителей полей используется знак табуляции. Например, фрагмент лога:

```
01.08.2006 10:55:18 57733 3517913292 73517913292 2479292 73512479292 3 62 5 57733 0
01.08.2006 14:08:33 75980 3517744152 73517744152 2479292 73512479292 4 34 5 75980 0
```

Логи должны быть разбиты по АТС и по часовым интервалам, при этом имя файла формируется по шаблону dd_НН, файл помещается в zip-архив с именем dd_НН.zip. Логи группируются в каталоги по годам и месяцам. Например, дерево файлов лога АТС.

```
2005 |
    |-01 |
        |-01_00.zip
        |-01_01.zip
    .....
    |-02 |
        |-01_00.zip
    .....
```

Пример подобных логов вы можете посмотреть [здесь](#). На уровень выше расположен скрипт **billing.pl** конвертации логов АТС в формат модуля. Вы можете модифицировать скрипт под формат логов вашей АТС. Скрипт написан на Perl, для запуска на Win платформе необходима установка Perl-интерпретатора, вы можете взять его с сайта <http://www.activestate.com>. Разумеется, конвертер может быть реализован на любом другом языке программирования.

Логи каждой АТС должны быть выложены в отдельный каталог, доступный серверу биллинга по FTP, либо через файловую систему. Примеры конвертеров логов доступны на [Wiki](#).

Настройка загрузки и обработки логов

Для каждой АТС на вкладке **Источники** модуля должен быть заведён свой источник. В зависимости от метода доступа к логам у источника устанавливается тип **FTP**, либо **Локальная или сетевая папка**.

При указании FTP-источника в поле Сервер вводится DNS-имя, либо IP-адрес FTP-сервера, его порт (если не указан, берётся 21) и каталог, в котором хранятся логи данного источника.

Формат адреса сервера: <HOST>[:<PORT>][PATH], параметры в квадратных скобках не обязательны к указанию.

Например: ftp.bitel.ru/pub/projects/bgbilling/phone/log/source_1. Это реальный каталог с логами, вы можете использовать их для тестирования.

В конфигурации источника может быть указана переменная **resource.categories=<коды категорий ресурсов через запятую>**. Про ресурсы номеров описано далее. Данная опция, по сути, определяет номерную ёмкость АТС. Обработчик выводит ошибки в журнал ошибок при установленной данной опции и обнаружении звонков в логах данного источника с номеров, относящихся к указанным категориям, но при этом не соотнесённых ни одному пинту. Таким образом можно осуществлять простой аудит.

При создании источника типа **Локальная или сетевая папка** достаточно указать каталог с логами. Название источника используется для его идентификации в дальнейшем.

Загрузку логов производит процесс **DataLoader**. Для его запуска на UNIX-системе выполните скрипт **data_loader_start.sh**, предварительно в файле **data_loader.sh** необходимо прописать путь к Java-машине, например:

```
cd ${0%${0##*/}}.  
JAVA_HOME=/opt/java/jre
```

В Windows-системе выполните скрипт **data_loader_install.bat** для установки загрузчика в список служб и далее вы можете запускать службу **BGDat aLoader** через оснастку **Администрирование=>Службы**.

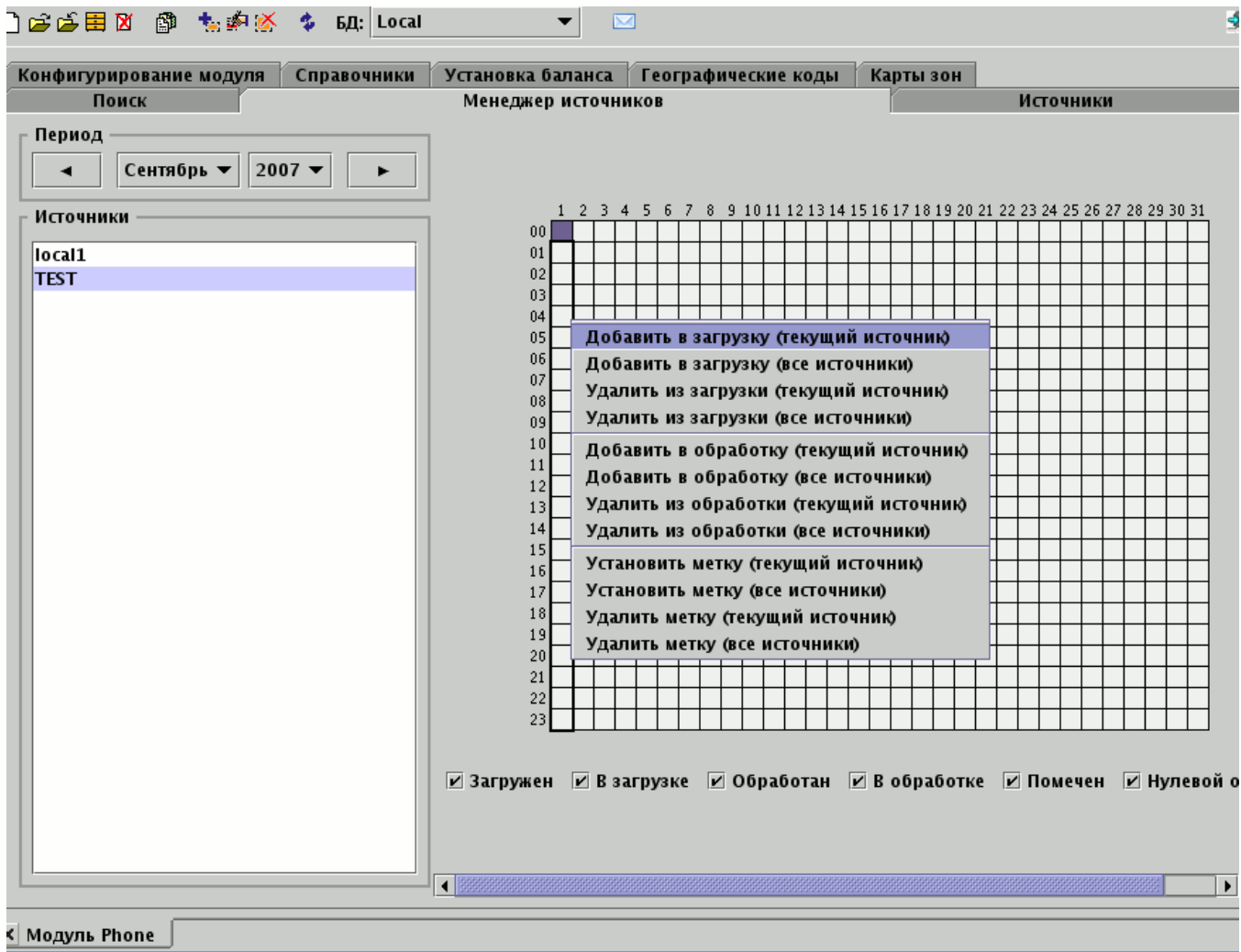
В случае успешного старта загрузчика логов в **log/dataloader.log** должен быть примерно такой текст.

```
Checking port 9033...  
Port is free starting the application...  
Starting DLProcessManager on 9033  
Creating socket on 9033
```

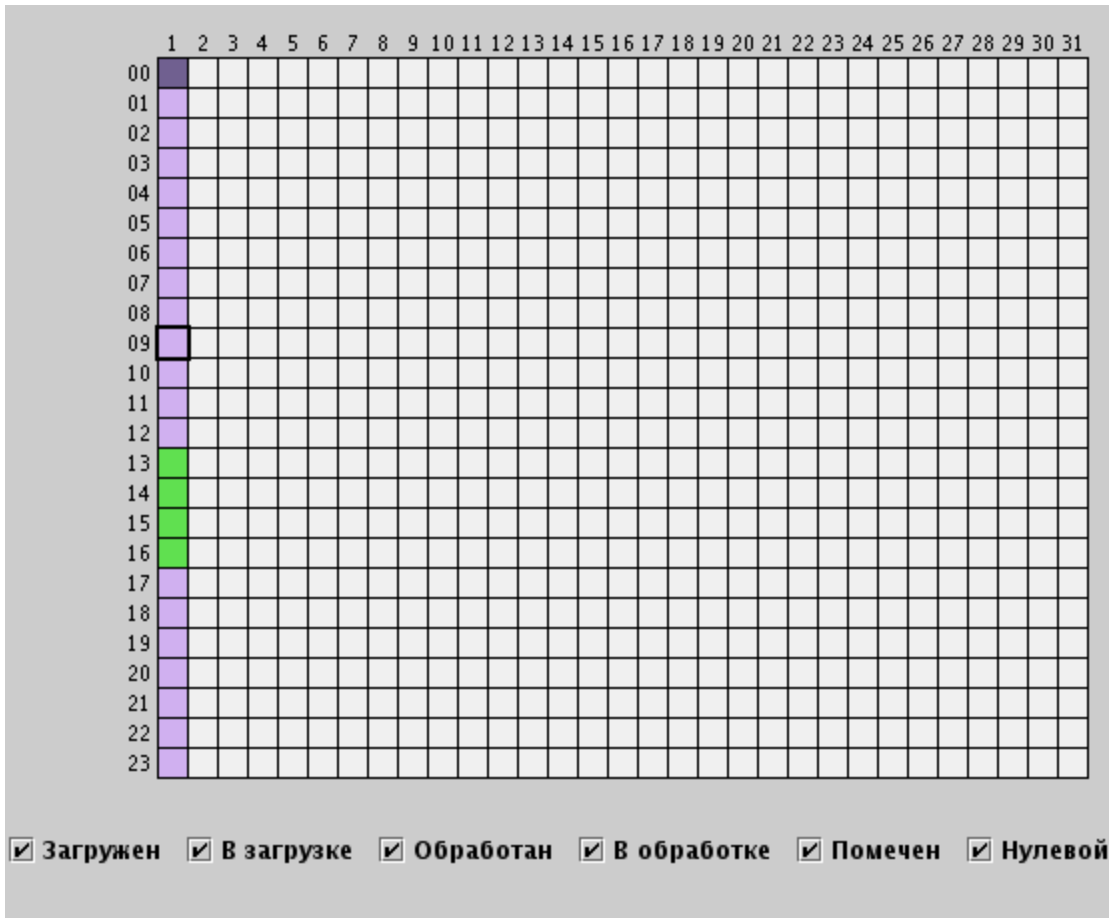
Если лог пуст, проверьте лог **log/dataloader.out** на предмет ошибок.

Процесс обработки логов включает в себя две стадии: загрузку первичных логов в базу данных и непосредственно обработку логов, при которой звонки соотносятся клиентам и тарифицируются. Загруженные логи можно переобработать неограниченное количество раз, количество сессий в базе не будет увеличиваться. При загрузке часового лога задание на его обработку генерируется автоматически.

Загрузка часового лога может быть инициирована вручную, либо автоматически. Для ручной инициации загрузки логов откройте вкладку **Менеджер источников** модуля.



По мере загрузки логов цвет часовых квадратов будет изменяться. Загрузка логов производится случайным образом, не по порядку часов. По мере загрузки логов загрузчик логов сам добавляет задания на обработку и производит обработку логов. Обработка логов выполняется по порядку часов. Для отслеживания изменения состояний логов нажимайте кнопку **Обновить** панели инструментов.



По завершению обработки сессии должны появиться в ответах договоров. При этом производится учёт абонентского и операторского трафика и его тарификация. Про настройку абонентов, операторов и их тарифов описано далее.

Если этого не произошло, обратитесь в **Сервис=>Журналы=>Журнал ошибок**. В выпадающем списке выберите ваш модуль телефонии, установите месяц и нажмите кнопку с галочкой.

Модуль
Phone

Период
 < Ноябрь 2007 >
 дни: с до

1 из 2865 [25]

ID	Дата	Ошибка
2233	2007-11-01 0:00	Не установлен тарифный план
2232	2007-11-01 0:00	Не установлен тарифный план
2231	2007-11-01 0:00	Не установлен тарифный план
2230	2007-11-01 0:00	Не установлен тарифный план
2229	2007-11-01 0:00	Не установлен тарифный план
2228	2007-11-01 0:00	Не установлен тарифный план
2227	2007-11-01 0:00	Не установлен тарифный план
2226	2007-11-01 0:00	Не установлен тарифный план
2225	2007-11-01 0:00	Не установлен тарифный план
2224	2007-11-01 0:00	Не установлен тарифный план
2223	2007-11-01 0:00	Не установлен тарифный план
2222	2007-11-01 0:00	Не установлен тарифный план
2221	2007-11-01 0:00	Не установлен тарифный план
2220	2007-11-01 0:00	Не установлен тарифный план
2219	2007-11-01 0:00	Не установлен тарифный план
2218	2007-11-01 0:00	Не установлен тарифный план
2217	2007-11-01 0:00	Не установлен тарифный план
2216	2007-11-01 0:00	Не установлен тарифный план
2215	2007-11-01 0:00	Не установлен тарифный план
2214	2007-11-01 0:00	Не установлен тарифный план
2213	2007-11-01 0:00	Не установлен тарифный план

Сведения о ошибке
 Contract: Z0212-07
 StartTime=01.11.2007 00:52:44 PORT_FROM=0 PORT_TO=0 NUM_FROM=73472900857 NUM_FROM_164=73472900857
 NUM_TO=89279256107 NUM_TO_164=79279256107

Найденные ошибки необходимо исправлять и запускать переобработку логов до полного исчезновения ошибок в журнале. Добавление логов в обработку производится в **Менеджере источников** путем выбора квадрата часов и активации пункта меню **Добавить в обработку** для выбранного источника.

После завершения отладки модуля возможна настройка автоматической загрузки логов телефонии. Это может быть сделано двумя способами:

1. По мере подготовки логов автоматически вызывать добавление задачи на загрузку лога запуском:

```
./data_loader.sh -eload=yyyy-MM-dd-НН-<source_id>
#,      2006-12-14 10      15
./data_loader.sh -eload=2006-12-14-10-15
```

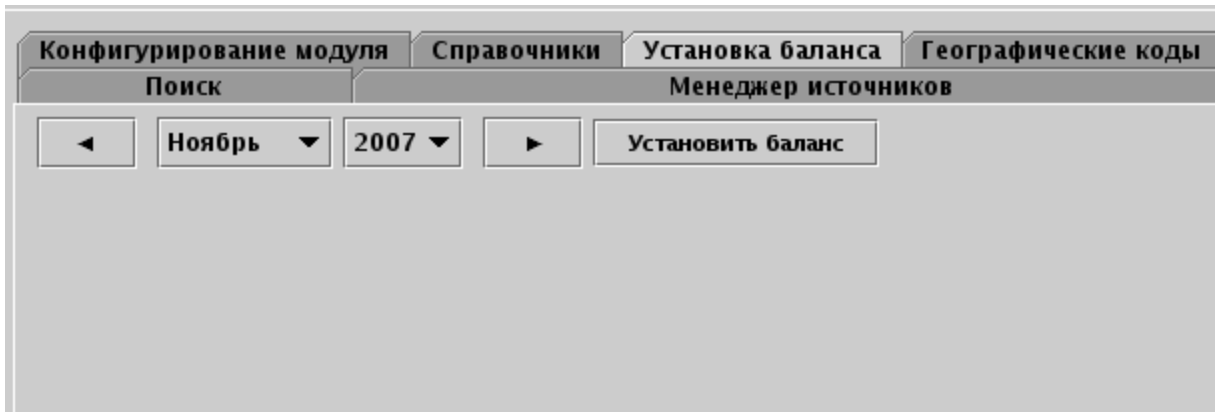
Также можно передать команду через сокет управления dataloader по TCP в виде (9033 по умолчанию):

```
load=yyyy-MM-dd-НН-<source_id>\n
```

Этот путь более правильный, подходит для полной автоматизации процесса "подачи" логов в биллинг.

2. Раз в час задача **Генератор заданий на загрузку**, запускаемая планировщиком, создаёт задание на загрузку очередного часового лога для всех источников FTP, либо Сетевая папка. Запуск задачи необходимо настроить в **Планировщике заданий** на 15 минут каждого часа. Вполне возможно, что к этому моменту логи ещё не будут добавлены, тогда загрузчик будет пытаться загрузить недостающие логи в течении 3х суток. Этот метод подходит для режима, когда логи подготавливает отдельный специалист, не связанный с биллингом.

Результирующими данными обработки логов являются логи сессий в договорах клиентов и операторов, привязанные к поинтам, либо правилам (см. далее). Для того, чтобы наработка сессий перешла в баланс договора используется вкладка модуля **Установка баланса**. Единственным параметром выступает дата месяца, за который необходимо установить баланс.

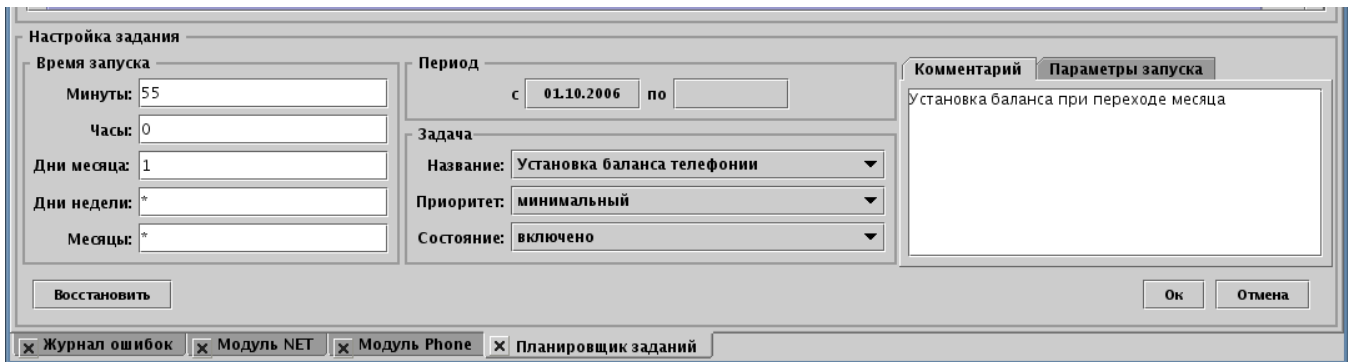
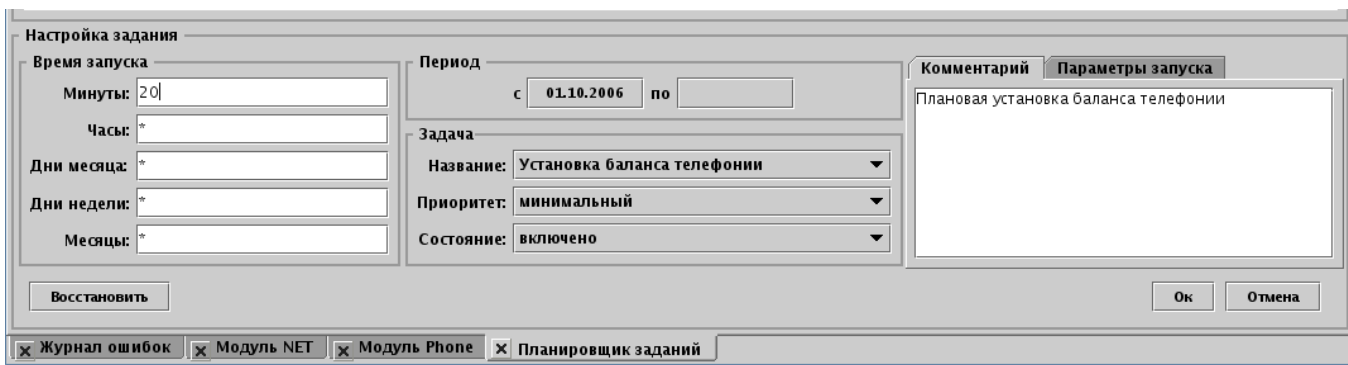


Для того, чтобы не делать установку вручную вы можете настроить задачу планировщика **Установка баланса телефонии**. При запуске задача берет предыдущий от текущего часа час и производит установку баланса за этот месяц.

Соответственно, необходимо настроить два экземпляра задачи в планировщике заданий. Одну - периодическую, например, каждый час. Другую - в 55 минут первого часа месяца для установки баланса за предыдущий месяц с учётом последнего загруженного часового лога. Примеры настройки периодической и единоразовой задач приведены на снимках ниже.

В параметрах запуска задачи должно быть установлено:

mid=< >

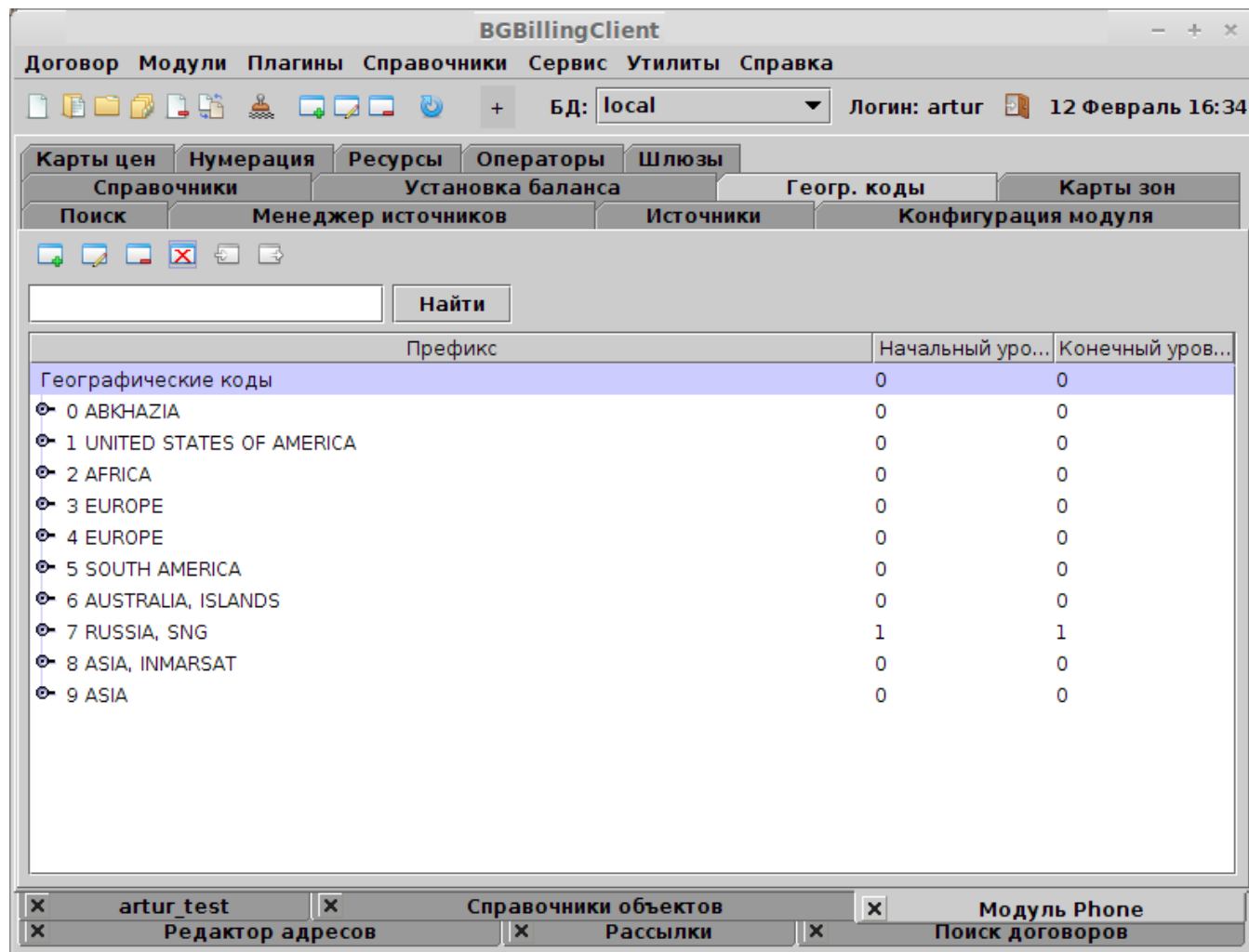


Географические коды, карты зон и цен

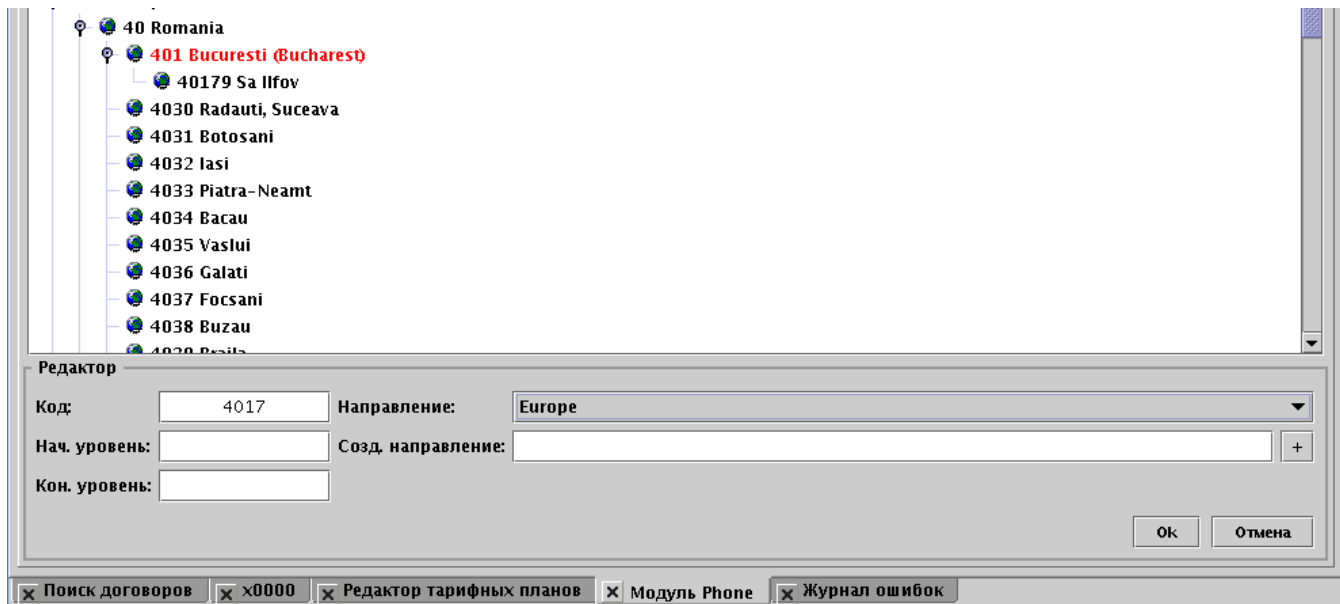
Тарификация с использованием справочника географических кодов является одним из вариантов тарификации для VoiceIP и Phone-модулей. Метод подходит при работе с операторами, разбивающими все множество префиксов мира на зоны с фиксированными ценами.

Справочник географических кодов - это перечень записей вида "префикс-направление", описывающий весь мир. Пример справочника можно загрузить [здесь](#). Перед началом работы с модулем необходимо произвести загрузку справочника на вкладке модуля **Географические коды**.

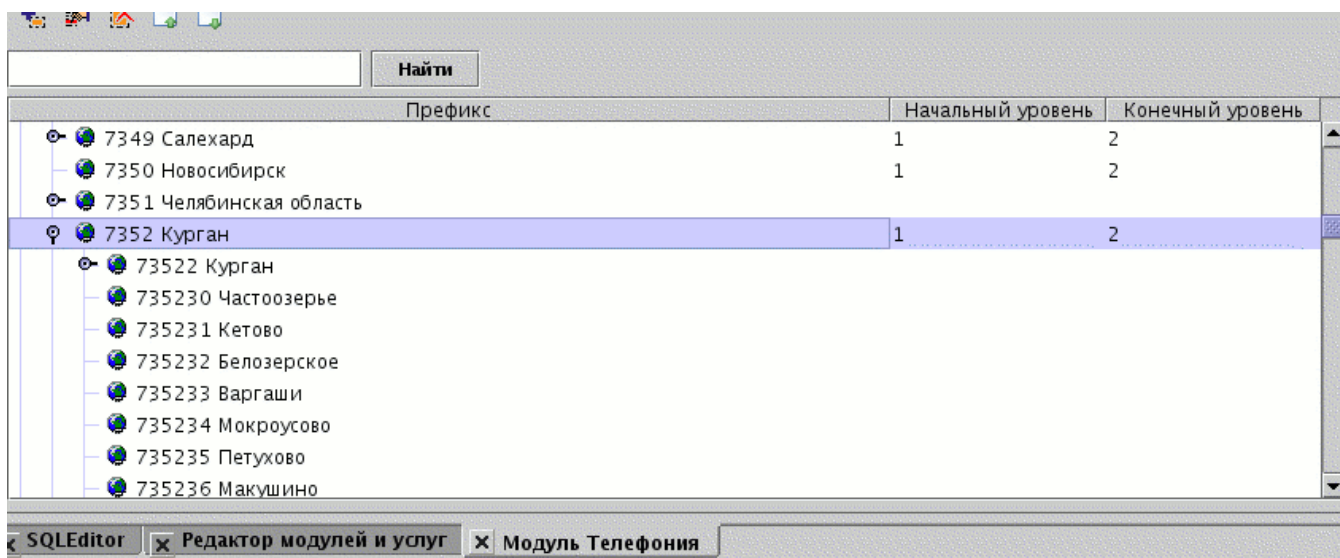
Для редактирования над справочником размещена панель инструментов.



Для добавления нового кода выберите родительский узел и нажмите **Новый элемент**.



Направление можно выбрать из списка существующих, либо создать новое, введя его название в текстовую область и нажав кнопку + справа от области. Следует обратить внимание на параметр **Нач. уровень** и **Кон. уровень**. Эти целые числа определяют до какой глубины будет уточняться направление при прохождении дерева. По умолчанию уровни не установлены, что означает уточнение до самого нижнего узла.



Уровни отображаются в дереве двумя числами после направления. Начальный и конечный уровни задают до какой глубины при прохождении дерева кодов будет уточняться направление. Глубина вложенности считается от нуля (корень дерева), т.е. 7 (Россия, СНГ) - это 1-ый уровень, 7352(Курган) - это второй уровень.

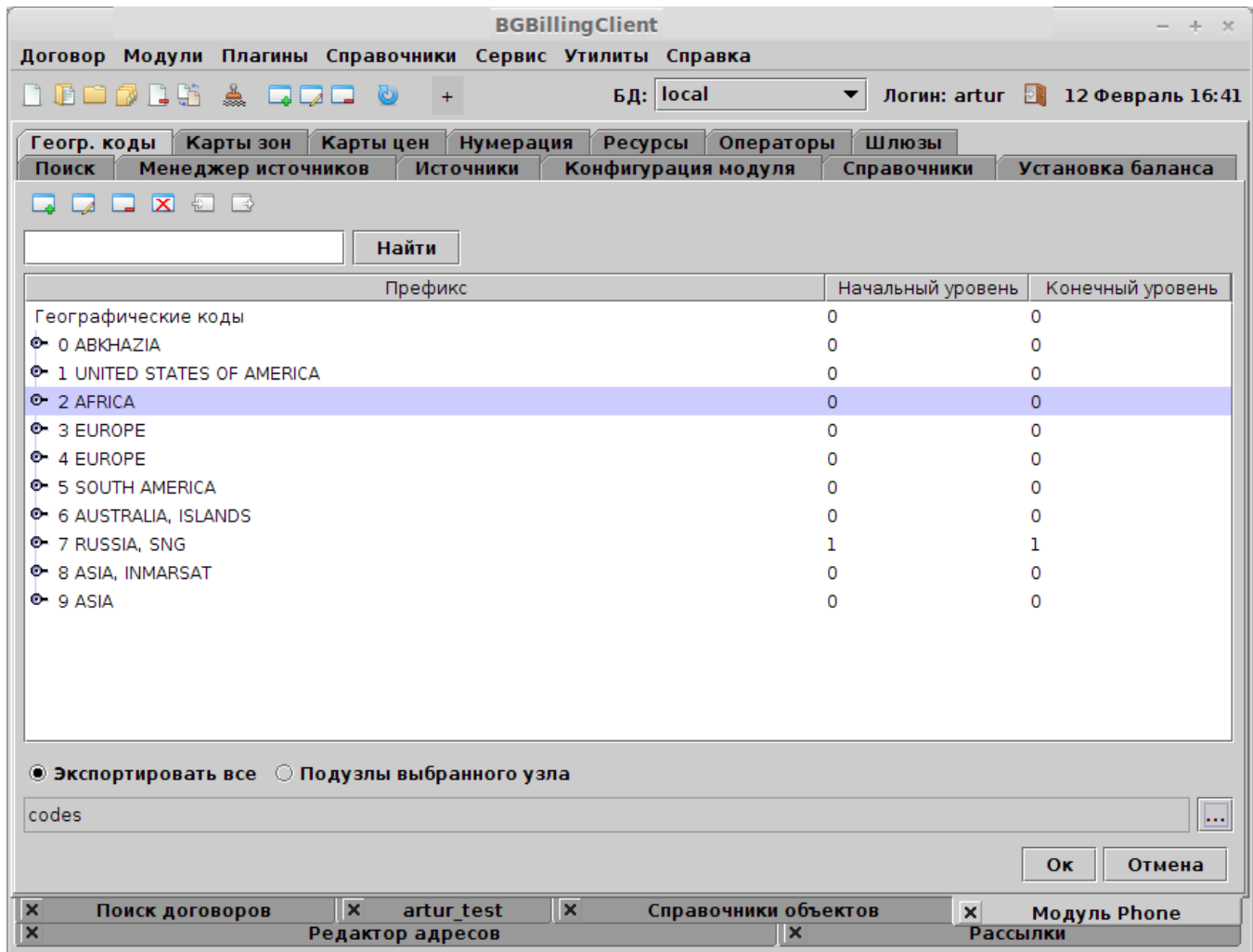
При установке уровней как на снимке сверху направление будет уточняться только до Курган, т.е. все звонки в Белозерское, Кетово и т.п. будут выглядеть как Курган.

Кнопка **Удалить** удаляет географический код.

Кнопка **Удалить все коды** удаляет все географические коды.

Перед удалением проверьте, не используется ли код в картах зон!

Кнопка **Экспорт** позволяет выгрузить все дерево, либо какой-то префикс в текстовый файл.

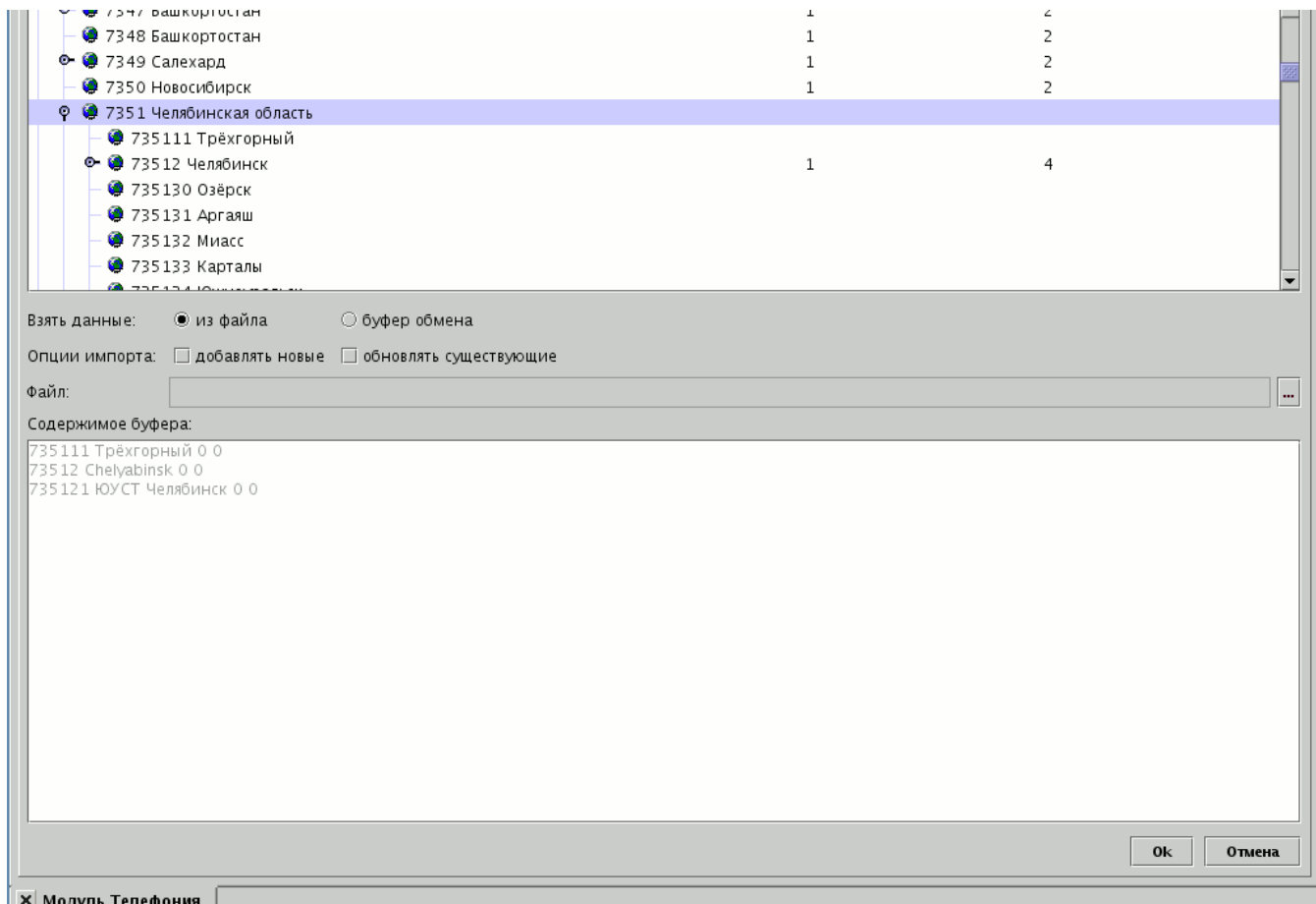


Справочник выгружается в текстовый файл с разделителями столбцов - табуляторами и имеет примерно следующий формат:

```
735111 0 0
73512 Chelyabinsk 0 0
735121 0 0
```

Последние 2 столбца - уровни определения направления. Их может и не быть.

Для импорта используется аналогичный файл, либо аналогичное содержимое буфера обмена. Можно импортировать как весь справочник, так и его часть.

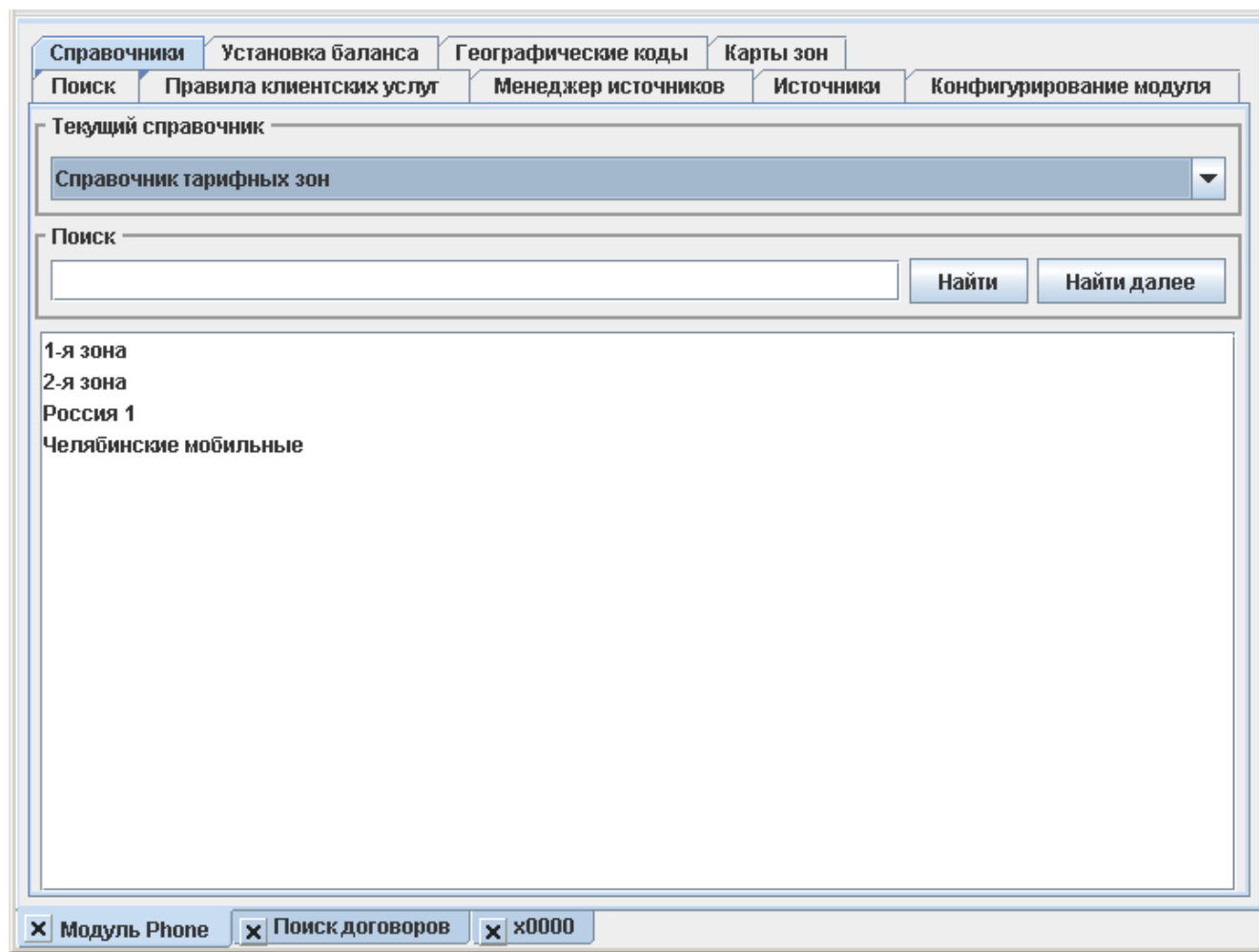


При импорте можно поставить одну или обе галочки **добавлять новые** и **обновлять существующие**. Обновляются направление и уровни префикса.

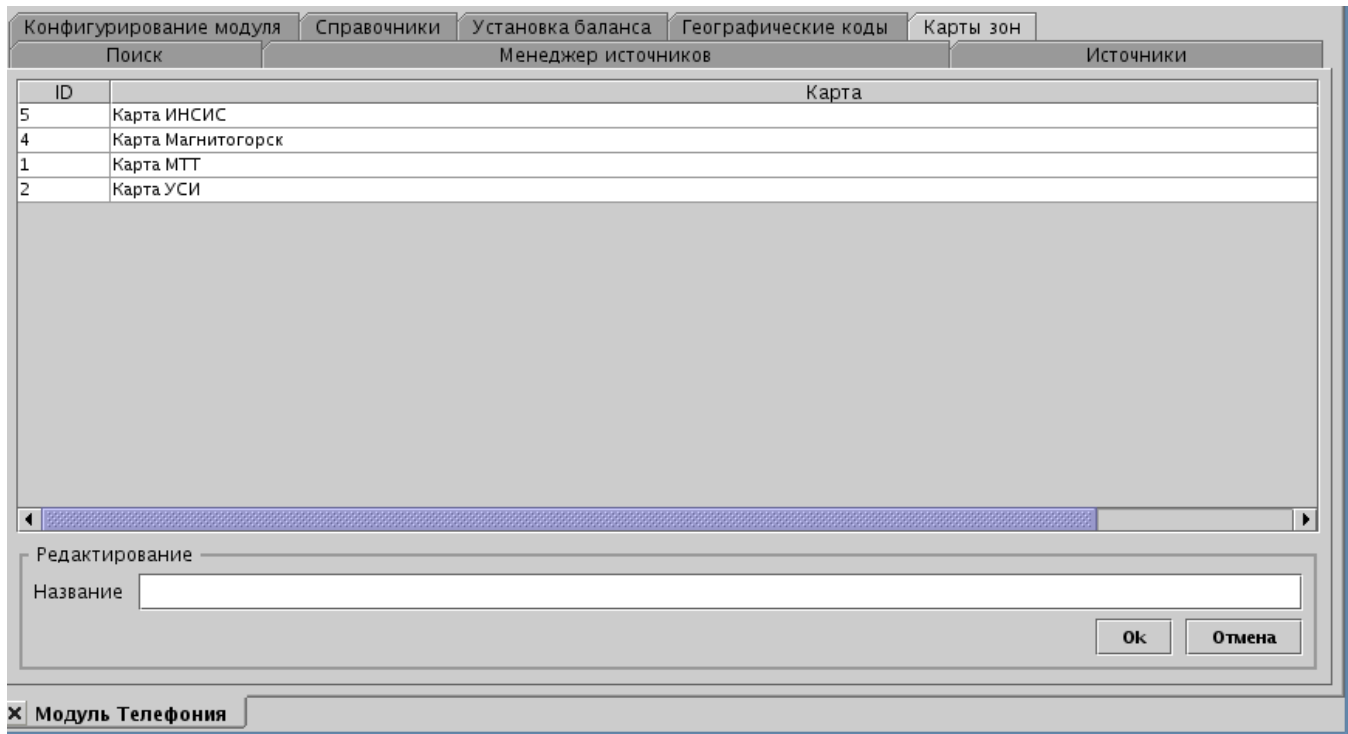
Карта зон

Карта зон - это разбиение справочника кодов по зонам - областям равной цены. Карт в модуле может быть несколько.

Перед созданием карты все используемые зоны должны быть описаны на вкладке **Справочники** модуля. Если МГМН-операторов с зоной тарификацией несколько - советуем начинать имя зоны названием оператора.



Создание карт производится на вкладке модуля **Карты зон** модуля. Для создания карты нажмите **Новый элемент** на стандартной панели инструментов и введите её название. Для открытия карты дважды кликните по ней мышью. Переименование - кнопка **Редактировать** панели инструментов.



Зоны присваиваются узлам дерева географических кодов. При присвоении зоны узлу-предку она автоматически распространяется на узлы-потомки.

"BG - биллинг" - универсальная биллинговая система (вер. 4.2 от 16.05.2007) UFANET

Договор Модули Справочники Сервис Автоматизация Утилиты Справка

БД: Local

Конфигурирование модуля Справочники Установка баланса Географические коды Карты зон

Поиск Менеджер источников Источники

Регион	Зона
Географические коды	
0 Справочные	Россия 1
1 North America, Canada	
2 Africa	
3 Europe	1-я зо
4 Europe	2-я зо
5 South America	
6 Australia, Islands	Между
7 Russia, SNG	1-я зо
8 Asia, Inmarsat	
9 Asia	

Импорт/Экспорт

Модуль Phone

Без зоны
 Россия 1
 Межгород 3 (5,78)
 Межгород 4 (6,62)
 13-я зона
 Межгород 1 (2)
 Межгород 2 (4,41)
 1-я зона
 2-я зона
 Челябинские мобильные
 Зона 13
 Межгород 5 (6,83)
 Межгород 6 (7,14)
 Международная 1 (10,75)
 Международная 2 (12,29)
 Международная 3 (13,51)
 Международная 4 (19,18)
 Международная 5 (19,96)
 Международная 6 (22,11)
 Международная 7 (22,41)
 Международная 8 (24,56)
 Международная 9 (30,09)
 Международная 10 (36,23)
 Международная 11 (39,75)

Экспорт

Для установки зон используется меню, вызываемое правой кнопкой мышки. Карту зон можно экспортировать целиком в XML-файл и импортировать из него также целиком. Файл имеет примерно следующий формат:

```
<?xml version="1.0" encoding="UTF-8"?>
<zones name=" 1">
  <zone name="1- ">
    <codes>
      <code id="735150"/>
      <code id="3"/>
      <code id="7"/>
      <code id="73512"/>
    </codes>
  </zone>
  <zone name=" 1">
    <codes>
      <code id="7341"/>
      <code id="91"/>
    </codes>
  </zone>
  <zone name=" 6 (22,11)">
    <codes>
      <code id="6"/>
    </codes>
  </zone>
</zones>
```

```
        </codes>  
    </zone>  
</zones>
```

Атрибут **name** в корневом узле **zones** может не указываться в импортируемом файле.

Карта цен

Карта цен - это привязка цен к географическим кодам. Карт цен в модуле может быть несколько.

Карта цен может быть удобна для тарификации мг/мн звонков, когда оператор мг/мн тарифицирует агента не по зонам, а по связке: префикс - цена или диапазон префиксов - цена.

При использовании карты цен направление вычисляется по географическим кодам.

ID	Карта цен
5	123
6	Оператор

Редактор

Название

OK Отмена

Карту цен аналогично карте зон можно импортировать и экспортировать в csv-файл формата:

```
7      10.0
7347  5.0
```

Или в таком:

```
7              10.0
7347          5.0
```

Во втором случае текстовое значение второго столбца будет привязано к префиксу в данной карте цен.

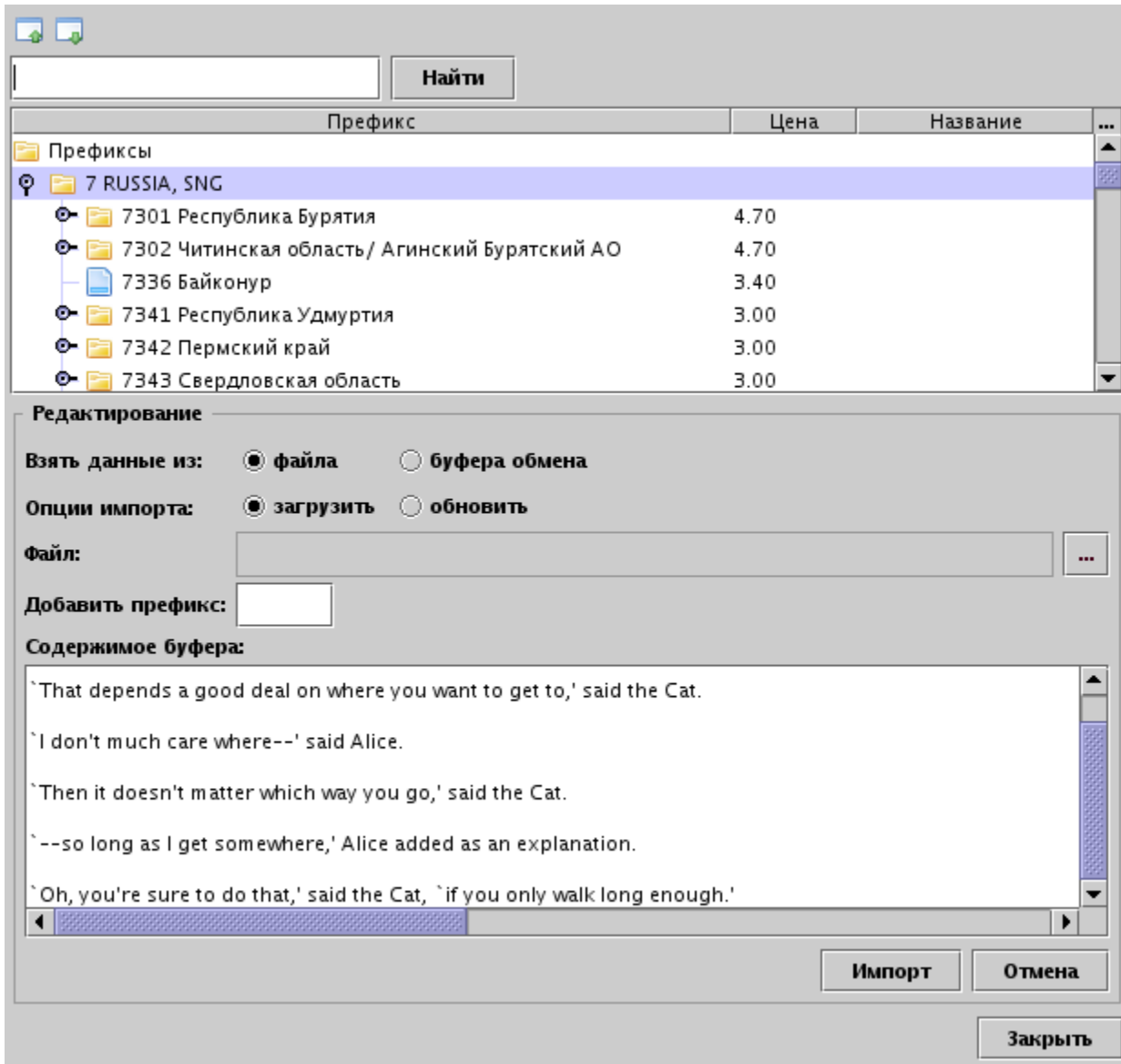
Столбцы в файле импорта должны быть разделены одним знаком табуляции.

При импорте можно указать дополнительный префикс - он будет добавлен перед каждым префиксом. Это может быть удобно, если переданный список цен оператора для МГ тарификации представлен без цифры 7, т.е. 917, а не 7917.

При импорте автоматически разделяются префиксы вида:

```
355(40-41, 43-49)    ALBANIA [1]    1,82
43 644, 650, 660, 680, 681, 688, 699, 711,720    (mob)    16,51
```

В качестве значения префикса может выступать префикс-диапазон, например: 374(9300-9328).

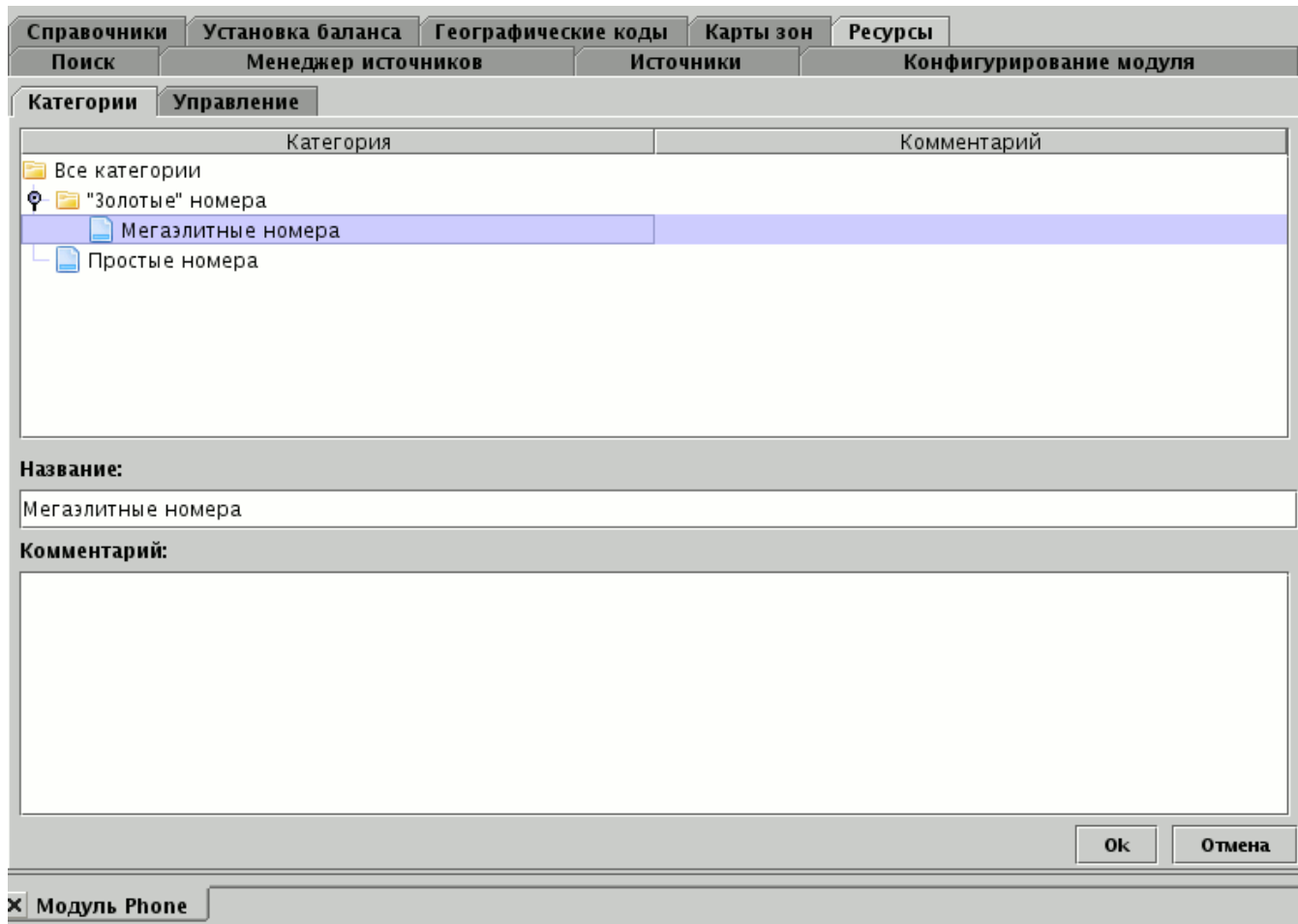


Помимо импорта возможно поэлементное редактирование карты цен стандартными кнопками добавления/редактирования/удаления.

Управление ресурсами номеров

Управление ресурсами позволяет отслеживать номерную ёмкость. Для управления ресурсами используется вкладка **Ресурсы** модуля.

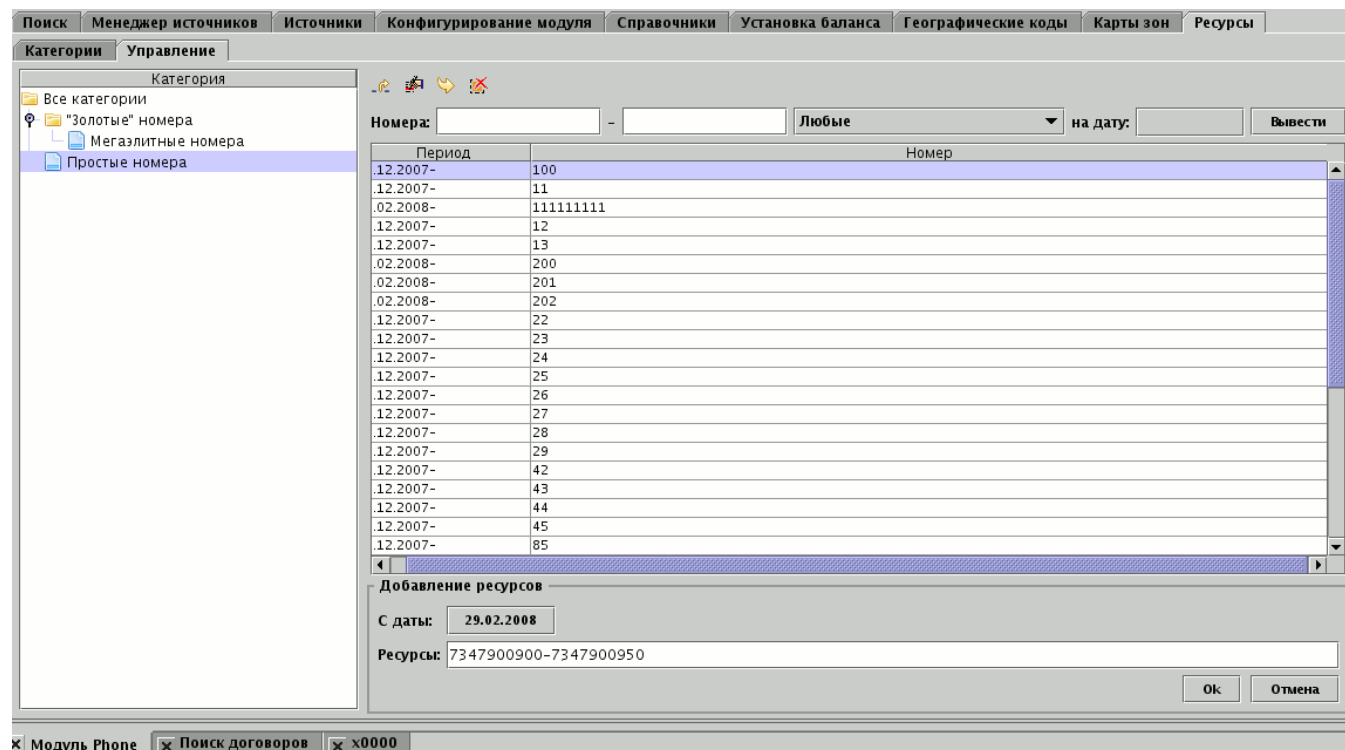
Категории ресурса позволяют произвести логическое деление всего объёма ресурсов. Например, на "Золотые" и обычные номера, либо на номера, доступные на разных АТС. Для редактирования категорий используется подвкладка Категории, для добавление категории выберите родительскую категорию и нажмите **Новый элемент** на стандартной панели инструментов.



На вкладке **Управление** производится непосредственное управление ресурсами.

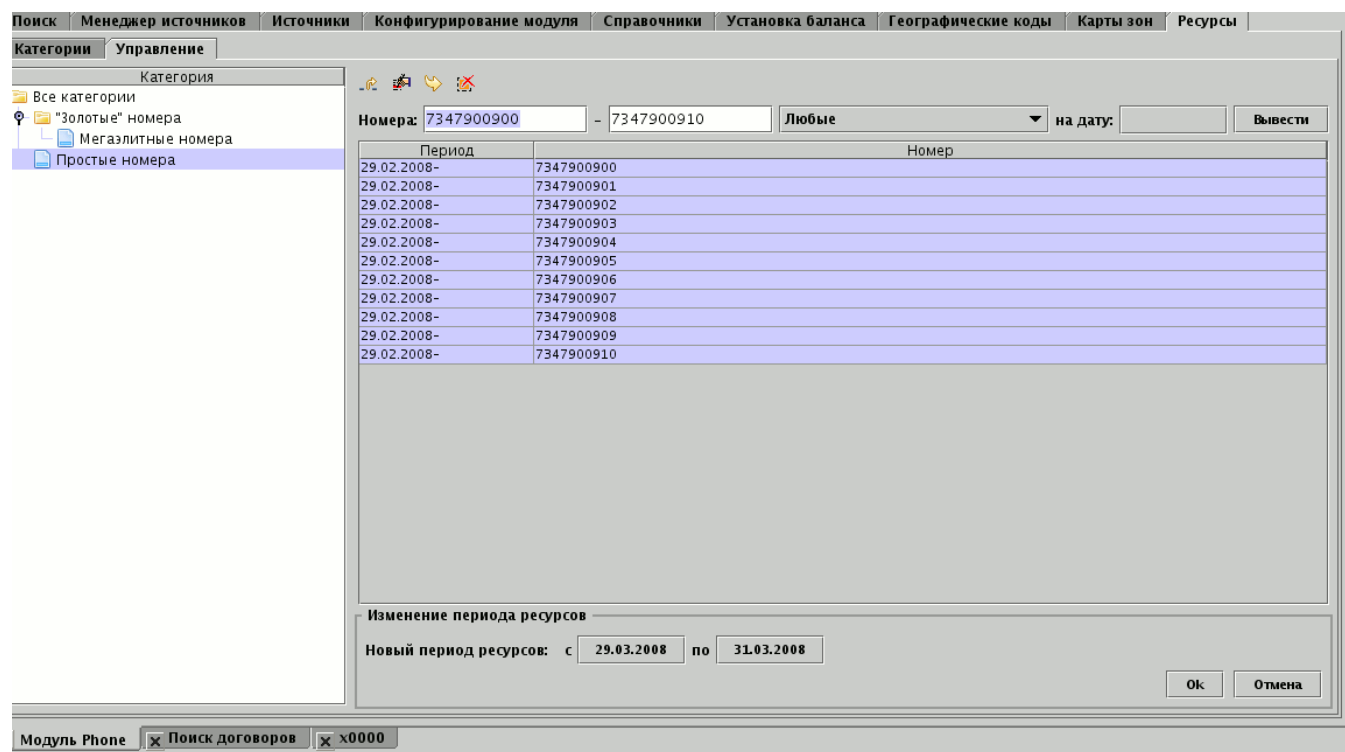
Добавление ресурсов

Для добавления ресурсов выберите категорию в дереве и нажмите кнопку **Открыть ресурсы** в панели инструментов над таблицей просмотра ресурсов.



В открывшемся редакторе введите диапазон номеров и дату, с которой они доступны.

После загрузки номерной ёмкости возможно изменить период действия для некоторых номеров. Для выбора поддиапазона ресурсов используется фильтр, далее комбинацией клавиш Ctrl+A выбираются все строки открывшейся таблицы и кнопкой **Изменить период ресурсов** открывается редактор.

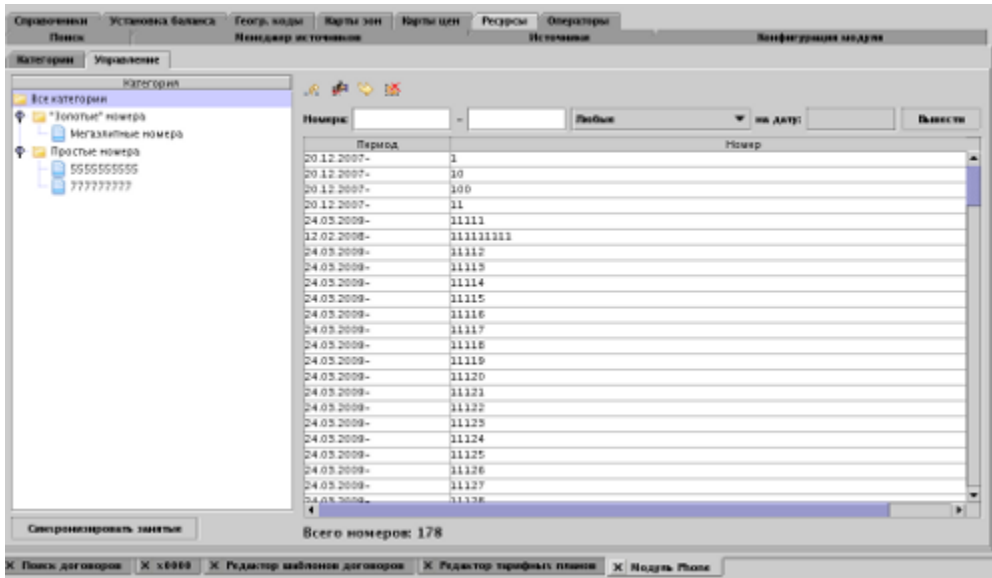


Аналогично осуществляется перенос ресурсов в другую категорию и удаление ресурсов. Удалить можно только ресурс, который не проставлен ни в одном договоре.

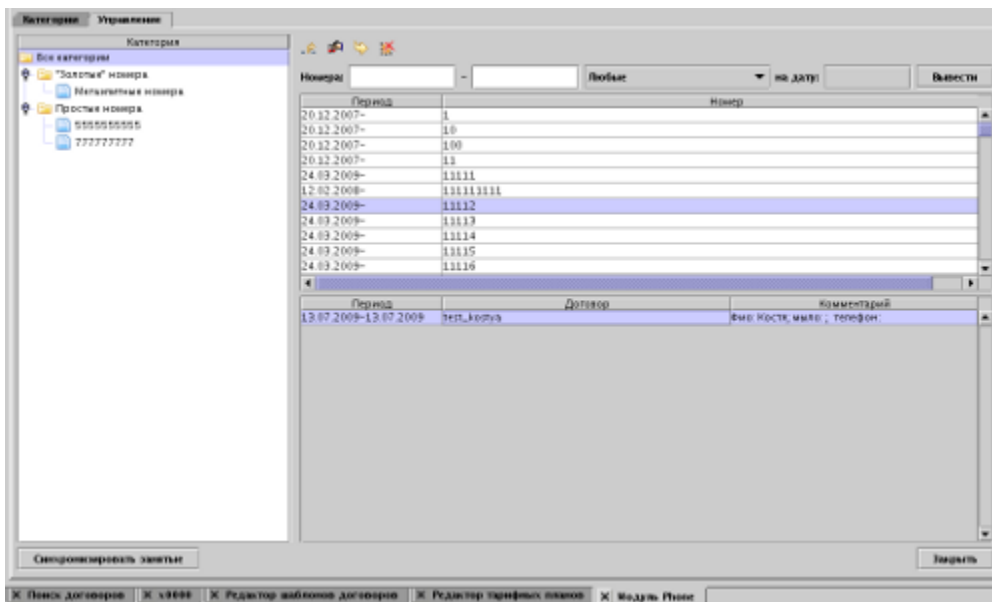
Слежение за ресурсами

При выборе категории отображаются ресурсы только выбранной категории, при выборе **Все категории** - отображаются все ресурсы. Возможен фильтр по диапазону номеров и статусу номера. Свободным считается номер, не выданный никакому договору. При установке фильтра по статусу номера необходима установка даты, на которую выводятся статусы.

Фильтр применяется нажатием кнопки **Вывести**.



При двойном клике мыши по строке с ресурсом открывается таблица, отображающая историю использования ресурса. Двойной клик в таблице истории открывает договор.



Кнопка **Синхронизировать занятые** производит обновление состояния ресурсов и истории их использования на основании поинтов договоров. При любом редактировании поинтов происходит автоматическое изменение состояния ресурса и истории его использования.

При установке флага

```
#  
phone.resource.reserve=1  
#  
#phone.resource.reserve.month.count=1
```

появляется возможность резервирования номеров на месяц, после закрытия номера.

Учёт абонентского трафика

Понятие поинта в модуле идентично абонентской точке доступа (порта, либо номера), предоставленной клиенту на АТС. К поинтам привязываются сессии. Для вывода всех поинтов, привязанных к договору, необходимо выбрать экземпляр модуля в дереве договора. Возможна сортировка поинтов по номеру/порту и периоду действия, а также возможна фильтрация поинтов на определенную дату.

The screenshot shows the BGBillingClient application window. The title bar reads "BGBillingClient". The menu bar includes "Договор", "Модули", "Плагины", "Справочники", "Сервис", "Утилиты", and "Справка". The status bar shows "БД: local", "Логин: artur", and "12 Февраль 17:23".

The main interface is divided into several sections:

- Left Panel (Tree View):** Shows a hierarchical structure of the account. The "Модули" (Modules) section is expanded, showing "Phone" as the selected module with the status "открыт". Other modules include "Mobi. Деньги", "MPS", "Payture", "Softkey", "VoIP", "Бухгалтерия", and "Новые абонлаты".
- Top Section:** Contains fields for "Договор" (Contract) with the value "artur_test [Гареев Артур Маратович]", "ID" (352984), and "Период" (Period) from "24.11.2009" to the current date.
- Right Panel (Table):** Displays a list of points under the "Пойнты" (Points) tab. The table has columns for "ID", "Описание" (Description), "Период" (Period), and "Комментарий" (Commentary). One point is visible with ID "16035" and description "22556" for the period "12.02.2014-...".

At the bottom, there is a taskbar with several open windows: "Редактор адресов", "Рассылки", "Поиск договоров", "artur_test", "Справочники объектов", and "Модуль Phone".

В свойствах поинта могут быть указаны параметры: **Номер(а)**, **Порт(ы)**, **Алиас** и **АТС**, к которой привязан абонент. Привязка к АТС (источнику) позволяет избежать дублирования звонков, при прохождении вызова абонента через несколько АТС. Алиас может быть не указан, тогда биллинг будет идентифицировать (отображать) поинт как **Номер(а) <перечень номеров> на <Название источника>**, либо **Порты(а) <перечень портов> на <Название источника>**.

Внесение нескольких номеров (портов) на один поинт может быть произведено через запятую, однако это лишает возможности просмотра сессий с разбивкой по номерам и данный способ учёта не рекомендуется.

Пойнты Правила Шлюзы

Редактирование Тарифы Абонплаты Управление списками

Источник

- den_test
- local1
- localK
- TEST
- ACBT-NN

Алиас 22556

Номер(а) 29 <<<

Порт(ы)

Период с 12.02.2014 по

Комментарий

Привязка к ip-адресу модуля IPN

Объект

Установить

Ок Отмена

На вкладке **Редактирование** задаются свойства поинта, его источник и период действия. Кнопка **Взять из пула ресурсов** рядом с полем ввода номера позволяет получить номер из базы ресурсов модуля.

Пойнты Правила

Категория

- Все категории
- Золотые номера
 - Мегазлитные номера
 - Простые номера

С

На дату: 29.02.2008 Вывести

7347900922
 7347900923
 7347900924
 7347900925
 7347900926
 7347900927
 7347900928
 7347900929
 7347900930
 7347900931
 7347900932
 7347900933
 7347900934
 7347900935
 7347900936
 7347900937
 7347900938
 7347900939
 7347900940
 7347900941

Доступно номеров: 87

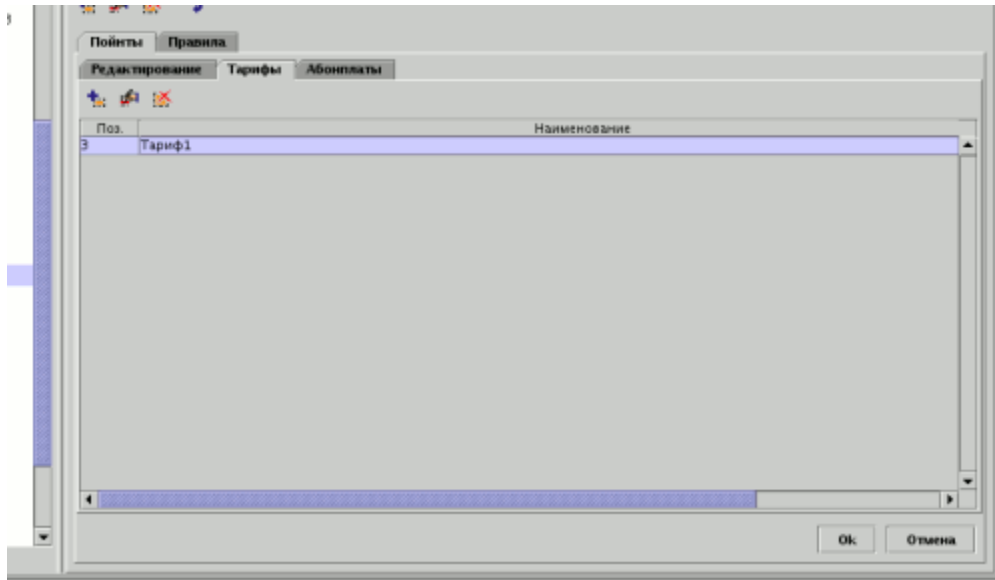
Ок Отмена

юв x0000

Вне зависимости от того, был ли номер получен из пула ресурсов, либо введён вручную, если он присутствует в пуле - при всех модификациях поинтов в истории использования ресурса будут изменяться записи. Для возвращения ресурса в пул, достаточно закрыть его в договоре.

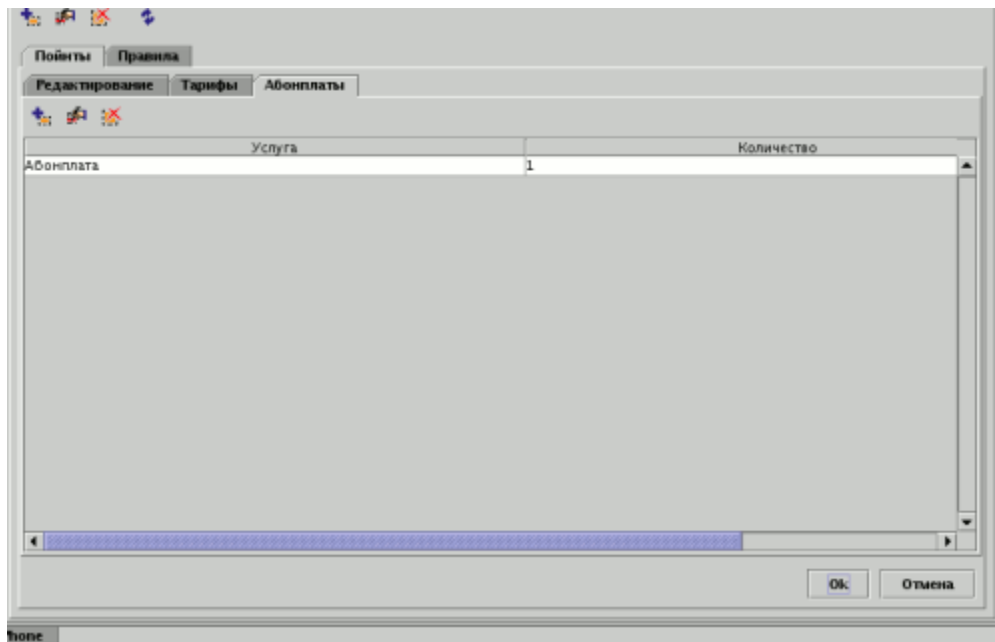
Обратите внимание, что должен быть указан номер абонента в формате E.164 (см. снимок). При сохранении поинта система автоматически проверяет отсутствие в модуле одинаковых номеров на двух разных договорах с пересекающимися периодами. При обнаружении конфликта система выдаст предупреждение и данные не будут изменены.

На вкладке **Тарифы** могут быть указаны тарифные планы для конкретного поинта.



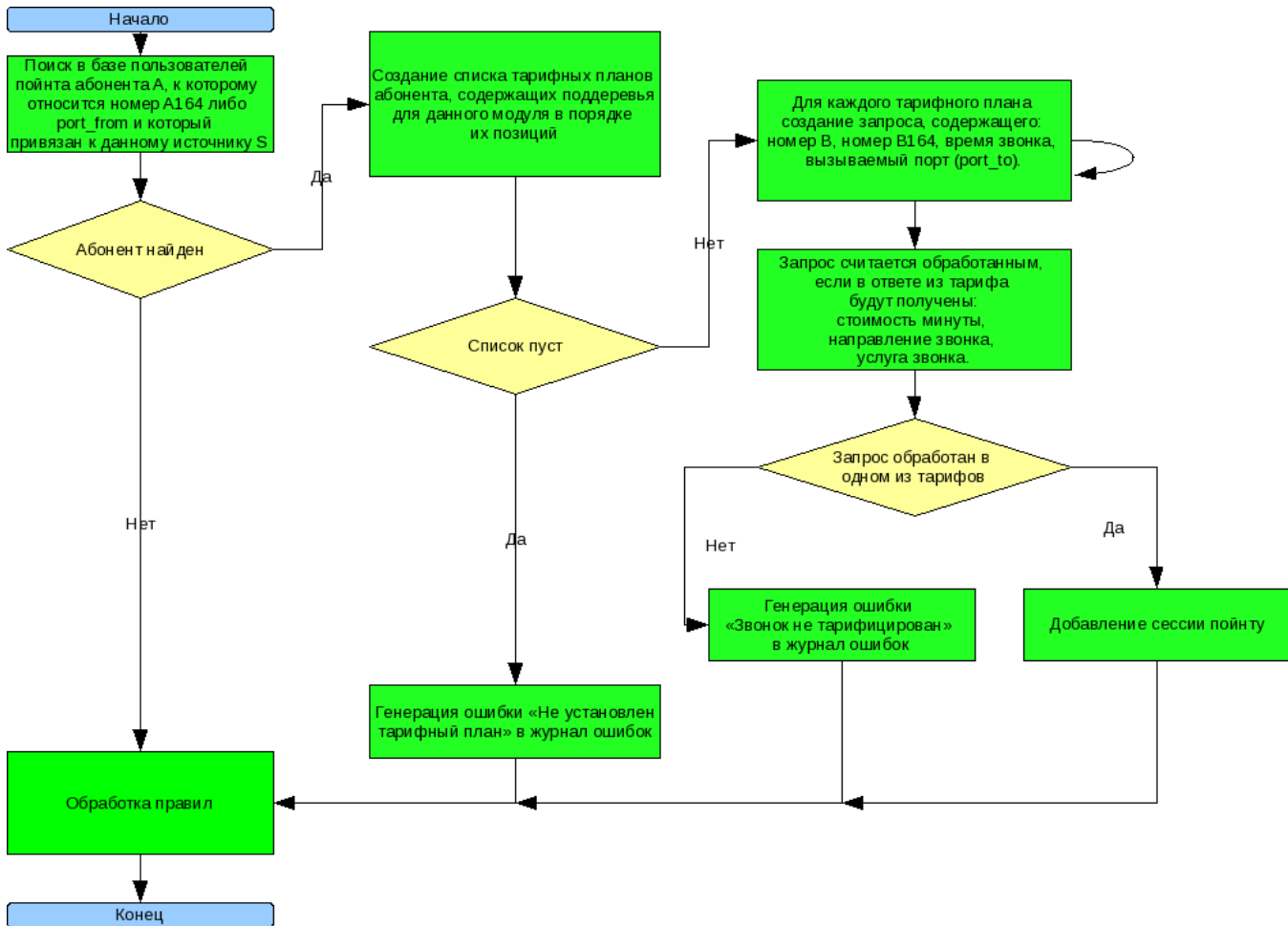
На вкладке **Абонплаты** к поинту могут быть привязаны абонентские платы модуля NPay. Для отображения в этой вкладке абонплат в конфигурации модуля телефонии должно быть прописано (<mid> - код экземпляра модуля NPay):

```
npay.mid=<mid>
```



Непосредственно к номеру могут быть привязаны абонплаты за различные дополнительные услуги, выделение линии и пр. Начисление абонентских плат производится средствами указанного в конфигурации экземпляра модуля.

При обработке часового лога выполняются следующие шаги для каждой записи. Обработка правил на данной схеме не детализирована, она будет рассмотрена далее.



Как видно из вышеописанной схемы после идентификации поинта производится поочерёдный "просмотр" тарифов его договора с целью определения стоимости минуты звонка, направления и услуги. Порядок просмотра тарифов задаётся позицией. Первыми в порядке просматриваются (если есть) тарифы поинта, персональные тарифы договора, содержащие поддерево для данного модуля, далее глобальные тарифы.

Соответственно в договоре может быть несколько тарифов на телефонии. Например: Местная связь, Зоновая связь, МГМН-связь, упорядоченных по позиции. При этом тариф МГМН-связи может меняться в зависимости от оператора, выбранного абонентом. При использовании хотчойс (выбора оператора префиксом преднабора) в договоре должны быть установлены все тарифы МГМН-операторов.

Ниже приведён снимок редактора тарифов договора с указанными несколькими тарифами.

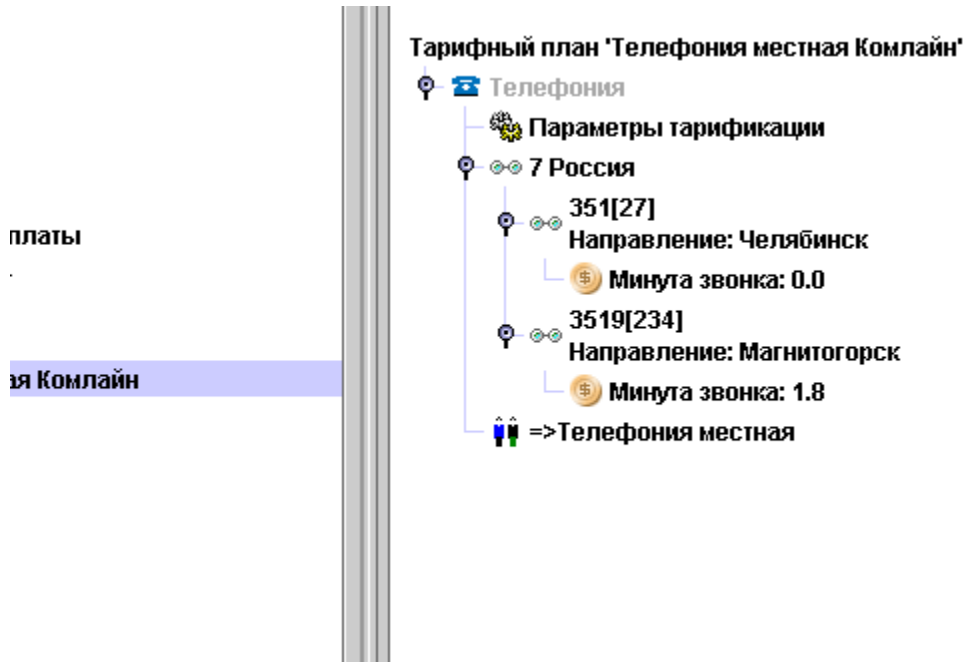
<input type="checkbox"/> Группы тарифов	0	Базовый (для всех и телефонии)	29.09.2006-
<input type="checkbox"/> Скрипт поведения	5	Телефония местная	01.09.2006-
<input type="checkbox"/> Пароль	10	Телефония Совинтел преселект с НДС	01.09.2006-
<input type="checkbox"/> Лимит	0		
<input type="checkbox"/> Режим	дебет		
<input type="checkbox"/> Группы			
<input type="checkbox"/> Организации			
<input type="checkbox"/> Городская сеть			
<input type="checkbox"/> ГТЮ			
<input type="checkbox"/> Телефон			
<input type="checkbox"/> Тарифные планы			
<input type="checkbox"/> Базовый (для всех и			
<input type="checkbox"/> Телефония Совинтел			
<input type="checkbox"/> Телефония местная			
<input type="checkbox"/> Баланс			
<input type="checkbox"/> Входящий остаток	-44791.52		
<input type="checkbox"/> Приход	44791.51		
<input type="checkbox"/> Нарabотка	43823.53		
<input type="checkbox"/> Расход	0.00		
<input type="checkbox"/> Исходящий остаток	-43823.54		
<input type="checkbox"/> Модули			
<input type="checkbox"/> Phone			

Далее следует описание правил построения тарифных планов, описание логики работы всех стандартных тарифных узлов можно найти [здесь](#).

Тарифы на местную связь

Тариф, содержащий поддерево с тарифами на местную связь, должен располагаться в договоре с минимальной позицией. Т.е. при тарификации звонка поинта он должен просматриваться первым. Следом за ним могут следовать тарифы зоновой и МГМН-связи.

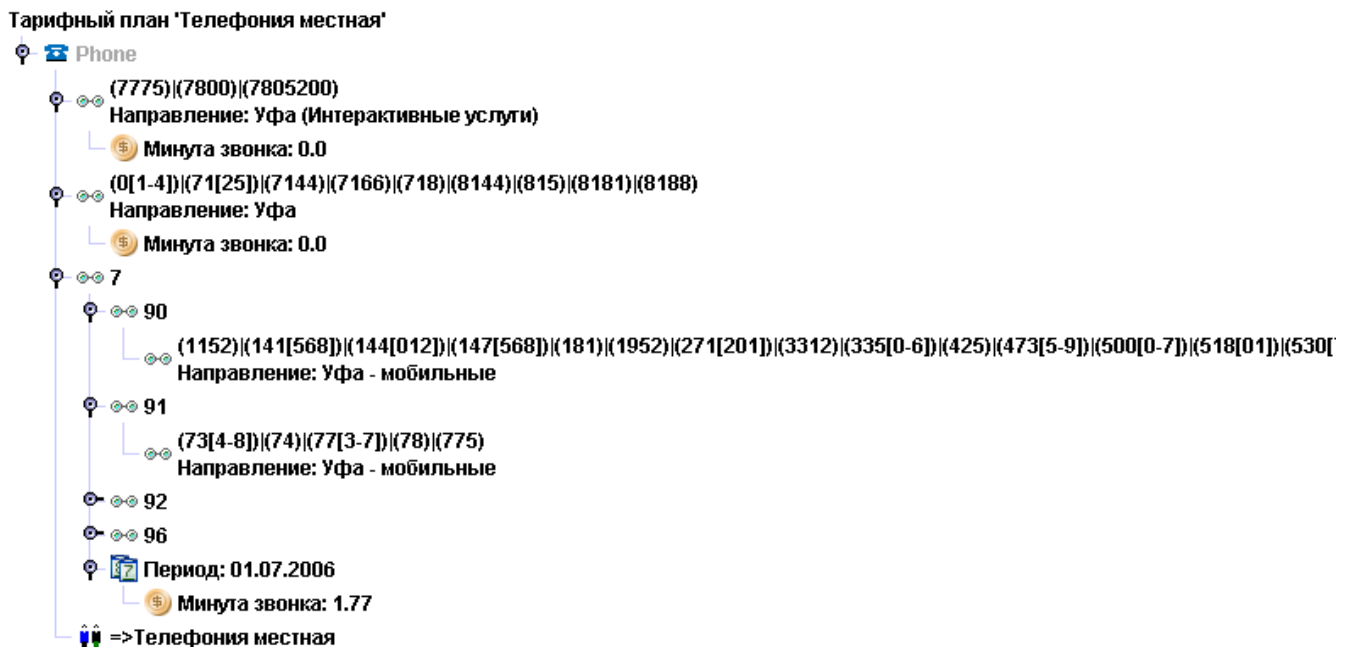
Рассмотрим простейший тариф на местную связь.



Для разбора номера используется узел **Часть префикса**.

Узел **Установка услуги** помещает в ответ тарифного запроса услугу данного звонка. Стоимость минуты может быть заключена в фильтр по типу времени, период. Узел **Параметры тарификации** также может быть добавлен в разные части префикса с различными режимами округления.

Вот пример более сложного местного, тарифа в котором также описываются сотовые префиксы:



Рассмотрим ещё одну линейку из двух тарифов на местную связь - **Комбинированный** и **Поминутный**. У оператора выделено два типа внутреннего трафика - **Внутрисетевой** и **Местный**, при этом цена внутрисетевого постоянна и равна 0.09, а цена местного в поминутном тарифе зависит от времени суток, а в комбинированном - бесплатна первые 450 минут, а далее по определённой цене.

В данных тарифах используется узел **Диапазон наработки**, позволяющий изменять стоимость услуги в зависимости от наработки по какой-либо зоне.

Оба тарифа расширены от базового тарифа **Телефония**, в котором определены 2 зоны в зависимости от префикса. Обратите внимание, что более длинный префикс должен стоять первым. Наследующие тарифы также дополнены поддеревом модуля абонлат, в котором определена стоимость абонлаты за телефонный номер в зависимости от тарифа.

The screenshot displays a web interface for managing tariffs. On the left, a sidebar titled "Тарифные планы" (Tariff Plans) lists several options: "Абонлаты телефония" (Phone subscriptions), "Телефония" (Phone), "Телефония Юр. Лица < 5 номеров" (Phone for legal entities < 5 numbers), "Телефония: Ф: Комби" (Phone: F: Combo), "Телефония: Ф: Неограниченный" (Phone: F: Unlimited), and "Телефония: Ф: Поминутный" (Phone: F: Pay-per-minute). The "Телефония" option is highlighted.

The main content area is titled "Наименование тарифного плана" (Tariff Plan Name) and shows "Телефония". Below this, there are buttons for "Поиск" (Search) and "Управление поддеревьями" (Manage tree). A search input field is present but empty.

The detailed view of the "Тарифный план 'Телефония'" (Tariff Plan 'Phone') is shown below. It features a tree structure with the following nodes:

- Телефония
 - Параметры тарификации
 - 7383275
 - Направление: Novosibirsk
 - Уст. зону: Внутрисетевая
 - 7383
 - Направление: Novosibirsk
 - Уст. зону: Местная

В комбинированном тарифе добавлены 2 узла типа **Зона** в одном из них жёсткая цена, в другом - два диапазона, определяющие 450 минут в месяц бесплатного местного трафика и далее по 0.18 рубля за минуту.

Абонплаты телефония
Телефония
Телефония Юр. Лица < 5 номеров
Телефония Ф: Комби
Телефония Ф: Неограниченный
Телефония Ф: Поминутный

Телефония: Ф: Комби

Поиск Управление поддеревьями

Тарифный план 'Телефония: Ф: Комби'

- Абонплаты [Тарифный план 'Абонплаты телефония']
- Телефония [Тарифный план 'Телефония']
 - Параметры тарификации
 - 7383275
 - Направление: Novosibirsk
 - Уст. зону: Внутрисетевая
 - 7383
 - Направление: Novosibirsk
 - Уст. зону: Местная
 - Зона: Внутрисетевая
 - =>Телефония внутрисетевая
 - Минута звонка: 0.09
 - Зона: Местная
 - Трафик Местная от 0 до 450 мин./месяц безусловно
 - Минута звонка: 0.0
 - Трафик Местная от 451 до 0 мин./месяц безусловно
 - Минута звонка: 0.18
 - =>Телефония местная

В поминутном тарифе внутрисетевой трафик также определён по 0.09, а цена местного зависит от времени суток:

Тарифные планы

- Абонплаты телефония
- Телефония
- Телефония Юр. Лица < 5 номеров
- Телефония: Ф: Комби
- Телефония: Ф: Неограниченный
- Телефония: Ф: Поминутный

Наименование тарифного плана

Телефония: Ф: Поминутный

Поиск Управление поддеревьями

Тарифный план 'Телефония: Ф: Поминутный'

- 🌐 Абонплаты [Тарифный план 'Абонплаты телефония']
- 📞 Телефония [Тарифный план 'Телефония']
 - 🌿 Параметры тарификации
 - 7383275
 - Направление: Novosibirsk
 - 7383
 - Направление: Novosibirsk
 - 🌐 Зона: Внутрисетевая
 - 💰 Минута звонка: 0.09
 - 👤 =>Телефония внутрисетевая
 - 🌐 Зона: Местная
 - 🕒 Тип времени: День
 - 💰 Минута звонка: 0.23
 - 🕒 Тип времени: Ночь
 - 💰 Минута звонка: 0.15
 - 👤 =>Телефония местная

Обратите внимание на то, что внутрисетевая и местная телефония считаются разными услугами и лягут двумя позициями в наработку договора.

Рассмотрим пример модернизации комбинированного тарифа, приведённого ранее, когда в бесплатную квоту входят и внутрисетевой и местный трафики. Базовый тарифный план, обратите внимание, что зонавая и местная телефония теперь установлены в одну зону.

Тарифный план 'Телефония'

- 📞 Телефония
 - 🌿 Параметры тарификации
 - 7383275
 - Направление: Novosibirsk
 - 🌐 Уст. зону: Местная
 - 👤 =>Телефония внутрисетевая
 - 7383
 - Направление: Novosibirsk
 - 🌐 Уст. зону: Местная
 - 👤 =>Телефония местная

Расширяющий комбинированный тариф, оплата превышения общей квоты в 450 минут идёт по разным ценам за внутрисетевой и местный трафики:

☎ Телефония [Тарифный план 'Телефония']

⚙ Параметры тарификации

7383275

Направление: Novosibirsk

🌐 Уст. зону: Местная

👤 =>Телефония внутрисетевая

☎ Трафик Местная от 0 до 450 мин./месяц безусловно

💰 Минута звонка: 0.0

☎ Трафик Местная от 451 до 0 мин./месяц безусловно

💰 Минута звонка: 0.09

7383

Направление: Novosibirsk

🌐 Уст. зону: Местная

👤 =>Телефония местная

☎ Трафик Местная от 0 до 450 мин./месяц безусловно

💰 Минута звонка: 0.0

☎ Трафик Местная от 451 до 0 мин./месяц безусловно

💰 Минута звонка: 0.18

Тарифы на МГМН-связь

Более сложными в реализации являются тарифы МГМН-операторов. Тарифы каждого оператора должны быть реализованы в отдельном плане. Существуют два основных способа тарификации: зонавая и по префиксам.

Тарификация по префиксам

Тарификация по префиксам предполагает внесение в тарифный план дерева префиксов для разбора звонков. Логика тарифа полностью идентична приведённым выше тарифам на местную связь. Например, тариф может выглядеть таким образом:

При построении подобных тарифов целесообразно делать общее дерево префиксов, устанавливая в нем разбивку по направлениям, далее его клиентскими тарифами, определяя цены. Например, базовое дерево тарифов может выглядеть следующим образом.

Тарифный план 'Телефония Совинтел базовый'

☎ —

Набор ограничений:

Часы => 0 - 7

>>>> Льготный

Набор ограничений:

Часы => 8 - 23

>>>> Бизнес

☎ ☎ 1 США

☎ ☎ 2

☎ ☎ 3

☎ ☎ 4

☎ ☎ 6

☎ ☎ 7

☎ ☎ 5

☎ ☎ 8

Направление: Saint Lucia

☎ ☎ 6

☎ ☎ 8

☎ ☎ 8

☎ ☎ 0

☎ ☎ 6

☎ ☎ 7

☎ ☎ (09)(29)

Направление: Dominican Republic

☎ ☎ 9

☎ ☎ ((125)|(154)|(204)|(250)|(289)|(306)|(403)|(416)|(418)|(450)|(506)|(514)|(604)|(613)|(647)|(705)|(709)|(778)|(780)|(807)|(819)|(867)|(886)|(88

Направление: CANADA

☎ ☎ (800)|(822)|(833)|(844)|(855)|(866)|(877)|(88[0128])|(900)

Направление: USA premium

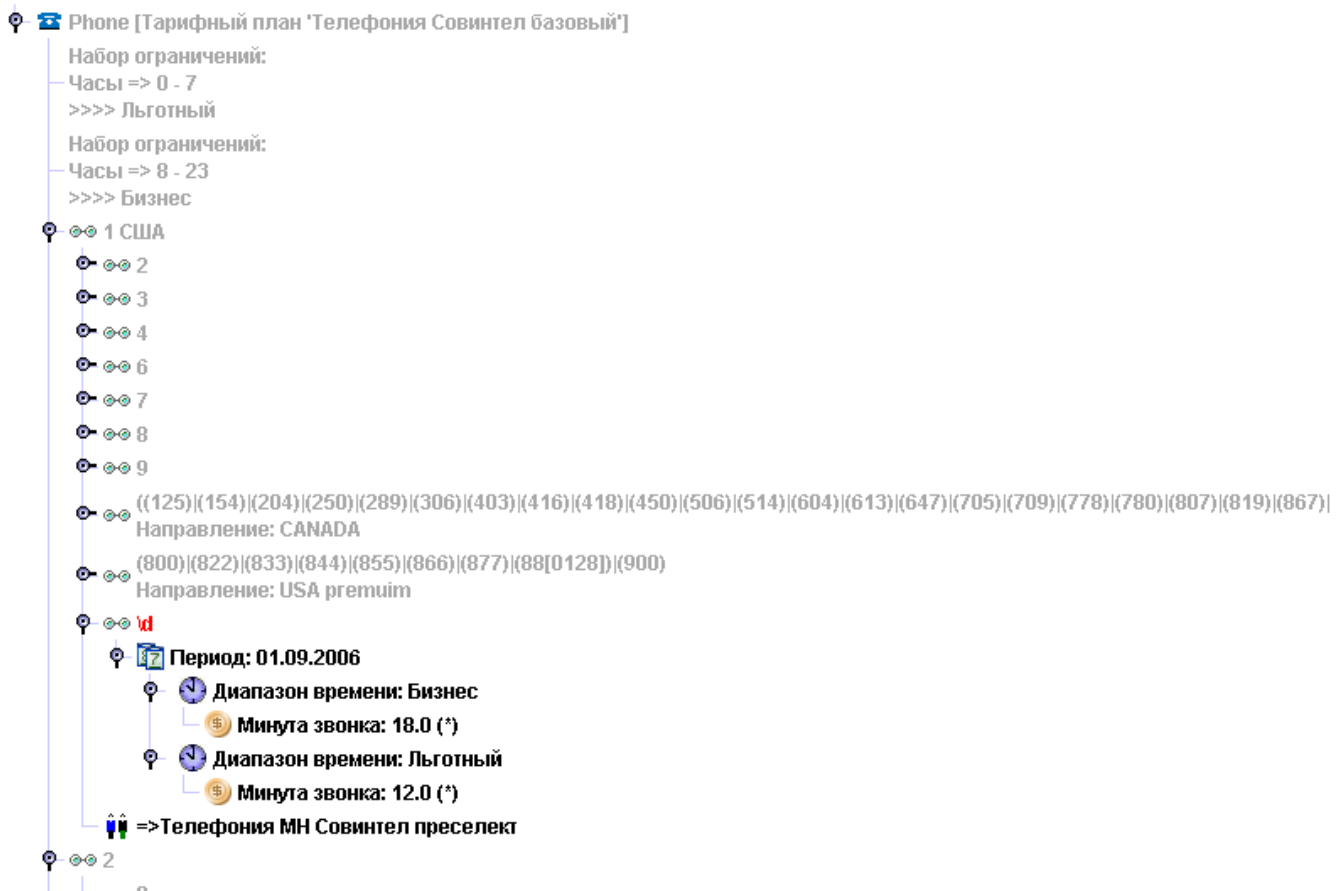
☎ ☎ 14

В верхней части дерева заведены два набора ограничений определяющих 2 диапазона времени тарификации: обычное и льготное время. Далее в наследованном дереве определяются цены для каждого из диапазонов.

Вместо **Набора ограничений** в последних версиях биллинга можно также использовать **Фильтр по типу времени**. Также возможно использование узла **Диапазон префиксов** совместно с узлами типа **Часть префикса**.

Тариф **Телефония Совинтел преселект** - наследуется от базового тарифа в нем добавляются цены

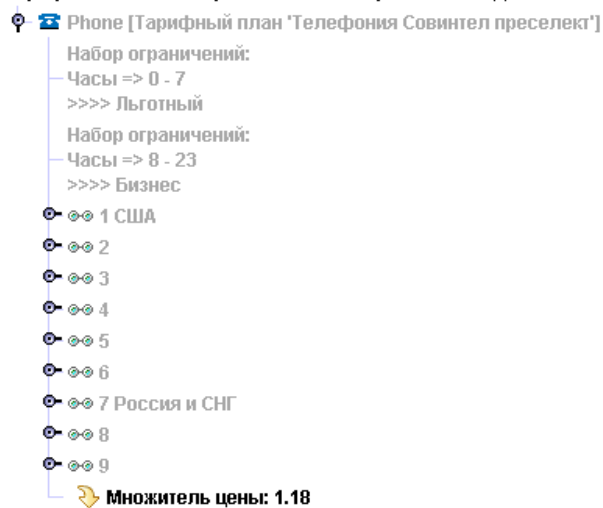
Тарифный план 'Телефония Совинтел преселект'



В данном тарифе определяются цены звонка и виды услуг. Т.к. в данном тарифе встречаются звонки МГ и МН, то узлов установки услуги множество.

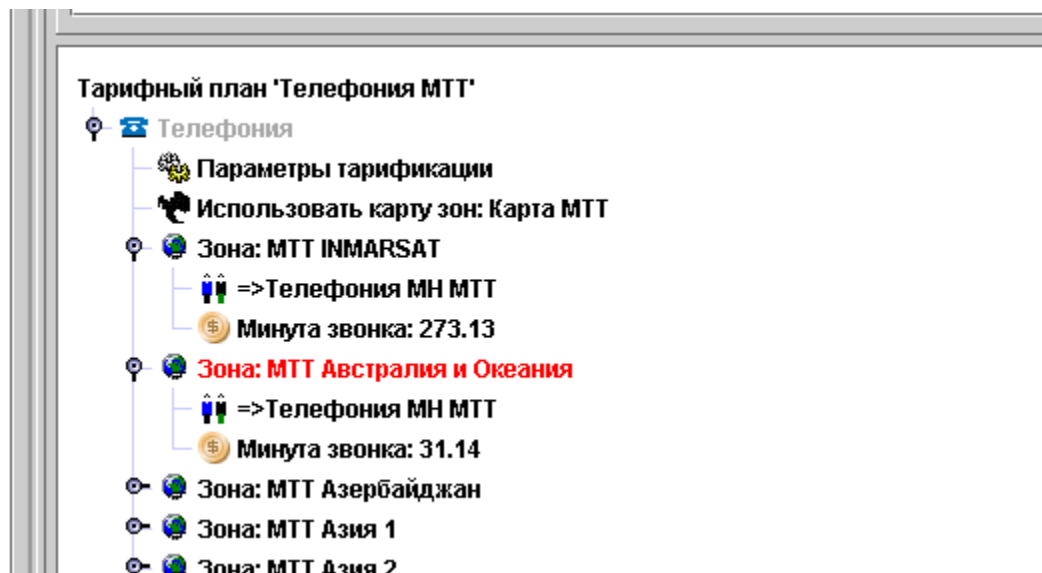
Тариф **Телефония Совинтел преселект с НДС** - наследуется от предыдущего, но в конце добавляется коэффициент умножения цены. Обратите внимание, что для того, чтобы он сработал, цены минут должны быть помечены галочкой **По умолчанию**.

Тарифный план 'Телефония Совинтел преселект с НДС'



Тарификация по зонам

В основу зонной тарификации ложится карта зон, соотносящая коды из справочника географических кодов с зонами равной цены. Про построение карт зон вы можете почитать [ранее](#). В тарифном плане с помощью узла **Использовать карту зон** указывается используемая карта и далее добавляются узлы **Зона** с указанием в каждой стоимости минуты и услуги.






Тарификация по карте цен

В основу этой тарификации ложится карта цен, соотносящая коды и правильные диапазоны из карты с ценами. В тарифном плане с помощью узла **Карта цен** указывается используемая карта. Если префикс найден в этой карте, то звонку также соотносится направление из географических кодов и **выполняются дочерние узлы**. Чтобы назначить цену из этой карты, внутри должен быть узел **Стоимости** с параметром "взять из карты цен".

Тарифный план 'Телефония: Оператор'

 Phone

  Использовать карту цен: Оператор

 Стоимость минуты взять из карты цен

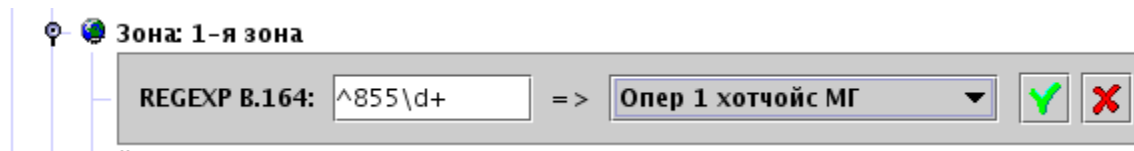
 => Совинтел МГ

Тарификация с несколькими МГМН-операторами

В случае когда МГМН-звонок клиента может быть терминирован на несколько операторов (преселект) встаёт проблема тарификации звонка с одинаковым конечным направлением, но разными ценами. Задача решается созданием отдельного тарифного плана для каждого из возможных операторов и установку их в договор клиента. При тарификации будут пройдены все тарифные планы до первого, в котором будет найдена цена, услуга и направление.

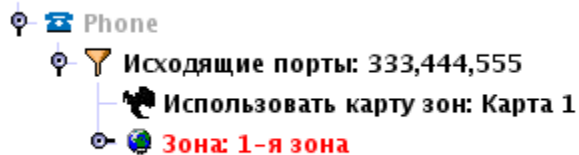
Для того, чтобы тарифный план отрабатывал только на "свои" звонки, можно использовать метод фильтрации по набранному номеру, либо по исходящему порту АТС.

В первом случае во все узлы **Установка услуги** дополнительно указывается **REGEXP**, совпадающий с преселект-набором данного оператора, например:



Во втором случае в начале тарифного плана устанавливается фильтр по портам АТС, исходящие звонки на которые относятся к данному оператору. Несколько портов могут быть указаны через запятую.

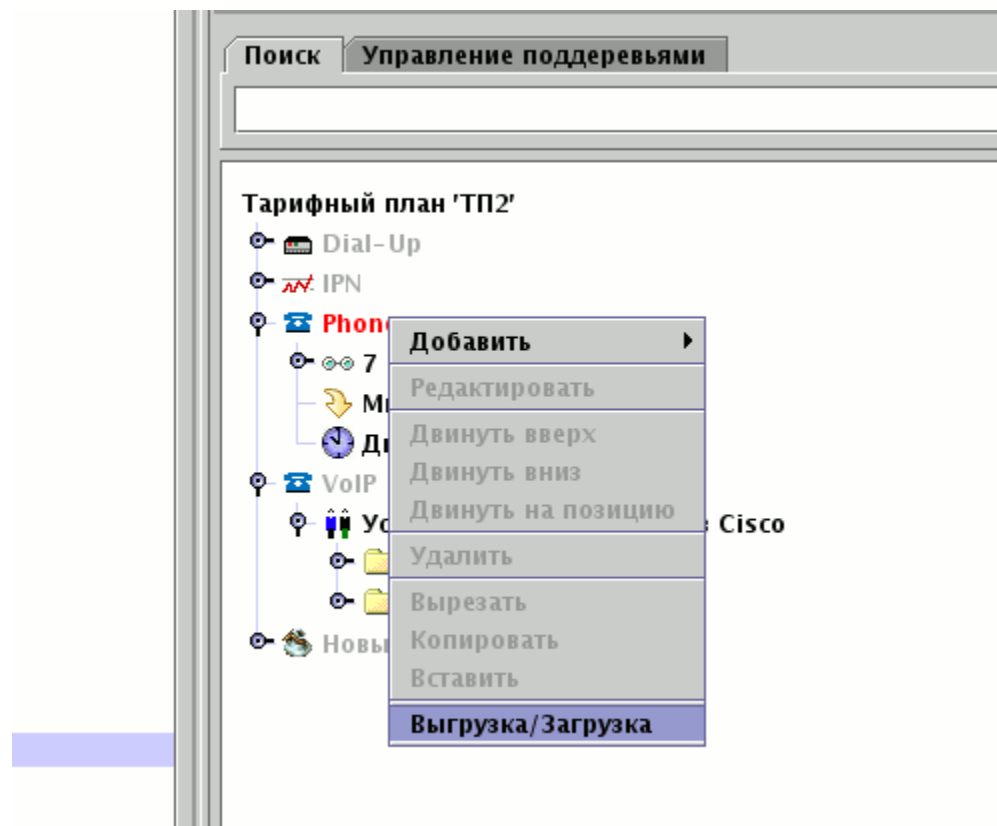
Тарифный план 'New tariff plan'



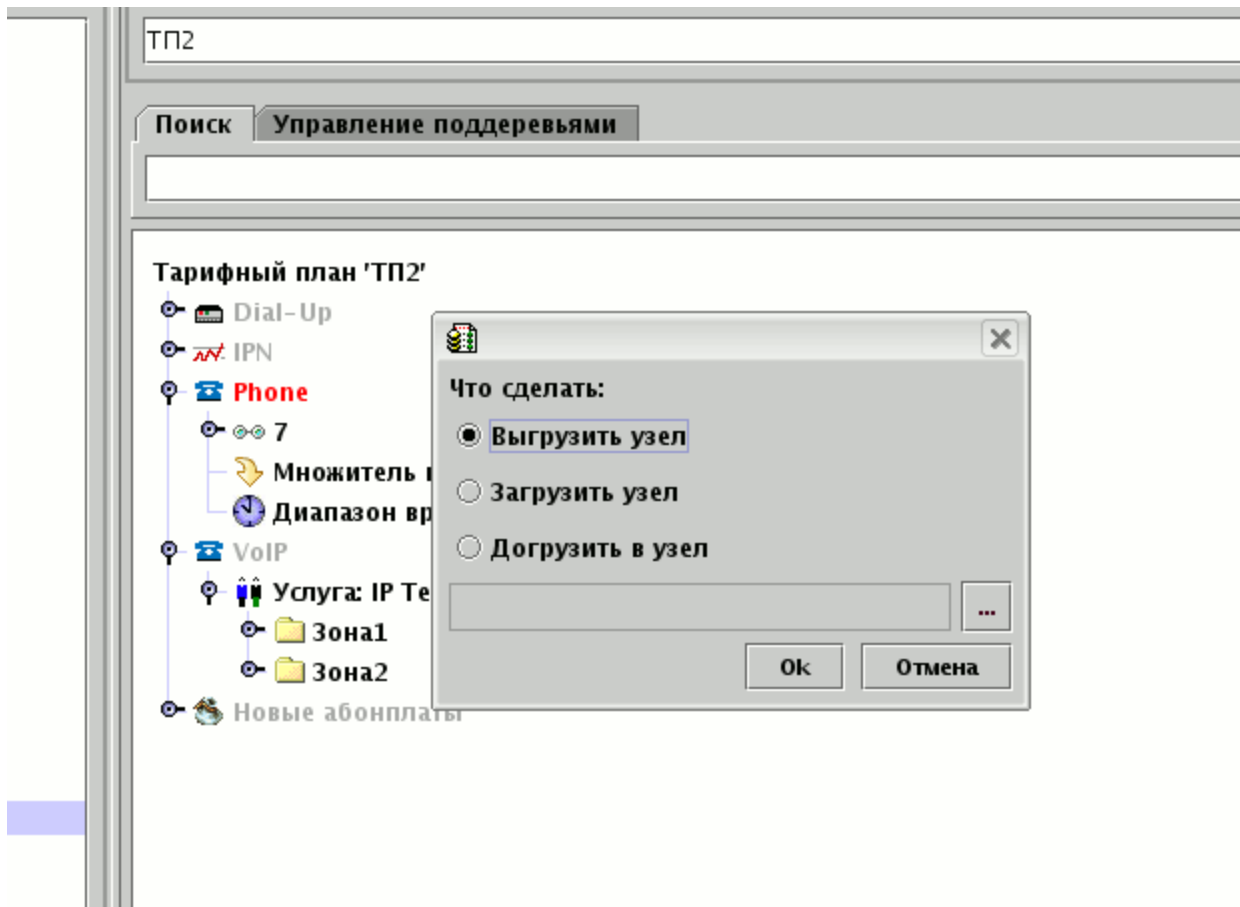
Импорт и экспорт тарифных планов голосовых модулей

Допускается импорт и экспорт только независимых модульных поддеревьев. Т.е. эти тарифы не должны наследовать и не должны наследоваться от других тарифов.

Тарифные планы голосовых модулей позволяют делать экспорт дерева в XML-формат и загрузку. Для произведения выгрузки, либо загрузки необходимо выбрать корневой узел тарифного дерева для модуля телефонии, либо узел **Услуга** для модуля VoiceIP. Далее нажать правую кнопку мыши, выбрать пункт **Выгрузка/Загрузка**.



В появившемся окне выбрать файл и желаемое действие. Выбор **Загрузить узел** перетирает все подузлы и выгружает их из файла, **Догрузить в узел** добавляет узлы из файла за существующими подузлами.



Для того, чтобы определить формат, необходимо создать часть нужного тарифа и выгрузить его. Все названия направлений и зон в файле с выгрузкой идентифицируются по имени. Если при загрузке направление или зона не будут найдены - биллинг создаст их. Поэтому следует быть внимательными с регистром и пробельными символами.

Загрузка тарифов удобна для программной генерации тарифа с его последующей загрузкой.

Специфичные тарифные узлы модуля

При конструировании тарифного плана вы можете использовать [стандартные узлы](#) и перечисленные ниже специфичные для модуля узлы тарифных планов.

Стоимость минуты звонка

Редактор и узел в дереве выглядят следующим образом:



The image shows a configuration editor for the 'Стоимость минуты звонка' (Call minute cost) parameter. It consists of a horizontal bar with the following elements from left to right: the text 'Стоимость', a dropdown menu currently showing 'минуты', an equals sign '=' followed by another dropdown menu, a text input field containing the value '0.33', a checkbox labeled 'по умолчанию' (by default), a green checkmark icon, and a red 'X' icon.


В первом выпадающем списке редактора задаётся стоимость либо **минуты**, либо **звонка**, в зависимости от того, что необходимо. Во втором списке выбирается символ "=", либо **взять из CDR**, что позволяет задать стоимость в узле дерева, либо указать необходимость взятия её из CDR-записи поля **Стоимость звонка**.

Установленная галочка **По умолчанию** определяет, что цена в запросе переопределяется только в случае, если она не установлена узлом по умолчанию предка. Если цена установлена безусловно, либо узлом по умолчанию в этом же дереве, она не переопределяется.

Фильтр по портам

Редактор и узел в дереве выглядят следующим образом:

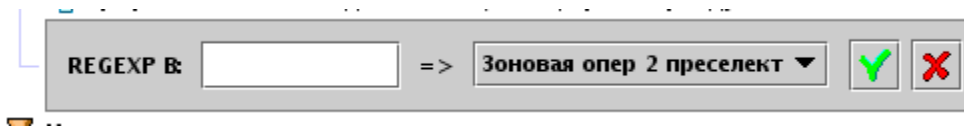
Исходящие порты:	<input type="text" value="5"/>	
Входящие порты:	<input type="text" value="6"/>	<input checked="" type="checkbox" value="true"/> <input type="checkbox" value="false"/>
Комментарий:	<input type="text" value="Комментарий"/>	

 **Исходящие порты: 5**
Входящие порты: 6
Комментарий

Фильтр пропускает запрос внутрь только, если исходящий порт вызова попадает в перечень.

Установка услуги

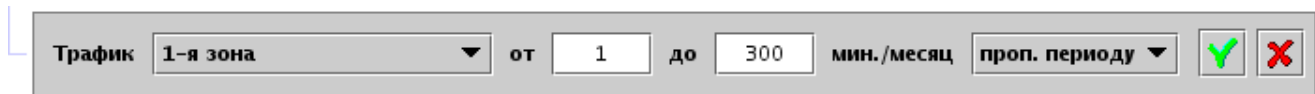
Редактор и узел в дереве выглядят следующим образом:



Узел помещает в тарифный запрос услугу. Поле **REGEXP В** позволяет задать [REGEXP](#)-выражение, с которым будет сравниваться набор клиента, услуга будет установлена только при совпадении регулярного выражения.

Диапазон наработки

Редактор и узел в дереве выглядят следующим образом:



The image shows a configuration panel for a traffic filter. It includes a label 'Трафик', a dropdown menu with '1-я зона' selected, the word 'от', a text input field with '1', the word 'до', a text input field with '300', the unit 'мин./месяц', another dropdown menu with 'проп. периоду' selected, and two buttons: a green checkmark and a red 'X'.

Фильтр пропускает запрос внутрь только, если наработка поинта по указанной зоне лежит в указанном диапазоне. При выборе **пропорционально периоду** объёмы определяются пропорционально доле в месяце периода действия поинта.

Отчёты в клиенте

Для вывода любого отчёта необходимо выбрать месяц и поинты и нажать кнопку с галочкой. Для выбора доступны только те поинты, период которых пересекается с выбранным периодом. Если фильтр по поинту не установлен, то выводится отчёт всех поинтов договора.

Возможна установка фильтра на вывод только платных звонков. Для сохранения, печати или отправки на почту отчёта нажмите кнопку с принтером, дискетой или конвертом.

В нижней области каждого отчёта отображаются количество сессий, суммарная длительность и стоимость. Многостраничные отчёты перематываются стандартным элементом управления.

Отчет по сессиям модуля Phone

Многостраничный отчет выводит сессии выбранного пункта (поинтов) за выбранный месяц. Сортировка - по времени звонка. Может быть сохранен в HTML и CSV-формате. Оформление задаётся шаблонами **phone_login_sessions.xml** и **phone_login_sessions_csv.xml** соответственно. Для сохранения больших отчетов рекомендуется CSV-формат. HTML-отчет также может быть отправлен на почту, либо распечатан.

Параметры | **Отчет** | **Карточки**

Период
 Декабрь 2007
 дни: с [] до []

Пойнты \ Правила
 Номер(а) [73519438769] на ZTE
 Номер(а) [73519438797] на ZTE

Логги
 Сессии | Наработка | Направления | Услуги | Детализация | Только платные

Пойнт	Направление	Услуга	Время входа	Длительность	Длит. округ.	Ст.
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 09:56:42	00:00:33 [33]	00:00:33 [33]	0.00
Номер(а) 73519438769	Новосибирская...	Телефония МГ МТТ	03.12.2007 10:15:50	00:00:42 [42]	00:01:00 [60]	5.42
Номер(а) 73519438769	Новосибирская...	Телефония МГ МТТ	03.12.2007 10:17:06	00:00:47 [47]	00:01:00 [60]	5.42
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 10:35:17	00:00:41 [41]	00:00:41 [41]	0.00
Номер(а) 73519438769	Новосибирская...	Телефония МГ МТТ	03.12.2007 10:38:26	00:00:39 [39]	00:01:00 [60]	5.42
Номер(а) 73519438769	Новосибирская...	Телефония МГ МТТ	03.12.2007 10:41:48	00:00:55 [55]	00:01:00 [60]	5.42
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 10:45:25	00:02:53 [173]	00:02:53 [173]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 11:10:58	00:02:12 [132]	00:02:12 [132]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 12:01:48	00:11:27 [687]	00:11:27 [687]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 12:46:31	00:01:35 [95]	00:01:35 [95]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 12:48:22	00:01:26 [86]	00:01:26 [86]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 13:17:08	00:00:08 [8]	00:00:08 [8]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 13:49:02	00:00:09 [9]	00:00:09 [9]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 14:55:08	00:00:09 [9]	00:00:09 [9]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:10:02	00:02:14 [134]	00:02:14 [134]	0.00
Номер(а) 73519438769	Новосибирская...	Телефония МГ МТТ	03.12.2007 15:13:14	00:06:03 [363]	00:07:00 [420]	5.42
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:20:43	00:04:45 [285]	00:04:45 [285]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:26:20	00:00:09 [9]	00:00:09 [9]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:37:06	00:05:22 [322]	00:05:22 [322]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:59:14	00:01:33 [93]	00:01:33 [93]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 16:05:46	00:00:52 [52]	00:00:52 [52]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 16:15:05	00:00:51 [51]	00:00:51 [51]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 16:17:12	00:01:22 [82]	00:01:22 [82]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 16:20:31	00:04:06 [246]	00:04:06 [246]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	04.12.2007 10:12:10	00:00:48 [48]	00:00:48 [48]	0.00

Кол-во сессий: 78 **Сумма (сек.):** 02:22:57 [8577] [143] **Сумма (сек. окр.):** 02:29:59 [8999] [150] **Сумма (руб.):** 161.68

Телефония

Поиск договоров | ВК-1056

При двойном клике по строке с сессией отображается строка из исходного CRD-лога, по которому была создана данная сессия.

Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 12:48:22	00:01:26 [86]	00:01:26 [86]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 13:17:08	00:00:08 [8]	00:00:08 [8]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 13:49:02	00:00:09 [9]	00:00:09 [9]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 14:55:08	00:00:09 [9]	00:00:09 [9]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:10:02	00:02:14 [134]	00:02:14 [134]	0.00
Номер(а) 73519438769	Новосибирская...	Телефония МГ МТТ	03.12.2007 15:13:14	00:06:03 [363]	00:07:00 [420]	5.42
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:20:43	00:04:45 [285]	00:04:45 [285]	0.00
Номер(а) 73519438769	Магнитогорск	Телефония местн...	03.12.2007 15:26:20	00:00:09 [9]	00:00:09 [9]	0.00

Исходный лог

Источник	Время входа	Длительность	С номера	С номера Е.164	На номер	На номера Е...	С порт
ZTE	03.12.2007 13:49:02	00:00:09 [9]	3519438769	73519438769	460644	73519460644	521

Кол-во сессий: 108 **Сумма (сек.):** 03:15:27 [11727] [195] **Сумма (сек. окр.):** 03:27:30 [12450] [208] **Сумма (руб.):** 308.02

Заккрыть

Наработка

Односторонний отчёт выводит суммарное количество звонков, длительность и стоимость сессий с группировкой по поинтам. Может быть сохранен в HTML и CSV-формате, распечатан, либо отправлен на почту. Шаблоны преобразования - **phone_login_direct.xml**, **phone_login_direct_csv.xml**.

Параметры Отчет Карточки

Период
Декабрь 2007
дни: с [] до []

Пойнты \ Правила
Пойнты
 Номер(a) [73519438769] на ZTE
 Номер(a) [73519438797] на ZTE

Все Сброс Инверт.

Логи
Сессии **Наработка** Направления Услуги Детализация Только платные 1 из 1 [25]

Пойнт/Правило	Количество	Длительность	Длит. округ.	Стоимость
Номер(a) 73519438797	30	00:52:30 [3150]	00:57:31 [3451]	146.34
Номер(a) 73519438769	78	02:22:57 [8577]	02:29:59 [8999]	161.68

Кол-во сессий: 108 Сумма (сек.): 03:15:27 [11727] [195] Сумма (сек. округ.): 03:27:30 [12450] [208] Сумма (руб.): 308.02

Телефония
Поиск договоров X BK-1056

Направления

Односторонний отчёт выводит суммарное количество звонков, длительность и стоимость сессий с группировкой по поинтам и направлениям. Может быть сохранен в HTML и в CSV-формате, распечатан, либо отправлен на почту. Шаблоны преобразования в HTML - **phone_login_direct.xml**, **phone_login_direct_csv.xml**. При двойном клике по строке таблицы выводится отчёт по сессиям с фильтром по выбранному в строке поинту и направлению.

Параметры | **Отчет** | **Карточки**

Период: Декабрь 2007
дни: с [] до []

Пойнты \ Правила

- Номер(а) [73519438769] на ZTE
- Номер(а) [73519438797] на ZTE

Логги: Сессии | **Наработка** | **Направления** | Услуги | Детализация | Только платные | 1 из 1 [25]

Пойнт	Направление	Количество	Длительность	Длит. округ.	Стоимость
Номер(а) 73519438797	Магнитогорск	21	00:30:31 [1831]	00:30:31 [1831]	0.00
Номер(а) 73519438797	Новосибирская об...	9	00:21:59 [1319]	00:27:00 [1620]	146.34
Номер(а) 73519438769	Магнитогорск	64	01:57:59 [7079]	01:57:59 [7079]	0.00
Номер(а) 73519438769	Новосибирская об...	13	00:22:48 [1368]	00:29:00 [1740]	157.18
Номер(а) 73519438769	Челябинская облас...	1	00:02:10 [130]	00:03:00 [180]	4.50

Кол-во сессий: 108 | Сумма (сек): 03:15:27 [11727] [195] | Сумма (сек. округ.): 03:27:30 [12450] [208] | Сумма (руб.): 308.02

Телефония

Поиск договоров | ВК-1056

Услуги

Односторонний отчёт выводит суммарное количество звонков, длительность и стоимость сессий с группировкой по поинтам и услугам. Может быть сохранен в HTML и в CSV-формате, распечатан, либо отправлен на почту. Шаблон преобразования в HTML - **phone_login_service.xml**, **phone_login_service_csv.xml**. При двойном клике по строке таблицы выводится отчёт по сессиям с фильтром по выбранному в строке поинту и услуге.

Параметры Отчет Карточки

Период
Декабрь 2007
дни: с до

Пойнты \ Правила
Пойнты
 Номер(а) [73519438769] на ZTE
 Номер(а) [73519438797] на ZTE

Все Сброс Инверт.

Телефония

Логи
Сессии **Наработка** Направления Услуги Детализация Только платные 1 из 1 [25]

Пойнт	Услуга	Количество	Длительность	Длит. округ.	Стоимость
Номер(а) 73519438797	Телефония местная	21	00:30:31 [1831]	00:30:31 [1831]	0.00
Номер(а) 73519438797	Телефония МГ МТТ	9	00:21:59 [1319]	00:27:00 [1620]	146.34
Номер(а) 73519438769	Телефония местная	64	01:57:59 [7079]	01:57:59 [7079]	0.00
Номер(а) 73519438769	Телефония МГ МТТ	13	00:22:48 [1368]	00:29:00 [1740]	157.18
Номер(а) 73519438769	Телефония зонавая...	1	00:02:10 [130]	00:03:00 [180]	4.50

Кол-во сессий: 108 Сумма (сек.): 03:15:27 [11727] [195] Сумма (сек. округ.): 03:27:30 [12450] [208] Сумма (руб.): 308.02

Поиск договоров x BK-1056

Детализация

Отчет выводит сессии, собранные в следующем формате:

```
1
N
  A: , ,
  B
1
K
  B: , ,
  B: , ,
  B....
```

При переходе отчёта по услуге, либо отчёта по номеру на несколько страниц, строки с суммами повторяются на каждой странице отчёта. Номера телефонов берутся непосредственно из звонков, привязанных к поинтам договора.

Отчет может быть сохранен в HTML и CSV-форматах. HTML-формат может быть распечатан или отправлен на почту. Шаблоны преобразования в HTML и CSV - [phone_login_service_sessions.xml](#) и [phone_login_service_sessions_csv.xml](#) соответственно.

Период
Декабрь 2007
дни: с [] до []

Логи
Сессии | Нарботка | Направления | Услуги | **Детализация** | Только платные | 4 из 5 [25]

Время входа	Направление	Номер телефона В	Длит. округ.	Стоимость
Услуга: Телефония местная Номер телефона А: 73519438769				
06.12.2007 11:33:47	Магнитогорск	73519376249	00:01:23 [83]	0.00000
06.12.2007 11:43:57	Магнитогорск	73519211731	00:09:00 [540]	0.00000
06.12.2007 12:09:56	Магнитогорск	73519247503	00:03:11 [191]	0.00000
06.12.2007 12:35:22	Магнитогорск	73519237795	00:01:01 [61]	0.00000
06.12.2007 12:55:55	Магнитогорск	73519376249	00:00:32 [32]	0.00000
06.12.2007 12:58:08	Магнитогорск	73519229123	00:01:46 [106]	0.00000
06.12.2007 13:02:51	Магнитогорск	73519249500	00:00:02 [2]	0.00000
06.12.2007 13:11:52	Магнитогорск	73519249500	00:00:02 [2]	0.00000
06.12.2007 13:17:28	Магнитогорск	73519376249	00:01:08 [68]	0.00000
06.12.2007 16:05:02	Магнитогорск	73519237795	00:00:44 [44]	0.00000
Итого по номеру: кол-во 64, сумма 0.00, окр. длит. (мин) 118				
Итого по услуге: кол-во 85, сумма 0.00, окр. длит. (мин) 149				
Услуга: Телефония МГ МТТ Номер телефона А: 73519438797				
03.12.2007 09:36:36	Новосибирская область	73832718988	00:01:00 [60]	5.42000
03.12.2007 09:37:17	Новосибирская область	73832718988	00:03:00 [180]	16.26000
04.12.2007 12:41:29	Новосибирская область	73832718988	00:07:00 [420]	37.94000
05.12.2007 10:39:50	Новосибирская область	73832718988	00:03:00 [180]	16.26000
05.12.2007 15:37:42	Новосибирская область	73832718988	00:04:00 [240]	21.68000
05.12.2007 16:07:52	Новосибирская область	73832717714	00:02:00 [120]	10.84000
06.12.2007 09:42:42	Новосибирская область	73832714964	00:01:00 [60]	5.42000
06.12.2007 11:27:52	Новосибирская область	73832718880	00:05:00 [300]	27.10000
06.12.2007 13:03:21	Новосибирская область	73832718880	00:01:00 [60]	5.42000
Итого по номеру: кол-во 9, сумма 146.34, окр. длит. (мин) 27				
Номер телефона А: 73519438769				
03.12.2007 10:15:50	Новосибирская область	73832717562	00:01:00 [60]	5.42000
03.12.2007 10:17:06	Новосибирская область	73832717959	00:01:00 [60]	5.42000

Кол-во сессий: 108 **Сумма (сек.):** 03:15:27 [11727] [195] **Сумма (сек. округ.):** 03:27:30 [12450] [208] **Сумма (руб.):** 308.02

Телефония
Поиск договоров | ВК-1056

Входящие сессии

Учет входящих сессий производится только при установке в конфигурации модуля переменной `store.incoming.calls=1`.

Параметры | Отчет | Карточка | HelpDesk | СММ | Документы

Период: Декабрь 2008
дни с: до:

Пойеты \ Пришла
Пойеты
 Номер(ы) (78463121532) из local1

Вс | Сброс | Искать

Логи
Сессии | Наработка | Направление | Услуги | Детализация | Входящие | 1 из 1 [25]

Номер(ы)	Пойет	Время входа
Номер(ы) 78463121532		01.12.2008 00:44:59
Номер(ы) 78463121532		01.12.2008 00:47:30
Номер(ы) 78463121532		01.12.2008 00:48:27

Итого: 3 сессии, 00:00:00 (0) [0] суммарно, 00:00:00 (0) [0] в среднем, 0 руб.

Phone | Бухгалтерия

Х Поиск договоров | Х x00 | Х x0000 | Х SQLEditor | Х Y1_Sovintel | Х Y1

Отчёты в Web-интерфейсе

Логика отчётов абсолютно идентична отчётам в клиенте биллинга.

Учёт операторского трафика

Редактирование правил

Для подключения модуля телефонии в договор добавьте одну или несколько услуг модуля к договору. Для модуля телефонии в данный момент наличие той или иной услуги модуля в договоре кроме подключения модуля к договору не означает.

Понятие правила в модуле означает фильтр, извлекающий звонки определённого типа из потока звонков первичного лога. Правила, как и поинты, привязываются к договору на соответствующей вкладке. Для редактирования используется собственная панель инструментов, расположенная над таблицей.

The screenshot displays a software interface for managing rules. On the left, a tree view lists various modules, with 'Phone' selected. The main window is divided into tabs: 'Свойства' and 'Услуги' at the top, and 'Пойнты' and 'Правила' below. The 'Правила' tab is active, showing a table with the following data:

Описание	Период	Комментарий
Местное завершение 1 на local1	01.08.2006 -	

At the bottom of the interface, there is a search bar labeled 'Поиск договоров' with the value 'x0000' entered.

Имя правила формируется из АТС + привязанной услуги, либо используется алиас, если он указан.

Пойнты Правила

Редактирование

АТС

local1
TEST

Комментарий

Основные Префиксы

Услуга
Местное завершение 1 ▼ Период с 01.08.2006 по

Алиас

С портов

На порты

REGEXP #A.164

NOT REGEXP #A.164

REGEXP #B.164 ^7351

NOT REGEXP #B.164

Объект
Точка на (ghf) ▼

Ok Отмена

Обязательными к заполнению являются код услуги, АТС, к которой привязано правило. Период действия правила влияет на выбор правил, участвующих в обработке лога за конкретную дату. Алиас используется для ввода идентификатора правила, если алиас не введён, правило называется в виде Услуга + АТС.

Правило представляет из себя набор полей фильтров, связанных условием И. Т.е. звонок будет отнесён к правилу только при совпадении всех заполненных в нем фильтров. Фильтры применяются к полям CDR, описанным в [формате лога](#).

В качестве фильтров могут быть указаны:

- 1) REGEXP-маска номера В.164 и А.164 из первичной информации;
- 2) REGEXP-маски, под которые не должны подпадать номера В.164 и А.164 (исключающие шаблоны)№⁹
- 3) порты (см. формат первичного лога) исходящие и входящие через запятую, условие срабатывает при совпадении порта хотя бы с одним из перечисленных;
- 4) диапазоны префиксов А.164 и В.164 номеров.

Редактирование

АТС

local1
TEST

Комментарий

Основные **Префиксы**

С префиксов

Исключить с префиксов

На префиксы
7351|28-29,2478,2479

Исключить на префиксы

Ok Отмена

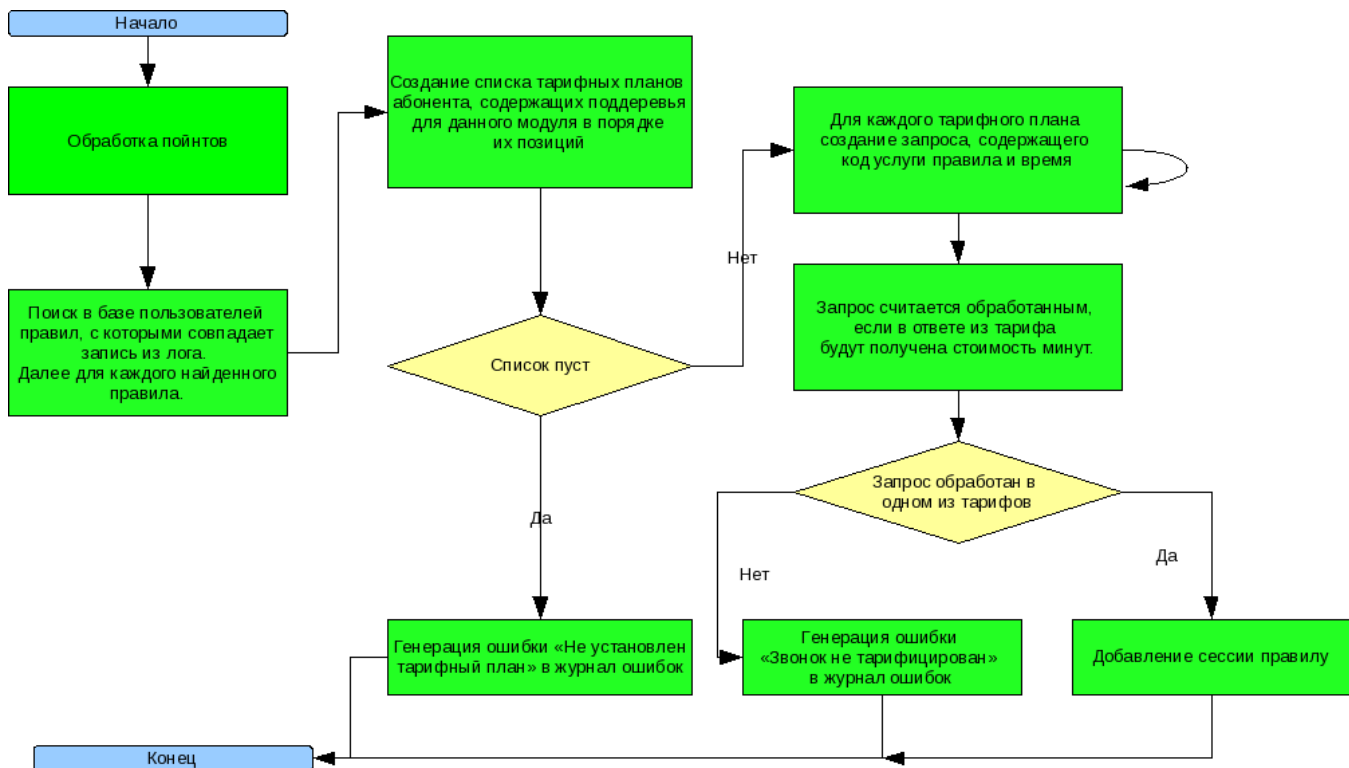
00

Диапазоны префиксов указываются следующим образом: <общий префикс>|диапазоны, либо единичные префиксы через запятую.

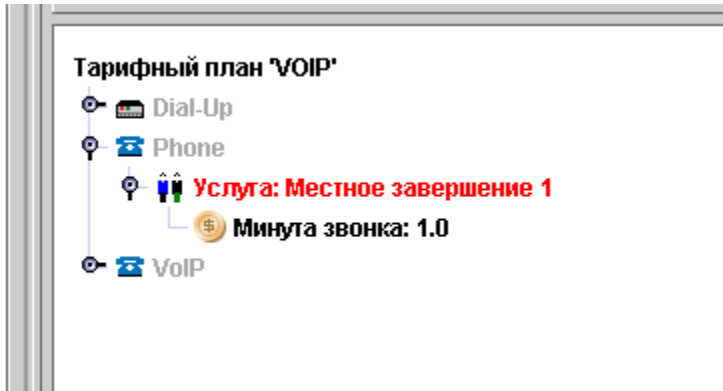
В приведённом на скриншоте примере указаны префиксы 735128-735129,73512478,73512479

5) диапазоны префиксов, в которые не должны попадать А.164 и В.164 номера.

Обработка правил производится для каждой CDR-записи часового лога после обработки поинтов. Производится последовательный просмотр правил, привязанных к обрабатываемой АТС и действующих в сутки, к которым относится лог. Обработка происходит по следующему алгоритму:



Как видно из схемы, в отличие от поинтов, в тарифном плане оператора указываются только услуги и цены за минуту. Например, тариф оператора может выглядеть так:



При заведении операторских договоров необходимо руководствоваться следующей логикой: правила добавляются тому, кто должен.

Например, мы завершаем звонки на некоего оператора через 5-ый порт некой АТС. Заводится договор ОПЕРАТОР Я. В редакторе модулей и услуг добавляется услуга "Завершение на некоего оператора". В правилах договора ОПЕРАТОР Я прописывается, что звонок на "некой АТС" с любого порта на 5-ый есть услуга "Завершение на некоего оператора". В тарифе определяется стоимость данной услуги.

В результате в договоре ОПЕРАТОР Я будет наработка по услуге "Завершение на некоего оператора" - ваш долг данному оператору за предоставленную им услугу.

В случае когда ОПЕРАТОР X завершает на вас и подключён он через некоторые порты АТС Y, то зачастую проще рассмотреть оператора как клиента, тем более, если стоимость завершения разнится на разные префиксы. В этом случае заводится поинт на данные порты и далее тарифы как для клиента, но установка услуги производится "Завершение на Z".

Услуга инициации вызова для какого-либо оператора решается также правилом. Но т.к. за данную услугу платит оператор вам, то правило добавляется в договор оператора.

При использовании в договоре одновременно поинтов и правил необходимо, чтобы тарифы правил по позициям были выше тарифов поинтов, иначе правило может обсчитаться по тарифу поинта.

Отчёты операторов

Для операторских правил доступны все [отчёты поинтов](#) за исключением отчёта по направлениям, т.к. операторским сессиям не сопоставляются направления. При сохранении больших отчётов по сессиям следует использовать исключительно CSV-формат.

Транзитные операторы

С точки зрения биллинга настройка тарификации транзитных операторов не отличается от настройки и тарификации операторских правил, за исключением того, что при тарификации в тарифе доступны узлы-префиксы, карта зон, т.е. возможно тарифицировать операторов по направлениям.

Операторские отчеты

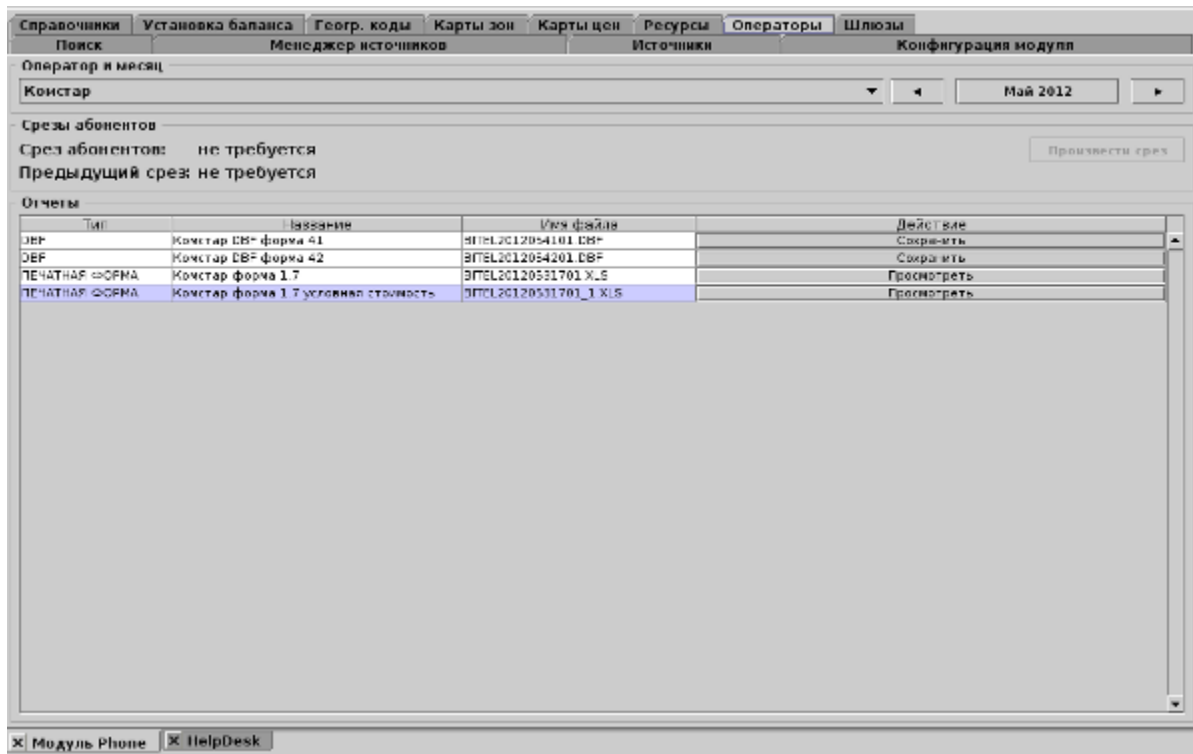
Общие сведения

В модуле телефонии возможно генерировать отчеты операторам дальней связи по агентским договорам. Форматы отчетов "защиты" в модуле, для добавления поддержки новых операторов обратитесь к разработчикам. Поддерживаемые форматы отчетности:

- Комстар;
- Совинтел (Вымпелком);
- МТТ;
- Инфолада;
- Ростелеком.

Описание поддерживаемых форматов вы можете найти на странице [Wiki](#).

Генерация отчетности осуществляется на вкладке **Операторы** модуля.



Для каждого оператора в конфигурацию модуля добавляются записи с ключами вида **operator.<n>.<param_rest>**, описывающие коды параметров, тарифов и прочую информацию для выгрузки отчетности.

Где:

- **<n>** - уникальный числовой код оператора в конфигурации;
- **<param_rest>** - наименование параметра

Отчёты из перечня операторской отчетности можно либо загрузить, либо просмотреть и распечатать прямо в интерфейсе клиента. В последующих разделах описывается настройка конфигурации для поддерживаемых форматов отчетности. Некоторые отчёты требуют выполнения при каждом съёме срезов абонентской базы.

Совинтел (ВымпелКом)

Поддержаны отчёты:

- Изменения по абонентам.
- Отчетность по абонентам.
- Отчет о реализованных услугах.
- Акт сдачи-приемки услуг.

Параметры конфигурации оператора.

Конфигурация

```
#
operator.<n>.title=
operator.<n>.class=bitel.billing.server.phone.oper.bean.sovintel.Operator
#
operator.<n>.code=
#
operator.<n>.contract.date="01" 2006 .
operator.<n>.min.account=
operator.<n>.act.city=
operator.<n>.act.preamble= « » ( - «»), ..., 001/01/01 «01» 2006 ., , ... ( - «»), ..., ,
«», - «», - :
operator.<n>.act.report.date="1" 2006 .
#
# (10, 12, 14)
#
operator.<n>.service.10=
#
operator.<n>.service.12=
#
operator.<n>.service.14=
# ( )
operator.<n>.subContract=1
# / ,
operator.<n>.tariff.list=
#
# ( BIL)
operator.<n>.bill.moduleId=
# - (10, 12, 14), : 500:14,501:12,502:10
operator.<n>.bill.positions=
operator.<n>.lastPayDay=20
#
operator.<n>.contract=FW000
#
# () : /
operator.<n>.param.name=
# () , : 25,27
operator.<n>.param.address=
#
operator.<n>.param.inn=
#
operator.<n>.param.kpp=
# ""
operator.<n>.param.diplomat=
# " "
operator.<n>.param.notResident=
# ""
operator.<n>.param.info=
operator.<n>.country.code=
operator.<n>.location.code=
```

МТТ

Поддержаны отчёты:

- Объем услуг междугородней связи, оказанных автоматическим способом (по направлениям);
- Объем услуг международной связи, оказанных автоматическим способом (по направлениям);
- Детализация уступаемых прав требования по оплате услуг связи.

Параметры конфигурации оператора.

Конфигурация

```
#
operator.<n>.title=
operator.<n>.class=bitel.billing.server.phone.oper.bean.mtt.Operator
#
#
operator.<n>.yield_rights.mg.sids=
#
operator.<n>.yield_rights.mn.sids=
#
operator.<n>.contract_groups=
# -
operator.<n>.yield_rights.contract=title
#
# yield_rights.contract.pid=
# -
operator.<n>.yield_rights.name=comment
#
# yield_rights.name.pid=
#
#
operator.<n>.service.mg.sids=
# -
operator.<n>.service.mg.mode=zone_map
#
operator.<n>.service.mg.mapId=
# Regexp B (E.164)
operator.<n>.service.mg.regexp=^( [012345689].*|771|772|7700|7701|7702|7705|7707|7760|7761|7762|7763|7777)
# Regexp- B (E.164)
operator.<n>.service.mg.notRegexp=
# L
#operator.<n>.service.mg.exception.1.code=
#operator.<n>.service.mg.exception.1.regexp=
#operator.<n>.service.mg.exception.1.notRegexp=
#
#
operator.<n>.service.mn.sids=
# -
operator.<n>.service.mn.mode=zone_map
#
operator.<n>.service.mn.mapId=
# Regexp B (E.164)
operator.<n>.service.mn.regexp=
# Regexp- B (E.164)
operator.<n>.service.mn.notRegexp=^( [012345689].*|771|772|7700|7701|7702|7705|7707|7760|7761|7762|7763|7777)
#
#operator.<n>.service.mn.exception.1.code=
#operator.<n>.service.mn.exception.1.regexp=
#operator.<n>.service.mn.exception.1.notRegexp=
```


Комстар

Поддержаны отчёты:

- Форма 41;
- Форма 42;
- Форма 1.7;
- Форма 1.7 условная стоимость;
- Форма 1.7 с разделением юр.лиц на бюджет и хозрасчёт.

Параметры конфигурации оператора.

Конфигурация

```
#
operator.<n>.title=
#
operator.<n>.class=ru.bitel.bgbilling.modules.phone.server.oper.komstar.Komstar
#
operator.<n>.oper.code=VTKM
#
# " " , " " , " "
operator.<n>.service.l=100500
#
#
operator.<n>.bill.module.id=5
# " " ( 1.7)
operator.<n>.service.mg=0
# " " ( 1.7)
operator.<n>.service.mn=0
# " " ( 1.7)
operator.<n>.service.vz=264
# , ( ).
# : , ( 1.7 .)
operator.<n>.split.ul.bud.listparameter=77,88
#
#-----
# 4.1
#-----
# () -, 4.1 ( ), -
operator.<n>.form.41.invoice.types=
# , , :
operator.<n>.form.41.inn.pid=
operator.<n>.form.41.kpp.pid=
operator.<n>.form.41.address.pid=
operator.<n>.form.41.title.pid=-1
#
#
operator.<n>.form.41.fz.need.total=1
operator.<n>.form.41.fz.contract.title=100000
operator.<n>.form.41.fz.invoice.title=01#111
#
#-----
# 4.2
#-----
# - ,
# id - 17
operator.<n>.form.42.bill.types=
# ,
operator.<n>.form.42.inn.pid=
operator.<n>.form.42.fio.pid=
operator.<n>.form.42.address.pid=
```

Ростелеком

Поддержаны отчёты:

- Форма 1.4;
- Форма 1.7;

Параметры конфигурации оператора.

Конфигурация

```
# 1.4 ( ), 1.7 ( , )
operator.6.title=
operator.6.oper.code=BITEL
operator.6.service.mg=84
operator.6.service.mn=104
operator.6.service.vz=83
operator.6.class=ru.bitel.bgbilling.modules.phone.server.oper.rostelecom.Rostelecom
# , ( ).
# : ,
# 1.4, 1.7
operator.6.split.ul.bud.listparameter=77,88
# regexp-, , 1.2 to_number_164
# : regexp.mobile=^79[0-9]+,^7666[0-9]+
operator.6.regexp.mobile=^79[0-9]+
```

Тарификация при работе по агентской схеме

Агентская схема предполагает, что клиент, совершая звонки, может потреблять услуги нескольких операторов, с которыми у оператора местной связи, обслуживающего клиента, заключены агентские договоры. Нарботка одного и того же телефонного номера должна быть отнесена к различным договорам, по каждому из которых, в общем случае, ведется свой баланс и работа с задолженностью.

Для поддержки агентской схемы в модуле телефонии используются независимые субдоговоры. Номер абонента добавляется на супердоговор, одновременно являющимся договором на услуги местной связи. В супердоговор заносятся тарифные планы на местную связь.

Далее к супердоговору соотносят один или несколько независимых субдоговоров с тарифами конкретных операторов. Номера супердоговора "наследуются" субдоговором, период субдоговора определяет период отношений данного оператора с клиентом.

При тарификации последовательно просматриваются тариф основного договора и тарифы всех независимых субдоговоров, исходя из чего звонок соотносится либо к супердоговору (местный звонок, отработал тариф местной связи), либо к одному из субдоговоров. На снимках экрана ниже изображена связка супердоговор-субдоговор, тарифы и отчеты в каждом из них.

Скриншот интерфейса программы, отображающий детали договора. В верхней части видны вкладки: Параметры, Отчет, Карточки, HelpDesk, CRM, Документы. В центре заголовок договора: Т1 | Гайдеко Илья Валерьевич. Справа указаны номер договора (32543) и период действия (с 26.01.2004 по). В центре экрана активна вкладка "Глобальные тарифы".

Пло	Наименование	Период действия	Комментарий
0	T1_Телефония местная	01.12.2008-	

В левой панели "Параметры договора" видны следующие значения:

- Объекты договора: 0
- Иерархия договоров: супердогов.
- Статус: активен
- Лимит: 0.00
- Режим: дебет
- Баланс (апрель 2009):
 - Входящий остаток: -0.18
 - Приход: 1000.00
 - Нарботка: 0.00
 - Расход: 300.00
 - Исходящий остаток: 699.82
- Тарифные планы: T1_Телефония местная
- Модули: Рфоле
- Бухгалтерия
- Группы
- Web: Пароль статистики, Управление лимитом, Скрытие поведения, Дел. действия
- Примечания: 0

Скриншот интерфейса программы, отображающий детали договора. В верхней части видны вкладки: Параметры, Отчет, Карточки, HelpDesk, CRM, Документы. В центре заголовок договора: Т1 | Гайдеко Илья Валерьевич. Справа указаны номер договора (32543) и период действия (с 26.01.2004 по). В центре экрана активна вкладка "Свойства".

В левой панели "Параметры договора" видны следующие значения:

- Объекты договора: 0
- Иерархия договоров: супердогов.
- Статус: активен
- Лимит: 0.00
- Режим: дебет
- Баланс (апрель 2009):
 - Входящий остаток: -0.18
 - Приход: 1000.00
 - Нарботка: 0.00
 - Расход: 300.00
 - Исходящий остаток: 699.82
- Тарифные планы: T1_Телефония местная
- Модули: Рфоле (выделено)
- Бухгалтерия
- Группы
- Web: Пароль статистики, Управление лимитом, Скрытие поведения, Дел. действия
- Примечания: 0

Параметры Отчет Карточки HelpDesk CRM Документы

Договор T1_Sovintel [] 32545 с 01.12.2008 по

Договор является субдоговором с независимым балансом для T1 [Гайдеко Илья Валерьевич]

Открыть супердоговор Освободить договор

- Параметры договора
- Объекты договора 0
- Иерархия договоров н.субдог.
- Статус активен
- Лимит 0.00
- Режим дебет
- Баланс (апрель 2009)
 - Входящий остаток -31.44
 - Приход 300.00
 - Наработка 0.00
 - Расход 0.00
 - Исходящий остаток 268.56
- Тарифные планы
 - T1_Телефония Совинт
- Модули
 - Phone
 - Бухгалтерия
- Группы
- Web
 - Пароль статистики
 - Управление лимитом
- Скрипт поведения
- Доп. действия
- Примечания 0

Скрытый

Поиск договоров X x69 X x000 X SQLEditor X T1_Sovintel X T1 X Редактор тарифных планов

Параметры Отчет Карточки HelpDesk CRM Документы

Договор T1_Sovintel [] 32545 с 01.12.2008 по

Глобальные тарифы Персональные тарифы Группы тарифов

Поз.	Наименование	Период действия	Комментарий
0	T1_Телефония Совинт	01.12.2008-	

- Параметры договора
- Объекты договора 0
- Иерархия договоров н.субдог.
- Статус активен
- Лимит 0.00
- Режим дебет
- Баланс (апрель 2009)
 - Входящий остаток -31.44
 - Приход 300.00
 - Наработка 0.00
 - Расход 0.00
 - Исходящий остаток 268.56
- Тарифные планы
 - T1_Телефония Совинт
- Модули
 - Phone
 - Бухгалтерия
- Группы
- Web
 - Пароль статистики
 - Управление лимитом
- Скрипт поведения
- Доп. действия
- Примечания 0

Скрытый

Поиск договоров X x69 X x000 X SQLEditor X T1_Sovintel X T1 X Редактор тарифных планов

Параметры Отчет Карточка НефDesk CRM Документы

Период: < Девябрь 2008 >

Дни с: до

Поиск / Правила

Номер(а) [78463121532] на ЮсА1

Все Сброс Инверт.

Поиск: Бухгалтерия

CRM: Поиск договоров x40 x4004 SQLEditor T1_Sovintel T1 Редактор тарифных планов

Логи

Сессии Наработка Направления Услуги Детализация Входящие 1 из 2 [25]

Поиск	Направление	Услуга	Время входа	Длительность	
Номер(а) 78463121532	Богороди Каваны	Совинтел МН	01.12.2008 00:16:06	00:00:40 [40]	00
Номер(а) 78463121532	Богороди Каваны	Совинтел МН	01.12.2008 00:17:02	00:00:40 [40]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:17:58	00:00:32 [32]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:18:50	00:01:11 [71]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:20:25	00:00:43 [43]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:21:26	00:00:32 [32]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:22:15	00:01:13 [73]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:23:44	00:00:31 [31]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:24:33	00:00:36 [36]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:25:26	00:00:31 [31]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:26:12	00:00:33 [33]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:27:01	00:00:32 [32]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:28:39	00:00:37 [37]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:29:41	00:00:32 [32]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:30:29	00:00:45 [45]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:32:32	00:01:13 [73]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:34:02	00:00:36 [36]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:34:53	00:01:13 [73]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:37:02	00:00:39 [39]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:37:55	00:01:05 [65]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:39:15	00:00:30 [30]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:40:01	00:00:32 [32]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:47:50	00:00:38 [38]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:48:49	00:00:38 [38]	00
Номер(а) 78463121532	Челюбинск	Совинтел МГ	01.12.2008 00:50:36	00:01:14 [74]	00

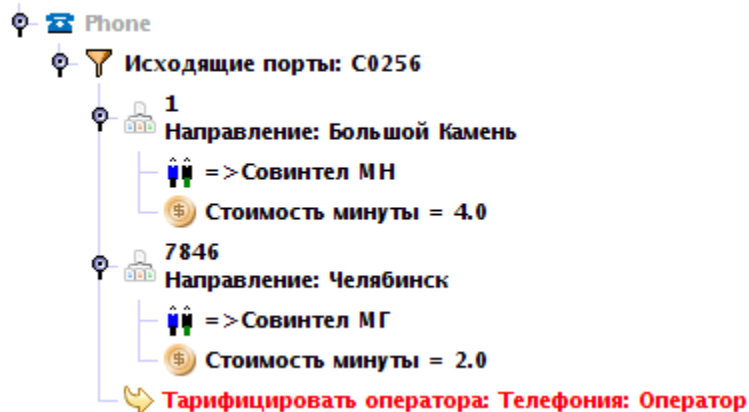
4

Кол-во сессий: 26 Сумма (ст.): 00:19:04 [1144] [19] Сумма (ст. авр.): 00:19:04 [1144] [19] Сумма (руб.): 40.82

Составление тарифов при агентской схеме

В тарифном плане тарифа агентского договора может быть дополнительно указана ссылка на тариф, используемый для вычисления цены для оператора-агента, по которым он приобретает трафик. По сути, это цена, по которой "продает" трафик оператор дальней связи и на которую в дальнейшем производится наценка оператором-агентом.

Тарифный план 'Т1_Телефония Совинтел'



Для этого используется узел тарифа **Тарифицировать оператора**. В операторском тарифе необходимо указать правила округления и цену. Вычисленное время сессии для оператора и стоимость для оператора используются для предоставления отчетности и вычисления агентского вознаграждения.

Отчетность

Отключение абонентов

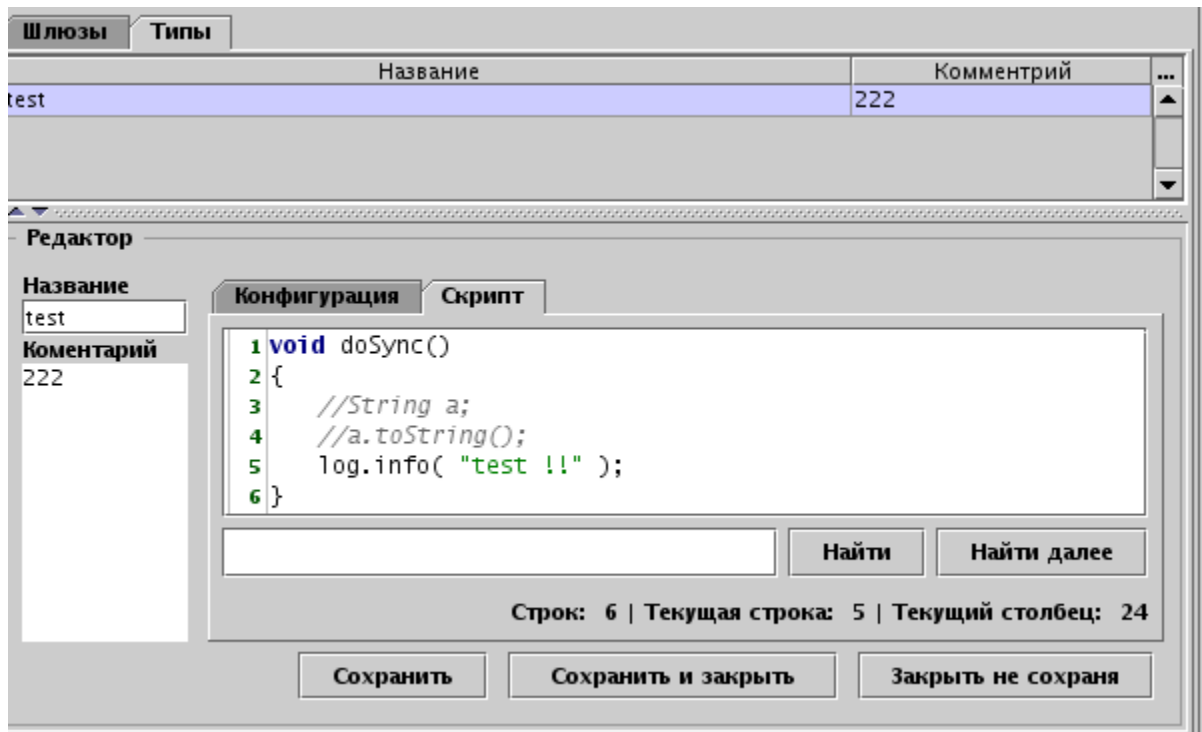
Для отключения абонентов используется статус модуля Phone. Этот статус можно менять вручную в договоре. При этом отработает скрипт смены статуса модуля Phone, который пошлет на АТС команды на закрытие абонента

Статус модуля Phone зависит от статуса договора . Типичная схема отключения должников: зайти в Монитор статуса , отфильтровать должников и поменять им статус договора. При этом произойдет смена статуса модуля Phone и отработает скрипт.

Потом по событию прихода платежа биллинг открывает статус договора , что также влечет за собой открытие статуса Phone.

Для настройки отключения абонентов делаем следующие шаги:

1) В настройках модуля Phone на Вкладке Шлюзы->Типы добавляем новый тип шлюза .



Тут мы задаем название, комментарий, конфигурацию и код шлюза на BeanShell

```
void doSync()
{
    //
}
```

В этом скрипте доступны объекты :

con - объект типа java.sql.Connection - соединение с БД;

gate - объект типа ru.bitel.bgbilling.modules.phone.common.bean.Gate - данные шлюза;

log - объект типа org.apache.log4j.Logger для логирования;

mid - код модуля телефония;

cid - код договора, для которого вызван скрипт;

status - статус договора (0 - открыт, 1 - закрыт).

2) Добавляем шлюз на вкладке Шлюзы->Шлюзы

Шлюзы Типы

Хост:порт	Тип	Комментрий	...
dddd:1	test	sa	▲

Хост и порт шлюза
 :

Ключевое слово

Тип шлюза

Комментарий Конфигурация

sa

Восстановить Ок Отмена

Тут задается хост, порт , ключевое слово, коментарий,конфигурация, тип шлюза (создан на предыдущем шаге).

3) Добавляем шлюз в договор.

Пойнты Правила Шлюзы

Управление

Статус

Шлюзы

Редактор

История смены статуса

дата	статус	
13.04.2010 18:47	закрыт	De
13.04.2010 18:46	открыт	De
13.04.2010 18:44	закрыт	De
13.04.2010 18:32	открыт	De
13.04.2010 18:32	закрыт	De
13.04.2010 18:14	открыт	По
13.04.2010 18:14	закрыт	По
13.04.2010 18:07	открыт	По
13.04.2010 18:06	закрыт	По
13.04.2010 18:06	открыт	Се
13.04.2010 18:04	открыт	Се
13.04.2010 18:01	открыт	Се
13.04.2010 18:00	открыт	Се
13.04.2010 17:58	открыт	Се
13.04.2010 17:57	открыт	Се
13.04.2010 17:56	открыт	Се

Ok Отмена

Теперь мы можем менять статус модуля Phone, при этом история смены статусов для каждого шлюза отображается в таблице справа . Лог выполнения скрипта можно посмотреть с помощью двойного щелчка мыши по конкретной строке таблицы.

Web-интерфейс Phone

Через Web-интерфейс модуля Phone имеется возможность просмотра сессий, наработки, наработки по направлениям, наработки по услугам, детализации и входящих звонков клиента по поинтам или правилам. Используя фильтры по дате, можно просмотреть интересующие данные как за конкретный день, так и за целый месяц.

Для просмотра сессий телефонии нужно выбрать поинты/правила, указать период. Если необходимо вывести только платные сессии, то нужно поставить галочку. Отчет по данному пункту можно загрузить с этой страницы, нажав на формат, в котором будет сформирован отчет.

Просмотр сессий

Поинты/Правила

Номер(а) 718463121532

Месяц: июль Год: 2009

Числа месяца: с 1 по 31

Только платные

Для выделения сразу нескольких поинтов/правил держите нажатой клавишу 'Ctrl'

Вывести

Отчет в виде: CSV, HTML, CSV/ZIP, HTML/ZIP, Одной страницей (После изменения фильтра сначала нажмите "Вывести")

Страница << 1 из 1 >>

Поинт	Направление	Услуга	Время входа	Длительность сеанса	Длительность сеанса окр.	Сумма, руб.	С номера / На номер
Номер(а) 718463121532		Совинтел МГ	01.07.2009 00:01:25	00:00:39 [39]	00:00:39 [39]	3.30000	78463121532 / 78462760707
Номер(а) 718463121532		Совинтел МГ	01.07.2009 00:01:25	00:01:05 [65]	00:01:05 [65]	2.16667	78463121532 / 78462760707
Номер(а) 718463121532		Совинтел МГ	01.07.2009 00:01:25	00:00:30 [30]	00:00:30 [30]	1.00000	78463121532 / 78462760707
Номер(а) 718463121532		Совинтел МГ	01.07.2009 00:01:25	00:00:32 [32]	00:00:32 [32]	1.06667	78463121532 / 78462760707
Номер(а) 718463121532		Совинтел МГ	01.07.2009 00:01:25	00:00:38 [38]	00:00:38 [38]	1.26667	78463121532 / 78462760707
Номер(а) 718463121532		Совинтел МГ	01.07.2009 00:01:25	00:00:38 [38]	00:00:38 [38]	1.26667	78463121532 / 78462760707
Итого сессий: 6				00:04:02 [242]	00:04:02 [242]	10.07	

Для просмотра наработки по телефонии нужно выбрать поинты/правила, указать период. Если необходимо вывести только платные сессии, то нужно поставить галочку. Отчет по данному пункту можно загрузить с этой страницы, нажав на формат, в котором будет сформирован отчет.

Наработка по Телефонии

Пойнты/Правила

Номер(а) 718463121532

Месяц: июль Год: 2009

Числа месяца: с 1 по 31

Только платные

Для выделения сразу нескольких
пойнтов/правил держите нажатой клавишу 'Ctrl'

Вывести

Страница << < 1 из 1 > >>

Пойнт	Количество	Длительность	Длительность округлённая	Сумма , руб.
Номер(а) 718463121532	6	00:04:02 [242]	00:04:02 [242]	10.06668
Итого:	6	00:04:02 [242]	00:04:02 [242]	10.07

Для просмотра наработки по направлениям нужно проделать те же действия, что и в просмотре наработки по телефонии.

Наработка по направлениям

Пойнты

Номер(а) 718463121532

Месяц: июль Год: 2009

Числа месяца: с 1 по 31

Только платные

Для выделения сразу нескольких
пойнтов/правил держите нажатой клавишу 'Ctrl'

Вывести

Страница << < 1 из 1 > >>

Пойнт	Направление	Количество	Длительность	Длительность округлённая	Сумма , руб.
Номер(а) 718463121532		1	00:00:39 [39]	00:00:39 [39]	3.30000
Номер(а) 718463121532		5	00:03:23 [203]	00:03:23 [203]	6.76668
Итого:		6	00:04:02 [242]	00:04:02 [242]	10.07

Для просмотра наработки по услугам нужно проделать те же действия, что и в просмотре наработки по телефонии. Отчет по данному пункту можно загрузить с этой страницы, нажав на формат, в котором будет сформирован отчет.

Наработка по услугам

Пойнты/Правила

Номер(а) 718463121532

Месяц: Год:

Числа месяца: с по

Только платные

Для выделения сразу нескольких поинтов/правил держите нажатой клавишу 'Ctrl'

[Вывести](#)

Страница [«](#) [<](#) 1 из 1 [>](#) [»](#)

Пойнт	Услуга	Количество	Длительность	Длительность округлённая	Сумма , руб.
Номер(а) 718463121532	Совинтел МГ	6	00:04:02 [242]	00:04:02 [242]	10.06668
Итого:		6	00:04:02 [242]	00:04:02 [242]	10.07

Для просмотра детализации по телефонии нужно выбрать поинты/правила, указать период. Если необходимо вывести только платные сессии, то нужно поставить галочку. Отчет по данному пункту можно загрузить с этой страницы, нажав на формат, в котором будет сформирован отчет. Отчет по данному пункту можно загрузить с этой страницы, нажав на формат, в котором будет сформирован отчет.

Детализация по телефонии

Пойнты/Правила

Номер(а) 718463121532

Месяц: Год:

Числа месяца: с по

Только платные

Для выделения сразу нескольких поинтов/правил держите нажатой клавишу 'Ctrl'

[Вывести](#)

Отчет в виде: [CSV](#), [HTML](#), [CSV/ZIP](#), [HTML/ZIP](#) (После изменения фильтра сначала нажмите "Вывести")

Страница [«](#) [<](#) 1 из 1 [>](#) [»](#)

Дата/время звонка	Направление	Номер телефона В	Округленная длительность, мин.	Стоимость звонка, руб.
Услуга: Совинтел МГ				
Номер телефона А: 78463121532				
01.07.2009 00:01:25		78462760707	00:00:39 [39]	3.30000
01.07.2009 00:01:25		78462760707	00:01:05 [65]	2.16667
01.07.2009 00:01:25		78462760707	00:00:30 [30]	1.00000
01.07.2009 00:01:25		78462760707	00:00:32 [32]	1.06667
01.07.2009 00:01:25		78462760707	00:00:38 [38]	1.26667
01.07.2009 00:01:25		78462760707	00:00:38 [38]	1.26667
Итого по номеру: 6		00:04:02 [242]	00:04:02 [242]	10.07
Итого по услуге: 6		00:04:02 [242]	00:04:02 [242]	10.07
Итого сессий: 6		00:04:02 [242]	00:04:02 [242]	10.07

Для отображения пункта "Входящие звонки" в Web-интерфейсе необходимо в конфигурации модуля добавить следующее:

```
web.incommingCall=true  
web.incommingCall.exception=2,3
```

Значение **web.incommingCall** может быть true или false. Соответственно, если указано true, то данный пункт будет виден для всех, а если false - то ни для кого. Значение **web.incommingCall.exception** - это исключение из правила выше, т.е. если стоит значение true в **web.incommingCall**, а в исключении перечислены id групп договоров, то для перечисленных групп договоров данный пункт в Web-меню не будет отображен, и наоборот, соответственно.

Для просмотра входящих звонков по телефонии нужно выбрать поинты, указать период.

Пойнты/Правила

```
dd
dd
ddd
ddf
ddf
dfsdfsdf
Номер(a) 11111111111111111111
Номер(a) 123123
Номер(a) 8463121532
```

Месяц: март Год: 2010

Числа месяца: с 1 по 31

Только платные

Для выделения сразу нескольких поинтов/правил держите нажатой клавишу 'Ctrl'

Вывести

Страница << < 1 из 1 > >>

Пойнт	Время входа	Длительность[в секундах]	С номера / На номер
dd	01.03.2010 00:44:59	00:00:00 [0]	79063427904/78463121532
dd	01.03.2010 00:47:30	00:00:02 [2]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:00 [0]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:00 [0]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:06 [6]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:54 [54]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:04 [4]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:02 [2]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:03 [3]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:04 [4]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:05 [5]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:06 [6]	78469976436/78463121532
dd	01.03.2010 00:47:30	00:00:00 [0]	78469976436/78463121532
dd	01.03.2010 00:48:27	00:00:00 [0]	78469976436/78463121532
Итого сессий: 14		00:01:26 [86] [1]	00:00:00 [0] [1]

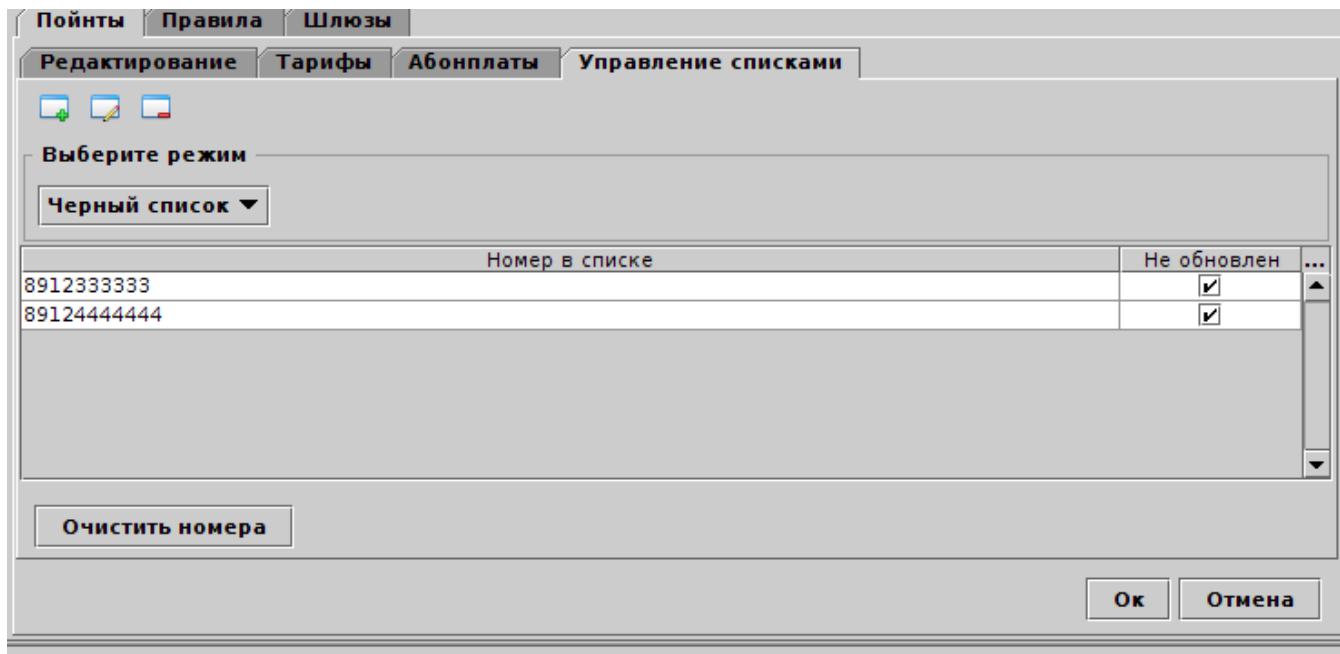
Для изменения логики отображения поинтов/правил можно в конфигурации прописать:

```
# / web "" .
# web.item.list.content.mode
# "all" - / (0...+)
# "last" - ( ...+) -
# "currperiod" - web (...)
web.item.list.content.mode=currperiod
```

Черно-белые списки

Опционально доступен функционал черно-белых списков. Включить его можно в [конфигурации](#). Пользователь сам может определять номера, вызовы с которых на его телефонный номер запрещены (черный список) или, наоборот, разрешены (белый список). На одном номере абонента можно включить либо черный, либо белый список. Управление оборудованием (АТС) осуществляется глобальным скриптом поведения, запускаемым по таймеру, который разрабатывается уже под конкретную АТС.

При редактировании поинта в клиенте доступна вкладка "Управление списками"



Тут можно добавлять/удалять/редактировать номера телефонов и выбрать тип списка - черный или белый.

Аналогичный интерфейс доступен абоненту в личном кабинете:

Управление черными/белыми списками

Выберите номер телефона:

- 11117
- 123
- 123123
- 156
- 8463121532
- 85

Редактировать

Режим списков:

Черный список ▼

Применить

Номера:

Добавить номер

Удалить

Удалить

Применить

Синхронизация черно-белых списков с АТС осуществляется с помощью глобального скрипта поведения. Вот примерная структура скрипта:

Код

```
import bitel.billing.server.util.*;
import ru.bitel.bgbilling.modules.phone.server.bean.*;
import ru.bitel.bgbilling.modules.phone.common.bean.*;
import java.util.*;
import java.net.*;
import java.io.*;

int mid = 0;
public void main( setup, con, conSlave )
{
    con.setAutoCommit( false );
    PhoneItemListManager pilm = new PhoneItemListManager( mid, con );

    //
    List changedModes = pilm.getToUpdateModes();
    for( PhoneItemListMode mode : changedModes )
    {
        updateModeOnBackend( mode );
        mode.setNewMode( false );
    }
}
```



```
        pilm.updateMode( mode );
    }

    //
    List deletedNumbers = pilm.getToDeleteNumbers();
    for( PhoneItemListNumber number : deletedNumbers )
    {
        deleteNumberOnBackend( number );
        pilm.deleteNumber( number );
    }

    //,
    List addedNumbers = pilm.getToUpdateNumbers();
    for( PhoneItemListNumber number : addedNumbers )
    {
        addNumberOnBackend( number );
        number.setNewNumber( false );
        pilm.updateNumber( number );
    }
    con.commit();
}

//
private void updateModeOnBackend( PhoneItemListMode mode )
{
    URL example = new URL("http://www.example.com/");
    URLConnection yc = example.openConnection();
    //TODO
}

//
private void deleteNumberOnBackend( PhoneItemListNumber number )
{
    URL example = new URL("http://www.example.com/");
    URLConnection yc = example.openConnection();
    //TODO
}

//
private void addNumberOnBackend( PhoneItemListNumber number )
{
    URL example = new URL("http://www.example.com/");
    URLConnection yc = example.openConnection();
    //TODO
}
}
```

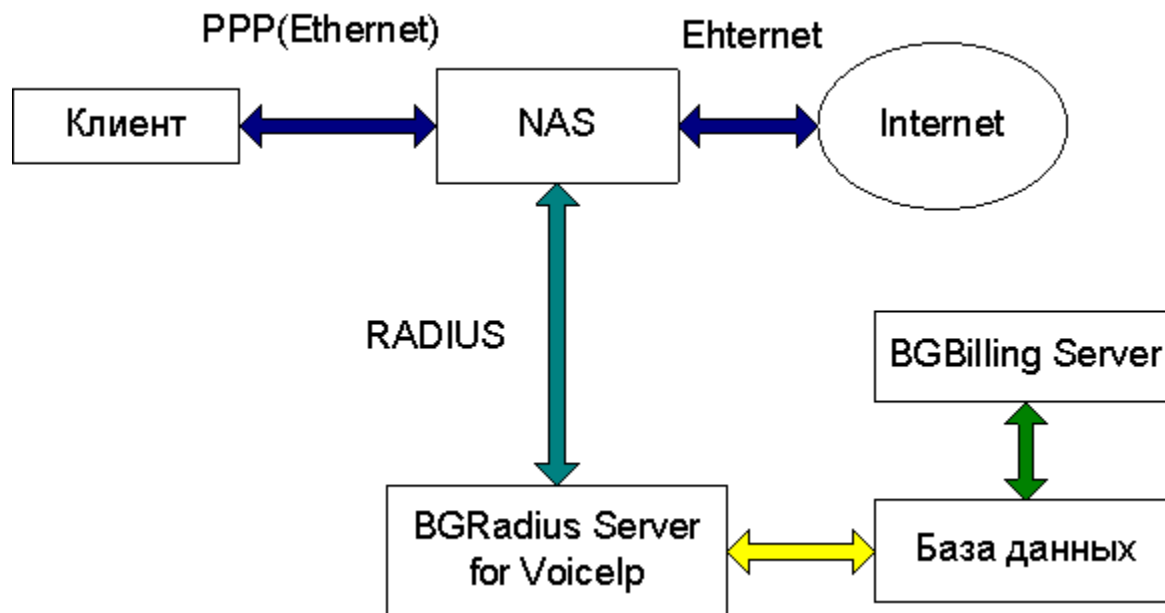
Модуль VoiceIP

Назначение модуля

Модуль VoiceIP предназначен для подсчёта стоимости звонков через IP-телефонию. В настоящее время он поддерживает оборудование, которое шлёт запросы **RADIUS** в стандарте CISCO, а также поддерживается оборудование QUINTUM. Так же как и модуль DialUP, этот модуль использует RADIUS-сервер, установка аналогична установке DialUP.

Базовые понятия и алгоритм работы модуля VoiceIP

Структура взаимодействия основных частей модуля VoiceIP изображена на рисунке.



NAS (Network Access Server) - сервер, через который происходит звонок клиента. В роли NASa могут выступать специализированные компьютеры типа Cisco, но практически может быть любое оборудование, способное обеспечивать звонки клиентов по IP и использующее аналогичный с Cisco h323 стандарт RADIUS-запросов.

Обрабатываются 3 типа запросов: авторизационный, старт и стоп.

[Базовые сведения о протоколе RADIUS и конфигурировании атрибутов.](#)

Протокол RADIUS основан на UDP, представляет из себя пакет определённого типа с набором атрибутов. Рассмотрим ход типового соединения по логу RADIUS запросов (radius.log).

В данном случае обмен запросами идёт между RADIUS-сервером и IVR-платформой, запрашивающей данные пользователя и проводящей авторизацию по телефону пользователя.

Код

```
01 13:13:21
Type=AUTHENTICATION_REQUEST
Attributes:
User-Password=
NAS-IP-Address=81.30.199.58
Service-Type=1
Calling-Station-Id=3472558528
cisco-avpair=h323-ivr-out\u61transactionID:69732
h323-conf-id=585DAE20 CFD411DC 9F39C1A6 BC2D9958
```

```
01 13:13:21
Type=AUTHENTICATION_ACCEPT
Process time:10
Attributes:
h323-credit-amount=99540.38
h323-return-code=0
```

```
01 13:13:26
Type=AUTHENTICATION_REQUEST
Attributes:
User-Password=
NAS-IP-Address=81.30.199.58
Service-Type=1
Calling-Station-Id=3472558528
Called-Station-Id=6767#79174444339
```

```
cisco-avpair=h323-ivr-out\u61transactionID:69732
h323-conf-id=585DAE20 CFD411DC 9F39C1A6 BC2D9958

01 13:13:26
Type=AUTHENTICATION_ACCEPT
Process time:10
Attributes:
h323-credit-amount=99540.38
h323-credit-time=7200
h323-return-code=0

01 13:14:36
Type=ACCOUNTING_REQUEST
Attributes:
User-Name=3472558528
NAS-IP-Address=81.30.199.58
Service-Type=1
Acct-Input-Octets=54043
Acct-Output-Octets=33311
Acct-Status-Type=2
Acct-Delay-Time=0
Acct-Session-Time=57
Acct-Input-Packets=1370
Acct-Session-Id=00033F15
Acct-Authentic=1
Acct-Output-Packets=1136
Calling-Station-Id=3472558528
Called-Station-Id=6767#79174444339
h323-gw-id=pool3.ufanet.ru
h323-voice-quality=31
h323-remote-address=81.30.206.222
h323-disconnect-cause=10
h323-connect-time=13:13:39.562 ESS Fri Feb 1 2008
h323-disconnect-time=13:14:36.387 ESS Fri Feb 1 2008
cisco-avpair=h323-incoming-conf-id\u61585DAE20 CFD411DC 9F39C1A6 BC2D9958
cisco-avpair=subscriber\u61RegularLine
cisco-avpair=session-protocol\u61cisco
cisco-avpair=gw-rxd-cdn\u61ton:4,npi:1,#:2900222
cisco-avpair=release-source\u614
cisco-avpair=remote-media-address\u6181.30.199.118
cisco-avpair=in-trunkgroup-label\u61TGI
cisco-avpair=gw-rxd-cgn\u61ton:2,npi:1,pi:0,si:3,#:3472558528
cisco-avpair=gw-final-xlated-cdn\u61ton:4,npi:1,#:6767#79174444339
cisco-avpair=gw-final-xlated-cgn\u61ton:2,npi:1,pi:0,si:3,#:3472558528
h323-conf-id=585DAE20 CFD411DC 9F39C1A6 BC2D9958
h323-setup-time=13:13:26.192 ESS Fri Feb 1 2008
h323-call-origin=originate
h323-call-type=VoIP
```

Каждый пакет содержит информацию о NAsе (**NAS-Identifier** и/или **NAS-IP-Address**), на основании которой RADIUS-сервер сопоставляет пришедший пакет NASу в модуле. При сопоставлении сначала производится поиск NASа с названием, идентичным атрибуту **NAS-Identifier** пакета, затем, если результат отрицательный, идёт поиск NASа с IP-адресом, равным **NAS-IP-Address**. Если пришедшему пакету NAS не сопоставлен в **radius.log**, выводится ошибка **NAS not found for Packet!!!**.

Обмен сообщениями с каждым NASом шифруется определённым кодовым словом - секретом. Секрет должен совпадать для NASа в биллинге и для конфигурации самого NASа. При несовпадении секретов проверка пароля будет все время выдавать неверный результат, т.к. секрет используется в шифрации пароля.

Идентификатором соединения в пределах NASа для RADIUS-сервера выступает атрибут **h323-conf-id**. Обратите внимание, что он идентичен для всех пакетов в пределах сессии. NAS должен контролировать, чтобы в один момент времени одинаковый **h323-conf-id** не проставлялся в RADIUS-пакетах, относящихся к разным сессиям.

Далее рассмотрим по пакетно обмен данными между NASом и RADIUS-сервером по ходу соединения.

1. AUTHENTICATION_REQUEST

Запрос авторизации отправляется NASом RADIUS-серверу и содержит помимо идентификационной информации соединения, указанной выше, информацию, идентифицирующую пользователя. Возможна идентификация пользователя по номеру звонящего, логину и паролю (запрашивает IVR) или любому другому атрибуту, режим поиска настраивается для типа логина.

При наличии пароль шифруется по алгоритму PAP с использованием секрета. При некорректном секрете пароль в PAP-режиме не расшифровывается и отображается в **radius.log** и в мониторе ошибок не в том виде, в котором был введён пользователем.

Передаётся последовательно два запроса авторизации. Первая авторизация передаётся до набора номера и содержит только идентифицирующую пользователя информацию.

Вторая авторизация высылается в случае успеха первой и содержит набранный номер, атрибут **Called-Station-Id**.

2. AUTHENTICATION_REJECT

Отказ в авторизации, код ошибки авторизации передаётся дополнительно в атрибуте **h323-return-code**. Детальную причину отказа в авторизации пользователя можно посмотреть в **Мониторе модуля** в режиме ошибок. Код ошибки авторизации указанный в квадратных скобках в мониторе совпадает со значением, передаваемым в атрибуте. Значение атрибута может быть использовано IVR системой для озвучивания причины ошибки.

3. AUTHENTICATION_ACCEPT

Пользователь авторизован.

Для первой авторизации это означает что он идентифицирован, пароль введён верно (если есть), остаток на балансе положителен, тариф установлен и т.п. В атрибутах передаётся **h323-return-code=0** (нет ошибок) и **h323-credit-amount=<остатку счета>**.

При второй авторизации это означает, что верны все условия первой авторизации, а кроме того на набранный номер есть цена в тарифе. Дополнительно к атрибутам первой авторизации передаются атрибут **h323-credit-time=<на сколько секунд разговора хватает баланса пользователя>**. Значение данного атрибута ограничивается сверху значением переменной **voip.max.time** конфигурации модуля.

Дополнительно в **AUTHENTICATION_ACCEPT** могут передаваться атрибуты, указанные в конфигурации типа логина (см. далее).

4. ACCOUNTING_REQUEST

Модулем обрабатываются запросы аккаунтинга только типа **Stop**. Атрибут **Acct-Status-Type** для них равен 2. Данный тип запросов передаёт на RADIUS-сервер информацию о завершении соединения. Время соединения берется из атрибута **Acct-Session-Time**.

5. ACCOUNTING_RESPONSE

Ответ RADIUS-сервера о том, что он получил запрос аккаунтинга. Ответ не содержит никаких атрибутов.

Возможны схемы работы с учётом только **Stop** пакетов, без авторизации. В этом случае RADIUS-сервер не может запретить звонок при отсутствии цены в тарифе и, чтобы не потерять его, помещает его в лог без определённого направления и с нулевой стоимостью.

Настройка модуля VoiceIP

Установите модуль на сервер с помощью утилиты **bg_installer**, обновите клиент. Затем создайте экземпляр модуля. Добавьте услугу **IP-телефония**. Услуга в понятии модуля - доступ к Voip-услугам определённого NASa.

Перезапустите клиента, откройте в меню **Модули** созданный экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, скопируйте туда приведённый ниже текст и сделайте её активной. Значения параметров указаны после символа комментария (#).

Конфигурация

```
#
contract.status.active.codes=0

#-
vendors=9=Cisco;2011=Huawei;2021=Unix PPP;529=Lucent;6618=Quintum;529=Ascend
#
password.length.min=5
password.length.max=10
#
password.length.auto=6
#
password.chars=1234567890
#     web
showyears=5
#XSL
xslt.1=voiceip_login_sessions.xsl
xslt.1.csv=voiceip_login_sessions_csv.xsl
reportTitle.1= VoiceIP
#XSL
xslt.2=voiceip_login_amount.xsl
xslt.2.csv=voiceip_login_amount_csv.xsl
reportTitle.2= VoiceIP
#XSL
xslt.3=voiceip_login_direct.xsl
xslt.3.csv=voiceip_login_direct_csv.xsl
reportTitle.3= VoiceIP .
#-     Web-
show.sessions.on.page=25
# Web-
web.menuItem1= VoiceIp
web.menuItem2= VoiceIp
web.menuItem3= VoiceIp
web.menuItem4= VoiceIp
#
top.nocard.login=10000
#-----
#
#
#closed.date.disabled.ActionWrapLogin=1
#
#closed.date.disabled.ActionRecalculateSessions=1
#
#closed.date.disabled.ActionSetBalance=1
#
#closed.date.disabled.ActionUpdateLoginInfo=1
#-----

##### RADIUS- #####
#l - , - "Service deny"
check.service=0
# "", 0 - ""
card.module.id=0
#
voip.max.time=1800
# RADIUS,
radius.attributes=Service-Type;Framed-Protocol;Framed-IP-Address;Framed-IP-Netmask;Framed-Routing;Filter-Id;
Framed-MTU;Framed-Compression;Login-IP-Host;Login-Service;Login-TCP-Port;Old-Password;Reply-Message;Callback-
Number;Callback-Id;Expiration;Framed-Route;Framed-IPX-Network;State;Class;Session-Timeout;Idle-Timeout;
```

```
Termination-Action;NAS-Identifier;Proxy-State;Framed-Pool
# Disconnect cause
#   Disconnect cause      ( , )
voip.ignore.dc=66
#   Disconnect-Cause     DC:HEX Color
voip.monitor.dc.color=10:00ff00;3:ff0000;22:ff00ff;11:ffff00
#
findmode.0.title=  User-Name=LOGIN
findmode.0.value=User-Name=LOGIN
findmode.1.title=  User-Name=ALIAS
findmode.1.value=User-Name=ALIAS
findmode.2.title=  Calling-Station-Id=ALIAS
findmode.2.value=Calling-Station-Id=ALIAS
#
find.order=0,1,2
#
# ASR(%)  ACD()      ,
#color.asr=0-30:#dd0000;30-100:#ffffff
#color.acd=0-60:#dd0000
#
# DC
#0 - , , radius-. .
#1 - , radius- Q931 ( , NAS)
#2 - Q931 ( , NAS)
voiceip.dc.column.mode=0
```

Параметры xslt.1, xslt.2 и xslt.3 должны указывать на [XSLT](#)-шаблоны преобразования для печати сессий клиента и отправки их на E-mail. Параметры reportTitle.1 - reportTitle.3 задают тему письма с отчётом и заголовок отчёта. Если все было сделано правильно, настройка модуля завершена.

Настройка режимов поиска и типов логинов

Режим поиска представляет собой запись в конфигурации модуля следующего вида:

```
findmode.< >.title=< >
findmode.< >.value=< >=<LOGIN|ALIAS>
```

Порядок использования режимов поиска задаётся переменной **find.order** в конфигурации модуля. Например:

```
#
findmode.0.title= User-Name=LOGIN
findmode.0.value=User-Name=LOGIN
findmode.1.title= User-Name=ALIAS
findmode.1.value=User-Name=ALIAS
findmode.2.title= Calling-Station-Id=ALIAS
findmode.2.value=Calling-Station-Id=ALIAS
#
find.order=0,1,2
```

означает, что первая попытка поиска идёт среди логинов, где логин равен атрибуту User-Name. Это, например, карточные пользователи. Далее идёт поиск по User-Name=ALIAS и Calling-Station-Id=ALIAS.

Режим поиска User-Name=LOGIN должен существовать всегда с кодом 0 для возможности использования карточной платформы.

Параметр **find.order** может быть переопределён в конфигурации NASa. Например, если у вас выделен отдельный шлюз под карточную платформу, на нем нет необходимости осуществлять поиск по телефону или текстовому логину.

Кроме того, возможно жёсткое задание режима поиска для конкретного запроса в скрипте предобработки RADIUS-запроса. Например, данный скрипт в случае прихода атрибутов User-Name и User-Password в запросе задаёт жёстко режим поиска = 0, т.е. User-Name=LOGIN.

При этом в конфигурации NASa может быть указан find.order, задающий поиск по номеру звонящего.

```
import java.sql.*;

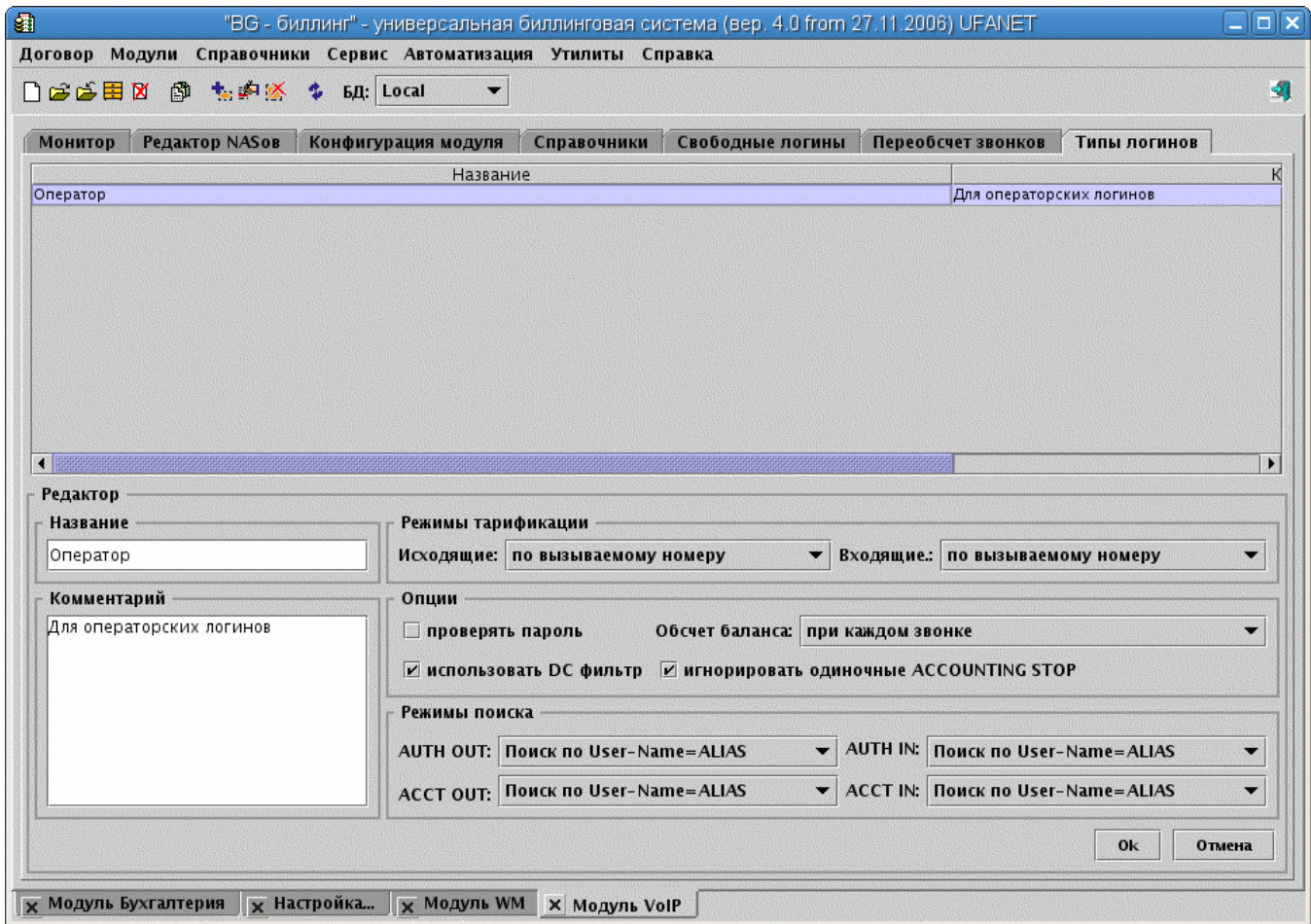
import bitel.billing.common.*;
import bitel.billing.server.radius.*;

userName = request.getStringAttribute( RadiusStandartAttributes.User_Name );
userPassword = request.getStringAttribute( RadiusStandartAttributes.User_Password );
if( userName != null && userPassword != null )
{
    request.setOption( "find.mode", new Integer( 0 ) );
}
```

Тип Voip-логина определяет поведение логина при его поиске и тарификации его сессий. Типы логинов введены, начиная с версии 4.0 и призваны упростить процесс заведения логинов Voip.

В типе указываются режимы поиска для различных видов запросов: авторизационных (входящих и исходящих) и аккаунтинга (входящего и исходящего).

При последовательном поиске по указанным в **find.order** режимам RADIUS определяет типы логинов, в которых указан данный режим поиска при данном типе запроса. Если список типов пуст поиск не производится.



Рассмотрим иные параметры типа логинов кроме режимов поиска:

1. Режимы тарификации - какой номер использовать при поиске цены входящего и исходящего звонков;

Для операторских логинов возможна тарификация входящих звонков по вызываемому номеру. При этом в тарифном плане оператора стоимость входящих звонков соответствует стоимости термации данного префикса данным оператором.

Получив отчёт входящих звонков оператора в отчёте договора, можно получить сумму, которую вы должны данному оператору за термацию им звонков.

2. Проверять пароль - галочка отсутствует у типов логинов с авторизацией по номеру звонящего;
3. Обсчёт баланса (при каждом звонке/никогда) - опция никогда не стоит у операторских логинов для снижения нагрузки на радиус. При этом после каждого звонка не обновляется наработка в балансе пользователя. Для обычных логинов стоит режим обчёта после каждого звонка.
4. Использовать DC фильтр - обнулять длительность звонка при **h323-dicsonnet-couse, попадающим в перечень из параметра voip.ignore.dc**. Данная опция обычно снимается для операторских логинов;
5. Игнорировать одиночные ACCOUNTING STOP - данная опция не учитывает STOP-пакеты, пришедшие без предварительной авторизации, и может быть использована для исключения дублирования звонков клиентов в случае их прохождения через несколько NASов.

В системе присутствует тип логина **По умолчанию**, устанавливаемый карточным логинам. В нем установлен следующий набор опций:

1. Все режимы поиска User-Name=LOGIN (0);
2. Проверка пароля;
3. Использование DC фильтра;
4. Обсчёт баланса при каждом звонке;
5. Стоимость исходящего по вызываемому номеру, входящего по номеру звонящего;
6. Игнорирование одиночных ACCOUNTING STOP-пакетов.

При необходимости возможно переопределить значение флага "Игнорировать одиночные ACCOUNTING STOP-пакеты" путем изменения значения опции **default.login.ignore.stop**, которая может принимать значения **true** (установлено по умолчанию) или **false**. Это может потребоваться, например, в случае, если в пакетах авторизации и аккаунтинга атрибуты **h323-conf-id** отличаются друг от друга и биллинг не может создать сессию по одиночному stop-пакету.

На вкладке **Атрибуты** можно указать передаваемые при авторизации данного типа логина атрибуты в следующем формате (пример, первый атрибут смысла не имеет, второй используется в MVTS для SIP-аккаунтов):

```
Cisco-AVPair=lcp:interface-config=ip unnumbered Loopback135
cisco-avpair=xpgk-ep-number=${ALIAS}
```

В атрибутах могут быть использованы следующие макро-подстановки:

- **\${ALIAS}** - первый алиас voip логина;
- **\${BALANCE_MODE}** - 0 - кредитовый режим, 1 - дебетовый;
- **\${MIN_COST}** - стоимость минуты звонка, десятичный разделитель - точка.

Если значение подстановки не может быть вычислено, атрибут не будет передан. В дополнение к определённым атрибутам обязательно передаются **h323-credit-amount** (остаток на счёте), **h323-credit-time** (максимальная длительность звонка по направлению в секундах) и **h323-return-code** (код ошибки, либо 0 при успешной авторизации).

Для типа логина **По умолчанию** передаваемые атрибуты указываются в конфигурации модуля в следующем виде:

```
default.login.attributes=<ATTR_NAME_1>=<ATTR_VALUE_1>;<ATTR_NAME_2>=<ATTR_VALUE_2>
```

Например:

```
default.login.attributes=h323-billing-model=${BALANCE_MODE};h323-ivr-in=tariff:${MIN_COST}
```

В данном параметре можно использовать те же макроподстановки, что и в атрибутах для типов логинов.

При каждой модификации типов логинов, конфигурации модуля или NASов необходимо перегружать RADIUS-сервер.

Настройка NASов модуля VoiceIP

Основная информация, которую должна содержать конфигурация NASa - это его идентификатор, IP-адрес и секрет (необходим для шифрования пароля пользователя, должен совпадать на RADIUS, клиенте и сервере).

Кроме того, к нему должна быть привязана услуга и определена конфигурация определения направления звонка.

gate	192.168.184.254	-01.01.2008	gate
huaw1	81.30.210.254	-01.09.2003	900111(Huawei Expert)
oracle	192.168.184.94	25.01.2006-01.01.2008	
pool-1	213.24.120.34	-01.08.2003	Unix
pool3	81.30.199.58	-01.09.2003	900111(Cisco53x)
pool7	213.59.17.253	-01.12.2005	797588(Cisco36x)
pool9	213.24.120.254	-01.09.2003	797588(Cisco53x)
stat	213.24.120.68	-01.01.2008	Unix
ufps	81.30.199.235	-	Unix (основной для тестов)
vg1	81.30.199.34	-01.09.2003	900111(Cisco53xx)
wfi	192.168.1.1	-01.01.2008	server.office.bitel.ru

Редактор NAS

Идентификатор: pool.bitel.ru | Вендор: Cisco

Адрес (X.X.X.X): 1.1.1.1 | Секрет: *****

Период: с 25.01.2008 по

Комментарий:

Дата	Название	Пользователь
------	----------	--------------

Создать | Открыть | Удалить | Сохранить | Отменить | Установить

Восстановить | Ок | Отмена

Модуль Dial-Up

Настройка сервера доступа происходит в два шага.

Сначала кнопкой **Новый элемент** создаётся новый NAS, ему прописывается идентификатор (что будет приходить от него в атрибуте **NAS-Identifier**), IP-адрес (что будет приходить от него в атрибуте **NAS-IP-Address**), RADIUS-секрет, вендор и комментарий. Список вендоров задаётся в конфигурации модуля. После нажатия кнопки **Ок** должна появиться новая строка в таблице.

Вендор определяет код вендора, указываемый для h323-атрибутов, для большинства оборудования он должен быть Cisco.

Двойным кликом мыши откройте её для редактирования и кнопкой **Создать** добавьте текстовую конфигурацию. Название - произвольное.

Возможна также настройка общих конфигураций для NASов. Создание общих конфигураций происходит аналогично описанному далее, за исключением того, что они могут использоваться в нескольких NASax, при этом локальные настройки имеют больший приоритет (т.е. если оба флага присутствуют и во включенной общей, и в выбранной локальной конфигурациях, то будет использован локальный).

Конфигурация | **Общая конфигурация** | Скрипт преобработки

Дата	Название	Пользователь
14.12.2009 17:23...	ишо	Кирилл Березин
14.12.2009 17:19...	первый тестовый шаблон!	Кирилл Березин

Создать | Открыть | Удалить | Сохранить | Отменить | Установить

Содержимое конфигурации должно содержать следующие данные:

1) Привязанную к NASy услугу: **service**=<код услуги>;

2) Необходимо настроить какие звонки считать исходящими, а какие - входящими для пользователя. Для определения направления используются атрибуты **h323-call-type** и **h323-call-origin** из RADIUS-запроса. Значения этих атрибутов, соответствующие каждому типу звонка, необходимо указать через дробь и в нижнем регистре (даже если в запросе указан верхний регистр).

Ниже приведена конфигурация, которая может быть использована для популярного Гейткипера Aqua:

```
auth.in=voip/originate
auth.out=voip/answer
acct.in=voip/originate
acct.out=voip/answer
```

В этом случае авторизационные запросы с атрибутами **h323-call-type=Voip h323-call-origin=originate** будут считаться исходящими, **h323-call-type=Voip h323-call-origin=answer** входящими.

При необходимости можно указывать несколько сочетаний данных атрибутов через ;. Например:

```
acct.out=voip/answer;voip/proxy
```

STOP-пакет исходящего звонка может содержать как сочетание Voip Answer, так и Voip Proxy. Процесс подборки данных атрибутов эмпирический и зависит от вашего оборудования.

Если через ваш шлюз идут только исходящие вызовы, можно определить следующую конфигурацию. В этом случае всякий вызов будет воспринят как исходящий.

```
auth.out=all/all
acct.out=voip/originate
```

С версии 5.1 есть возможность отрезания реалма типа @xxx от атрибута, используемого как имя пользователя. Для этого в конфигурацию необходимо добавить, например:

```
realm.attr=User-Name
realm.value=ipnet.ru,voipnet.ru
```

Для идентификации сессии можно использовать значение атрибута Acct-Session-Id, для этого в конфигурацию добавить:

```
sip=1
```

Для вывода ошибок, при невозможности определить направление звонка, в конфигурацию нужно добавить:

```
acct.logError=1
```

Для полной обработки входящих звонков в конфигурацию нужно добавить:

```
auth.income.full=1
```

Следующий параметр определяет формат суммы в поле H323_credit_amount

```
balance.summa.format.pattern=#####.##
```

В мониторе логов модуля в столбце DC выводится значение атрибута h323-disconnect-cause. В зависимости от настроек модуля (см. раздел по настройке модуля VoiceIP) там может выводиться описание пришедшего атрибута в соответствии с протоколом Q931. При необходимости в конфигурации NAS можно переопределить выводимое значение (например, можно задать перевод для описания), либо задать свое, если значение не отображается по умолчанию.

```
voiceip.dc.< >=<>
```

Подробные инструкции по интеграции BGRadiusVoip с различными NASами доступны в Wiki.

Скрипт предобработки RADIUS-запроса

На вкладке **Скрипт предобработки** может быть написан [BGS скрипт предобработки](#), определяющий перечисленные ниже параметры каким-либо сложным способом, для которого недостаточен формат конфигурации. Параметры, заданные скриптом, более приоритетные, чем взятые из конфигурации.

- Тип звонка - `request.setFlag(int callTypeFlag);`
- Тип поиска - `request.setOption("find.mode", int mode);`
- Услуга звонка - `request.setOption("service", int service);`

Вы можете посмотреть примеры скриптов предобработки запросов в [Wiki](#).

Скрипты обработки RADIUS-запросов кэшируются RADIUS-сервером при первом исполнении. Для сброса кэша необходимо перезапустить RADIUS-сервер, либо выполнить команду в каталоге RADIUS-сервера.

Для Linux:

```
./radius.sh flush_script_cache
```

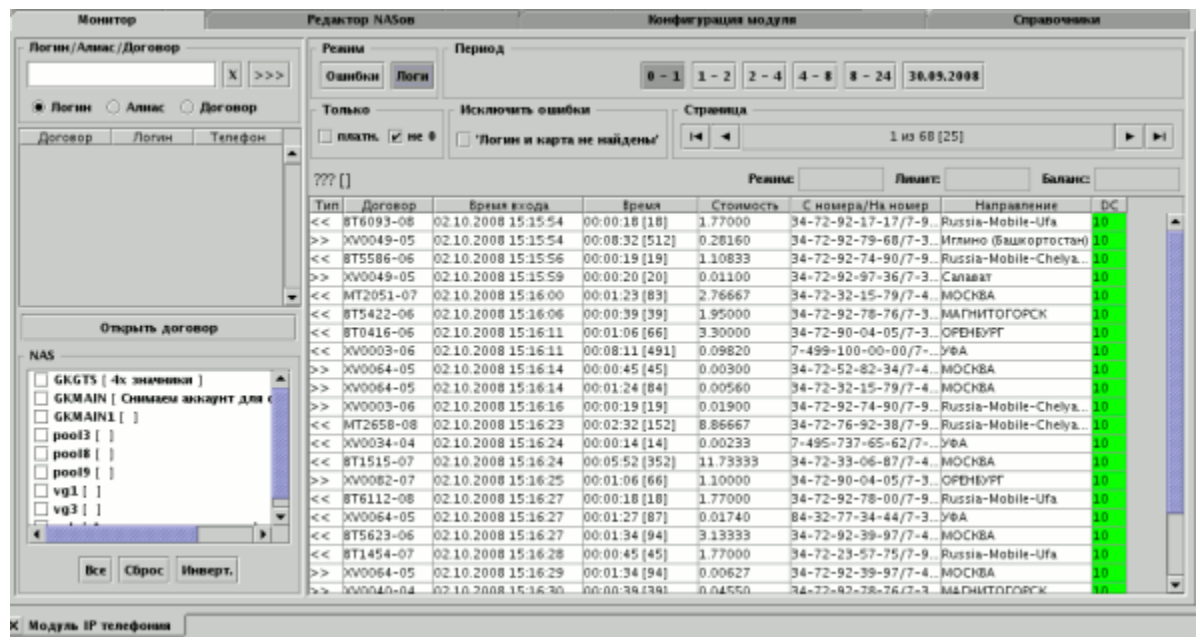
Для MS Windows:

```
radius.bat flush_script_cache
```

Логи работы скриптов вы можете посмотреть в файле **BGBillingServer/log/script.log**

Монитор

В отличие от монитора DialUP модуля здесь отсутствует режим **Текущие**, т.к. модуль фиксирует соединения только по STOP-пакету RADIUSa. В режиме логов есть фильтр для выделения звонков с ненулевой длительностью и только платных. В первом столбце таблицы сессий отображается направление звонка "<<" - исходящий, ">>" - входящий.



В последнем столбце выводится Disconnect-Cause - причина разрыва, цвета ячейки можно задать в конфигурации модуля. Принципы выбора периодов, фильтрации сессий и ошибок по логину идентичны монитору модуля DialUp.

Перечень типов ошибок в мониторе и их расшифровка:

Таблица. Таблица кодов ошибок

Код ошибки	Название	Описание
1	Неверный пин-код карты	Карта найдена, но введённый пароль не соответствует её пин-коду.
2	Неверный пароль логина	Логин найден, но введённый пароль не соответствует его паролю.
3	Тарифные планы не найдены	У договора не установлен ни один тарифный план для данного модуля на день авторизации.
4	Ошибка баланса	Остаток баланса договора меньше лимита договора.
6	NAS не найден	RADIUS-пакету не сопоставлен NAS в модуле.
7	Не найден код услуги	В конфигурации NASa не определены коды услуг.
8	Карта просрочена	Истёк период годности карты.
9	Карта заблокирована	Карта не передана дилеру.
10	Карта активирована на баланс	Карта уже активирована для пополнения баланса.
11	Цена не найдена	На одну или несколько услуг, заявленных в конфигурации NASa, не найдена цена в тарифном плане.
12	Соединение не может быть установлено.	В пакете либо нет атрибута h323-conf-id, либо нет логина (атрибуты User-Name или Calling-Station-Id).
13	Тип звонка не определён	Тип Voip-звонка не установлен на основании анализа пакета.
14	Логин и карта не найдены	Не найдены ни логин, ни карта по данным пакета.
15	Нет денег на звонок	Недостаточно средств для минуты звонка.
16	Звонок не тарифицируется	Данный тип вызовов не тарифицируется в биллинге (указано в типе логина).
17	Невозможно активировать карту на этом NASe	Карта не может быть активирована на данном NASe, её услуга активации не прописана в параметре card.activate.service конфигурации NASa.
26	Доступ заблокирован	Доступ логина установлен в Запрещён .


32	Префикс Запрещён	Звонок запрещён узлом Запрещение звонка .
33	Договор не активен	Статус договора не позволяет авторизовать логин.
35	Истек срок жизни карточного договора	У договора, для создания которого активирована карта, закончился период действия.
40	Направление не найдено	В тарифном плане не указано направление звонка.

Возможно изменение текста отображаемой ошибки, для этого в конфигурации модуля указывается:

```
error.message.code.<code>=<>
```

Где **<code>** - код ошибки, **<название>** - отображаемое в таблице название.

Настройка RADIUS-сервера для VoiceIP

BGRadiusVoip обновляется как обычное [серверное приложение биллинга](#). Необходимо обновить приложение перед первым запуском. Необходимо перезагружать BGRadiusVoip при любом изменении конфигурации модуля, типов логинов, либо конфигурации NASов. Считывание  этих конфигураций происходит только при старте RADIUS-сервера.

Установка BGRadiusVoip на платформу Linux

1) Извлеките **BGRadiusVoip** из архива и скопируйте в каталог **/usr/local**;

2) Перейдите в каталог **/usr/local/BGRadiusVoip**;

3) Удалите все **.ini**, **.bat** и **.exe** файлы:

```
rm -f *.*.bat & rm -f *.*.exe & rm -f *.*.ini
```

4) Откройте для редактирования файл **radius.sh** и пропишите в нем путь к Java-машине, например, так:

```
...  
cd ${0%${0##*/}}.  
  
JAVA_HOME=/opt/java/jdk16  
  
if [ -z "$JAVA_HOME" ]; then  
    echo "The JAVA_HOME environment variable is not defined"  
    echo "This environment variable is needed to run this program"  
    exit 1  
fi  
...  
...
```

5) Проверьте **.sh** файлы на наличие символов **^M**, если символы присутствуют их можно удалить вручную, либо воспользоваться утилитой:

```
dos2unix *.sh
```

6) Установите права запуска для всех ***.sh** файлов:

```
chmod 744 *.sh
```

7) Возьмите из каталога **BGRadiusVoip/script** скрипт запуска **bgradius_voip** и скопируйте его в каталог **/etc/init.d**, установите права на исполнение (см. выше);

8) Выясните текущий уровень запуска системы командой:

```
[root@gate init.d]# runlevel  
N 3
```

9) Создайте линк для автоматического запуска RADIUS-сервера:

```
ln -s /etc/init.d/bgradius_voip /etc/rc5.d/S99bgradius_voip
```

10) Произведите настройку **radius.properties** и запустите RADIUS-сервер (см. далее).

Установка BGRadiusVoip на платформу Windows

Для установки BGRadiusVoip на платформу Windows на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить RADIUS, стоит Java-машина. Если её нет, установите версию не меньше 1.6.0. Загрузить можете с нашего сайта;
- 2) Загрузите с сервера RADIUS-сервер для VoiceIp;
- 3) Распакуйте архив на диск C: ;
- 4) Установите переменную окружения **BGRAD_HOME_VOIP=C:\BGRadiusVoip**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;
- 5) Установите службу BGRadiusVoip, для чего запустите файл **radius_install.bat**;
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **radius_uninstall.bat** .

Настройка файла radius.properties

Откройте файл **radius.properties** и произведите настройку:

```
#-
processor.class=ru.bitel.bgbilling.modules.voiceip.radius.VoiceIpRadiusProcessor
# VoiceIP
processor.mid=XXX

# ( 7.0)
auth.host=*
acct.host=*

#
auth.port=1812
acct.port=1813

#
admin.port=1955

#
db.driver=com.mysql.jdbc.Driver
db.url=jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false
db.user=bill
db.pswd=bgbilling

# MQ
mq.url=failover:(nio://127.0.0.1:61616)
mq.user=bill
mq.pswd=bgbilling
```

Установите переменную **processor.mid**=<числовой код экземпляра модуля VoiceIP>. При необходимости скорректируйте параметры доступа к базе данных и к **MQ**-серверу. Параметры **auth.port** и **acct.port** - порты, на которых сервер будет слушать авторизацию и аккаунт. Установлены значения по умолчанию. Параметр **admin.port** определяет порт, на котором работающий процесс BGRadiusVoip открывает сокет управления, данный порт не должен быть доступен с других машин. Если на этой же машине установлены другие сервера, использующие такие же порты, их следует изменить.

Изменения описанных в данном разделе параметров в radius.properties применяются только после перезапуска BGRadiusVoip. Все иные настройки могут быть изменены в ходе работы приложения и применяются при редактировании NASов и их конфигураций, редактировании конфигурации ядра (система алармов), конфигурации модуля.

Управление BGRadiusVoip

Для запуска и останова сервера RADIUS для VoiceIp используйте:

- 1) для Windows: консоль запуска и управления службами, служба BGRadiusVoip;
- 2) для UNIX: скрипты **radius_start.sh** и **radius_stop.sh**.

После запуска посмотрите логи в папке **BGRadiusVoip/log**.

- **radius.log** - распечатка пакетов запросов и ответов;
- **radius.out** - выходной поток, критичные ошибки;
- **connection.log** - лог хода соединения, обчетов.

Если запуск прошёл успешно, в логе **connection.log** должен вывестись список загруженных NASов, указанных вами в модуле Voip.

В **radius.log** должно быть сообщение вида:

```
INFO 18.05.2004 13:04:41 Starting radius auth_port:1812 acct_port:1813 admin_port:1899
INFO 18.05.2004 13:04:41 Init processor
class: bitel.billing.server.processor.voiceip.VoiceIpProcessor
mid: 6

INFO 18.05.2004 13:04:42 Starting PortListener port=1812|type=AUTH_LISTENER
INFO 18.05.2004 13:04:42 Starting PortListener port=1813|type=ACCOUNT_LISTENER
INFO 18.05.2004 13:04:42 Starting PortListener port=1899|type=ADMIN_LISTENER
```

Это свидетельствует о том, что сервер запущен и ожидает пакеты.

Если сервер не запустился, ищите причину в файле **radius.out**. В него пишутся все критические ошибки.

С работающего RADIUS-сервера возможно получение с сервера списка соединений и статуса. Это достигается запуском скрипта **radius.bat(.sh)** с параметрами. Список параметров можно получить простым запуском **radius.bat(.sh)**. Ниже приведена выводимая при этом справка:

```
Usage: [start|stop|help|status|ps|kill|flush_script_cache]
Parameters:
  help|? - show this help
  start - starting RADIUS server
  stop - stopping RADIUS server
  status - current connections status
  flush_script_cache - flush BGS script cache
##### Only for DialUp RADIUS #####
  ps - active connections list
  kill [-port <#port>] [-nas ] [-login <#login>]
      - kill connections by filter
  kill doesn't work with empty params list

Example: radius.sh start
Example: radius.sh kill -nas supernas.bayan.com -login 11
```

radius.sh(.bat) status - краткий статус сервера

```
[bill@bill-2 radius_voip]$ ./radius_status.sh
version 4.2 from 16.05.2007
19.02.2008 17:32:34 Request for minutes:
    Account for min: 98; for five min: 619; for ten min: 1253
    Auth for min: 39; for five min: 191; for ten min: 381
Waiting connections: 63
Started: 05.09.2007 06:10:13 Uptime: 167 d 12:22:20
Memory total: 291 176 448; max: 807 796 736; free: 40 794 064
Trees in cache: 28 Connections pool status Idle: 35; Active: 0; maxActive: 300; maxIdle: 100
```

Выводимая информация построчно:

1. Версия, номер и время сборки BGRadiusVoip;
2. Текущее время;
3. Количество запросов аккаунтинга в минуту, пять минут и десять минут;

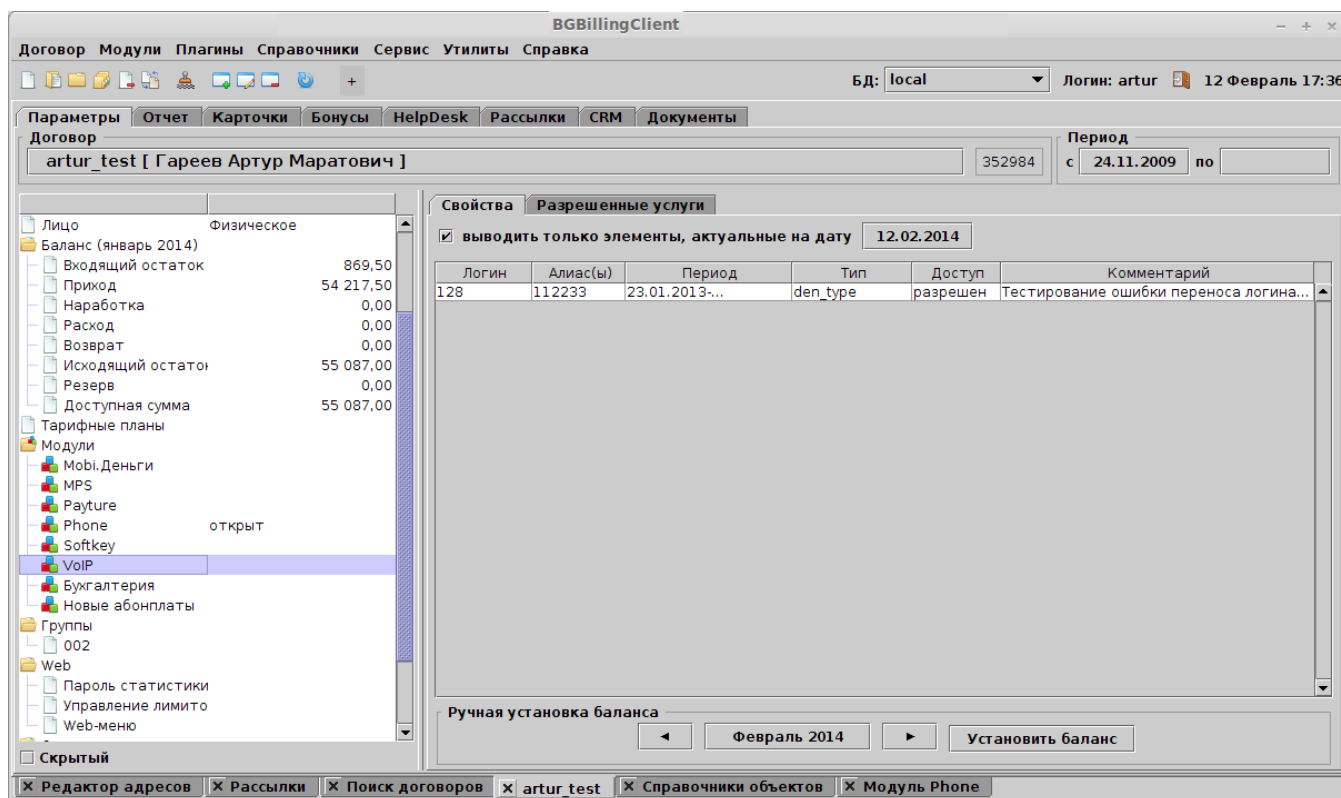
4. Количество запросов авторизации в минуту, пять минут и десять минут;
5. Время старта и uptime BGRadiusVoip;
6. Зарезервированная память, максимально доступная память и свободная в зарезервированной области память;
7. Количество тарифных деревьев в кэше, число соединений с БД, простаивающих, активных, максимально допустимых активных и максимально допустимых простаивающих. Настраивается в **radius.properties**.

radius.sh(.bat) flush_script_cache - сброс кэша [скриптов](#) преобработки RADIUS-запросов.

Клиентская настройка

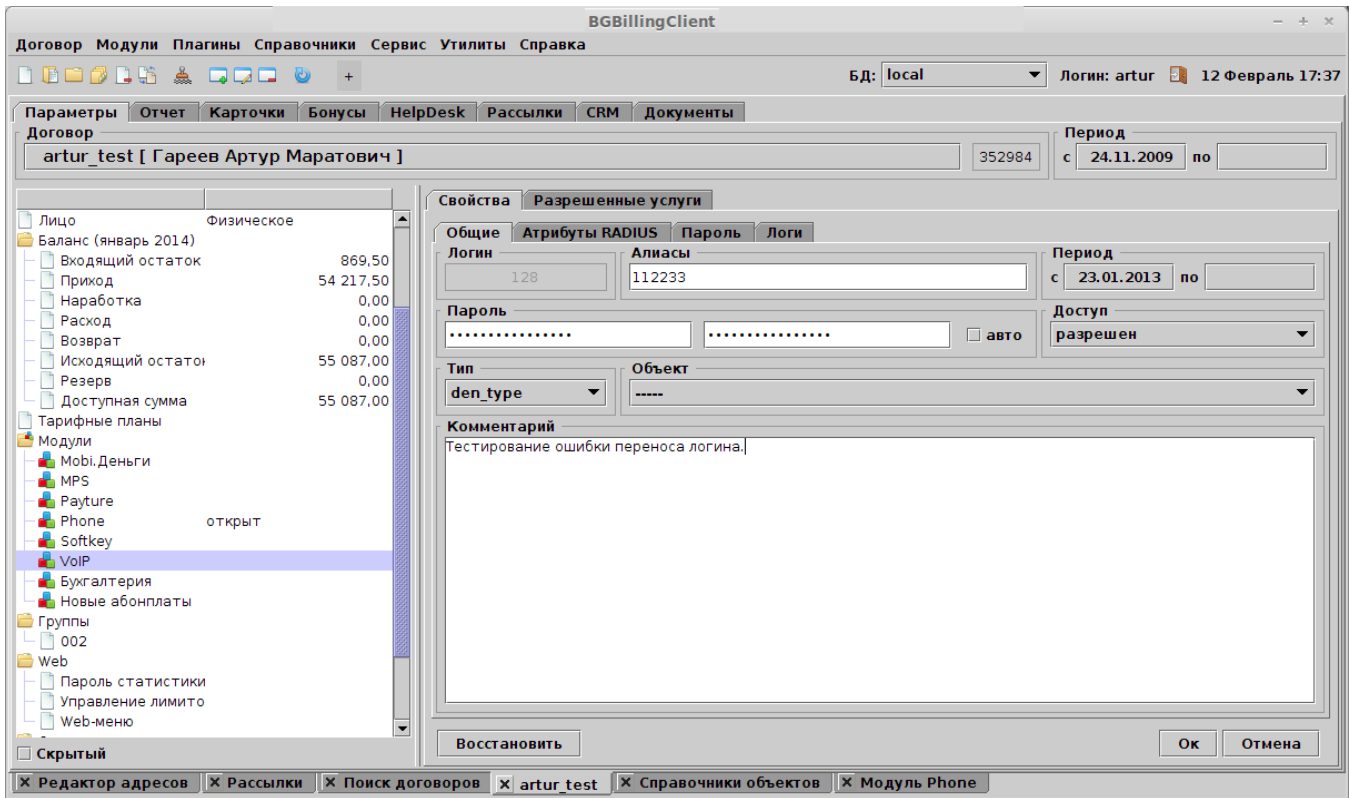
Для подключения модуля к договору добавьте в него экземпляр модуля, выбрав узел **Модули** дерева договора и выбрав модуль из списка установленных модулей. Добавление разрешённых услуг модуля имеет смысл при установке опции **check.service=1** в конфигурации модуля. Клиент не сможет авторизоваться, если в договоре не будет хотя бы одной из прописанных на NASE услуг.

Добавление логинов производится в вкладке свойства модуля.

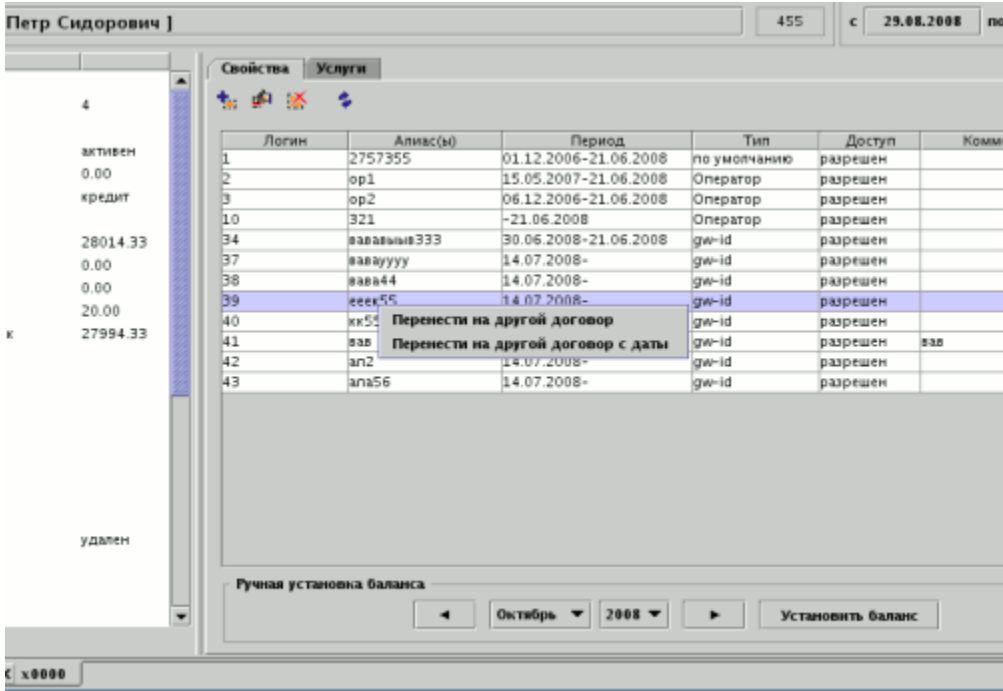


Числовые логины выдаются последовательно и не могут быть изменены. Они могут быть использованы для идентификации, так же как **Алиасы**. У каждого логина могут быть установлены несколько **Алиасов** (перечисленных через пробел), в которых могут быть, например, номера телефонов клиентов, их IP-адреса, либо другие идентификаторы на гейткиперах. Отличия алиаса в том, что в нем могут быть указаны произвольные значения (при отсутствии конфликтов с алиасами других логинов).

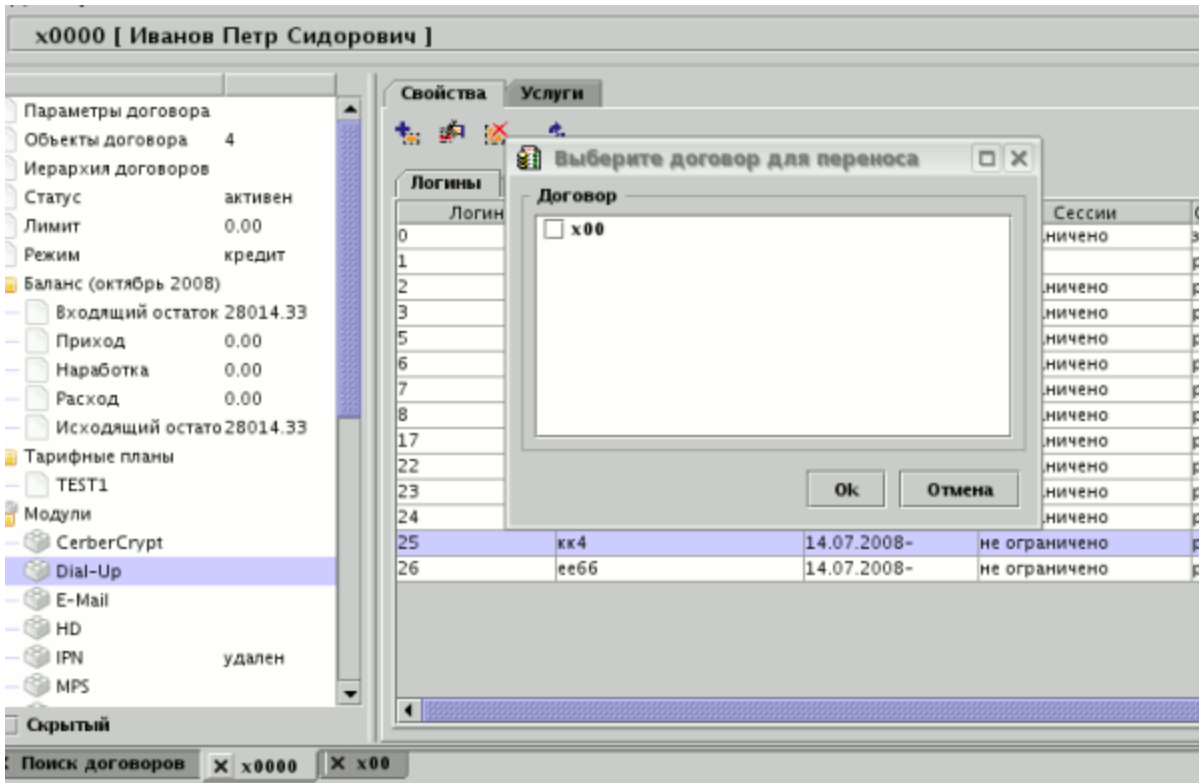
Каждому логину указывается тип, определяющий его поведение при поиске и обчёте. Поиск логина может производиться путём сопоставления произвольных атрибутов RADIUS-запроса с числовым логином либо алиасом, все зависит от настройки режимов поиска. Обратите внимание на поле ручной установки баланса в нижней части таблицы, он необходим для типов логинов, для которых не осуществляется обчёт после каждого звонка.



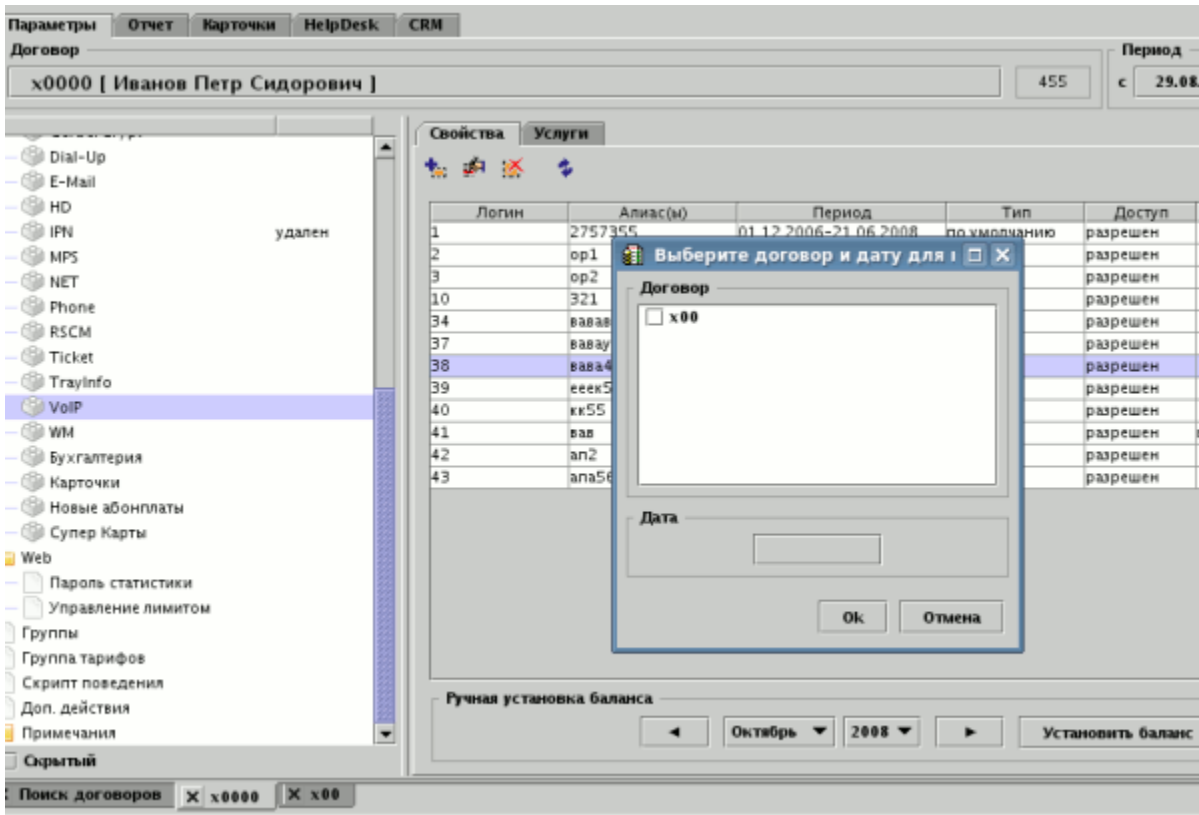
При выборе записей в таблице логинов и нажатии правой кнопкой мыши отображается всплывающее меню.



Пункт **Перенести на другой договор** переносит логин из одного договора в другой (карточка должна быть открыта в текущий момент), при этом перемещается и вся наработка по данному логину. При переносе переносится наработка по логину только на период действия логина.



Пункт **Перенести на другой договор с даты** закрывает логин на существующем договоре и открывает аналогичный логин на целевом договоре с другим периодом. Все алиасы и свойства логина копируются. Возможен перенос только логина с неустановленной датой закрытия.



В обоих режимах перераспределение наработки между договорами происходит автоматически, но для этого должен быть запущен планировщик заданий.

Интеграция VoiceIP с модулем "Карточки"

Производится аналогично [модулю DialUP](#).

Поиск по базе карточек производится при авторизации, если последовательно не был найден клиент с использованием указанных режимов поиска. При поиске карты используется значение поля User-Name, которое сравнивается с логином карты. Если карта найдена и не активирована, то создаётся договор с шаблоном, указанным в карте, а в него заносится логин с типом [По умолчанию](#).

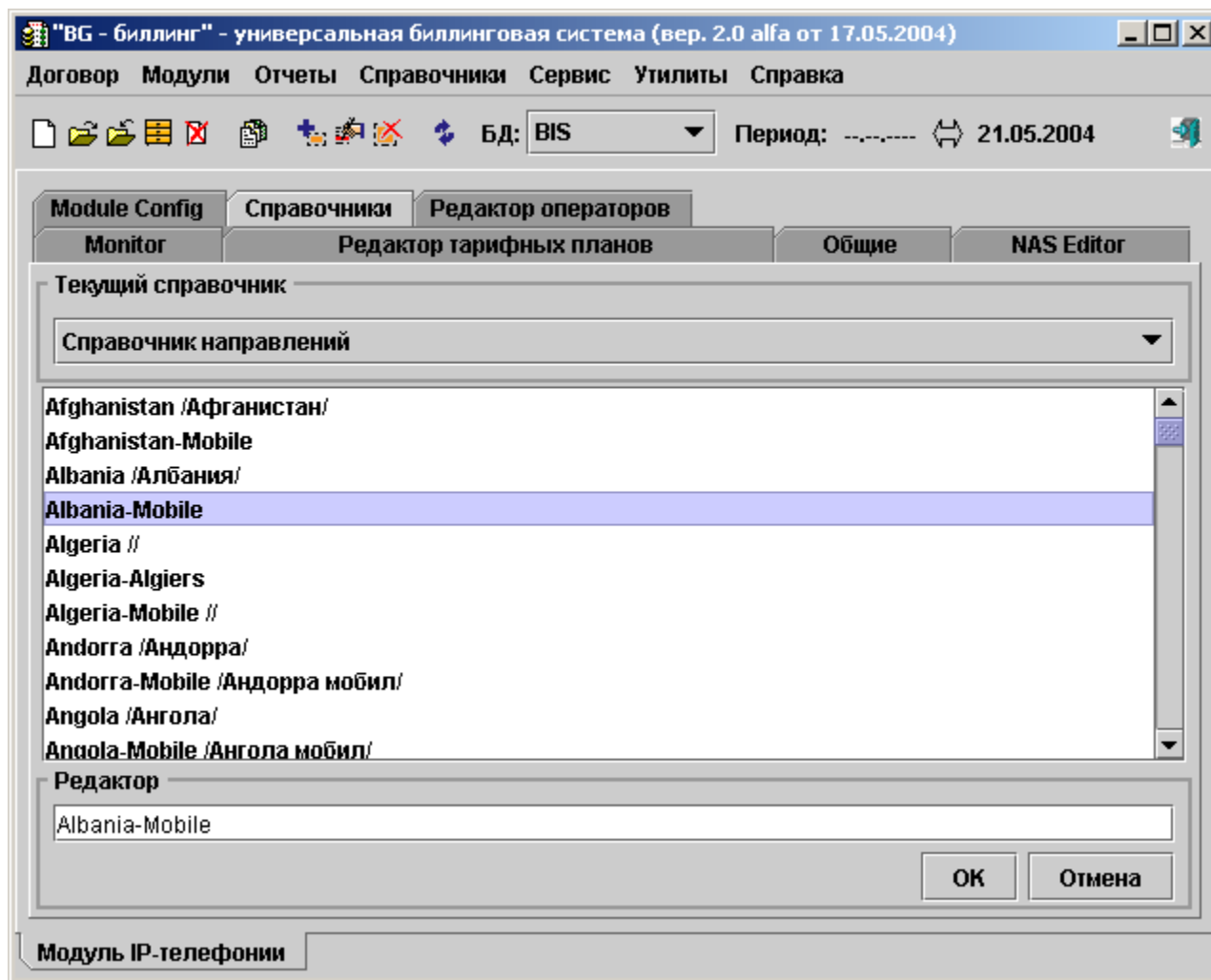
При следующем звонке RADIUS-сервер должен найти этот логин, для этого в режимах поиска должен присутствовать метод User-Name=LOGIN.

Настройка тарифных планов VoiceIP

Определение стоимости звонка производится по префиксу и типу звонка (входящий/исходящий). Кроме того, для отчётности все звонки делятся по направлениям. Для создания нового направления откройте вкладку **Модули=>Модуль IP телефонии=>Справочники**. Логика поиска тарифа соответствует [Алгоритму 1](#).

В данном модуле единственный справочник - направлений. Для того чтобы редактировать его используйте панель инструментов.

Для добавления направлений вы можете также использовать контекстные редакторы в дереве тарифного плана.



Задача тарификации в модуле - определение стоимости минуты, направления звонка и параметров тарификации (опционально). Исходными параметрами тарифного запроса выступают:

- **Услуга звонка** - определяется в конфигурации NASa;
- **Тип звонка** - входящий/исходящий;
- **Номер** - в зависимости от типа звонка и типа логина либо входящий, либо исходящий;
- **Время начала звонка** - может быть использовано для определения льготных периодов.

Последовательно запрос передаётся во все персональные тарифные планы договора, затем в глобальные тарифы, установленные для договора. Просмотр тарифов идёт в порядке позиций, выбираются только тарифные планы с периодом действия, включающим дату начала звонка и содержащие тарифное поддерево для данного экземпляра модуля.

Обработка прекращается, если тариф вернёт в запросе следующие данные:

- **Стоимость минуты звонка;**
- **Направление звонка;**
- **Параметры тарификации** (не обязательно).

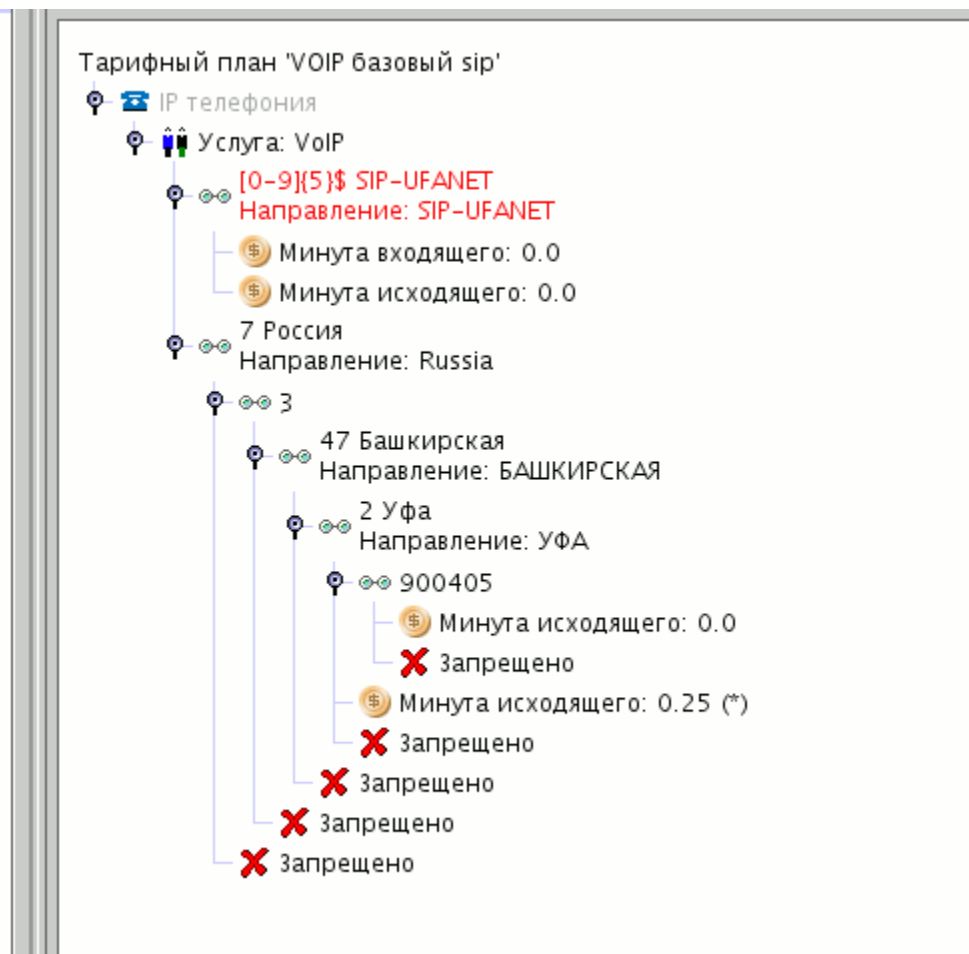
Верхний узел в каждом тарифе - типа **Услуга**. В нем нужно определить услугу, для которой определяется стоимость. В модуле может быть несколько услуг, услуга привязывается к NASy (см. конфигурацию NASов). Таким образом, вы можете дифференцировать стоимость звонка через разные шлюзы.

Далее тарифный план может быть построен двумя путями. Описание логики работы узлов вы можете найти [здесь](#).

Тарификация по префиксам в модуле VoiceIP

Разбор префикса производится непосредственно в тарифном плане, с использованием узлов **Часть префикса**, **Диапазон префиксов**.

Приведённый ниже тариф устанавливает стоимость входящих и исходящих вызовов с пятизначных номеров как нулевую. Направление на Россию, Башкирию и Уфу заблокированы с помощью узла **Запрещение звонка**. Этот узел специфичен для модуля VoiceIP - при прохождении запроса через него в момент авторизации пользователя выдаётся ошибка авторизации **Prefix deny**.



Следующий тариф определяет стоимость звонков по всем направлениям с помощью узла **Часть префикса**, в последних версиях биллинга часть из этих узлов может быть заменена **Диапазоном префиксов**. На направление USA определена льготная и обычная цена, с использованием узла **Фильтр по времени**. В последних версиях этот узел также может быть заменён **Фильтром по типу времени**.

Тарифный план 'VOIP базовый'

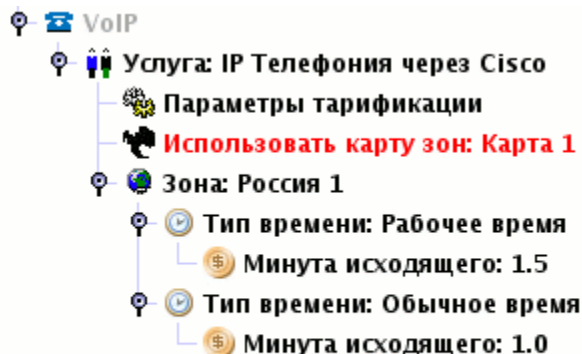
- Dial-Up
- IP телефония
- Услуга: VoIP
 - [0-9][5]\$ SIP-UFANET
 - Направление: SIP-UFANET
 - 1 США и Канада
 - (20[23])(20[5-9])(210)(21[3-9])(22[01])(22[3-5])(22[7-9])(23[12])(23[4-7])(24[03])(24[7-9])(25[1-4])(25[68])(26[0-2])(26[79])(27[01])(27[5-8])(28[0-3])(28[5-7])(30[1-5])(30[7-9])(310)(311)(312)(313)(314)(315)(316)(317)(318)(319)(320)(321)(322)(323)(324)(325)(326)(327)(328)(329)(330)(331)(332)(333)(334)(335)(336)(337)(338)(339)(340)(341)(342)(343)(344)(345)(346)(347)(348)(349)(350)(351)(352)(353)(354)(355)(356)(357)(358)(359)(360)(361)(362)(363)(364)(365)(366)(367)(368)(369)(370)(371)(372)(373)(374)(375)(376)(377)(378)(379)(380)(381)(382)(383)(384)(385)(386)(387)(388)(389)(390)(391)(392)(393)(394)(395)(396)(397)(398)(399)(400)(401)(402)(403)(404)(405)(406)(407)(408)(409)(410)(411)(412)(413)(414)(415)(416)(417)(418)(419)(420)(421)(422)(423)(424)(425)(426)(427)(428)(429)(430)(431)(432)(433)(434)(435)(436)(437)(438)(439)(440)(441)(442)(443)(444)(445)(446)(447)(448)(449)(450)(451)(452)(453)(454)(455)(456)(457)(458)(459)(460)(461)(462)(463)(464)(465)(466)(467)(468)(469)(470)(471)(472)(473)(474)(475)(476)(477)(478)(479)(480)(481)(482)(483)(484)(485)(486)(487)(488)(489)(490)(491)(492)(493)(494)(495)(496)(497)(498)(499)(500)(501)(502)(503)(504)(505)(506)(507)(508)(509)(510)(511)(512)(513)(514)(515)(516)(517)(518)(519)(520)(521)(522)(523)(524)(525)(526)(527)(528)(529)(530)(531)(532)(533)(534)(535)(536)(537)(538)(539)(540)(541)(542)(543)(544)(545)(546)(547)(548)(549)(550)(551)(552)(553)(554)(555)(556)(557)(558)(559)(560)(561)(562)(563)(564)(565)(566)(567)(568)(569)(570)(571)(572)(573)(574)(575)(576)(577)(578)(579)(580)(581)(582)(583)(584)(585)(586)(587)(588)(589)(590)(591)(592)(593)(594)(595)(596)(597)(598)(599)(600)(601)(602)(603)(604)(605)(606)(607)(608)(609)(610)(611)(612)(613)(614)(615)(616)(617)(618)(619)(620)(621)(622)(623)(624)(625)(626)(627)(628)(629)(630)(631)(632)(633)(634)(635)(636)(637)(638)(639)(640)(641)(642)(643)(644)(645)(646)(647)(648)(649)(650)(651)(652)(653)(654)(655)(656)(657)(658)(659)(660)(661)(662)(663)(664)(665)(666)(667)(668)(669)(670)(671)(672)(673)(674)(675)(676)(677)(678)(679)(680)(681)(682)(683)(684)(685)(686)(687)(688)(689)(690)(691)(692)(693)(694)(695)(696)(697)(698)(699)(700)(701)(702)(703)(704)(705)(706)(707)(708)(709)(710)(711)(712)(713)(714)(715)(716)(717)(718)(719)(720)(721)(722)(723)(724)(725)(726)(727)(728)(729)(730)(731)(732)(733)(734)(735)(736)(737)(738)(739)(740)(741)(742)(743)(744)(745)(746)(747)(748)(749)(750)(751)(752)(753)(754)(755)(756)(757)(758)(759)(760)(761)(762)(763)(764)(765)(766)(767)(768)(769)(770)(771)(772)(773)(774)(775)(776)(777)(778)(779)(780)(781)(782)(783)(784)(785)(786)(787)(788)(789)(790)(791)(792)(793)(794)(795)(796)(797)(798)(799)(800)(801)(802)(803)(804)(805)(806)(807)(808)(809)(810)(811)(812)(813)(814)(815)(816)(817)(818)(819)(820)(821)(822)(823)(824)(825)(826)(827)(828)(829)(830)(831)(832)(833)(834)(835)(836)(837)(838)(839)(840)(841)(842)(843)(844)(845)(846)(847)(848)(849)(850)(851)(852)(853)(854)(855)(856)(857)(858)(859)(860)(861)(862)(863)(864)(865)(866)(867)(868)(869)(870)(871)(872)(873)(874)(875)(876)(877)(878)(879)(880)(881)(882)(883)(884)(885)(886)(887)(888)(889)(890)(891)(892)(893)(894)(895)(896)(897)(898)(899)(900)(901)(902)(903)(904)(905)(906)(907)(908)(909)(910)(911)(912)(913)(914)(915)(916)(917)(918)(919)(920)(921)(922)(923)(924)(925)(926)(927)(928)(929)(930)(931)(932)(933)(934)(935)(936)(937)(938)(939)(940)(941)(942)(943)(944)(945)(946)(947)(948)(949)(950)(951)(952)(953)(954)(955)(956)(957)(958)(959)(960)(961)(962)(963)(964)(965)(966)(967)(968)(969)(970)(971)(972)(973)(974)(975)(976)(977)(978)(979)(980)(981)(982)(983)(984)(985)(986)(987)(988)(989)(990)(991)(992)(993)(994)(995)(996)(997)(998)(999)(1000)
 - Направление: USA /США/
 - Период: 20.09.2006
 - Набор ограничений:
 - Месяцы => 11; Числа месяца => 3 - 4
 - Дни недели => 6 - 7
 - Часы => 0 - 8, 20 - 23
 - Месяцы => 12; Числа месяца => 24
 - Месяцы => 10; Числа месяца => 11
 - Месяцы => 5; Числа месяца => 1, 9
 - Месяцы => 2; Числа месяца => 23
 - Месяцы => 3; Числа месяца => 8
 - Месяцы => 1; Числа месяца => 1 - 9
 - Месяцы => 6; Числа месяца => 12
 - Минута исходящего: 3.0
 - Набор ограничений:
 - Минута исходящего: 4.5
 - (204)(306)(343)(403)(416)(418)(437)(438)(450)(460)(474)(506)(514)(519)(579)(600)(604)(613)(647)(705)(709)(778)(780)(807)(819)(825)(867)(879)(902)(905)(942)(226)(250)(289)
 - Направление: CANADA /КАНАДА/
 - Минута исходящего: 4.5
 - (212)(347)(646)(718)(917)
 - Направление: USA-NEW YORK (CITY) /США НЬЮ-ЙОРК/
 - Минута исходящего: 4.5
 - 242 Bahamas
 - Направление: BAHAMAS /БАГАМСКИЕ ОСТРОВА/
 - 357 Bahamas-Mobile
 - Направление: BAHAMAS-MOBILE
 - 359 Bahamas-Mobile
 - Направление: BAHAMAS-MOBILE
 - 427 Bahamas-Mobile
 - Направление: BAHAMAS-MOBILE
 - 457 Bahamas-Mobile
 - Направление: BAHAMAS-MOBILE
 - 477 Bahamas-Mobile
 - Направление: BAHAMAS-MOBILE

Построение тарифов по методу непосредственного разбора целесообразно для небольших тарифов (терминация трафика оператора), либо для тарифов, которые не могут быть приведены к перечню зон с равными ценами.

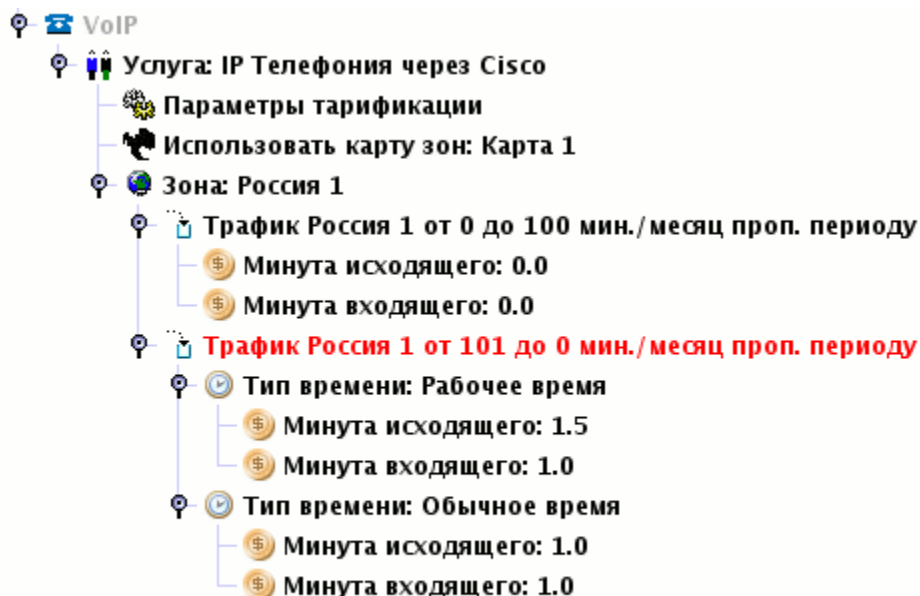
Тарификация по зонам модуля VoiceIP

В тарифном плане возможно использование фильтра по **типам времени** и зонной тарификации. Зонная тарификация основывается на справочнике географических кодов и картах зон, логика работы их совершенно идентична **модулю телефонии**.

Пример тарифного плана, использующего обе эти возможности, приведён на скриншоте. Использование типов времени особенно удобно в больших тарифных деревьях. Справочники географических кодов и карт зон модулей Телефонии и IP-телефонии могут быть импортированы и экспортированы друг в друга.



Следующий пример тарифа предоставляет квоту на зону Россия 1 в 100 минут в месяц на каждый логин. Флаг **Пропорционально периоду** означает, что если логин был активен только часть месяца, квоты будут пропорционально уменьшены. При превышении квоты по зоне стоимость минуты различается в зависимости от времени.



Тарификация по карте цен модуля VoiceIP

Тарификация по карте цен аналогична [модулю телефонии](#).

Смешанный режим

Подходит при наличии некоторых исключений из зонового тарифа. Перед зонами размещаются части префикса. В них необходимо устанавливать фиктивную зону для предотвращения повторной тарификации в узле **Зона**. Направление берётся из справочника тарифных зон.

й)
й)
!
И
леком
леком (руб)
леком
леком 001
ртный)

Тарифный план 'Голос Ангарск КВАНТ'

- ☎ Телефония
 - 👤 Услуга: Телефония
 - 👤 Услуга: Телефония (область)
 - ⚙️ Параметры тарификации
 - 🗺️ Использовать карту зон: Общая
 - 📞 7
 - 886
 - 📍 Зона: без Зоны
 - 💰 Минута исходящего: 15.00
 - 86
 - 📍 Зона: без Зоны
 - 💰 Минута исходящего: 10.00
 - 852
 - 📍 Зона: без Зоны
 - 💰 Минута исходящего: 10.00
 - 🌐 Зона: Австралия, Океания
 - 🌐 Зона: Азия 1
 - 🌐 Зона: Азия 2
 - 🌐 Зона: Америка 1
 - 🌐 Зона: Америка 2
 - 🌐 Зона: Африка
 - 🌐 Зона: Европа 1
 - 🌐 Зона: Европа 2
 - 🌐 Зона: Европа, мобильные телефоны
 - 🌐 Зона: Инмарсат
 - 🌐 Зона: Иркутская область

Модификация стоимости звонка

С 4.1 версии в наследованных тарифах возможно использование **Множителя цены**. Принципы его использования можно посмотреть [здесь](#).

С 4.3 версии биллинга в наследованных тарифах возможно использование узла Изменение стоимости. При прохождении тарифного запроса через данный узел стоимость минуты звонка, либо всего звонка целиком может быть увеличена. Ниже приведён пример, как в расширенном тарифе стоимость каждого звонка увеличивается на 10 рублей. Данная функция может быть использована для любого типа тарифов.

The screenshot shows a configuration tree for a VoIP tariff plan. The tree structure is as follows:

- VoIP [Тарифный план 'VOIP']
 - Услуга: IP Телефония через Cisco
 - Параметры тарификации
 - Использовать карту зон: Карта 1
 - Зона: Россия 1
 - Трафик Россия 1 от 0 до 100 мин./месяц проп. периоду
 - Трафик Россия 1 от 101 до 0 мин./месяц проп. периоду
 - Тип времени: Рабочее время
 - Минута исходящего: 1.5
 - Минута входящего: 1.012345

A dialog box is open over the 'Тип времени: Рабочее время' node, showing a modification of 10 units 'за минуту' (per minute). The dialog includes a plus sign, a text input field with '10', a dropdown menu with 'за минуту', and confirmation (green checkmark) and cancellation (red X) buttons. A dropdown menu is also visible below the dialog, showing options: 'за минуту' (selected), 'за минуту', and 'за сессию'.

The screenshot shows the same configuration tree as above, but with an additional parameter added under the 'Тип времени: Рабочее время' node:

- Тип времени: Рабочее время
 - Минута исходящего: 1.5
 - Минута входящего: 1.012345
 - Изменение стоимости: +10.0 за сессию**

The 'Изменение стоимости' parameter is highlighted in red, indicating it is the active modification.

Импорт/экспорт тарифов

Тарифные планы IP-телефонии могут быть экспортированы и импортированы из XML-формата. Данная возможность описана [здесь](#).

Работа с операторами

Модуль IP-телефонии позволяет организовать полноценную работу с операторами Voip с учётом терминции и оригинации трафика.

Старая схема

Логины операторов заводятся аналогично обычным логинам, но в типе операторского логина устанавливаются опции обчёта входящих звонков по вызываемому номеру. В тарифных планах операторов стоимость входящего звонка соответствует той цене, которую мы платим оператору за звонок через него. Она должна быть установлена отрицательной.

Стоимость в тарифном плане оператора соответствует той цене, которую оператор должен нам за терминацию его звонка.

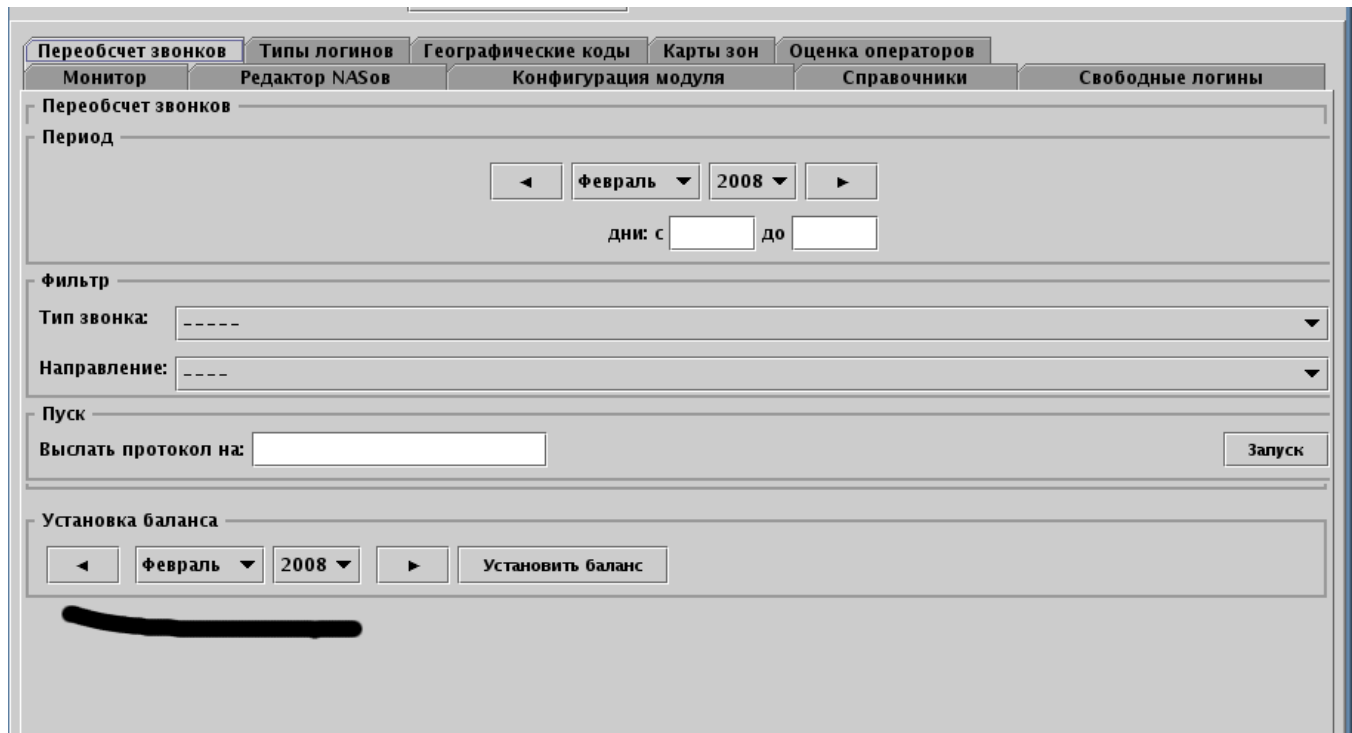
Таким образом, положительный баланс оператора означает его долг нам, а отрицательный - нашу задолженность ему. Платежи оператора заносятся как приходы. Наши платежи оператору заносятся как расходы.

В свойствах логинов операторов необходимо отключить обсчёт баланса, для его обсчёта используется ручной режим.



На скриншоте изображена панель запуска ручной установки баланса, её необходимо производить каждый раз, как только требуется узнать текущий баланс оператора. В случае, если вы не устанавливали баланс за прошлый месяц, необходимо сначала установить баланс предыдущего месяца, а затем текущий.

С 4.4 версии возможна групповая установка баланса для всего модуля на вкладке **Переобсчёт звонков**. Необходимо выбрать месяц и нажать **Установить баланс**.

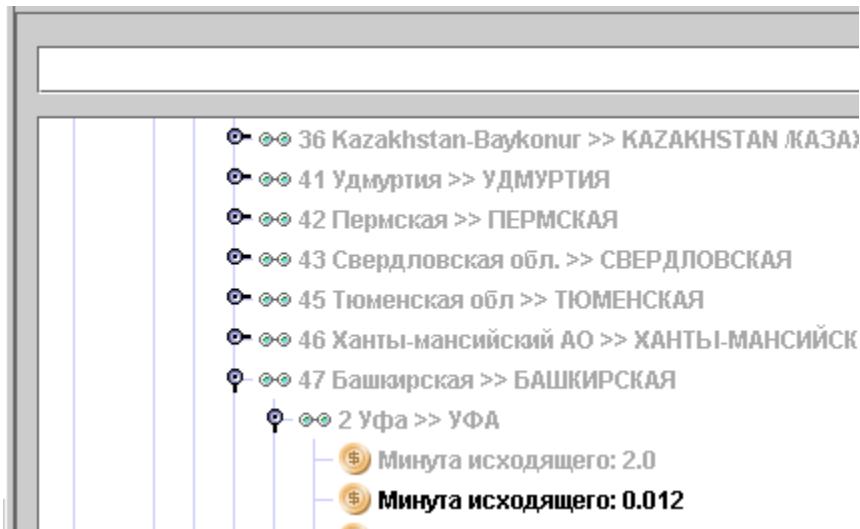


Также возможна настройка задачи планировщика **Установка балансов VoiceIP**. В параметрах задачи должно быть указано **mid**=<код экземпляра модуля VoiceIP>. При выполнении задача берет месяц предыдущего от текущего часа и производит установку балансов на этот месяц.

Более простым является вариант использования положительных цен, в таком случае баланс не используется вовсе, а для сверок берутся значения отчётов по входящим и исходящим сессиям оператора.

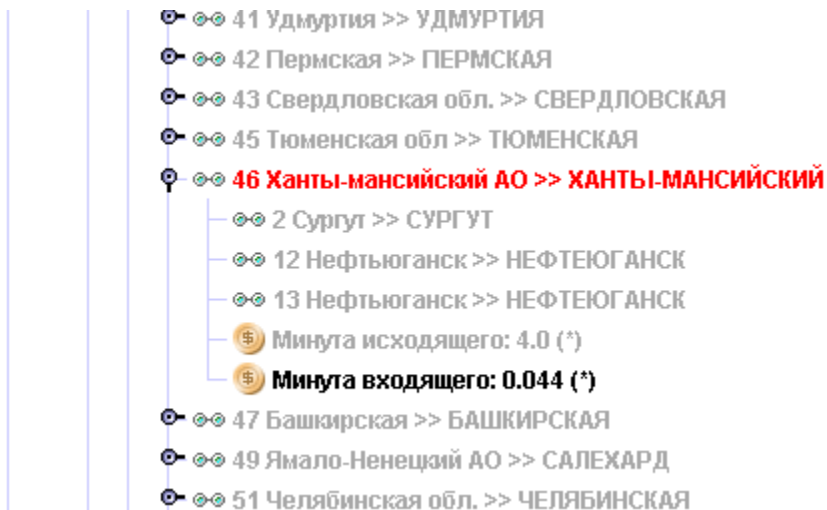
Примерная организация тарифных планов может выглядеть следующим образом: создаётся БАЗОВЫЙ тариф, включающий в себя дерево всех направлений и префиксов + стоимость исходящих звонков. Этот базовый тариф может использоваться для тарификации карточных договоров.

Для каждого оператора устанавливается этот тариф, после чего он расширяется и устанавливаются входящие цены оригинации на данного оператора + исходящие цены звонков по различным направлениям для договора.



Обратите внимание, что цена исходящего звонка из базового тарифа перекрыта ценой исходящего звонка из тарифа-расширения.

Также в тарифах операторов могут использоваться "Цены по умолчанию".



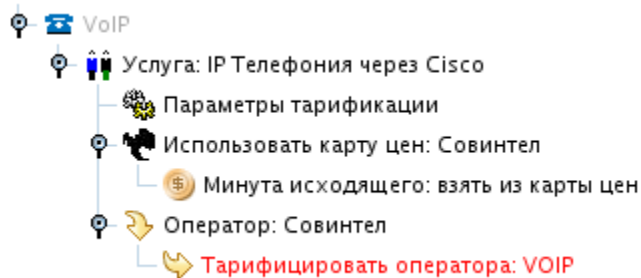
Цена по умолчанию проставляется в тарифный запрос только, если в нем уже нет другой цены. Например, в данном случае происходит разбор тарифа по тарифному дереву, т.е. обработка запроса идёт до совпавшей ветке и если звонок идёт на .4613., то запрос будет помечен обработанным на узле "Нефтьюганск", после чего все остальные узлы дерева префиксов будут пропускать его без обработки. Направление звонка также будет установлено в "НЕФТЬЮГАНСК".

Если установить простую цену в конце 46 узла, то она будет прописывать себя во все проходящие через нее запросы, т.е. если мы поместим спец. цену на 12 узел, она будет перетёрта в дальнейшем при прохождении цены в конце 46 узла.

Однако, т.к. цена помечена звёздочкой, т.е. ставится только по умолчанию, то она будет установлена только, если в запросе нет других цен.

Новая схема

Тарификация стоимости звонка для оператора осуществляется во время тарификации звонка клиента. Для этого используется узел тарифа **Тарифицировать оператора**, в котором указывается ссылка на тариф оператора. Из него извлекается подходящая цена и правила округления.



Чтобы разделить тарификацию на разных операторов используется узел **Оператор**. Операторы задаются в конфигурации модуля:

```
#
operator.1.title=VTK
#(id) -
operator.1.cid=135121
# c 813 813
operator.1.account.813=813
operator.2.title=
operator.2.cid=135122
# 4
operator.1.account.4=all
```

Для привязки звонка к оператору необходимо устанавливать его код в скрипте предобработки запроса наса:

Код

```
import bitel.billing.server.radius.*;

prefix = "remote-media-address";
length = prefix.length();

attributes = request.getVendorAttributes( Vendors.CISCO, CiscoVendor.Cisco_AVPair );
if( attributes != null )
for( it = attributes.iterator(); it.hasNext(); )
{
    ra = it.next();
    value = ra.getStringValue();

    if( value.startsWith( prefix ) )
    {
        if( value.endsWith( "77.82.17.33" ) )
        {
            request.setOption( "operator", 1 );
            break;
        }
        else if( value.endsWith( "172.36.104.61" ) )
        {
            request.setOption( "operator", 2 );
            break;
        }
    }
}
```

Т.е. в зависимости от содержимого пакета нужно установить `request.setOption("operator", <код оператора>)`.

Также следует учитывать, что скрипт предобработки запроса обрабатывает на каждом запросе, при этом установка кода оператора обрабатывает на запросе аутентификации и на Stop-пакете. Если установка произойдет оба раза, то запомнится значение обработки Stop-пакета.

Код оператора сохраняется в звонке, так что при переобсчете ветка Оператор также будет учитываться.

Построение тарифа для оператора почти не отличается от построения тарифа для абонента, за тем исключением, что в тарифе оператора необязательно указывать направление.

Для создания наработки на операторских договорах необходимо в конфигурации модуля указать привязку: код договора - услуги, которые принадлежат этому оператору и указать код услуги модуля, на который будет ложиться наработка в договоре оператора.

```
#      (id ) 455
# ,
operator.1.cid=455
#      4
operator.1.account.4=10,11
#
#operator.1.account.4=all
```

Для того чтобы наработка на договоре оператора обсчитывалась нужно настроить задачу планировщика **Установка балансов VoiceIP**. При выполнении задача берет месяц предыдущего от текущего часа и производит установку балансов на этот месяц.

Оценка операторов

Для оценки стоимости звонка через разных операторов начиная с 4.3 версии в модуле добавлена вкладка **Оценка операторов**. Необходимо выбрать группу операторских договоров, префикс и запустить поиск.

В договорах должен быть установлен операторский тариф.

Договор	Префикс	Направление	Стоимость
XV0008-04 (IPLC)	7351	ЧЕЛЯБИНСКАЯ	
XV0014-04 (ООО "Мегаполис-Лайн")	7351	ЧЕЛЯБИНСКАЯ	
XV0002-06 (Воип Трейд (Был ЗАО "Воип Эксчейн...))	7351	ЧЕЛЯБИНСКАЯ	
XV0043-04 (ООО КомТелСвязь)	7351	ЧЕЛЯБИНСКАЯ	
XV0042-04 (БИС-Информ)	7351	ЧЕЛЯБИНСКАЯ	
XV0041-04 (ИКЦ Экспресс)	7351	ЧЕЛЯБИНСКАЯ	
XV0026-04 (ЕС Телеком)	7351	ЧЕЛЯБИНСКАЯ	
XV0020-04 (Альфа-Телеком (Maxifon))	7351	ЧЕЛЯБИНСКАЯ	
XV0022-06 (ООО Наука-Связь)	7351	ЧЕЛЯБИНСКАЯ	
XV0023-04 (ООО "Телекол")	7351	ЧЕЛЯБИНСКАЯ	
XV0025-04 (ОСС Урал)	7351	ЧЕЛЯБИНСКАЯ	
XV0029-04 (ООО "Комлайн")	7351	ЧЕЛЯБИНСКАЯ	
XV0030-04 (ООО "Кросстел.Ру")	7351	ЧЕЛЯБИНСКАЯ	
XV0032-04 (ООО «Инфоцентр»)	7351	ЧЕЛЯБИНСКАЯ	
XV0033-04 (ООО «Глобал Он Лайн»)	7351	ЧЕЛЯБИНСКАЯ	
XV0045-04 (ООО «Телекомтрейд»)	7351	ЧЕЛЯБИНСКАЯ	
XV0036-04 (ООО Юниверсум)	7351	ЧЕЛЯБИНСКАЯ	
XV0044-04 (ЗАО «Международная компания свя...))	7351	ЧЕЛЯБИНСКАЯ	
Оператор(тест-тест-тест) (тест)	7351	ЧЕЛЯБИНСКАЯ	
XV0048-05 (ЗАО Самарское предприятие Промс...))	7351	ЧЕЛЯБИНСКАЯ	
XV0059-05 (ЗАО Облком)	7351	ЧЕЛЯБИНСКАЯ	
XV0062-05 (Inter Netz / Nauk)	7351	ЧЕЛЯБИНСКАЯ	
XV0065-05 (ТТК центр)	7351	ЧЕЛЯБИНСКАЯ	
XV0068-05 ("OSS-ZVEZDA")	7351	ЧЕЛЯБИНСКАЯ	
XV0048-07 (ООО "Промсвязьмонтаж")	7351	ЧЕЛЯБИНСКАЯ	
XV0074-05 ("Лексико" Lexico Consulting L.C.)	7351	ЧЕЛЯБИНСКАЯ	
UFANET_OPER (Фиктивный договор для временн...))	7351	ЧЕЛЯБИНСКАЯ	
XV0082-07 (Оренбургская Городская сеть)	7351	ЧЕЛЯБИНСКАЯ	
XV0017-06 (ЗАО Евро-Тел)	735	ЧЕЛЯБИНСКАЯ	
XV0011-06 (ЗАО "Глобус-Телеком")	735	ЧЕЛЯБИНСКАЯ	
XV0067-06 (ЗАО "Корвет-ОСС (Был ЗАО "...	735	ЧЕЛЯБИНСКАЯ	

Для каждого оператора будет найдена цена входящего звонка по данному префиксу, далее будет выведена сводная таблица.

Отчёты VoiceIP

Доступны оперативные отчёты по **Сессиям**, **Наработке** и **Направлениям**.

Для получения отчёта по сессиям - выберите один или несколько логинов и нажмите кнопку вывода отчёта (галочка). Отображаются только те логины, период которых пересекается с указанным для поиска периодом. Дополнительно можно установить фильтр по номеру звонящего, при этом будут выведены только звонки, где в номере звонящего есть указанная подстрока. Также можно установить фильтр для вывода только платных звонков, только входящих, либо только исходящих. Двойной клик мышью в строке отчёта по сессиям показывает RADIUS-лог по текущей сессии.

Скриншот интерфейса отчётов VoiceIP. Вкладка "Сессии".

Тип	Направление	Время входа	Длительность	Окр. длитель...	Ст. мин.	Ст. звонка	С номера / На номер
<<	Chelyabinsk region	14.05.2007 16:38:08	00:00:00 [0]	00:00:00 [0]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:49:19	00:00:00 [0]	00:00:00 [0]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:49:38	00:00:00 [0]	00:00:00 [0]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:50:49	00:00:00 [0]	00:00:00 [0]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:53:38	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:07	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:07	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:08	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:08	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:09	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:10	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:11	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:54:12	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:55:02	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:55:02	00:00:10 [10]	00:00:10 [10]	0.00000	0.00000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:55:03	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:55:03	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:55:04	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 17:55:04	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 18:01:56	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 18:03:15	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 18:03:52	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510
<<	Chelyabinsk region	14.05.2007 18:06:58	00:00:10 [10]	00:00:10 [10]	1.50000	0.25000	3472543228 / 73510

Сводные данные:

Коп-во сессий:	27	Время [сек.][мин.]:	00:03:50 [230] [4]	Время окр.[сек.][мин.]:	00:03:50 [230] [4]	Сумма:	2.75
----------------	----	---------------------	--------------------	-------------------------	--------------------	--------	------

Для вывода отчёта по наработкам достаточно выбрать месяц и нажать кнопку вывода. Будут отображены суммарные количества сессий, суммарное время по всем логинам договора. Здесь также действуют фильтры по номеру, направлениям звонка и платности.

Скриншот интерфейса отчётов VoiceIP. Вкладка "Наработка".

Логин	Сессий	Время	Окр. время	Наработка
1[2757355]	0	00:00:00 [0]	00:00:00 [0]	0.00
2[op1]	27	00:03:50 [230]	00:03:50 [230]	2.75
3[op2]	1	00:00:00 [0]	00:00:00 [0]	0.00
10[321]	0	00:00:00 [0]	00:00:00 [0]	0.00
12[]	0	00:00:00 [0]	00:00:00 [0]	0.00

Сводные данные:

Коп-во сессий:	28	Время [сек.][мин.]:	00:03:50 [230] [3]	Время окр.[сек.][мин.]:	00:03:50 [230] [3]	Сумма:	2.75
----------------	----	---------------------	--------------------	-------------------------	--------------------	--------	------

Отчет по направлениям показывает наработку выбранных логинов в разрезе конечных пунктов звонка (направлений), либо исходящих пунктов звонка для входящих звонков. Чтобы посмотреть все звонки логина по данному направлению, получите отчёт по направлениям и дважды кликните по строке с интересующим вас направлением. Будет выдан отчёт по сессиям с сессиями только по этому направлению.

Параметры Отчет Карточки HelpDesk CRM Документы

Период: Ноябрь 2008
дни: с [] до []

Логин: 1 []

Логги: Сессии Нарabотка **Направления** Маска по номеру: [] 1 из 1 [25]

Все Входящие Исходящие Только платные Сортировать по: **Названию**

Направление	Количество	Время	Окр. время	Стоимость	ASR	ACD
Chelyabinsk region	2	00:01:06 [66]	00:01:06 [66]	5.50000	100%	00:00:33

Кол-во сессий: 2 Время [сек.][мин.]: 00:01:06 [66] [1] Время окр.[сек.][мин.]: 00:01:06 [66] [1] Сумма: 5.50

Dial-Up IPN Phone VoIP Бухгалтерия

Параметр **ASR** рассчитывается как $ASR = \frac{\text{КОЛИЧЕСТВО_НЕНУЛЕВЫХ_ЗВОНКОВ}}{\text{ОБЩЕЕ_ЧИСЛО_ЗВОНКОВ}}$ по направлению. **ACD** рассчитывается как $ACD = \frac{\text{СУММАРНАЯ_ОКРУГЛЕННАЯ_ДЛИТЕЛЬНОСТЬ_ЗВОНКА}}{\text{КОЛИЧЕСТВО_ЗВОНКОВ}} * 100\%$ по направлению.

Данные параметры помогают оценить качество связи по направлению и особенно интересны в отчёте операторского договора. Для большего удобства ячейки имеют цветовую индикацию. Цвета столбцов с ASR и ACD задаются в конфигурации модуля следующим образом:

```
color.asr=0-30:dd0000;30-100:ffffff
color.acd=0-60:dd0000
```

Здесь 0-30, 30-100, 0-60 - диапазоны параметров, через двоеточие после них указывается цвет ячейки.

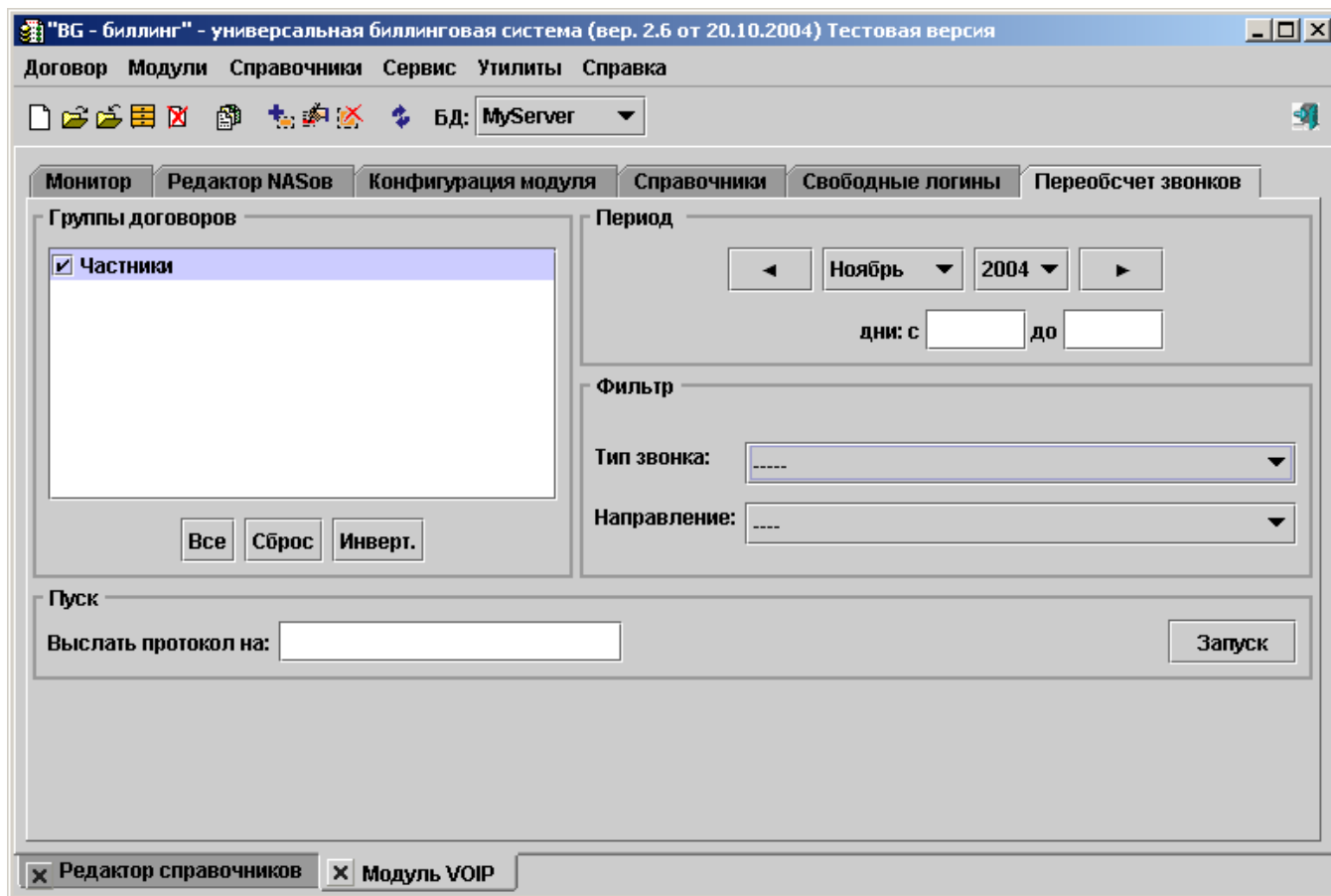
Также имеется возможность выгрузки отчётов в CSV и HTML-формате. Для это необходимо нажать на кнопку **Сохранить** (кнопка с дискетой) и указать интересующий формат файла.

Web-интерфейс VoiceIP

Предоставляет доступ к тем же отчётам, что и в АРМ администратора клиенту через страницу статистики. Можно изменять название пунктов меню редактируя параметры **web.menuItem1-web.menuItem3** в конфигурации модуля.

Переобсчёт сессий

Данная возможность используется для пересчёта сессий задним числом. Например, при неверно указанных ценах. Для пересчёта сессий зайдите на вкладку **Переобсчёт звонков**, затем выберите группы договоров, чьи сессии нужно переобсчитать, типы звонков и т.п.



Отчет о переобсчёте будет выслан на указанный E-mail. Не забудьте проверить опции почты в настройке сервера биллинга.

Создание договоров абонентом через Web

Модуль поддерживает создание договоров самим пользователем через специальную страницу.

URL страницы активации:

http://<адрес сервера биллинга>:8080/bgbilling/pubexecuter?action=CreateContract&module=voiceip&mid=<код модуля>

Шаблон страницы - файл **voice_create_contract.xml** в нем необходимо подправить параметр формы mid на правильный код модуля.

Дополнительно необходимо указать в конфигурации модуля:

```
#
web.register.contract.pattern=
#
web.register.fio.param=
# e-mail
web.register.email.param=
# " "
web.register.refbook.param=
# , 200000-250000;300;200-201
web.register.alias.range=
#
web.register.login.type=
```

Модуль Voice

Назначение модуля

Модуль предназначен для

- обчёта телефонных соединений обычной телефонии и IP-телефонии на основании текстовых логов CDR, позволяет вести учёт в соответствии с последними требованиями законодательства РФ.
- обчёта стоимости звонков через IP-телефонию с поддержкой оборудования, которое шлёт запросы [RADIUS](#). Используется свой RADIUS-сервер.
- Управлении доступом на оборудовании (отключение и подключение абонента).

В модуле реализован учёт и обчёт услуг, предоставляемых клиентам и учёт операторских взаиморасчётов.

Базовые понятия и алгоритм работы модуля Voice

Обработка логов CDR

Структурную схему обработки логов CDR в биллинге можно отобразить так:

Тут **BGVoiceProcessor** это отдельное приложение, которое занимается обработкой логов.

Работа по протоколу Radius

Для работы по протоколу Radius структура взаимодействия основных частей модуля Voice изображена на рисунке:

NAS (Network Access Server) - сервер, через который происходит звонок клиента. В роли NASа могут выступать специализированные компьютеры типа Cisco или любое другое оборудование поддерживающее Radius.

Обрабатываются следующие виды Radius-запросов:

- авторизационный(access),
- accounting(start/stop/update)

[Базовые сведения о протоколе RADIUS и конфигурировании атрибутов.](#)

Протокол RADIUS основан на UDP, представляет из себя пакет определённого типа с набором атрибутов. Рассмотрим ход типового соединения по логу RADIUS запросов (radius.log).

```
Time: 12.09.2017 12:18:30
Packet type: Accounting-Request
Identifier: 198
Authenticator: {29 3C ED 92 11 62 3F B4 45 60 B5 83 EC 3B 8F FD}
Attributes:
  NAS-IP-Address=192.168.103.98
  NAS-Port=0
  Acct-Status-Type=1
  Acct-Delay-Time=0
  Acct-Session-Id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  Called-Station-Id=44179312622020
  Called-Station-Id=44179312622020
  Login-Service=0
  Calling-Station-Id=9612170985
  cisco-avpair=h323-call-id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  cisco-avpair=src-gw-ip=192.168.103.100
  cisco-avpair=src-gw-name=9612170985
  cisco-avpair=src-number-in=9612170985
  cisco-avpair=src-number-out=9612170985
  cisco-avpair=dst-gw-ip=192.168.103.98
  cisco-avpair=dst-gw-name=44179312622020
  cisco-avpair=dst-number-in=44179312622020
  cisco-avpair=dst-number-out=44179312622020
  h323-conf-id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  h323-conf-id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  h323-setup-time=2017-09-12T12:18:26.100131+0300
  h323-call-origin=originate
```

```
Time: 12.09.2017 12:18:30
Packet type: Access-Request
Identifier: 201
Authenticator: {54 F5 0C 3D 7F FF D8 B3 0A EF 3C 91 5B C4 C9 D1}
Attributes:
  User-Name=79612170985
  User-Password=3otg ZzT
  NAS-IP-Address=192.168.103.98
  NAS-Port=0
  NAS-Port-Id=ISDN 3/0:D:14
  Acct-Session-Id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  NAS-Port-Type=44
  Called-Station-Id=79312622020
  Calling-Station-Id=79612170985
  h323-conf-id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  cisco-avpair=h323-call-type=outgoing
  cisco-avpair=h323-ivr-out=transactionID:1234-42232
  h323-gw-id=FREESW
  h323-prompt-id=79312622020
  h323-call-origin=outgoing
```

```
Time: 12.09.2017 12:18:30
```

```
Packet type: Access-Accept
Identifier: 201
Authenticator: {E8 6C BD C2 6D CF ED 8D 49 DD EC 45 93 83 F4 EF}
Attributes:
  Callback-Number=192.168.103.100
  h323-credit-amount=499324.60
  h323-credit-time=8609044
  h323-return-code=0
Process time auth: 0
Time: 12.09.2017 12:24:39
```

```
Packet type: Accounting-Request
Identifier: 228
Authenticator: {76 4C 55 00 2B 5B 3F D3 75 3C 6A CE 46 20 E0 D0}
Attributes:
  NAS-IP-Address=192.168.103.98
  NAS-Port=0
  Acct-Status-Type=2
  Acct-Delay-Time=0
  Acct-Session-Id=e6889347-45b8-4094-a74a-014cddb35ff2
  Called-Station-Id=44479312622020
  Acct-Session-Time=366
  Login-Service=0
  Calling-Station-Id=79612170985
  cisco-avpair=h323-call-id=e6889347-45b8-4094-a74a-014cddb35ff2
  cisco-avpair=src-gw-ip=192.168.103.100
  cisco-avpair=src-gw-name=79612170985
  cisco-avpair=src-number-in=79612170985
  cisco-avpair=src-number-out=79612170985
  cisco-avpair=dst-gw-ip=192.168.103.100
  cisco-avpair=dst-gw-name=44479312622020
  cisco-avpair=dst-number-in=44479312622020
  cisco-avpair=pdd-time=0
  cisco-avpair=dst-number-out=44479312622020
  h323-conf-id=a18a094e-630d-436b-ab3d-82ad11f8fa6e
  h323-setup-time=2017-09-12T12:18:26.180100+0300
  h323-call-origin=originate
  h323-connect-time=2017-09-12T12:18:28.420120+0300
  h323-disconnect-time=2017-09-12T12:24:34.560111+0300
  h323-disconnect-cause=10
```

Каждый пакет содержит информацию о NAsе (**NAS-Identifier** и/или **NAS-IP-Address**), на основании которой RADIUS-сервер сопоставляет пришедший пакет NASу в модуле. При сопоставлении сначала производится поиск NASа с идентификатором, идентичным атрибуту **NAS-Identifier** пакета, если результат отрицательный, идёт поиск NASа с IP-адресом, равным **NAS-IP-Address**. Если пришедшему пакету NAS не сопоставлен в **radius.log**, выводится ошибка **NAS not found for Packet!!!**.

Обмен сообщениями с каждым NASом шифруется определённым кодовым словом - секретом. Секрет должен совпадать для NASа в биллинге и для конфигурации самого NASа. При несовпадении секретов проверка пароля будет все время выдавать неверный результат, т.к. секрет используется в шифровании пароля.

Далее описана логика работы модуля. Большинство параметров описанных ниже (**h323-conf-id**, **h323-return-code**, **h323-credit-time**, **h323-credit-amount**, **Calling-Station-Id**, **Called-Station-Id**) могут быть заменены в конфигурации NAS на другие. По умолчанию используются те, которые тут описаны.

Идентификатором соединения в пределах NASа для RADIUS-сервера выступает атрибут **Acct-Session-Id**. Обратите внимание, что он идентичен для всех пакетов в пределах сессии. NAS должен контролировать, чтобы в один момент времени одинаковый **Acct-Session-Id** не проставлялся в RADIUS-пакетах, относящихся к разным сессиям.

Для тех, кто переходит с модуля VoiceIP. Значения всех параметров по умолчанию остались такие же как в VoiceIP. Кроме одного: **h323-conf-id** заменен на **Acct-Session-Id** по умолчанию. Это можно исправить в конфигурации NAS-а.

Далее рассмотрим по пакетно обмен данными между NASом и RADIUS-сервером по ходу соединения.

1. AUTHENTICATION_REQUEST

Запрос авторизации отправляется NASom RADIUS-серверу и содержит помимо идентификационной информации соединения, указанной выше, информацию, идентифицирующую пользователя. Возможна идентификация пользователя по номеру звонящего (**Calling-Station-Id**), логину и паролю (запрашивает IVR) или любому другому атрибуту, режим поиска настраивается на NAS.

При наличии пароль шифруется по алгоритму PAP с использованием секрета. При некорректном секрете пароль в PAP-режиме не расшифровывается и отображается в **radius.log** и в мониторе ошибок не в том виде, в котором был введён пользователем.

Передаётся последовательно два запроса авторизации. Первая авторизация передаётся до набора номера и содержит только идентифицирующую пользователя информацию.

Вторая авторизация высылается в случае успеха первой и содержит набранный номер, атрибут **Called-Station-Id**.

2. AUTHENTICATION_REJECT

Отказ в авторизации, код ошибки авторизации передаётся дополнительно в атрибуте **h323-return-code**. Детальную причину отказа в авторизации пользователя можно посмотреть в **Мониторе модуля** в режиме ошибок. Код ошибки авторизации указанный в квадратных скобках в мониторе совпадает со значением, передаваемым в атрибуте. Значение атрибута может быть использовано IVR системой для озвучивания причины ошибки.

3. AUTHENTICATION_ACCEPT

Пользователь авторизован.

Для первой авторизации это означает что он идентифицирован, пароль введён верно (если есть), остаток на балансе положителен, тариф установлен и т.п. В атрибутах передаётся **h323-return-code=0** (нет ошибок) и **h323-credit-amount=<остатку счета>**.

При второй авторизации это означает, что верны все условия первой авторизации, а кроме того на набранный номер есть цена в тарифе. Дополнительно к атрибутам первой авторизации передаются атрибут **h323-credit-time=<на сколько секунд разговора хватает баланса пользователя>**. Значение данного атрибута ограничивается сверху значением переменной **voip.max.time** конфигурации модуля.

Дополнительно в **AUTHENTICATION_ACCEPT** могут передаваться атрибуты, указанные в конфигурации типа логина (см. далее).

4. ACCOUNTING_REQUEST START

Атрибут **Acct-Status-Type** для них равен 2. Данный тип запросов передаёт на RADIUS-сервер информацию о старте соединения. Radius создает сессию, которая становится видна в **Мониторе модуля**.

5. ACCOUNTING_REQUEST UPDATE

Атрибут **Acct-Status-Type** для них равен 3. Данный тип запросов передаёт на RADIUS-сервер информацию о текущем статусе соединения. Radius обчитываем сессию, обновляет в ней время соединения и стоимость. Время соединения берется из атрибута **Acct-Session-Time**.

6. ACCOUNTING_REQUEST STOP

Атрибут **Acct-Status-Type** для них равен 2. Данный тип запросов передаёт на RADIUS-сервер информацию о завершении соединения. Время соединения берется из атрибута **Acct-Session-Time**.

7. ACCOUNTING_RESPONSE

Ответ RADIUS-сервера о том, что он получил запрос аккаунтинга(**START/UPDATE/STOP**) . Ответ не содержит никаких атрибутов.

Возможны схемы работы с учётом только **accounting**-пакетов(или даже только **Stop** пакетов), без авторизации. В этом случае RADIUS-сервер не может запретить звонок при отсутствии цены в тарифе.

Настройка модуля

[Установите](#) модуль на сервер, создайте экземпляр. Определите в **Редакторе модулей и услуг** услуги, обсчитываемые этим модулем.

В конфигурации модуля можно указать следующие общие настройки (для любой схему начисления):

Конфигурация

```
#
contract.status.active.codes=0
# - id ( ).
radius.agent.sub.contract.order=
```

Настройка обсчета по логам CDR.

Специфичные настройки при обсчете логов CDR :

Конфигурация

```
# ,  
cdr.audit.resource.categories=15
```

Настройка обчета по протоколу Radius.

Специфичные настройки при обчете по протоколу Radius:

Конфигурация

```
# (      NAS).  
radius.search.mode.pattern.1.rule=Calling-Station-id=PHONE  
radius.search.mode.pattern.2.rule=Called-Station-id=PHONE
```

Общий алгоритм первичной настройки модуля Voice.

Настройка модуля voice немного различается для CDR и Radius схем. И это будет отражено ниже.

Общий алгоритм первичной настройки модуля Voice для обсчета логов CDR.

Настройка клиентского обсчета по логам CDR.

1. Настраиваем [абонентский тип аккаунта](#). Тут в зависимости от схемы мы выбираем галочками те поля, которые будут видны в нем. Для клиентского обсчета в большинстве случаев нам надо выбрать галочку "номер" и устройство чтобы выбрать АТС, к которой привязан номер.
2. Настраиваем [Ресурсы номеров](#). Из них вы будете назначать номера на договор.
3. Добавляем [тип устройства](#). Это фактически тип вашей АТС.
Тут мы заполняем название. Надо поставить галочку **источник данных CDR**. В большинстве случаев надо выбрать [Обработчик логов CDR](#). Обработчик логов можно выбрать один из стандартной поставки или сделан новый под ваш тип АТС.
4. Добавляем [устройство](#) с типом, который добавлен в предыдущем шаге.
В большинстве случаев тут надо указать только название и путь к файлам логов. Путь - это где лежат логи cdr для этого устройства. Их будет обрабатывать обработчик логов CDR, указанный в типе данного устройства.
5. Добавляем [клиентские аккаунт](#) на договор с типом, который мы завели ранее. Тут нам надо будет выбрать устройство, номер телефона. Вид формы редактора аккаунта и наличие полей в ней может меняться в зависимости от настроек типа аккаунта.
6. Добавляем [Тариф для модуля Voice](#). Вид тарифа может различаться, но вначале для отладки рекомендуется начать с самого простого минимального тарифа, например добавить тарификацию по одному префиксу пока. Пример можно взять отсюда - [Тарифы на местную связь](#).
Потом уже когда все заработает, то усложнять тариф, вводить карты цен и т.п.
7. Добавить [тариф на договор](#).

Настройка операторского обсчета по логам CDR.

1. Настраиваем [операторский тип аккаунта](#). Тут в зависимости от схемы мы выбираем галочками те поля, которые будут видны в нем. Обычно это Regexr номера или regexr портов.
2. Добавляем [тип устройства](#). Это фактически тип вашей АТС.
Тут мы заполняем название. Надо поставить галочку **источник данных CDR**. В большинстве случаев надо выбрать [Обработчик логов CDR](#). Обработчик логов можно выбрать один из стандартной поставки или сделан новый под ваш тип АТС. Если у вас уже есть тип устройства(вы его например добавляли для клиентского обсчета, то этот пункт можно пропустить).
3. Добавляем [устройство](#) с типом, который добавлен в предыдущем шаге. Если у вас уже есть устройство(вы его например добавляли для клиентского обсчета, то этот пункт можно пропустить).
В большинстве случаев тут надо указать только название и путь к файлам логов. Путь - это где лежат логи CDR для этого устройства. Их будет обрабатывать обработчик логов CDR, указанный в типе данного устройства.
4. Добавляем [операторский аккаунт](#) на договор с типом, который мы завели ранее. Тут нам надо будет выбрать устройство, номер телефона. Вид формы редактора аккаунта и наличие полей в ней может меняться в зависимости от настроек типа аккаунта.
5. Добавляем [Тариф для модуля Voice](#). Вид тарифа может различаться, но вначале для отладки рекомендуется начать с самого простого минимального тарифа, например добавить тарификацию по одному префиксу пока. Пример можно взять отсюда - [Тарифы на местную связь](#).
Потом уже когда все заработает, то усложнять тариф, вводить карты цен и т.п.
6. Добавить [тариф на договор](#).
7. Визуальный пример настройки можно посмотреть тут - [Учет операторского трафика для CDR](#).

Последующие общие действия.

После обработки логов вы можете

1. Настроить и запустить приложение [BGVoiceProcessor](#).
2. Произвести [Обработка логов CDR в модуле Voice](#).
3. Изучить ошибки на [вкладке Ошибки](#).
Сделать какие-то правки и повторить процесс заново и снова изучить ошибки.
4. Проверить результат нормально обработанных записей в [отчетах модуля Voice](#) на договоре.
5. Если там сессий нет, то снова возвращаться в ошибки и изучать.

Автоматическая настройка загрузки логов

Если все отлажено и ручная загрузка логов работает, то можно приступить к автоматической загрузке логов.

Есть 2 варианта автоматической загрузки

1) Загрузка предыдущего дня в начале суток следующего дня с помощью задачи планировщика **Генерация заданий на обработку логов**. Она описана [тут](#).

В этом случае логи загружаются на машину биллинга каким-то внешним скриптом или [глобальным скриптом](#) биллинга [работающий в планировщике](#). А задача **Генерация заданий на обработку логов** в начале каждого суток запускает обработку логов предыдущих суток. Можно делать и каждый час обработку логов, но каждый раз будут перерабатываться логи за весь день.

2) Загрузка логов с помощью [Runtime обработчик логов CDR](#).

В этом случае файлы со звонками обрабатываются по мере появления в папке и перемещаются в другую папку.

Общий алгоритм первичной настройки модуля Voice для обсчета по Radius.

Абонентский обсчет

1. Настраиваем [абонентский тип аккаунта](#). Тут в зависимости от схемы мы выбираем галочками те поля, которые будут видны в нем. Для клиентского обсчета в большинстве случаев нам надо выбрать галочку устройство чтобы выбирать NAS, к которой привязан номер. И так же выбрать либо номер либо логин в зависимости от типа поиска который у нас будет настроен на NAS(его мы настроим ниже).
2. Если у нас номер телефона нужен на абоненте(а не только логин например) и мы его отметили на предыдущем шаге, то Настраиваем [P есурсы номеров](#). Из них вы будете назначать номера на договор.
3. Добавляем [тип устройства](#). Это фактически тип вашей NAS, через которую будут идти звонки . Нужно поставить галочку **Radius NAS**. Тут мы заполняем название. При первой настройке пока можно попробовать не задавать никаких обработчиков. Radius протокол работает без каких-либо обработчиков. Но обработчик протокола Radius может понадобиться если в процессе внедрения выяснится, что нужно дополнительно поменять какие-то атрибуты Radius в запросе или еще что-то. Возможно нужно будет заполнить конфигурацию типа устройства, более подробно описано тут - [Настройка NAS-ов для Radius-запросов](#). Вначале можно начать с пустой конфигурации и после получения первого Radus запроса и сообщений в логах понять что нужно заполнить и поправить.
4. Добавляем [устройство](#) с типом, который добавлен в предыдущем шаге. Это и есть наш NAS, с которого будет приходить Radius-запросы. Тут мы заполняем либо название либо ip в зависимости от настроек поиска устройства(по **NAS-IP-Address** или **NAS-Identifier**). Более подробно о настройке можно прочитать тут - [Настройка NAS-ов для Radius-запросов](#).
5. Добавляем [клиентские аккаунт](#) на договор с типом, который мы завели ранее. Тут нам надо будет выбрать номер телефона или логин. Вид формы редактора аккаунта и наличие полей в ней может меняться в зависимости от настроек типа аккаунта. Для первой настройке можно явно выбирать устройство(NAS), как настроить более сложные случаи (например абонент может выйти с любого NAS или привязан к устройству, которое находится под NAS) - описано в документации - [Настройка NAS-ов для Radius-запросов](#).
6. Добавляем [Тариф для модуля Voice](#). Вид тарифа может различаться, но вначале для отладки рекомендуется начать с самого простого минимального тарифа, например добавить тарификацию по одному префиксу пока. Пример можно взять отсюда - [Тарифы на местную связь](#). Потом уже когда все заработает, то усложнять тариф, вводить карты цен и т.п.
7. Добавить [тариф на договор](#).

Операторский обсчет

1. Настраиваем [абонентский тип аккаунта](#). Для операторского обсчета по Radius используется абонетский тип аккаунта. Тут в зависимости от схемы мы выбираем галочками те поля, которые будут видны в нем. Для клиентского обсчета в большинстве случаев нам надо выбрать галочку устройство чтобы выбирать NAS, к которой привязан номер. Остальные данные зависит от схемы поиска, который у нас будет настроен на NAS(его мы настроим ниже).
2. Добавляем [тип устройства](#) если его еще нет(например вы могли его уже настроить для абонентов). Это фактически тип вашей NAS, через которую будут идти звонки . Нужно поставить галочку **Radius NAS**. Тут мы заполняем название. При первой настройке пока можно попробовать не задавать никаких обработчиков. Radius протокол работает без каких-либо обработчиков. Но обработчик протокола Radius может понадобиться если в процессе внедрения выяснится, что нужно дополнительно поменять какие-то атрибуты Radius в запросе или еще что-то. Возможно нужно будет заполнить конфигурацию типа устройства, более подробно описано тут - [Настройка NAS-ов для Radius-запросов](#). Вначале можно начать с пустой конфигурации и после получения первого Radus запроса и сообщений в логах понять что нужно заполнить и поправить.
3. Добавляем [устройство](#) с типом, который добавлен в предыдущем шаге. Это и есть наш NAS, с которого будет приходить Radius-запросы. Тут мы заполняем либо название либо ip в зависимости от настроек поиска устройства(по **NAS-IP-Address** или **NAS-Identifier**). Более подробно о настройке можно прочитать тут - [Настройка NAS-ов для Radius-запросов](#).
4. Добавляем [клиентские аккаунт](#) на договор с типом, который мы завели ранее. Вид формы редактора аккаунта и наличие полей в ней может меняться в зависимости от настроек типа аккаунта. Для первой настройке можно явно выбирать устройство(NAS), как настроить более сложные случаи (например оператор может выйти с любого NAS или привязан к устройству, которое находится под NAS) - описано в документации - [Настройка NAS-ов для Radius-запросов](#).
5. Добавляем [Тариф для модуля Voice](#). Вид тарифа может различаться, но вначале для отладки рекомендуется начать с самого простого минимального тарифа, например добавить тарификацию по одному префиксу пока. Пример можно взять отсюда - [Тарифы на местную связь](#). Потом уже когда все заработает, то усложнять тариф, вводить карты цен и т.п.
6. Добавить [тариф на договор](#).
7. Если у нас оператор ищется по логину, то он находится и обсчитывается как обычный абонент. Или же можно сделать поиск оператора скриптами или создавать сессию для оператора при обычном Radius запросе абонента, это как описано в этой статье - [Учет операторского трафика для Radius](#).

Последующие общие действия.

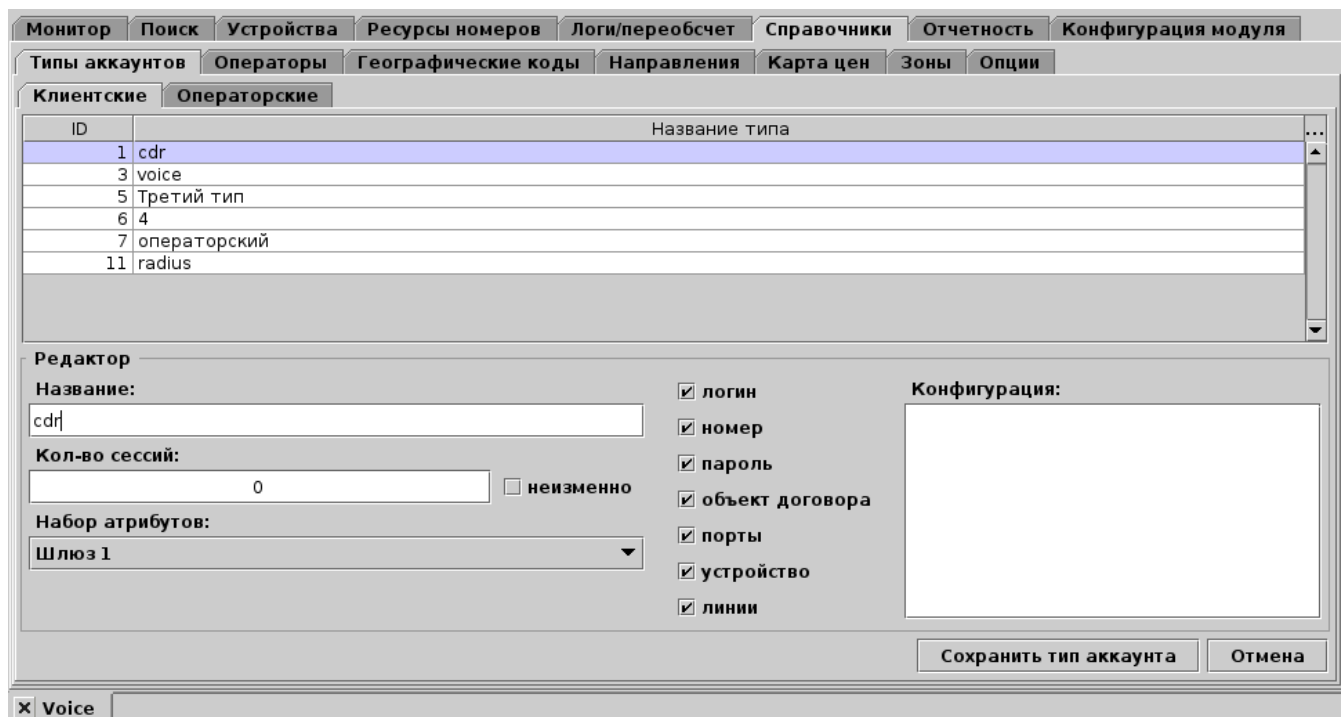
1. Настроить и запустить приложение [BGVoiceRadius](#).
2. Настроить так чтобы Radius-запросы приходили на BGVoiceRadius.
3. Получить ошибки в логах BGVoiceRadius или на вкладке ошибок в [Монитор соединений](#).
4. Проверить сессии в текущих сессиях и в завершенных в [Монитор соединений](#).
5. Проверить сессии в [отчетах модуля Voice](#) на договоре.

Сущности модуля Voice

Типы аккаунтов

Клиентские типы аккаунтов

Типы аккаунтов представляют из себя некоторый набор параметров, которые можно указать в аккаунте абонента.



Кол-во сессий - максимальное одновременно возможное количество сессий по одному аккаунту договора данного типа. Параметр позволяет выставить значение по умолчанию, которое будет выставлено у всех аккаунтов данного типа. Галочка **неизменно** - означает то, что его нельзя изменить в аккаунте.

Флаги в нижней области определяют какие параметры должны быть указаны и будут доступны в аккаунте договора.

- **Логин**- задавать логин в аккаунте.
- **Номер**- задавать номер телефона в аккаунте;
- **Пароль** - задавать пароль в аккаунте.
- **Порты** - задавать порты в аккаунте.
- **Устройство** - задавать устройство в аккаунте.

В конфигурации типа аккаунта могут быть определены переменные:

1. **title.pattern**=(**<шаблон имени сервиса>**), в шаблоне имени возможны переменные:

- **\${login}** - логин;
- **\${number}** - номер телефона.

По умолчанию там используется **\${login}**.

2. **const.device.id**=**<постоянный код устройства для всех аккаунтов данного типа>**.

Операторские типы аккаунтов

Операторские типы аккаунтов представляют из себя некоторый набор параметров, которые можно указать в аккаунте оператора. В данный момент они используются только при обчете по логам CDR.

Монитор Поиск Устройства Ресурсы номеров Логи/переобсчет Типы аккаунтов Справочники Конфигурация модуля

Клиентские Операторские

ID	Название типа
3	звонки по номеру В
8	Звонки по порту

Редактор

Название: Звонки по порту

Конфигурация

с портов с портов Regexp

На порты На порты Regexp

Regexp A

Not Regexp A

Regexp B

Not Regexp B

Сохранить тип аккаунта Отмена

Поиск договоров Voice AG003

Аккаунты

Клиентские аккаунты

Для предоставления абоненту совершать звонки необходимо добавить в договор аккаунт сконфигурированного заранее типа. Все аккаунты договора отображаются в сводной таблице.

The screenshot shows the BGBillingClient interface. The main window displays a table of accounts for contract TSN-2017-100. The table has the following columns: Аккаунт, Устройство, Тип, Номер, Период, Статус, Состояние, Комментарий, and id. Two accounts are listed:

Аккаунт	Устройство	Тип	Номер	Период	Статус	Состояние	Комментарий	id
7381269016	First: ID3 [localhost:6080] cdr			7381269016 08.09.2017...	открыт	отключен		5760
7381269020	Cisco: radius_test [81.3... voice			7381269020 08.09.2017...	открыт	отключен		5761

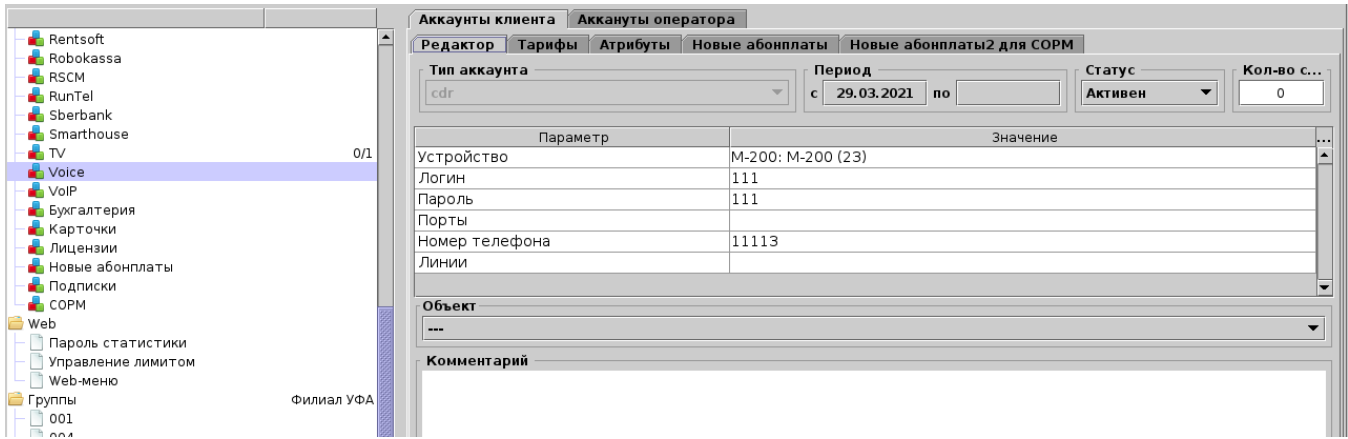
Столбцы таблицы:

- **Аккаунт** - сгенерированное на основании шаблона из типа сервиса название аккаунта договора;
- **Устройство** - устройство, к которому привязан аккаунт;
- **Тип** - тип аккаунта;
- **Период** - период действия;
- **Статус** - текущий статус сервиса;
- **Состояние** - реальное состояние сервиса на устройстве.
- **Комментарий** - комментарий.
- **id** - идентификатор аккаунта в БД.

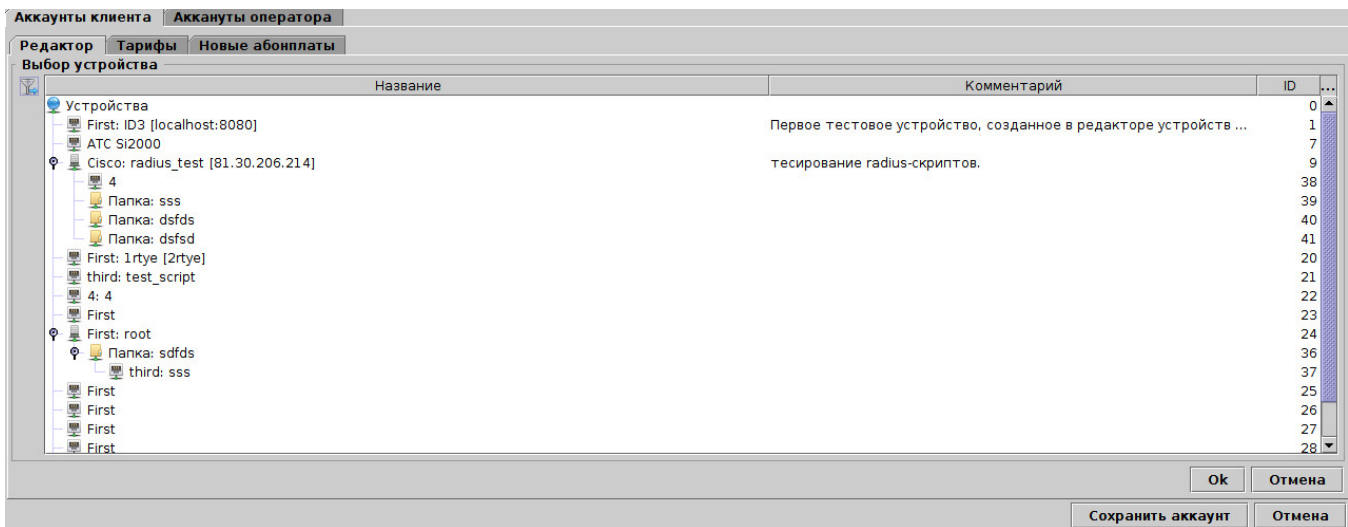
Статус аккаунта может принимать следующие значения: **открыт** и **заблокирован**. Статус аккаунта переключается вручную.

Состояние аккаунта может быть "подключён", либо "отключен", отображает реальное состояние, которое было установлено на устройстве. Между состояниями "подключен" и "отключен" модуль осуществляет автоматическое переключение в зависимости от состояния баланса (больше, либо меньше лимита), статуса договора, статуса аккаунта. Аккаунт подключен, когда в договоре установлен активный для него статус, сервис в статусе открыт и баланс больше лимита.

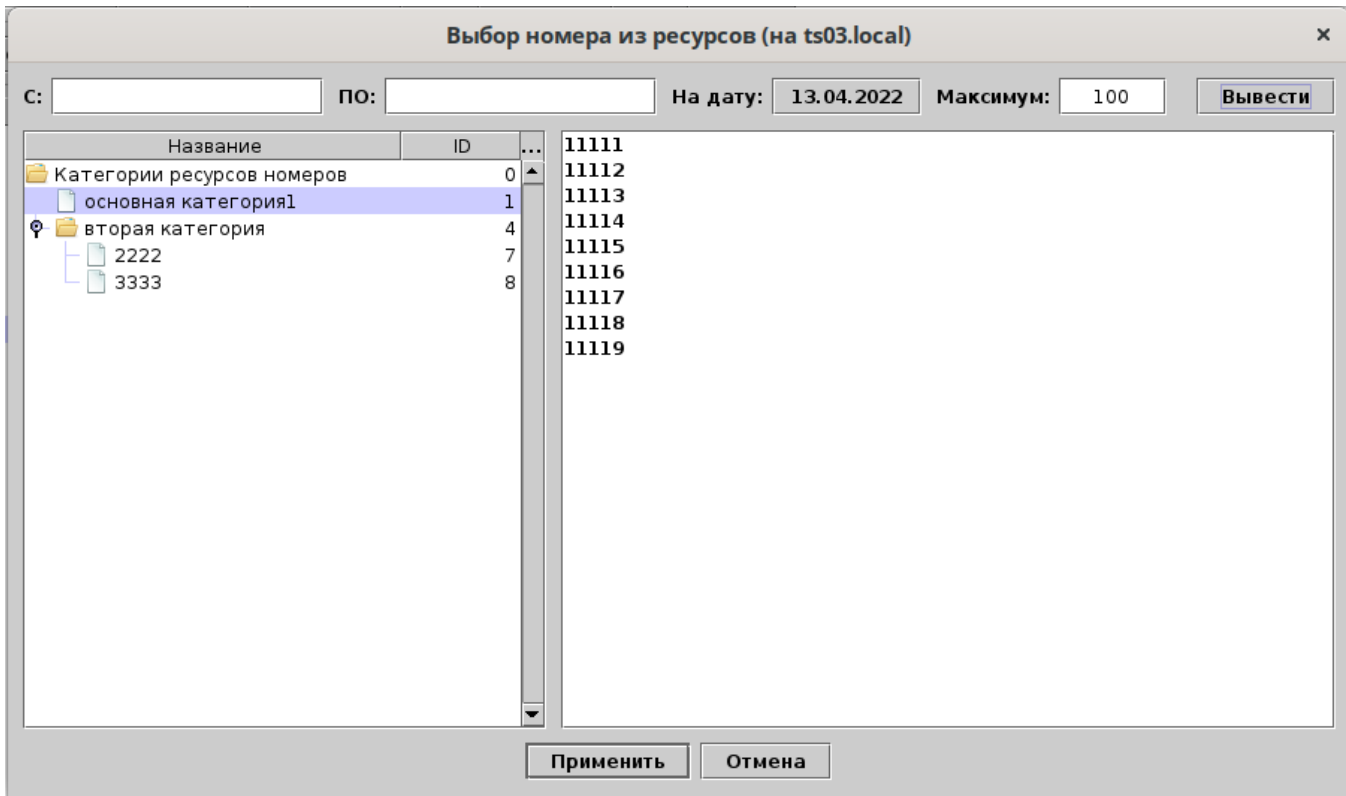
Редактор аккаунта:



В зависимости от настроек типа (ссылка!) в редакторе аккаунта могут присутствовать различные поля. Обязателен тип, период, статус, количество сессий, устройство. Устройство может быть указано постоянным для всех аккаунтов одного типа с помощью переменной в конфигурации типа сервиса. Вызов редактора соответствующих параметров происходит по двойному клику по строке с параметром.



Номер телефона при сохранении проверяется на вхождение в ресурс(ссылка!) .Для того, чтобы выбрать свободный номер из ресурсов, нужно нажать на кнопку "<<<" возле номера.



Тут можно выбрать категорию ресурса. И отфильтровать результаты можно по диапазону номеров (с и по) и дате. Можно задать максимальное количество выводимых номеров.

Так же могут быть доступны поля логин и пароль (либо оба сразу либо по отдельности).

Логин	111
Пароль	111

И поле - порты.

Порты	
-------	--

Порты задаются через запятую.

Операторские аккаунты

Аккаунты оператора используется только при обработке логов CDR. Для протокола Radius используются обычные аккаунты для операторского об~~с~~чета.

Для предоставления возможности оператору совершать звонки необходимо добавить в договор аккаунт сконфигурированного заранее типа. Все операторские аккаунты договора отображаются в сводной таблице.

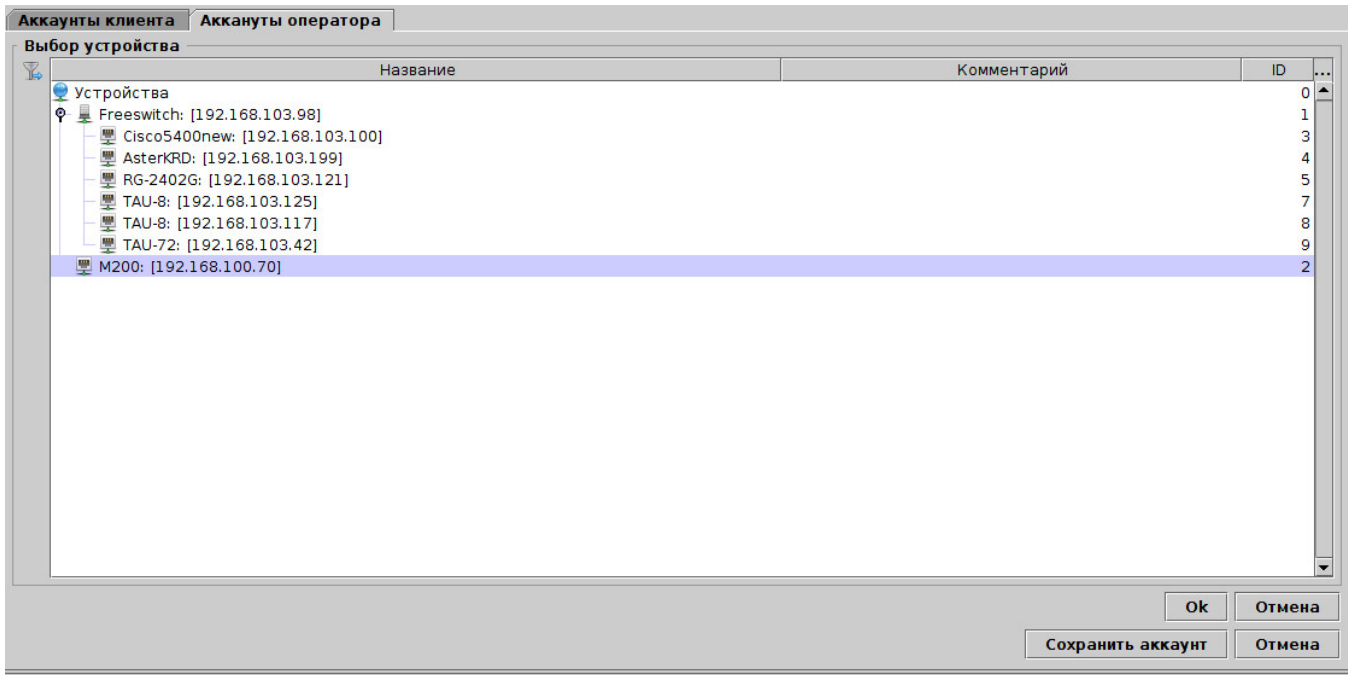
Аккаунты клиента	Аккануы оператора		
id	Имя	Устройство	Период
19064	Исходящие С 5-6-й порт на МР24	M200: [192.168.100.70]	01.07.2017-...
19067	Входящие С 5-6-й порт на МР24	M200: [192.168.100.70]	01.07.2017-...

Редактор операторского аккаунта:

Аккаунты клиента	Аккануы оператора		
Редактор операторского аккаунта			
Тип аккаунта Звонки по порту	Наименование Входящие С 5-6-й порт на МР24	Тип звонка Входящие	Период с 01.07.2017 по
Устройство M200: [192.168.100.70] <<<			
С портов	На порты		
С портов Regexp ^C12700[5,6]...	На порты Regexp		
Regexp A	No Regexp A		
Regexp B	No Regexp B		
Сохранить аккаунт Отмена			

В зависимости от настроек типа (ссылка!) в редакторе аккаунта могут присутствовать различные поля. Обязателен тип аккаунта, наименование, тип звонка, период, устройство.

Для того чтобы выбрать устройство нужно нажать на кнопку "<<<" возле него.



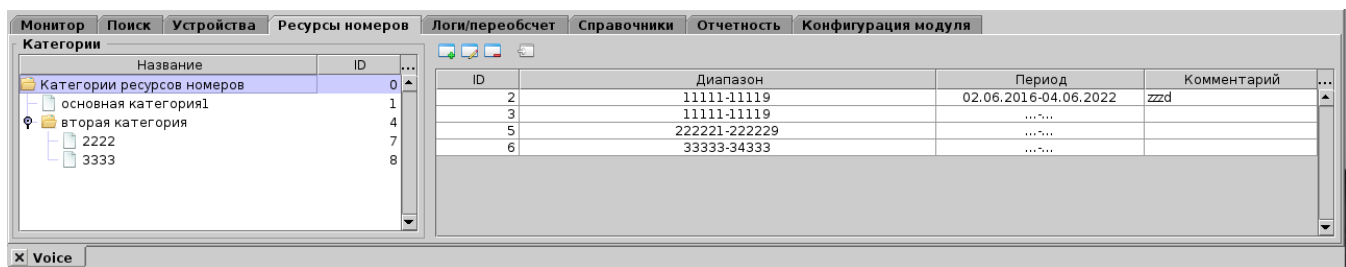
Остальные поля :

- С портов
- На порты
- С портов Regexp
- На порты Regexp
- Regexp A
- No Regexp A
- Regexp B
- No regexp B

Их наличие зависит от настройки типа операторского аккаунта.

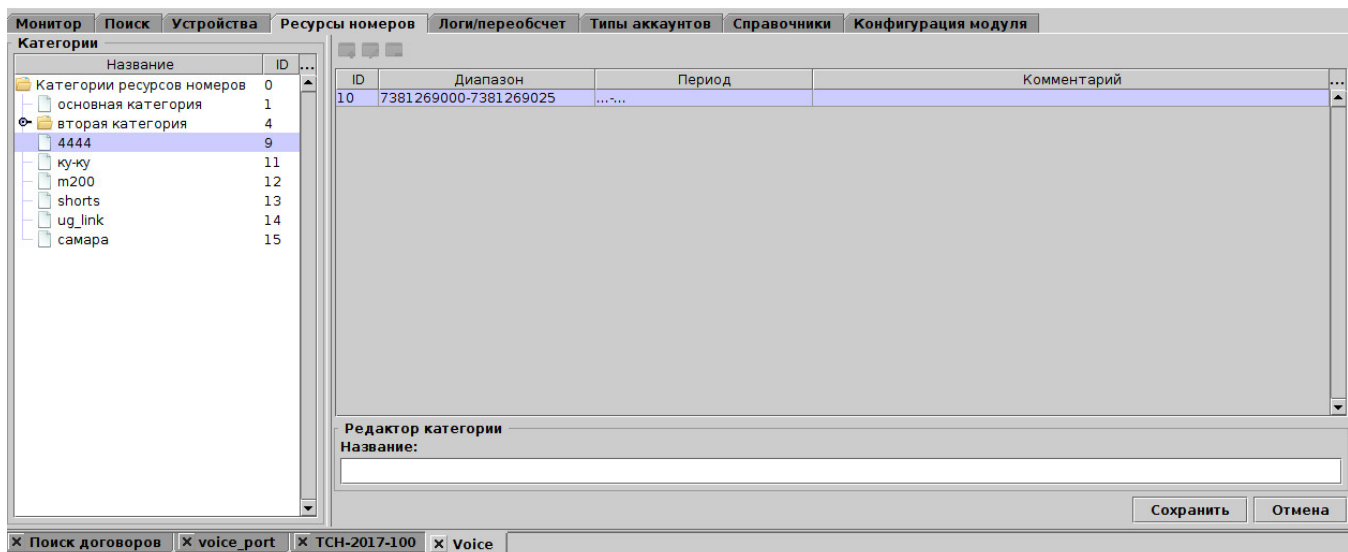
Ресурсы номеров

Ресурсы номеров настраиваются в одноименной вкладке модуля.

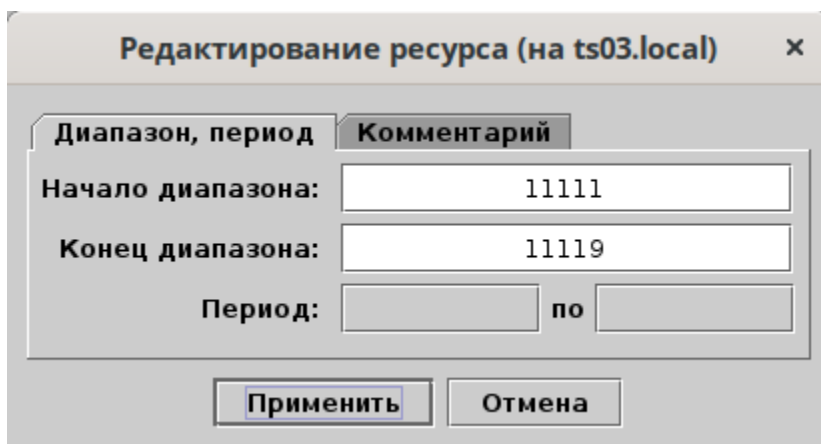


Слева выбирается категория. Справа задаются диапазоны номеров.

Категории добавляются и редактируются с помощью стандартной панели инструментов сверху.



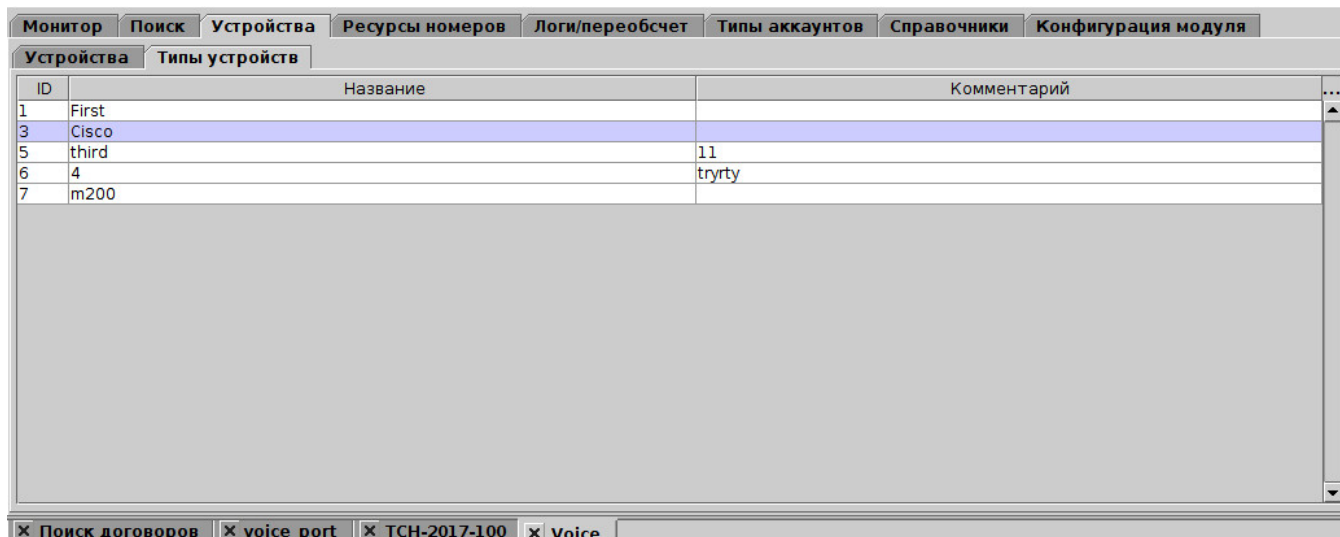
Диапазоны номеров добавляются с помощью дополнительной панели инструментов справа (на скриншоте она не активна, так как у нас уже открыт редактор).



Типы устройств модуля Voice

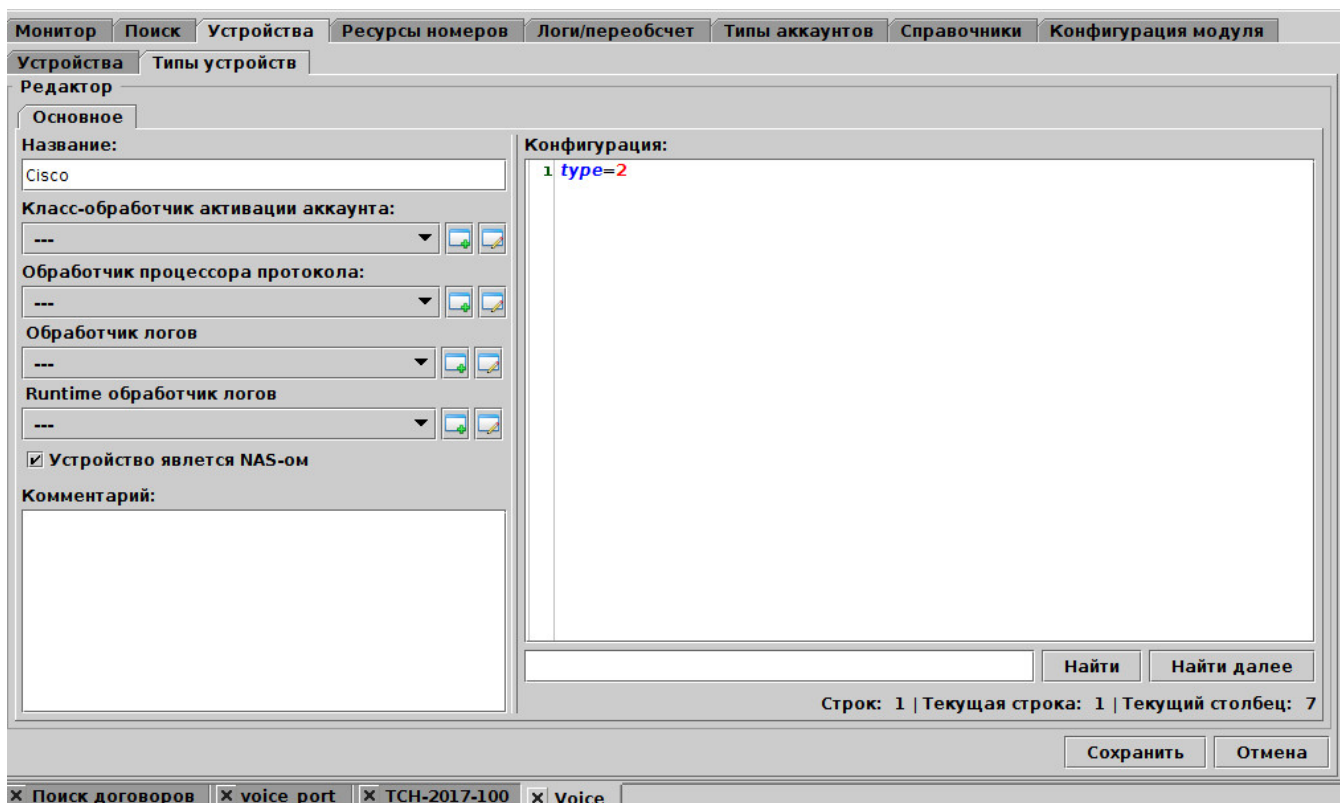
Тип устройства позволяет задавать общие параметры для группы одинаковых устройств.

Типы устройств редактируются на вкладке **Устройства->Типы устройств**.



ID	Название	Комментарий
1	First	
3	Cisco	
5	third	11
6	4	tryrty
7	m200	

Редактор типа выглядит так:



Редактор

Основное

Название: Cisco

Класс-обработчик активации аккаунта: ---

Обработчик процессора протокола: ---

Обработчик логов: ---

Runtime обработчик логов: ---

Устройство является NAS-ом

Комментарий:

Конфигурация:

```
1 type=2
```

Строк: 1 | Текущая строка: 1 | Текущий столбец: 7

Сохранить Отмена

Тут в можно задать следующие параметры:

- **Название** - название типа устройства.
- **Класс** - обработчик активации аккаунта.
- **Обработчик процессора протокола** - динамический класс, реализующий определённый интерфейс по управлению устройством для блокировки/разблокировки аккаунта, изменения его параметров;
- **Обработчик логов** - динамический класс, реализующий определённый интерфейс по обработке логов CDR.
- **Runtime обработчик логов** - динамический класс, реализующий определённый интерфейс по обработке логов CDR в runtime-режиме.
- **Устройство является NAS-ом** - указывается для типов устройств, которые используются как для voip-телефонии

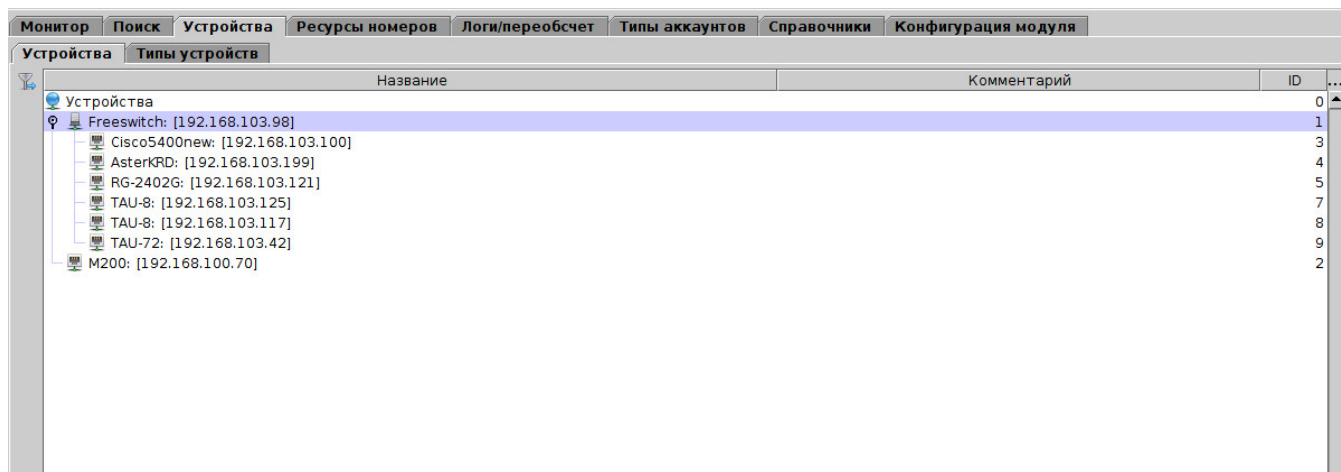
- **Комментарий**
- **Конфигурация**

Устройства модуля Voice

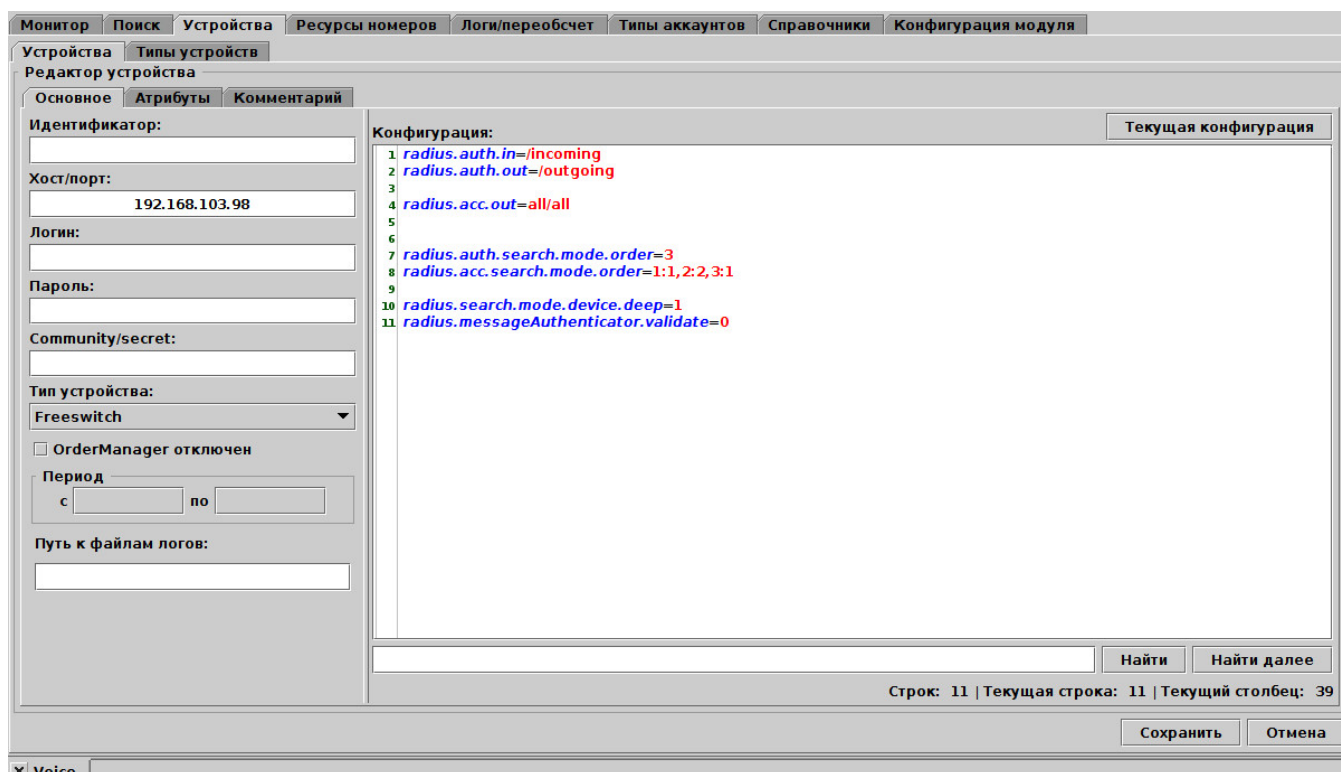
Устройства модуля Voice могут быть 2-х видов

- Источники данных для обработки логов CDR.
- NAS для предоставления VOIP-доступа по протоколу Radius.

Устройства представлены в виде дерева:



Редактор устройства имеет следующий вид:



В устройстве определяются параметры:

Идентификатор - используется для VOIP Radius для поиска устройства по Radius-параметру NAS-Identifier.

Хост/порт - Хост и порт устройства. Используется либо для управления доступом либо для поиска устройства по Radius-параметру NAS-IP-Address.

Логин - логин, может использоваться при управлении доступом.

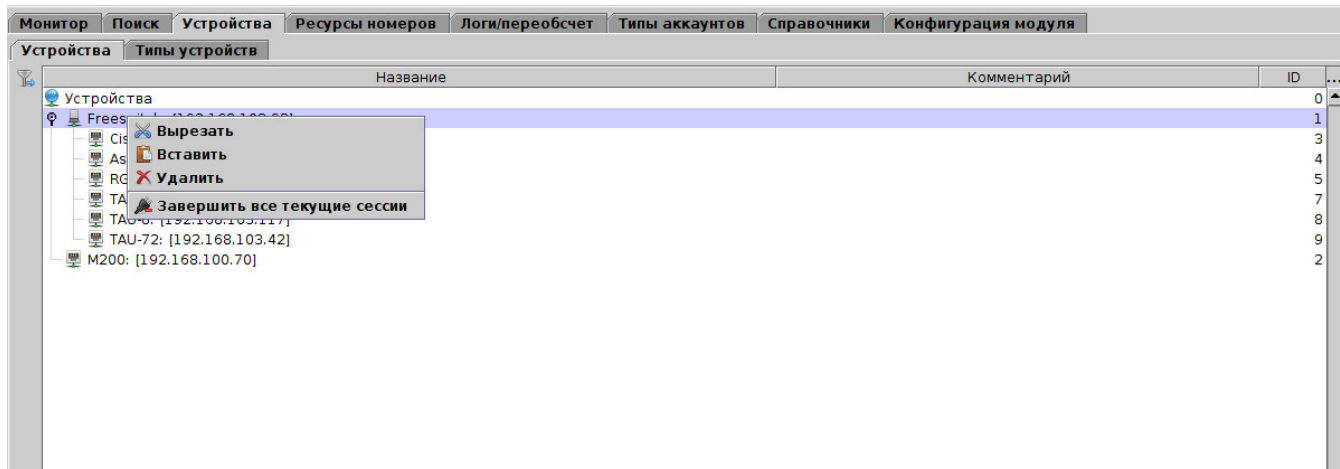
Пароль - пароль, может использоваться при управлении доступом.

Community/secret - Используется для VOIP Radius схем.

В конфигурации устройства указывается текстовая конфигурация, которая также зависит от процессора, использующего устройство.

Конфигурация каждого устройства **наследует все параметры конфигурации своего типа устройства**. Т.е. параметр, указанный в типе устройства будет доступен во устройствах данного типа.

В контекстном меню на каждом устройстве доступны следующие пункты:



Завершить все текущие сесии - завершает все текущие сессии на этом устройстве. Имеет смысл только для VOIP Radius схем.

Также есть команды - **Вырезать**, **Вставить**, **Удалить**, позволяющие редактировать дерево устройств.

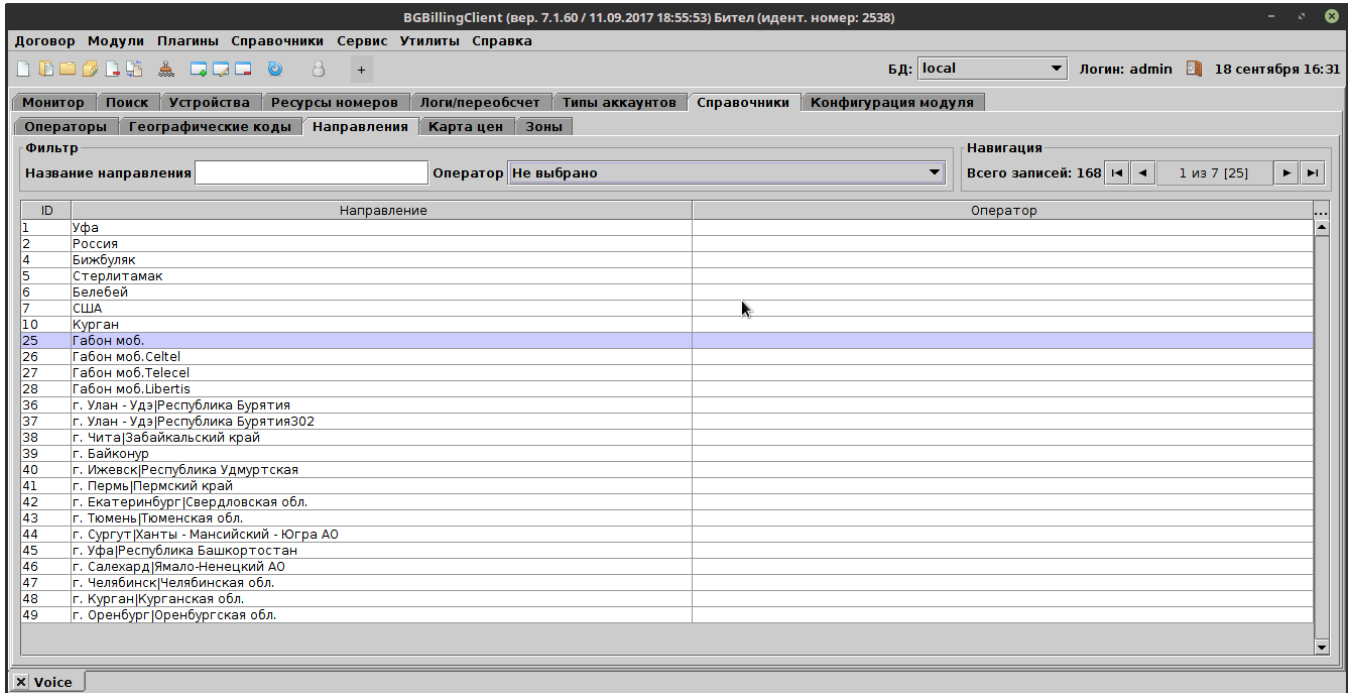
Географические справочники

Справочник направлений

При тарификации каждому звонку сопоставляется определенное направление, которое затем показывается в детализации для абонента, в отчетах операторов и т.д.

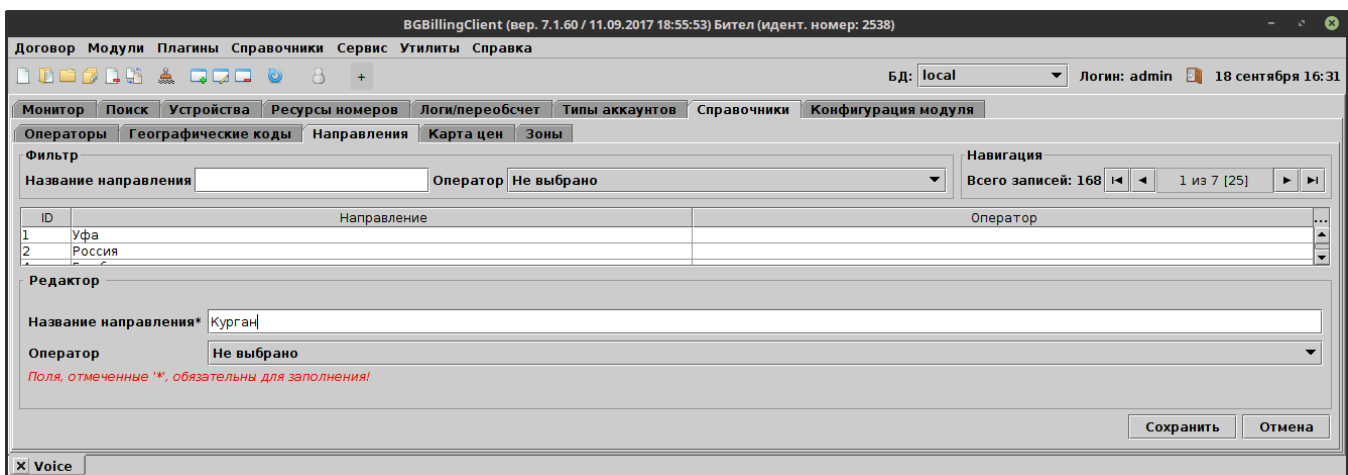
Чтобы открыть справочник направлений, необходимо зайти в меню **Модули** -> **<Название_экземпляра_модуля_Voice>**, далее перейти на вкладку **Справочники** -> **Направления**.

На данной вкладке выводится список направлений с возможностью фильтрации.



Фильтровать можно по названию и/или принадлежности к определенному оператору. Для фильтрации по названию достаточно начать вводить название направления.

Для ручного добавления/редактирования направления используется стандартная панель инструментов (кнопки **Новый элемент** и **Редактировать**). При этом открывается окно редактора направлений, в котором можно ввести название для нового/редактируемого направления, а также осуществить привязку направления к определенному оператору. Данная привязка предназначена для того, чтобы в операторских отчетах можно было выводить именно то название направления, которое требует оператор.

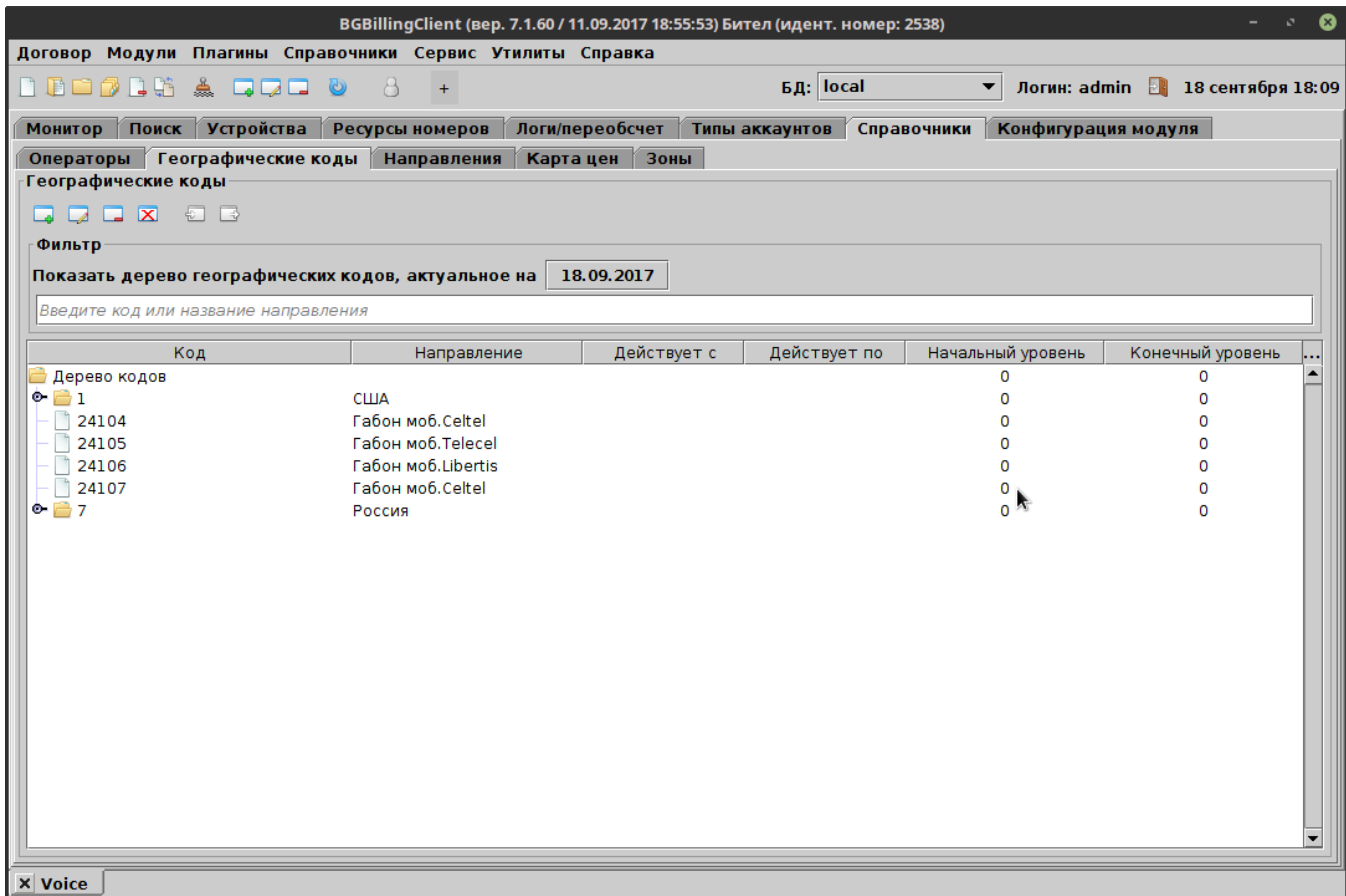


Обязательным для заполнения является поле "Название направления" - оно отмечено звездочкой.

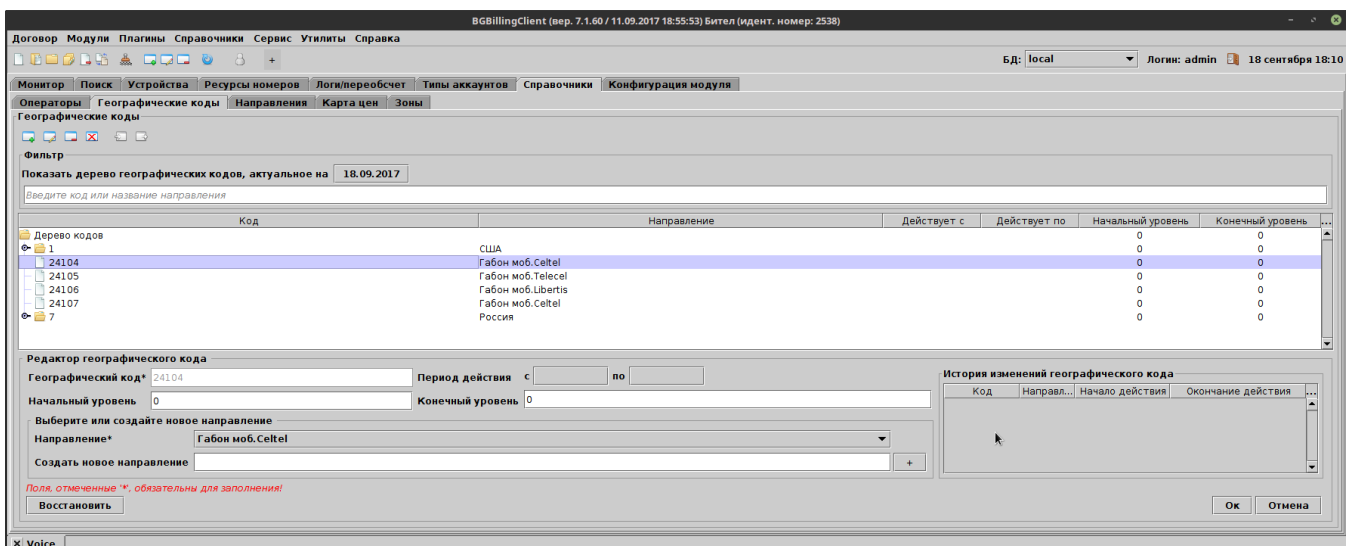
Справочник географических кодов

Географический код представляет собой запись вида "префикс-направление", характеризующаяся дополнительными атрибутами.

Для доступа к справочнику географических кодов необходимо зайти в меню **Модули** -> **<Название_экземпляра_модуля_Voice>**, далее перейти на вкладку **Справочники** -> **Географические коды**. Общий вид справочника географических кодов представлен на следующем изображении.

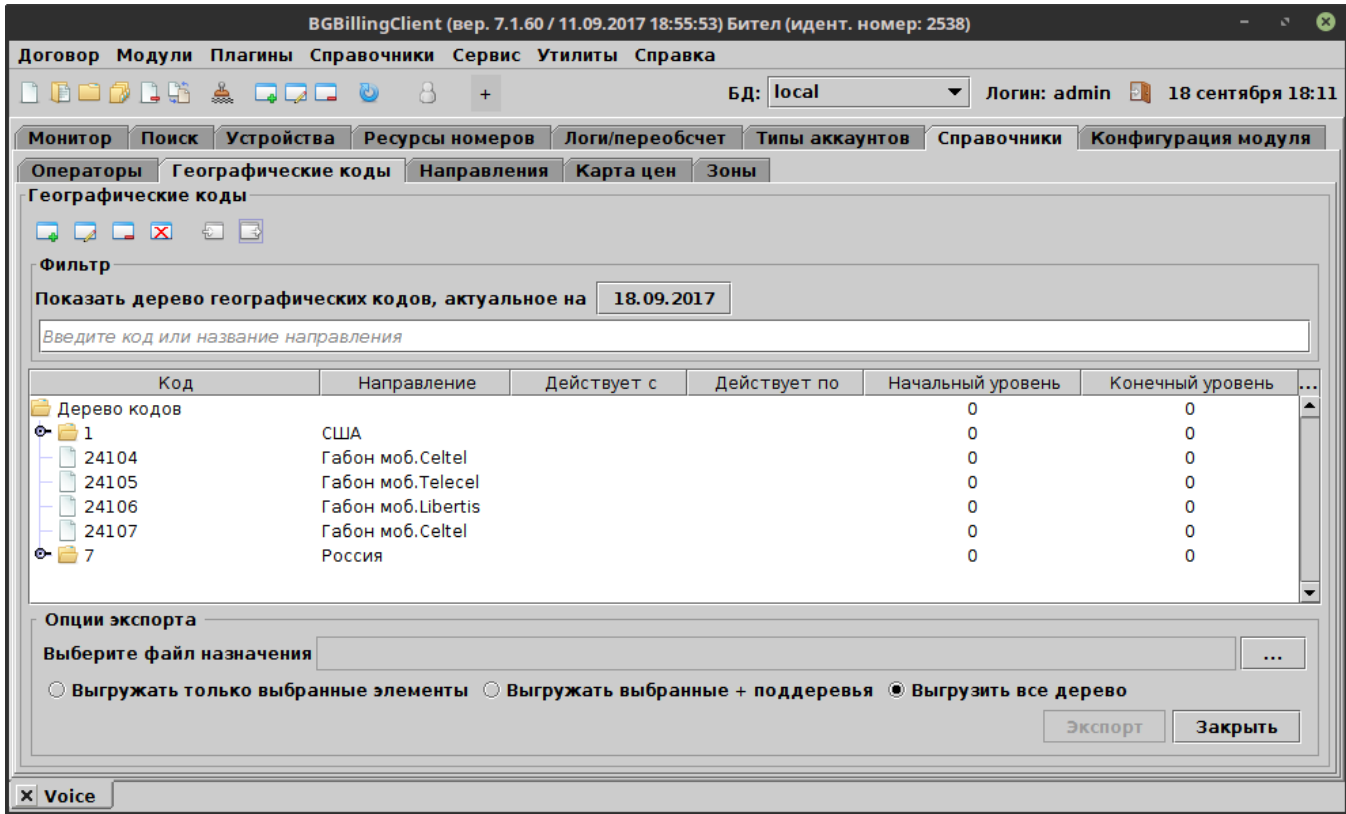


Для работы с отдельными элементами географических кодов используется панель инструментов, расположенная над фильтром. С помощью кнопки **Удалить элемент** можно удалить один или более выбранных элементов из дерева. Кнопка **Удалить все элементы** позволяет очистить дерево, не выбирая предварительно элементы. При нажатии кнопки **Новый элемент**, **Редактировать** на панели инструментов географических кодов открывается редактор, представленный на изображении ниже.



При редактировании существующего элемента поле Географический код будет недоступно. Таким образом, для уже существующего элемента можно поменять период действия, начальный и конечный уровень, а также направление. При отсутствии нужного направления в выпадающем списке можно создать новое направление, не переходя для этого в справочник направлений. Параметры **Начальный уровень** и **Конечный уровень** определяют до какой глубины будет уточняться направление при прохождении дерева. По умолчанию уровни не установлены (равны 0), что означает уточнение до самого нижнего узла. Глубина вложенности считается от нуля (корень дерева).

При нажатии кнопки **Экспорт** открывается окно с опциями для выгрузки выбранных (или всех сразу) географических кодов в файл. Изображение настроек выгрузки представлено ниже.



С помощью кнопки "..." выбирается файл назначения. Переключатели:

- **Выгружать только выбранные элементы** - в файл назначения попадут только выбранные в дереве географические коды. Если у выбранного элемента есть поддеревя элементов, то они не будут выгружены.
- **Выгружать выбранные + поддеревья** - в файл назначения попадут выбранные в дереве географические коды, а также, при наличии, все дочерние элементы.
- **Выгрузить все дерево** - в файл назначения попадут все элементы дерева географических кодов, не смотря на выбранные элементы.

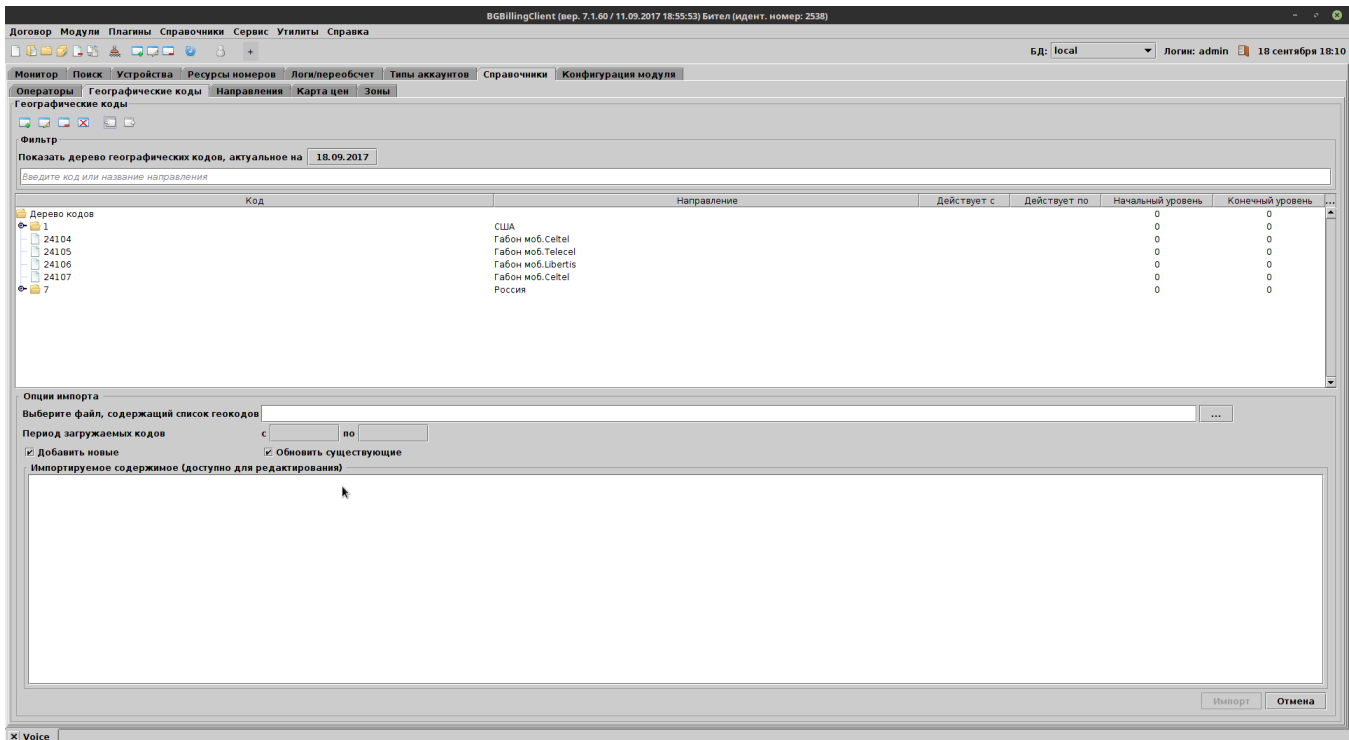
Кнопка **Экспорт** становится доступной только после того, как будет выбран файл назначения.

Формат выгружаемых данных:

<код> <название направления> <дата начала действия> <дата окончания действия> <начальный уровень> <конечный уровень>

Разделителем полей является символ табуляции ("t"), кодировка "UTF-8". При отсутствии данных (например, даты начала/окончания действия) выводится пустая строка.

При нажатии кнопки Импорт открывается окно с опциями для загрузки географических кодов. Окно с настройками представлено ниже.



Системой поддерживаются следующие форматы для загрузки:

- <код> <название направления>
- <код> <название направления> <начальный уровень> <конечный уровень>
- <код> <название направления> <дата начала действия> <дата окончания действия>
- <код> <название направления> <дата начала действия> <дата окончания действия> <начальный уровень> <конечный уровень>

Разделителем полей ожидается знак табуляции. Строки в файле могут соответствовать разным форматам. При необходимости можно задать единый период действия для всех загружаемых географических кодов. Данный период будет применяться для всех загружаемых строк, в которых отсутствует явное указание периода действия. При отсутствии какого-либо из параметров выставляется значение по умолчанию: 0 для уровней, null для периода (открытые даты).

В текстовую область **Импортируемое содержимое** попадает содержимое загружаемого файла. Данная область доступна для редактирования. Это значит, что можно оперативно подправить данные перед их импортом. Также можно вставить импортируемые данные из буфера обмена. Кнопка **Импорт** становится доступной только при наличии данных в текстовой области **Импортируемое содержимое**.

Следует иметь в виду, что каждому географическому коду должно соответствовать определенное направление. Это значит, что при отсутствии указанного в файле направления в справочнике направлений такая строка не будет импортирована. Опция **Добавить новые** позволяет автоматически "на лету" добавлять отсутствующие направления в справочник направлений. Опция **Обновить существующие** обновляет данные у уже имеющегося географического кода на те, которые содержатся в файле.

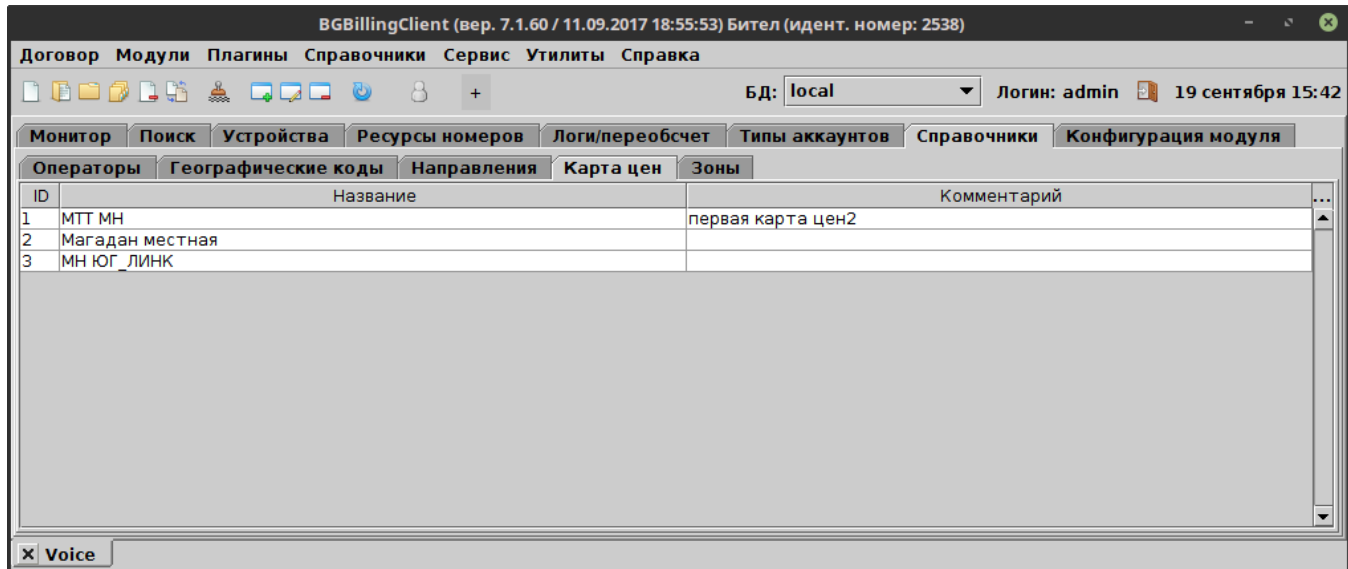
При возникновении каких-либо ошибок при загрузке будет выведено окно, в котором можно посмотреть какие строки не были загружены.

Карты цен

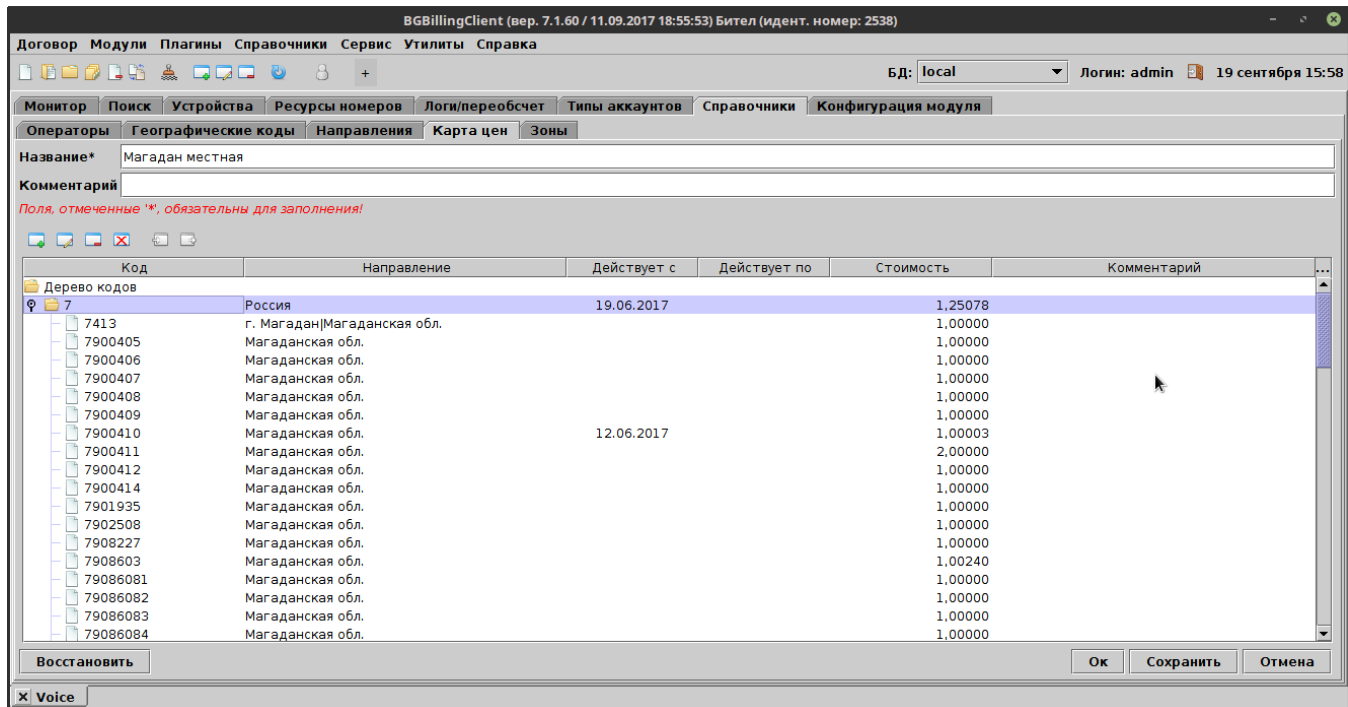
Карта цен представляет собой список географических кодов, в котором каждому элементу сопоставлена определенная цена. При использовании карты цен направление вычисляется по географическим кодам.

Для открытия справочника карт цен, необходимо перейти в меню **Модули** -> **<Название_экземпляра_модуля_Voice>**, далее перейти на вкладку **Справочники** -> **Карты цен**.

В появившейся панели отображается таблица с списком карт цен. Для добавления/удаления/редактирования карты цен используется стандартная панель инструментов.



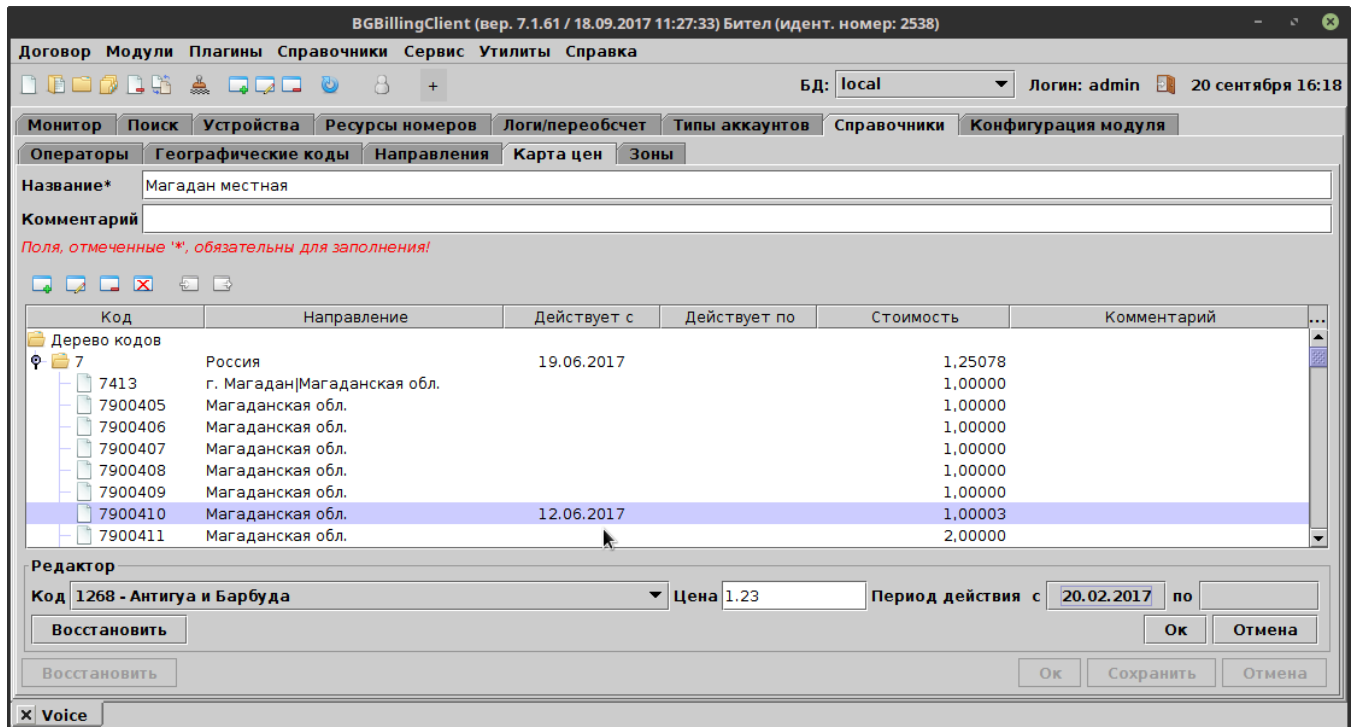
При создании новой или редактировании имеющейся карты цен открывается окно редактора карты цен.



Обязательным для заполнения является поле **Название**.

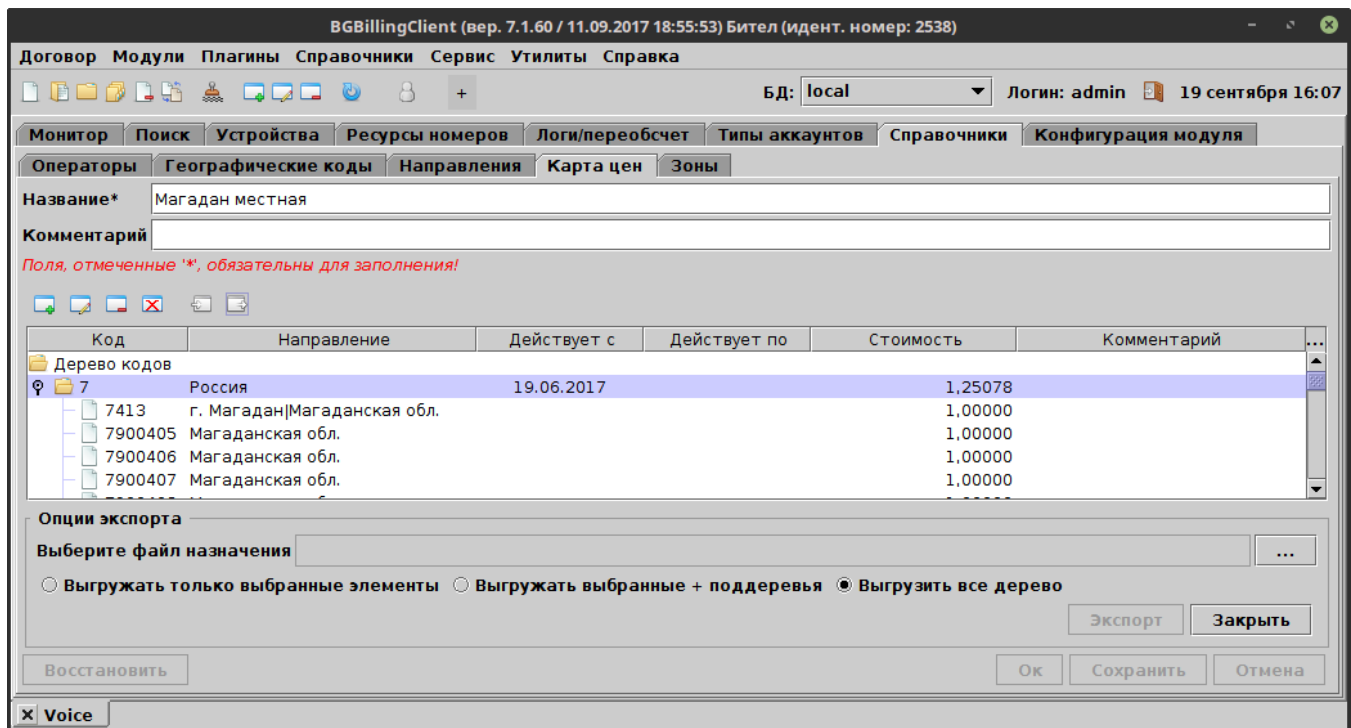
Для работы с элементами карты цен предусмотрена отдельная панель инструментов, расположенная над деревом элементов. С помощью панели можно создавать/редактировать/удалять выбранные/удалять все элементы, а также импортировать/экспортировать содержимое карты цен.

При нажатии кнопки **Новый элемент/Редактировать выделенный** откроется редактор элемента карты цен, в котором для выбранного из выпадающего списка географического кода (недоступно при редактировании элемента) можно задать период действия, стоимость.



С помощью кнопки **Удалить элемент** можно удалить один или более выбранных элементов из дерева. Кнопка **Удалить все элементы** позволяет очистить дерево, не выбирая предварительно элементы.

При нажатии кнопки **Экспорт** открывается окно с опциями для выгрузки выбранных (или всех сразу) элементов в файл. Изображение настроек выгрузки представлено ниже.



С помощью кнопки "..." выбирается файл назначения. Переключатели:

- **Выгрузить только выбранные элементы** - в файл назначения попадут только выбранные в дереве географические коды. Если у выбранного элемента есть поддеревья элементов, то они не будут выгружены.

- **Выгрузить выбранные + поддерева** - в файл назначения попадут выбранные в дереве географические коды, а также, при наличии, все дочерние элементы.
- **Выгрузить все дерево** - в файл назначения попадут все элементы дерева географических кодов, не смотря на выбранные элементы.

Кнопка **Экспорт** становится доступной только после того, как будет выбран файл назначения.

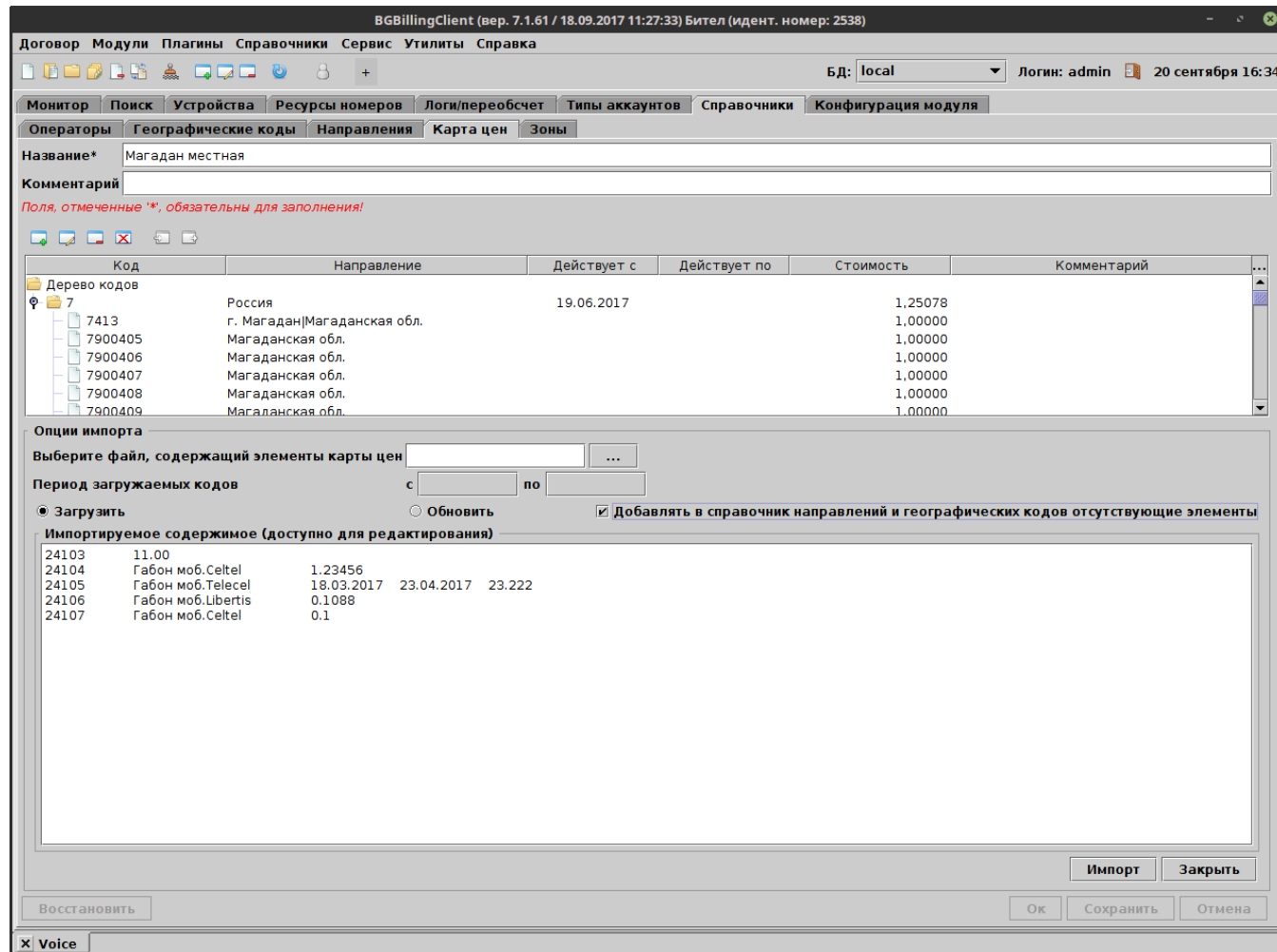
Формат выгружаемых данных:

<код> <название направления> <дата начала действия> <дата окончания действия> <стоимость>

Разделителем полей является символ табуляции ("t"), кодировка "UTF-8". При отсутствии данных (например, даты начала/окончания действия) выводится пустая строка.

При нажатии кнопки **Импорт** открывается окно с опциями для загрузки элементов в карту цен. Обращаем внимание, что импортировать элементы можно только в сохраненную карту цен! Перед загрузкой данных в новую карту цен воспользуйтесь кнопкой **Сохранить**.

Окно с настройками представлено ниже.



Системой поддерживаются следующие форматы для загрузки:

- <код> <стоимость>
- <код> <дата начала действия> <дата окончания действия> <стоимость>
- <код> <название направления> <стоимость>
- <код> <название направления> <дата начала действия> <дата окончания действия> <стоимость>

Разделителем полей ожидается знак табуляции. Строки в файле могут соответствовать разным форматам. При необходимости можно задать единый период действия для всех загружаемых элементов. Данный период будет применяться для всех загружаемых строк, в которых отсутствует явное указание периода действия. При отсутствии в справочнике географических кодов указанного в импортируемом файле кода строка будет проигнорирована. Для автоматического добавления отсутствующих географических кодов в справочник установите опцию **Добавлять в справочник направлений и географических кодов отсутствующие элементы**.

В текстовую область **Импортируемое содержимое** попадает содержимое загружаемого файла. Данная область доступна для редактирования. Это значит, что можно оперативно подправить данные перед их импортом. Также можно вставить импортируемые данные из буфера обмена. Кнопка **Импорт** становится доступной только при наличии данных в текстовой области **Импортируемое содержимое**.

Переключатель **Загрузить** полностью заменяет содержимое карты цен на импортируемые данные, а **Обновить** - обновляет данные у имеющихся элементов.

Зоны и карты зон

Сессии модуля Voice

Сессия модуля voice представляет собой звонок. Сессии бывают 2-х видов:

- Завершенные
- Текущие

Завершенные сессий это звонки, которые уже закончились. В независимости от того был создан звонок (по Radius-запросу или из логов CDR), после завершения они попадают в завершенные сессии. В случае создания звонка из CDR он сразу создается завершенным.

Для Radius сессий они становятся завершенными после прихода stop-пакета, или принудительного завершения из биллинга.

Текущие сессии это звонки, которые идут в данный момент. Просмотреть их можно в [Мониторе](#) и в [отчете на договоре](#) в отдельной вкладке. Они создаются только при прихода Radius start запросов или Radius update запросов(при условии что по этой сессии не было еще start).

Настройка и установка серверов

В модуле представлены два типа серверов:

- BGVoiceProcessor - осуществляет обработку логов CDR.
- BGVoiceRadius - обрабатывает Radius запросы от NAS.

Установка серверов Voice

Установка BGVoiceProcessor

Установка на платформу Linux

- 1) Извлеките **BGVoiceProcessor** из архива и скопируйте в каталог **/opt/bgbilling**;
- 2) Перейдите в каталог **/opt/bgbilling/BGVoiceProcessor**;
- 3) Удалите все **.ini**, **.bat** и **.exe** файлы:

```
rm -f ./*.bat & rm -f ./*.exe & rm -f ./*.ini
```

- 4) Откройте для редактирования файл **setenv.sh** и пропишите в нем путь к Java-машине, например так:

```
...  
    cd ${0%${0##*/}}.  
  
    JAVA_HOME=/opt/java/jdk8  
  
    if [ -z "$JAVA_HOME" ]; then  
        echo "The JAVA_HOME environment variable is not defined"  
        echo "This environment variable is needed to run this program"  
        exit 1  
    fi  
    ...
```

- 5) Установите права запуска для всех ***.sh** файлов:

```
chmod 744 *.sh
```

- 6) Настройте автозапуск **BGVoiceProcessor**. (/etc/systemd/system/multi-user.target.wants/bgvoice_processor.service)

```
[Unit]  
Description=BGVoiceProcessor Server  
After=network.target activemq.service  
Requires=activemq.service  
  
[Service]  
Type=forking  
TimeoutStartSec=120s  
ExecStart=/opt/bgbilling/BGVoiceProcessor/processor.sh start  
ExecStop=/opt/bgbilling/BGVoiceProcessor/processor.sh start  
User=bgbilling  
Group=bgbilling  
  
[Install]  
WantedBy=multi-user.target
```

- 7) Произведите настройку **voice-application.xml**
- 8) Обновите **BGVoiceProcessor** при помощи скрипта **/opt/bgbilling/BGVoiceProcessor/update.sh**
- 9) Для запуска и останова сервера **BGVoiceProcessor** используйте **systemctl start bgvoice_processor.service** и **systemctl stop bgvoice_processor.service**

Установка на платформу Windows

Для установки **BGVoiceProcessor** на платформу Windows на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить **BGVoiceProcessor** стоит Java-машина. Если её нет, установите версию не меньше 1.8.
- 2) Загрузите с сервера **BGVoiceProcessor**;

- 3) Распакуйте архив на диск C.;
- 4) Установите переменную окружения **BGVOICE_PROCESSOR_HOME =C:\BGVoiceProcessor**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;
- 5) Установите службу **BGVoiceProcessor**, для чего запустите файл **processor_install.bat**;
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **processor_uninstall.bat**;
- 7) Обновите как обычные [серверные приложения биллинга](#) ;
- 8) Для запуска и останова сервера **BGVoiceProcessor** используйте консоль запуска и управления службами, служба **BGVoiceProcessor**.

Установка BGVoiceRadius

Установка на платформу Linux

- 1) Извлеките **BGVoiceRadius** из архива и скопируйте в каталог **/opt/bgbilling/**;
- 2) Перейдите в каталог **/opt/bgbilling/BGVoiceRadius**;
- 3) Удалите все **.ini**, **.bat** и **.exe** файлы:

```
rm -f ./*.bat & rm -f ./*.exe & rm -f ./*.ini
```

- 4) Откройте для редактирования файл **setenv.sh** и пропишите в нем путь к Java-машине, например так:

```
...  
    cd ${0%${0##*/}}.  
  
    JAVA_HOME=/opt/java/jdk8  
  
    if [ -z "$JAVA_HOME" ]; then  
        echo "The JAVA_HOME environment variable is not defined"  
        echo "This environment variable is needed to run this program"  
        exit 1  
    fi  
    ...
```

- 5) Установите права запуска для всех ***.sh** файлов:

```
chmod 744 *.sh
```

- 6) Возьмите из каталога **BGVoiceRadius/script** скрипт запуска **bgvoice_radius** и скопируйте его в каталог **/etc/init.d**, установите права на исполнение (см. выше). Если вы изменили каталог установки или переименовывали **BGVoiceRadius**, скорректируйте скрипт.

- 7) Выясните текущий уровень запуска системы командой:

```
[root@gate init.d]# runlevel  
N 3
```

- 8) Добавьте службу в автоматический запуск.

```
chkconfig --add bgvoice_radius  
chkconfig --level {lev} bgvoice_radius on
```

где {lev} - требуемый уровень запуска.

- 9) Произведите настройку **voice-application.xml**; (Ссылка).
- 10) Обновите как обычные [серверные приложения биллинга](#);
- 11) Для запуска и останова сервера **BGVoiceRadius** используйте скрипты **radius_start.sh** и **radius_stop.sh**.

Установка на платформу Windows

Для установки **BGVoiceRadius** на платформу Windows на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить **BGVoiceRadius** стоит Java-машина. Если её нет, установите версию не меньше 1.8.
- 2) Загрузите с сервера **BGVoiceRadius**;
- 3) Распакуйте архив на диск C:;

- 4) Установите переменную окружения **BGVOICE_RADIUS_HOME =C:\BGVoiceRadius**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;
- 5) Установите службу **BGVoiceRadius**, для чего запустите файл **radius_unstall.bat**;
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **radius_uninstall.bat**;
- 7) Обновите как обычные [серверные приложения биллинга](#) ;
- 8) Для запуска и останова сервера **BGVoiceRadius** используйте консоль запуска и управления службами, служба **BGVoiceRadius**.

Общая часть конфигурации Voice-серверов.

Рассмотрим общую часть XML-конфигурации обоих серверов. Файл voice-application.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<application context="app">
  <param name="app.name" value="BGVoiceApplication"/>
  <param name="app.id" value="2"/>
  <param name="db.driver" value="com.mysql.jdbc.Driver"/>
  <param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-
8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&
queryTimeoutKillsConnection=true"/>
  <param name="db.user" value="bill"/>
  <param name="db.pswd" value="bgbilling"/>

  <param name="mq.url" value="failover:(tcp://localhost:61616)"/>
  <param name="mq.user" value="bill"/>
  <param name="mq.pswd" value="bgbilling"/>

  <param name="moduleId" value="" />
  <bean name="app" class="ru.bitel.bgbilling.modules.voice.runtime.VoiceLogProcessor" />
  ..
</application>
```

Параметры:

- **app.name** определяет имя приложения, оно используется, например в системе алармов;
- **app.id** - уникальный числовой идентификатор приложения среди всех приложений биллинга с данным параметром в XML-конфигурации, значение его не должно меняться всё время жизни системы;
- **moduleId** - код экземпляра модуля Voice, к которому относится сервер.

Далее следуют стандартные параметры с настройкой доступа к [серверу БД](#) и к [MQ-серверу](#) (серверам).

Настройка BGVoiceProcessor

BGVoiceProcessor - это сервер, который занимается обработкой логов CDR.

Пример типовой конфигурации voice-application.xml, которая идет в дистрибутиве:

voice-application.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<application context="app">
  <param name="app.name" value="BGVoiceApplication"/>
  <param name="app.id" value="2"/>
  <param name="db.driver" value="com.mysql.jdbc.Driver"/>
  <param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&queryTimeoutKillsConnection=true"/>
  <param name="db.user" value="bill"/>
  <param name="db.pswd" value="bgbilling"/>

  <param name="mq.url" value="failover:(tcp://localhost:61616)"/>
  <param name="mq.user" value="bill"/>
  <param name="mq.pswd" value="bgbilling"/>

  <param name="moduleId" value="" />
  <bean name="app" class="ru.bitel.bgbilling.modules.voice.runtime.VoiceLogProcessor" />

  <context name="logs">
    <bean name="logRecordProcessor" class="ru.bitel.bgbilling.modules.voice.accounting.mediation.process.LogRecordProcessor">
      <param name="initialDelay" value="10"/>
      <param name="delay">20</param>
    </bean>
  </context>

  <context name="logsRuntime">
    <bean name="logRuntimeProcessor" class="ru.bitel.bgbilling.modules.voice.accounting.mediation.runtime.LogRuntimeProcessor">
      </bean>
    </context>
  </context>
</application>
```

Настройка BGVoiceRadius

BGVoiceRadius - сервер обрабатывающий radius-запросы.

Пример типовой конфигурации voice-application.xml, которая идет в дистрибутиве:

```
<?xml version="1.0" encoding="UTF-8"?>
<application context="app">
  <param name="app.name" value="BGVoiceApplication"/>
  <param name="app.id" value="1"/>
  <param name="db.driver" value="com.mysql.jdbc.Driver"/>
  <param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-
8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&
queryTimeoutKillsConnection=true"/>
  <param name="db.user" value="bill"/>
  <param name="db.pswd" value="bgbilling"/>

  <param name="mq.url" value="failover:(tcp://localhost:61616)"/>
  <param name="mq.user" value="bill"/>
  <param name="mq.pswd" value="bgbilling"/>

  <param name="moduleId" value="" />
  <bean name="app" class="ru.bitel.bgbilling.modules.voice.runtime.VoiceRadius" />

  <!-- Radius access- -->
  <param name="datalog.radius.access.dir" value="data/radius/access" />
  <!-- Radius accounting -->
  <param name="datalog.radius.accounting.dir" value="data/radius/accounting" />

  <context name="radius">
    <bean name="radiusProcessor" class="ru.bitel.bgbilling.modules.voice.radius.
VoiceRadiusProcessor"/>
    <scheduledExecutorService name="access_hrlydtlggr" corePoolSize="1" />

    <bean name="accessDataLogger" class="ru.bitel.bgbilling.modules.voice.radius.
RadiusHourlyDataLogger">
      <param name="scheduledExecutor">access_hrlydtlggr</param>
      <param name="directoryName" value="datalog.radius.access.dir" />
    </bean>

    <!-- access- -->
    <bean name="radiusListener" class="ru.bitel.bgbilling.modules.voice.radius.VoiceRadiusListener">
      <constructor>
        <!-- (), . - -->
        <param name="host" value="" />
        <!-- , -->
        <param name="port" value="1815"/>
        <!-- -->
        <param name="byteBufferCapacity">512 * 1024</param>
        <!-- -->
        <param name="processor">radiusProcessor</param>
        <!-- , RadiusListener.Mode.authentication -->
        <param name="mode">RadiusListener.Mode.authentication</param>
        <!-- dataLogger -->
        <param name="dataLogger">accessDataLogger</param>
        <!-- - -->
        <param name="threadCount">10</param>
        <!-- -->
        <param name="maxQueueSize">200</param>
      </constructor>
    </bean>

    <scheduledExecutorService name="accounting_hrlydtlggr" corePoolSize="1" />
    <bean name="accountingDataLogger" class="ru.bitel.bgbilling.modules.voice.radius.
RadiusHourlyDataLogger">
      <param name="scheduledExecutor">accounting_hrlydtlggr</param>
      <param name="directoryName" value="datalog.radius.accounting.dir" />
    </bean>
  </context>
</application>
```

```
<!-- accounting- -->
<bean name="radiusListener" class="ru.bitel.bgbilling.modules.voice.radius.VoiceRadiusListener">
  <constructor>
    <!-- (), . - -->
    <param name="host" value=""/>
    <!-- , -->
    <param name="port" value="1816"/>
    <!-- -->
    <param name="byteBufferCapacity">512 * 1024</param>
    <!-- -->
    <param name="processor">radiusProcessor</param>
    <!-- , RadiusListener.Mode.accounting -->
    <param name="mode">RadiusListener.Mode.accounting</param>
    <!-- dataLogger -->
    <param name="dataLogger">accountingDataLogger</param>
    <!-- - -->
    <param name="threadCount">10</param>
    <!-- -->
    <param name="maxQueueSize">200</param>
  </constructor>
</bean>

</context>

</application>
```

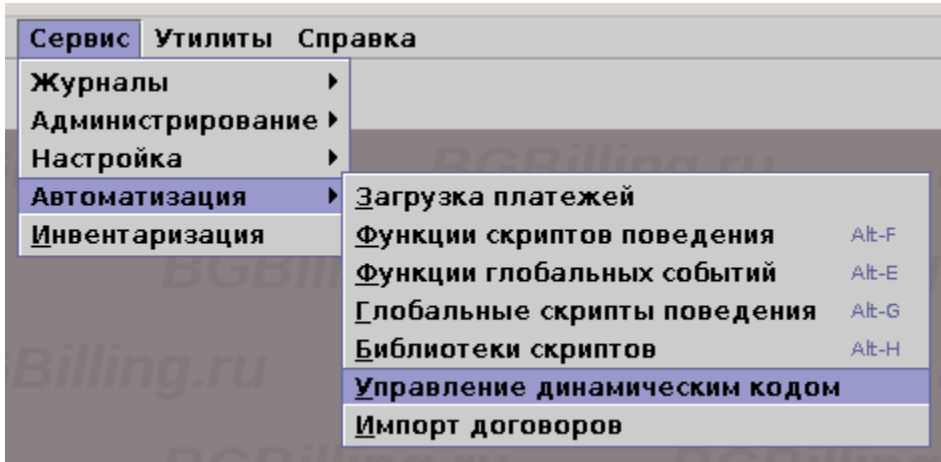
Все необходимые параметры прокомментированы в данном примере.

Обработка логов CDR в модуле Voice

Для обработки логов должен быть запущен сервер обработки логов.

Обработку и загрузку логов осуществляет специальный класс **обработчик логов(Mediator)**.

Его необходимо подготовить, загрузить и скомпилировать. Для этого в клиенте биллинга переходим в раздел



```
M200Mediator.java X
ru.ntl.bgbilling.modules.voice.mediator.m200.M200Mediator
1 package ru.ntl.bgbilling.modules.voice.mediator.m200;
2
3 import java.io.File;
4 import java.nio.file.Files;
5 import java.nio.file.Path;
6 import java.nio.file.Paths;
7 import java.time.LocalDateTime;
8 import java.time.format.DateTimeFormatter;
9 import java.util.Calendar;
10 import java.util.Date;
11 import java.util.regex.Matcher;
12 import java.util.regex.Pattern;
13
14 import org.apache.log4j.Logger;
15
16 import ru.bitel.bgbilling.modules.voice.accounting.mediation.Mediator;
17 import ru.bitel.bgbilling.modules.voice.accounting.mediation.VoiceRecord;
18 import ru.bitel.bgbilling.modules.voice.accounting.mediation.VoiceRecordProcessor;
19 import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDevice;
20 import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDeviceType;
21 import ru.bitel.bgbilling.server.util.Setup;
22 import ru.bitel.common.ParameterMap;
23 import ru.bitel.common.TimeUtils;
24 import ru.bitel.common.Utils;
25
26 /**
27  * Медиатор m200. Берет из папки файлы вида 2020/09/20/01.log
28  */
29 public class M200Mediator
30     implements Mediator
```

Добавляем новый класс или находим существующий

Для добавления кликаем по иконке на тулбаре



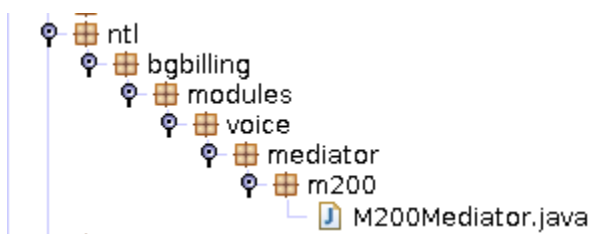
В открывшемся окне вводим название пакета и Имя класса

Пакет:

Имя класса (ресурса):

Шаблон:

Нажимаем Ок и находим в дереве наш класс



Двойным кликом открываем класс для редактирования и копируем в окно редактора содержимое класса.

M200Mediator.java X

ru.ntl.bgbilling.modules.voice.mediator.m200.M200Mediator

```
1 package ru.ntl.bgbilling.modules.voice.mediator.m200;
2
3 import java.io.File;
4 import java.nio.file.Files;
5 import java.nio.file.Path;
6 import java.nio.file.Paths;
7 import java.time.LocalDateTime;
8 import java.time.format.DateTimeFormatter;
9 import java.util.Calendar;
10 import java.util.Date;
11 import java.util.regex.Matcher;
12 import java.util.regex.Pattern;
13
14 import org.apache.log4j.Logger;
15
16 import ru.bitel.bgbilling.modules.voice.accounting.mediation.Mediator;
17 import ru.bitel.bgbilling.modules.voice.accounting.mediation.VoiceRecord;
18 import ru.bitel.bgbilling.modules.voice.accounting.mediation.VoiceRecordProcessor;
19 import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDevice;
20 import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDeviceType;
21 import ru.bitel.bgbilling.server.util.Setup;
22 import ru.bitel.common.ParameterMap;
23 import ru.bitel.common.TimeUtils;
24 import ru.bitel.common.Utils;
25
26 /**
27  * Медиатор m200. Берет из папки файлы вида 2020/09/20/01.log
28  */
29 public class M200Mediator
30     implements Mediator
31 {
```

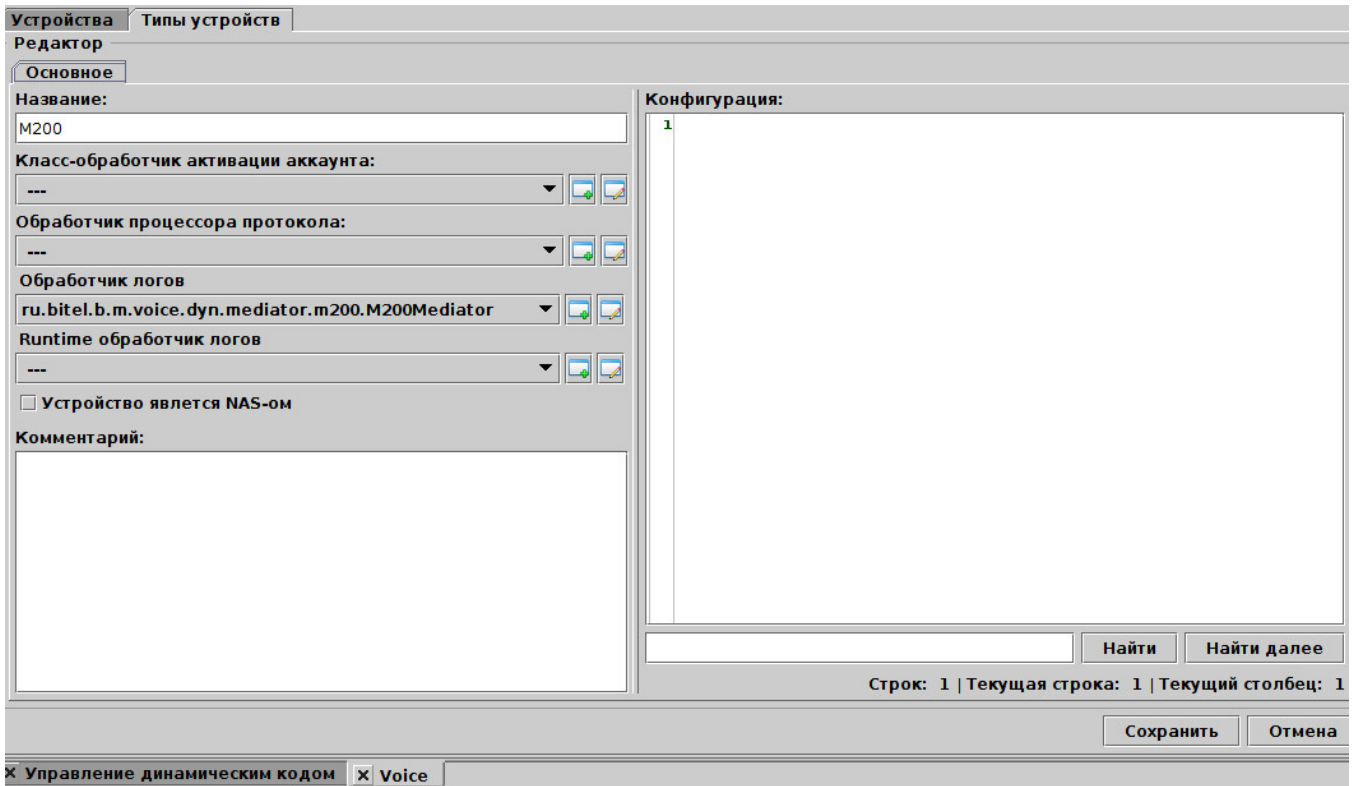
Далее нажимаем на кнопку

Сохранить и скомпилировать

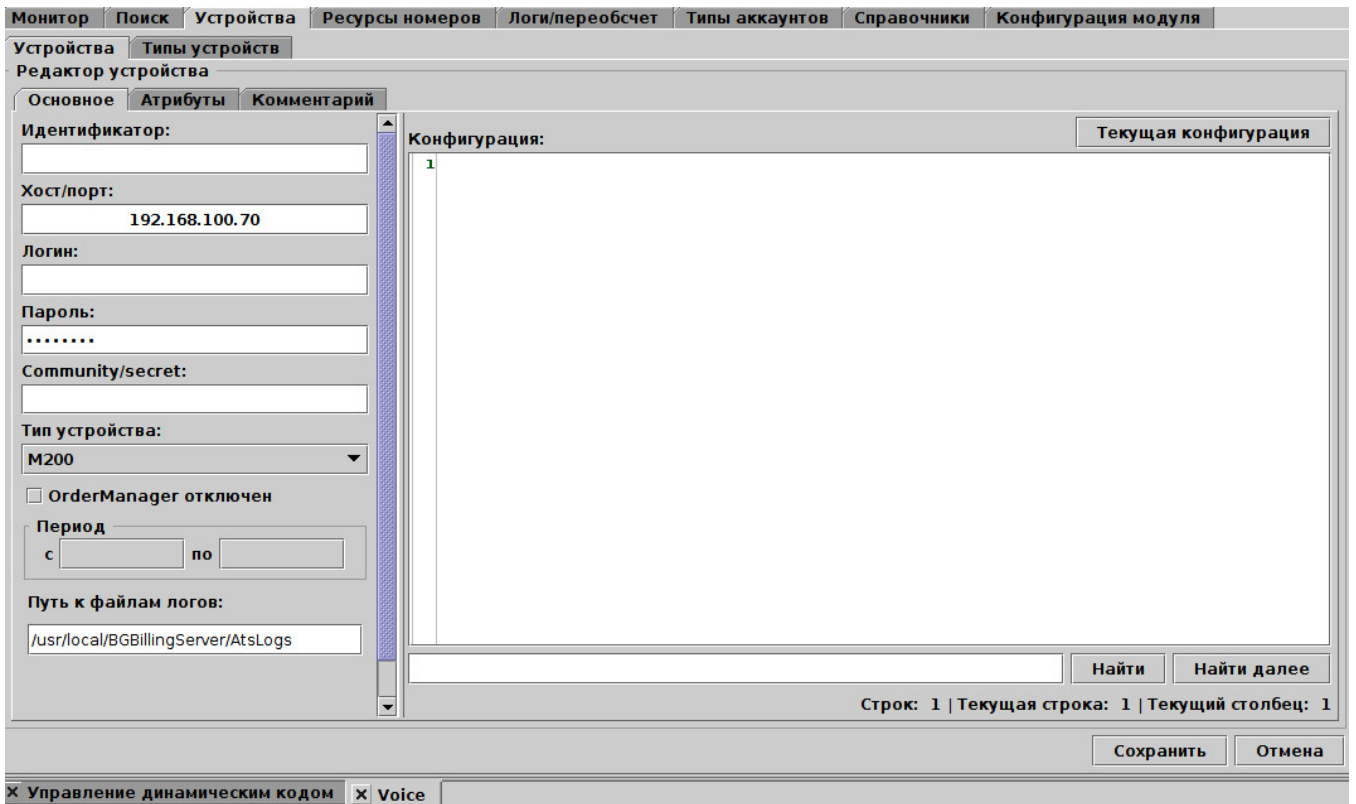
Если компиляция проходит без ошибок то можно переходить к следующим шагам.

Для обработки логов мы должны вначале добавить [тип устройства](#) и [устройство](#) данного типа.

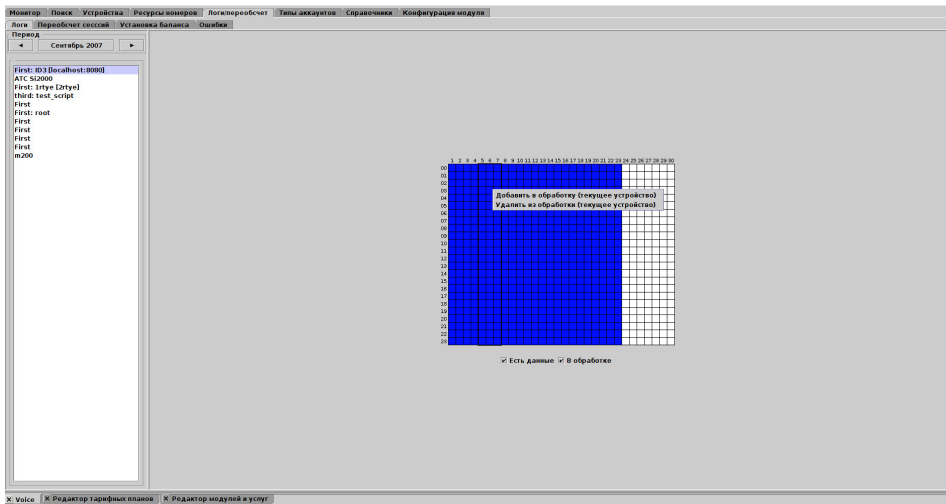
В [типе устройства](#) мы должны прописать класс [обработчика логов](#):



В **устройстве** мы должны прописать путь к файлам логов :



Далее на вкладке **Логи** мы можем добавлять логи для устройств:

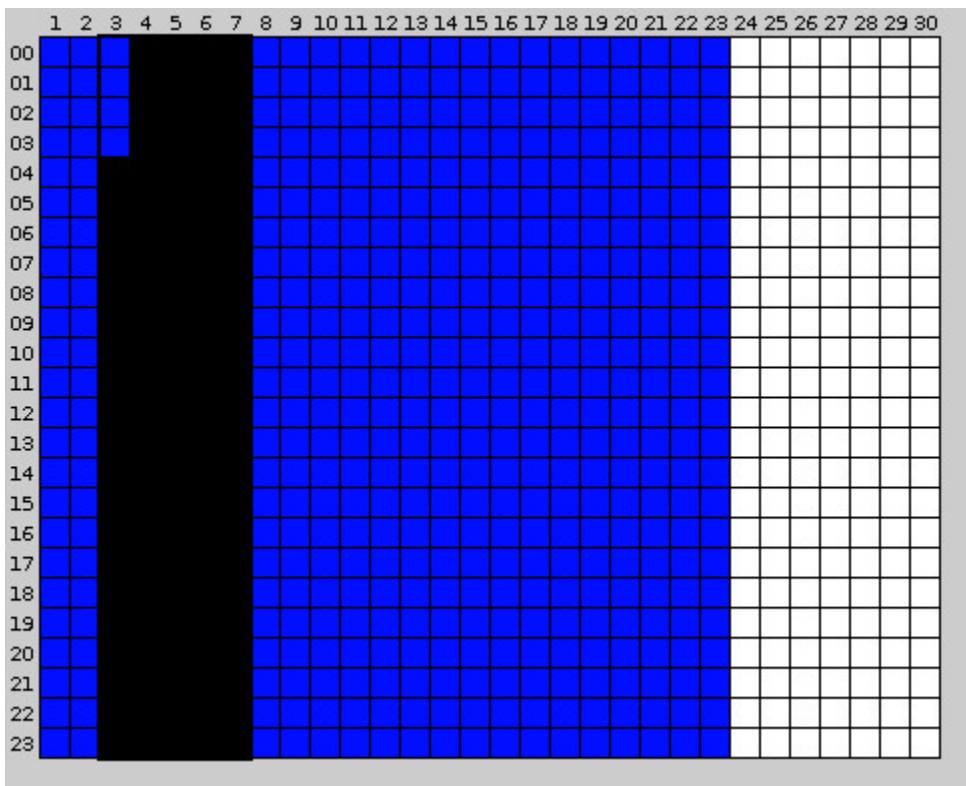


Слева вы выбираем дату и источник. Справа смотрим наличия логов в таблице, в которой по вертикали идут часы, а по горизонтали дни месяца. Синими цветом показываются те часы, для которых есть логи.

Минимальная единица обработки логов - 1 день. Для выделения некоторого диапазона дней доступно 2 пункта(при нажатии правой кнопкой мыши):

- Добавить в обработку(текущее устройство)
- Удалить из обработки(текущее устройство).

Эти пункты соответственно добавляют логи в обработку. После добавления логи начинают обрабатываться и состояние обработки выделяется цветом:



В данном случае черным цветом обозначены часы, который добавлены на обработку но еще не обработаны. После окончания обработки можно перейти на вкладку **Ошибки** :

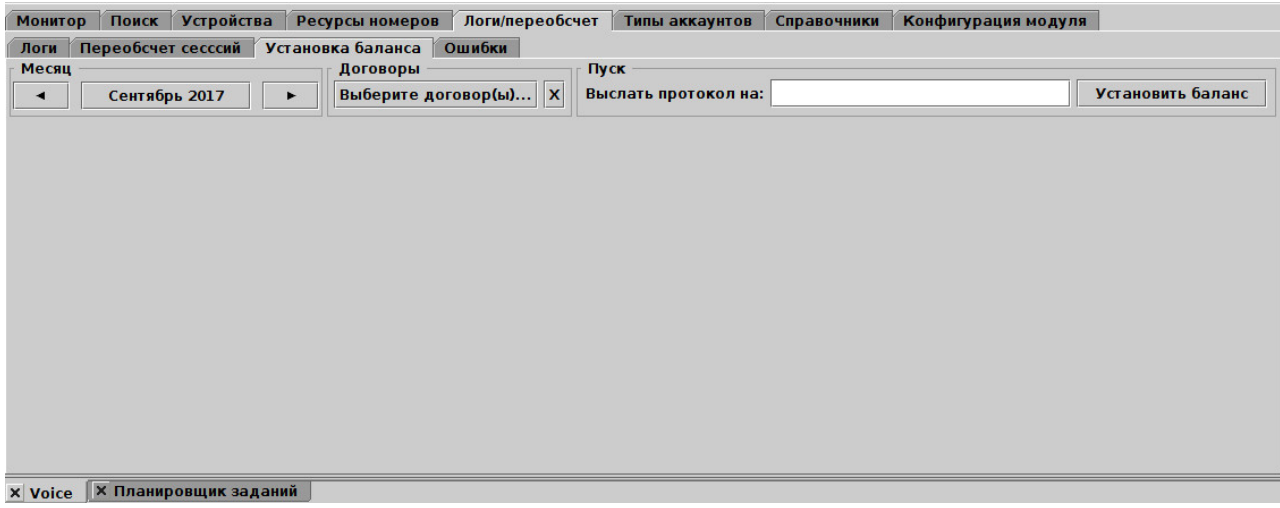
id	Договор	Аккаунт	С номера	На номер	Час	Старт	Ошибка	С порта	На порт
1149923			78462787632	78469931710	23.09.2007 23:00:00	23.09.2007 23:00:08	[1] Аккаунт не найден	C0358	C0155
1149924			78469931710	78462787632	23.09.2007 23:00:00	23.09.2007 23:00:08	[1] Аккаунт не найден	C0358	C0358
1149925			78462787632	78469975223	23.09.2007 23:00:00	23.09.2007 23:00:12	[1] Аккаунт не найден	C1056	C1051
1149926			78462787632	78469931710	23.09.2007 23:00:00	23.09.2007 23:00:16	[1] Аккаунт не найден	C0455	C0155
1149927			78469308845	78462769113	23.09.2007 23:00:00	23.09.2007 23:00:03	[1] Аккаунт не найден	C0352	C1054
1149928			78463120431	78462793533	23.09.2007 23:00:00	23.09.2007 23:00:20	[1] Аккаунт не найден	C0954	C0352
1149929			78462787632	78469931710	23.09.2007 23:00:00	23.09.2007 23:00:23	[1] Аккаунт не найден	C0954	C0155
1149930			78462787632	78469975223	23.09.2007 23:00:00	23.09.2007 23:00:25	[1] Аккаунт не найден	C1056	C1051
1149931			78469931710	78462787632	23.09.2007 23:00:00	23.09.2007 23:00:26	[1] Аккаунт не найден	C0353	C0358
1149932			78462787632	78462244444	23.09.2007 23:00:00	23.09.2007 23:00:12	[1] Аккаунт не найден	C0358	C0352
1149933			78462787632	78469931710	23.09.2007 23:00:00	23.09.2007 23:00:30	[1] Аккаунт не найден	C0955	C0155
1149934			78469990728	78463120644	23.09.2007 23:00:00	23.09.2007 23:00:33	[1] Аккаунт не найден	C0353	C0454
1149935			78469931710	78462787632	23.09.2007 23:00:00	23.09.2007 23:00:41	[1] Аккаунт не найден	C0353	C0358
1149936			78463121210	78469988114	23.09.2007 23:00:00	23.09.2007 23:00:52	[1] Аккаунт не найден	C1251	C0155
1149937			78462787632	78462244444	23.09.2007 23:00:00	23.09.2007 23:00:40	[1] Аккаунт не найден	C0955	C0255
1149938			78469990728	78463120644	23.09.2007 23:00:00	23.09.2007 23:00:45	[1] Аккаунт не найден	C0158	C0358
1149939			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:47	[1] Аккаунт не найден	C1051	C1051
1149940			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:46	[1] Аккаунт не найден	C1051	C1051
1149941			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:45	[1] Аккаунт не найден	C1051	C1051
1149942			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:44	[1] Аккаунт не найден	C1051	C1051
1149943			78469999999	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:16	[1] Аккаунт не найден	C0157	C0454
1149944			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:42	[1] Аккаунт не найден	C1051	C1051
1149945			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:16	[1] Аккаунт не найден	C0954	C0957
1149946			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:42	[1] Аккаунт не найден	C1051	C1051
1149947			78460000000	78469976828	23.09.2007 23:00:00	23.09.2007 23:00:17	[1] Аккаунт не найден	C0954	C0957

Процесс добавления логов на обработку можно автоматизировать с помощью задачи планировщика **Генерация заданий на обработку логов** модуля Voice. В параметрах запуска нужно поставить

```
# ,
devices=2
```

Результирующими данными обработки логов являются логи сессий в договорах клиентов и операторов, привязанные к клиентским или операторским аккаунтам.

Для того, чтобы наработка сессий перешла в баланс договора используется вкладка модуля **Установка баланса**. Единственным параметром выступает дата месяца, за который необходимо установить баланс и договор.



Если договор не указан, то баланс устанавливается для всех.

Процесс установки баланса можно автоматизировать с помощью задачи планировщика **Установка Баланса**. Она будет запускаться в заданное время и устанавливать баланс по всем договорам. Рекомендуем настраивать два экземпляра задачи: одна задача должна запускать с той частотой с которой у вас снимаются логи с АТС, а вторая раз в месяц, например 1 числа после загрузки последних логов за предыдущий месяц и производить финальную установку статуса за прошедший месяц (у этой задачи нужно установить в конфиге параметр **minus.days=1**)

Обработчик логов CDR

Обработка логов CDR происходит с помощью обработчика в динамическом коде, он реализует парсинг логов.

В поставке модуля есть

1. **ru.bitel.bgbilling.modules.voice.dyn.mediator.phone.DefaultMediator** - разбирает логи в [формате старого модуля Phone](#). Сделан для совместимости.
2. **ru.bitel.bgbilling.modules.voice.dyn.mediator.m200.M200Mediator** - разбирает логи в формате ATC M-200.
3. **ru.bitel.bgbilling.modules.voice.dyn.mediator.asterisk.AsteriskMediator** - разбирает логи в формате ATC Asterisk.

Если вы будете использовать какой-то из этих обработчиков и собираетесь его менять, то рекомендуем сделать его копию в другом пакете и использовать, тогда они не будут перетираться при обновлении.

Новые медиаторы будут описаны в нашей Wiki или документации.

Так же возможно написать свой Mediator для любой ATC. Вот описание интерфейса Mediator-a:

```
/**
 *
 *
 */
public interface Mediator
{
    /**
     *
     * @param setup
     * @param moduleId
     * @param device
     * @param deviceType
     * @param config
     * @return
     * @throws Exception
     */
    public Object init( Setup setup, int moduleId, VoiceDevice device, VoiceDeviceType deviceType, ParameterMap
config )
        throws Exception;

    /**
     *
     * VoiceRecord record = processor.next();
     *
     * @param processor
     * @param hour
     * @throws Exception
     */
    public void readHourDataLog( VoiceRecordProcessor processor, Date hour )
        throws Exception;

    /**
     * . data - 31 ,
     *
     * @param month
     * @param data
     */
    public void getLogExists( Date month, int [] data );
}
```


Runtime обработчик логов CDR

Модуль Voice поддерживает обработку логов CDR по мере появления их в папке. Для этого в типе устройства надо поставить - Runtime обработчик логов CDR.

Чтобы он работал нужно раскомментировать вот эту секцию в файле voice-application.xml приложения BGVoiceProcesso:

```
<context name="logsRuntime">
  <bean name="logRuntimeProcessor" class="ru.bitel.bgbilling.modules.voice.accounting.mediation.runtime.
LogRuntimeProcessor">
  </bean>
</context>
```

Это обработчик работает обычно в паре с [обработчиком логов CDR](#). Runtime обработчик берет логи новые в какой-либо папке, обрабатывает и перекладывает их в другую папку, разложенную по дням и часам.

А обычный обработчик логов CDR используется потом для обработки этих логов повторно в ручном режиме в случае каких-либо корректировок.

Runtime обработчик периодически опрашивает папку в которой появляются логи, и частота опроса задается настройка в устройстве:

```
log.runtime.cronTab=0 0/2 * * * ?
```

Тут задается расписание в формате cron в формате cron библиотеки Quartz. А данном примере запускается каждую вторую минуту(0,2,4,6...). Если не задать эту настройку на устройстве, то Runtime обработчик логов работать не будет.

Как примеры доступны некоторые обработчики в стандартной поставке и так же можно создавать свои.

В стандартной поставке есть пример пары связанных обработчиков логов для ATC Asterisk:

ru.bitel.bgbilling.modules.voice.dyn.mediator.asterisk.AsteriskMediator - обработчик логов CDR.

ru.bitel.bgbilling.modules.voice.dyn.mediator.asterisk.AsteriskRuntimeMediator - runtime обработчик логов CDR.

Также можно создавать свои обработчики. Runtime обработчик должен реализовывать интерфейс `ru.bitel.bgbilling.modules.voice.accounting.mediation.runtime.RuntimeMediator`:

```
/**
 * Runtime CDR.
 *
 */
public interface RuntimeMediator
{
    /**
     *
     * @param setup
     * @param moduleId
     * @param device
     * @param deviceType
     * @param config
     * @return Object. null
     * @throws BGException
     */
    Object init( Setup setup, int moduleId, VoiceDevice device, VoiceDeviceType deviceType, ParameterMap config
    )
        throws BGException;

    /**
     * Map .
     */
    SortedMap<Date, List<VoiceRecord> > getNewRecords()
        throws BGException;
```


Настройка NAS-ов для Radius-запросов.

Для настройки NAS мы должны вначале добавить [тип устройства](#) и [устройство](#) данного типа(ссылки!).

В [типе устройства](#) может быть указан [Обработчик процессора протокола](#).

Монитор Поиск Устройства Ресурсы номеров Логи/пересчет Типы аккаунтов Справочники Конфигурация модуля

Устройства Типы устройств

Редактор

Основное

Название: Freeswitch

Класс-обработчик активации аккаунта: ---

Обработчик процессора протокола: ru.bitel.b.m.voice.dyn.radius.softswitch.ProtocolHand...

Обработчик логов: ---

Runtime обработчик логов: ---

Устройство является NAS-ом

Комментарий:

Конфигурация:

```
1 #по умолчанию cisco так как большинство параметров оттуда.
2 vendor.code=9
3
4 #radius.attr.calling.station.id.vendor=27880
5 #radius.attr.calling.station.id.code=2
6 #radius.attr.called.station.id.vendor=27880
7 #radius.attr.called.station.id.code=17
8
9
10 #radius.attr.identifier.vendor=27880
11 #radius.attr.identifier.code=26
12
13 radius.attr.identifier.vendor=9
14 radius.attr.identifier.code=24
15
16 #radius.attr.credit.time.vendor=9
17 #radius.attr.credit.time.code=102
18 #radius.attr.credit.time.code=103
19 #radius.attr.credit.amount.code=101
20
21
22 #radius.attr.call.type.code=
23
24 radius.messageAuthenticator.validate=0
```

Строк: 24 | Текущая строка: 24 | Текущий столбец: 39

Найти Найти далее

Сохранить Отмена

Управление динамическим кодом Voice

В настройке типа в первую очередь надо указать код вендора

```
# cisco c .
vendor.code=9
```

Это параметр по умолчанию(если не указан) имеет значение 9(Cisco).

Еще нужно указать ряд radius-параметров, которые используются в логике работы модуля . Все они имеют вид

```
radius.attr.X.vendor=
radius.attr.X.code=
```

Где в первой строке указывают vendor, во второй code атрибута, X - это имя атрибута.

Все эти атрибуты имеют значение по умолчанию:

```
#####H323-credit-time#####
# vendor.code(9-cisco)
radius.attr.credit.time.vendor=
# H323-credit-time Cisco
radius.attr.credit.time.code=102

#####H323-return-code#####
# vendor.code(9-cisco)
radius.attr.error.vendor=
# H323-return-code Cisco
radius.attr.error.code=103
```

```

##### H323-credit-amount#####
# vendor.code(9-cisco)
radius.attr.credit.amount.vendor=
# H323-credit-amount Cisco
radius.attr.credit.amount.code=101

##### Acct-Session-Id #####
# -1( )
radius.attr.identifier.vendor=-1
# Acct-Session-Id
radius.attr.identifier.code=44

##### Calling-Station-Id #####
# -1( )
radius.attr.calling.station.id.vendor=-1
# Calling-Station-Id
radius.attr.calling.station.id.code=31

##### Called-Station-Id#####
# -1( )
radius.attr.called.station.id.vendor=-1
# Called-Station-Id
radius.attr.called.station.id.code=30

##### H323_call_type#####
# vendor.code(9-cisco)
radius.attr.call.type.vendor=
# H323_call_type Cisco
radius.attr.call.type.code=27

##### H323-call-origin#####
# vendor.code(9-cisco)
radius.attr.call.origin.vendor=
# H323_call_origin Cisco
radius.attr.call.origin.code=26

##### ( ) #####
# vendor.code(9-cisco)
radius.attr.connect.time.vendor=
# ( ).
radius.attr.connect.time.code=

```

cisco. .

Вот пример рабочей конфигурации для FreeSwitch(в нестандартном варианте):

```

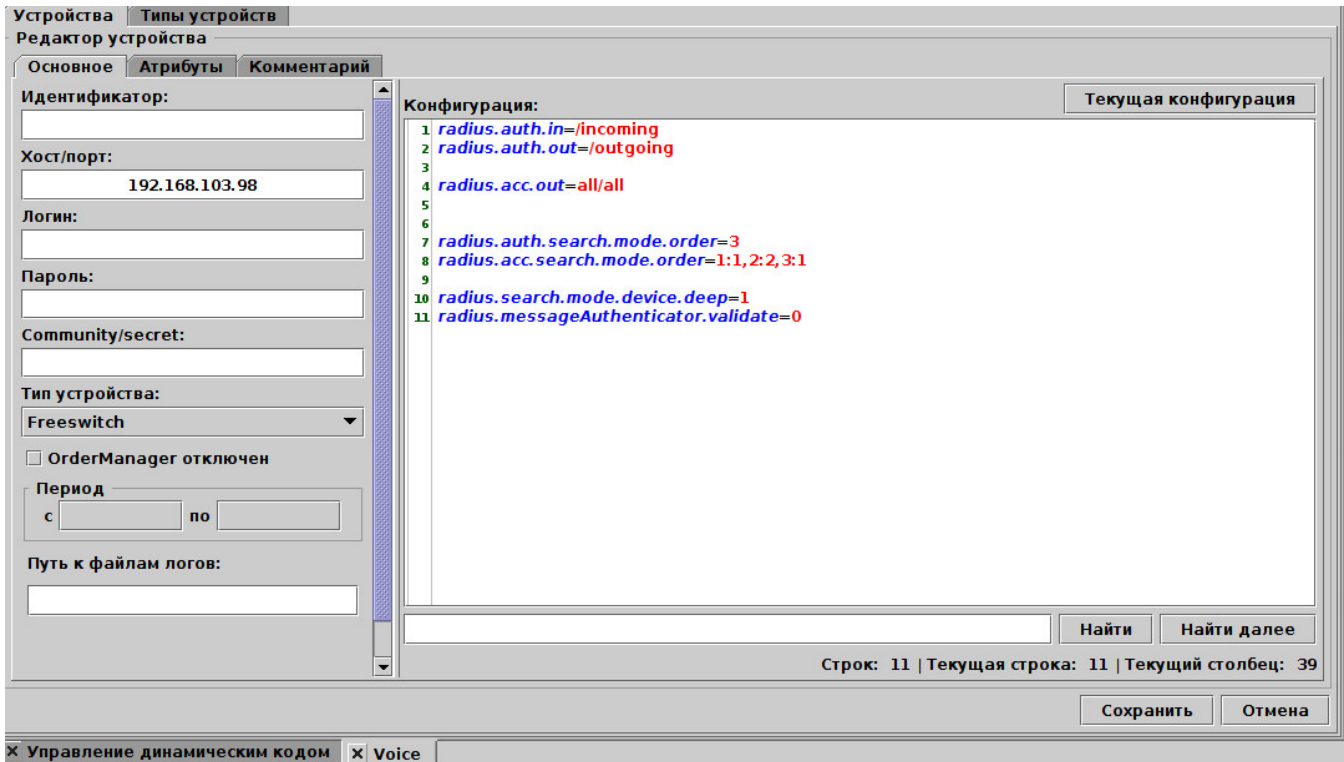
# -1 Cisco
radius.attr.identifier.vendor=9
#H323_conf_id
radius.attr.identifier.code=24

```

Как видно что в данном случае в качестве идентификатора вместо Acct-Session-Id использовали Cisco-ский атрибут H323-conf-id. Остальное все по умолчанию.

И указываем галочку **Устройство является NAS-ом.**

Далее заводим устройство данного типа :



Тут указываем или ip в поле Хост/порт или идентификатор. Поиск **NAS** происходит по Radius-Атрибутам:

- **NAS-IP-Address** - поиск по полю Хост/порт.
- **NAS-Identifier** - поиск по полю идентификатор.

Режимы поиска аккаунта

Вначале в **конфигурации модуля** мы задаем возможные режим поиска аккаунта в модуле.

```
radius.search.mode.pattern.< >.rule=< Radius->=<PHONE | LOGIN>
```

Где PHONE это номер телефона из аккаунта, а LOGIN - логин.

Например

```
radius.search.mode.pattern.1.rule=Calling-Station-id=PHONE
radius.search.mode.pattern.2.rule=Called-Station-id=PHONE
radius.search.mode.pattern.3.rule=User-Name=PHONE
radius.search.mode.pattern.4.rule=User-Name=LOGIN
```

Потом мы задаем уже в конфигурации устройства (или типа устройства) как эти режимы будут применяться на устройстве :

```
#
radius.auth.search.mode.order=exp[ ,exp]
#
radius.acct.search.mode.order=exp[ ,exp]
```

Где exp имеет вид

```
code[ :type]
```

code - это код режима, поиск из конфигурации, описанный выше. type - тип звонка (1 - исходящий, 2 - входящий). Тип звонка не обязателен (про него будет описано ниже).

Например:

```
radius.auth.search.mode.order=3
radius.acct.search.mode.order=1:1,2:2,3:1
```

По умолчанию происходит поиск аккаунтов на **NAS**, которые удовлетворяют заданным критериям. И так же на устройстве, которое является родительским для NAS - это позволяет сделать режим, когда вы не хотите указывать NAS явно на абоненте, т.е абонент может выходить с любого из NAS-ов, тогда имеет смысл объединить все NAS в отдельную папку в дереве устройств и указывать эту папку как `device.const` в типе аккаунта.

Отдельно можно указать чтобы аккаунты искались так же до дочерних устройствах(расположенные ниже в [дереве устройств](#)) данного NAS, это настройка указывается в типе устройства :

```
# . 1- , 0( ) -.
radius.search.mode.device.deep=1
```

Она означает что надо искать аккаунт не только на устройстве NAS, но и сразу его потомков.

Тут есть отличие как работает поиск в access и accounting. Для access мы просто находим один аккаунт, который удовлетворяет первый совпадающему в списке режиму. Далее поиск прекращается, проверяется баланс и т.п для этого аккаунта чтобы выдать ему access или reject.

Для accounting ищутся все аккаунты удовлетворяющие заданным в списке режимам и для каждого из них создается отдельная сессия. Это сделано для того, когда в случае звонка абонент-абонент, создать исходящую сессию на одном абоненте и исходящую на другом(подробнее о определении типа звона написано ниже).

Определение типа звонка

Типа звонка (входящий или исходящий) определяется в следующем порядке (каждый способ может переопределить предыдущий):

1. Определяем типа звонка на основе атрибутов.
2. Скрипт [обработки процессора протокола](#).
3. Направление указанное в настройке поиска account-a.

1-ый способ. Для определения типа звонка (входящий или исходящий) на основе атрибутов radius-запроса(1-ый способ) используются такие настройки :

```
#
radius.auth.in=voip/originate
#
radius.auth.out=voip/answer
# accounting-
radius.acct.in=voip/originate
# accounting-
radius.acct.out=voip/answer
```

Для определения типа используются атрибуты **h323-call-type** и **h323-call-origin**(эти параметры можно поменять на другие в настройке, описанной выше) из RADIUS-запроса. Значения этих атрибутов, соответствующие каждому типу звонка, необходимо указать через дробь и в нижнем регистре (даже если в запросе указан верхний регистр). В этом случае авторизационные запросы с атрибутами **h323-call-type=Voip h323-call-origin=originate** будут считаться исходящими, **h323-call-type=Voip h323-call-origin=answer** входящими. Еще есть специальное значение **all**. Например вот так можно указать что все вызовы идут как исходящие

```
radius.auth.out=all/all
radius.acct.out=all/all
```

Так же любой из атрибутов может быть пустым. Например

```
radius.auth.in=/incoming
```

2-ой способ. Так же тип звонка можно определить в [обработчике протокола](#) с помощью установки опции `VoiceNas.CALL_TYPE`. Пример обработчика, где все звонки делаются исходящими:

```

package ru.bitel;
import ru.bitel.bgbilling.kernel.network.radius.RadiusPacket;
import ru.bitel.bgbilling.modules.voice.access.om.ProtocolHandler;
import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDevice;
import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDeviceType;
import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceSession;
import ru.bitel.bgbilling.modules.voice.radius.VoiceNas;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.ParameterMap;
import ru.bitel.common.sql.ConnectionSet;
public class Prot3
    implements ProtocolHandler
{
    @Override
    public void preprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception
    {
        //
        request.setOption( VoiceNas.CALL_TYPE, VoiceSession.CALL_TYPE_OUTGOING );
    }
    @Override
    public void preprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception
    {
        //
        request.setOption( VoiceNas.CALL_TYPE, VoiceSession.CALL_TYPE_OUTGOING );
    }
    @Override
    public void init( Setup setup1, int int2, VoiceDevice voiceDevice3, VoiceDeviceType voiceDeviceType4,
ParameterMap parameterMap5 )
        throws Exception
    {
    }
    @Override
    public void postprocessAccountingRequest( RadiusPacket radiusPacket1, RadiusPacket radiusPacket2,
ConnectionSet connectionSet3 )
        throws Exception
    {
    }
    @Override
    public void postprocessAccessRequest( RadiusPacket radiusPacket1, RadiusPacket radiusPacket2,
ConnectionSet connectionSet3 )
        throws Exception
    {
    }
}

```

3-тий способ. При указании режимов поиска на устройстве(описано выше) мы можем опционально задать тип звонка(1 - исходящий, 2 - входящий). В этом случае если аккаунт будет найден для указанной пары **режим:тип звонка**, то тип звонка возьмется из этой пары.

```
radius.acct.search.mode.order=1:1,2:2
```

В данном примере для 1-го режима поиска звонок помечается как исходящий, а для второго как входящий. При этом в конфигурации модуля эти режимы могут быть определены так:

```
radius.search.mode.pattern.1.rule=Calling-Station-id=PHONE
radius.search.mode.pattern.2.rule=Called-Station-id=PHONE
```

Если происходит звонок от одного абонента провайдера к другому, то он найдет 2 разных аккаунта, на один добавится исходящий звонок, на второй - входящий.

Обработчик процессора протокола модуля Voice

Обработчик протокола позволяет делать некоторую дополнительную логику при обработке Radius-запросов - например, изменить RADIUS-запрос перед его обработкой системой или установить дополнительные опции, которые поменяют логику обработки запроса.

Обработчик создается в динамическом коде(ссылка) и имеет следующий интерфейс:

```
/**
 *
 * @see RadiusProtocolHandler
 *
 */
public interface ProtocolHandler
    extends RadiusProtocolHandler
{
    public void init( Setup setup, int moduleId, VoiceDevice voiceDevice, VoiceDeviceType voiceDeviceType,
ParameterMap config )
        throws Exception;
}
```

init- это метод инициализации, который вызывается при создании обработчика.

Так же обработчик расширяет следующий интерфейс:

```
/**
 * RADIUS-
 */
public interface RadiusProtocolHandler
{
    /**
     * RADIUS- Access-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void preprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;
    /**
     * RADIUS- Access-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void postprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;
    /**
     * RADIUS- Accounting-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void preprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception;
    /**
     * RADIUS- Accounting-Request
     * @param request
     * @param response
     * @param connectionSet
     * @throws Exception
     */
    public void postprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
```



```
connectionSet )  
    throws Exception;  
}
```

Параметры, заданные скриптом, более приоритетные, чем взятые из конфигурации. Тут можно выставить такие параметры:

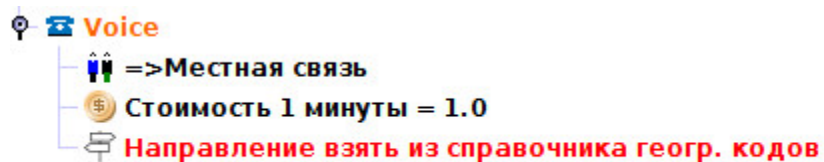
- Тип звонка - **request.setOption(VoiceNas.CALL_TYPE, VoiceSession.CALL_TYPE_OUTGOING);**
- Аккаунт оператора - **request.setOption(VoiceNas.OPERATOR_ACCOUNT, accountId);**

Тарификация модуля Voice.

Общие принцип работы тариффикатора:

1. Поиск стоимости идет от общего к частному. Т. е. более общие префиксы например, должны стоять в дереве выше, более детализированных. Таким образом в сначала стоимость возьмется из более общей ветки, а затем при нахождение более детализированной, перетрется из нее.
2. Узел **"Использовать карту зон"** должен быть выше узлов **"Зона"**
3. Узел **"Параметры тарификации"** должен быть выше узлов в которых эти параметры должны быть использованы

Рассмотрим простейший тариф модуля Voice.



Это несколько искусственный тариф для демонстрации минимальных требований к тарифу. Любой тариф должен содержать:

- **Услугу**
- **Стоимость**
- **Направление**

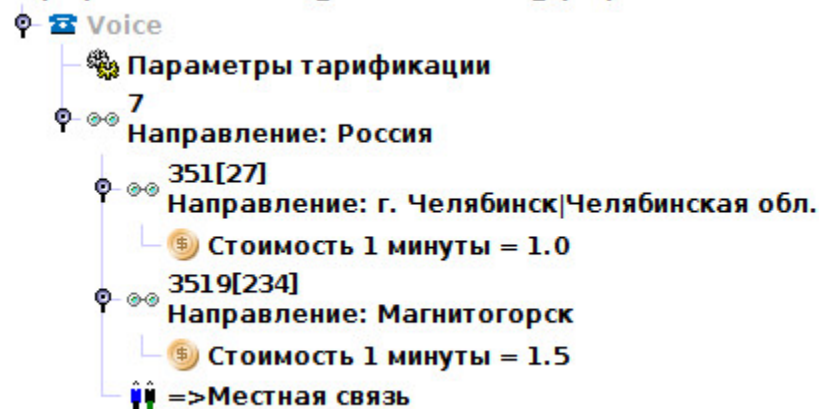
Направление в данном случае задается узлом **Установить направление** и может быть выбрано конкретным из списка или есть возможность чтобы направление искалось из справочника направлений. Так же направление могут подставлять явно или не явно другие узлы(часть префикса, карты цен, карты зон).

Тарифы на местную связь модуля Voice.

Тариф, содержащий поддерево с тарифами на местную связь, должен располагаться в договоре с минимальной позицией. Т.е. при тарификации звонка поинта он должен просматриваться первым. Следом за ним могут следовать тарифы зоновой и МГМН-связи.

Рассмотрим простейший тариф на местную связь.

Тарифный план 'voice_местная связь_префикс'

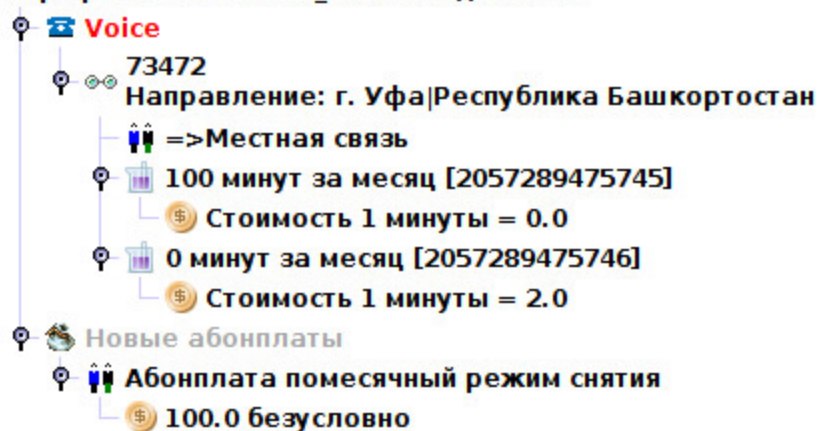


Для разбора номера используется узел **Часть префикса**.

Узел **Установка услуги** помещает в ответ тарифного запроса услугу данного звонка. Стоимость минуты может быть заключена в фильтр по типу времени, период. Узел **Параметры тарификации** также может быть добавлен в разные части префикса с различными режимами округления.

Вот пример тарифа с диапазоном наработки :

Тарифный план 'voice_Местная диапазон'



Тут первые 100 минут идут бесплатно(за них берет 100 рублей модуль прау), а начиная со 101-ой минуты цена по 2 рубля за минуту.

Тарифы на МГМН и зонную связь.

Более сложными в реализации являются тарифы МГМН-операторов. Тарифы каждого оператора должны быть реализованы в отдельном плане. Существуют следующие основные способы тарификации:

- По префиксам
- По карте цен
- По зонам

Тарификация по префиксам модуля Voice

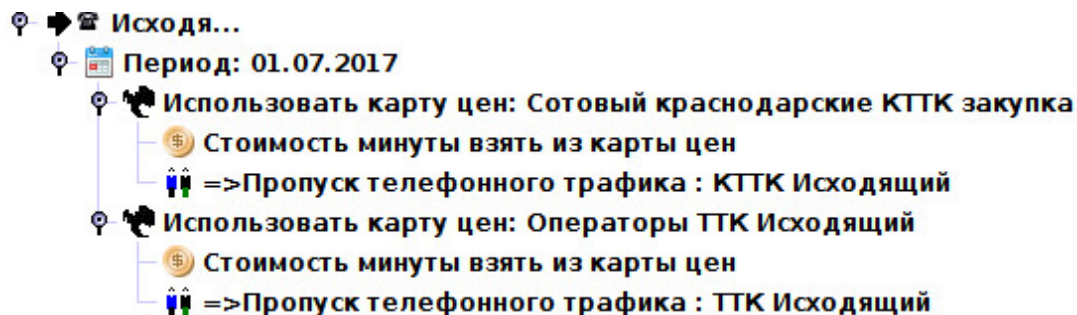
Тарификация по префиксам предполагает внесение в тарифный план дерева префиксов для разбора звонков. Логика тарифа полностью идентична приведённым выше тарифам на местную связь. Этот способ имеет смысл использовать только если у вас немного префиксов и цен.

При большом количестве префиксов целесообразнее использовать карту цен или карту зон.

Тарификация по карте цен модуля Voice

В основу этой тарификации ложится карта цена, соотносящая коды и правильные диапазоны из карты с ценами. В тарифном плане с помощью узла **Карта цен** указывается используемая карта. Если префикс найден в этой карте, то звонку также соотносится направление из географических кодов и **выполняются дочерние узлы**. Чтобы назначить цену из этой карты, внутри должен быть узел **Стоимости** с параметром "взять из карты цен".

Вот пример тарифа с картой цен:

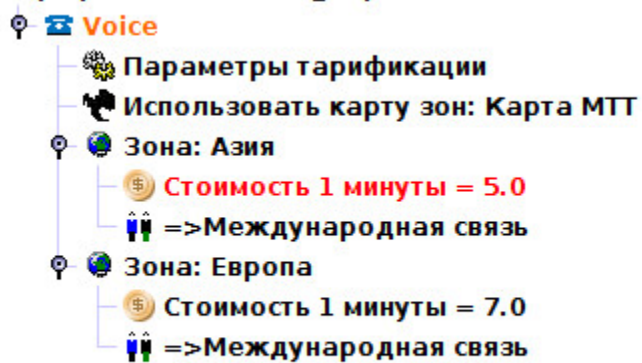


В данном тарифе стоит еще дополнительный узел **Тип Звонка**. Он пропускает только исходящие звонки.

Тарификация по зонам модуля Voice.

В основу зоновой тарификации ложится карта зон, соотносящая коды из справочника географических кодов с зонами равной цены. Про построение карт зон вы можете почитать [ранее](#). В тарифном плане с помощью узла **Использовать карту зон** указывается используемая карта и далее добавляются узлы **Зона** с указанием в каждой стоимости минуты и услуги.

Тарифный план 'voice_Карта зон'

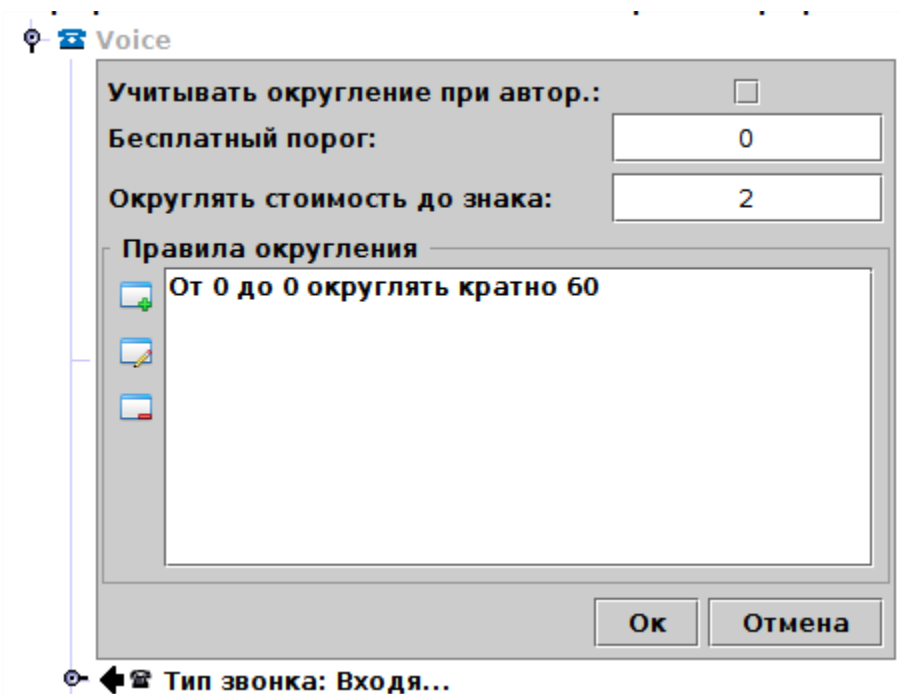
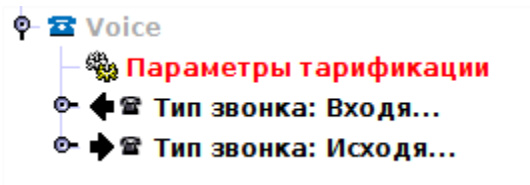


Тарификация с несколькими МГМН-операторами в модуле Voice

Импорт и экспорт тарифных планов модуля Voice

Специфичные тарифные узлы модуля Voice

Узел "Параметры тарификации"



На рисунке выше правило означает, что звонок с 0 секунд и до конца округлять кратно 60 секунд

Можно например, задать, что первую минуту с 0 до 60 секунд округлять до 60 секунд, а далее посекундная тарификация и т.д.

Клиентские отчеты модуля Voice

Основной отчет это отчет по сессиям.

Сессии: 21973 Время [сек.]: 860:45:43 [3098743]
Сумма: 0.00000

СессияID	Аккаунт	Тип	Направление	Алле...	ВреID	Старт	Стоп	Активность	Длительность	Стоимость	Час
6801731	78612...	<<	Входящий звонок	7861...	7861202...	01.09.2017 00:05:11	01.09.2017 00:05:21		00:00:10 [10]	0.00	01.09.2017 00:00:00
6801732	78612...	>>	Краснодар	7861...	7861279...	01.09.2017 00:09:29	01.09.2017 00:09:29		0.00	0.00	01.09.2017 00:00:00
6801736	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 00:13:05	01.09.2017 00:13:23		00:00:18 [18]	0.00	01.09.2017 00:00:00
6801738	78612...	<<	Входящий звонок	7989...	7861202...	01.09.2017 00:13:20	01.09.2017 00:13:24		00:00:04 [4]	0.00	01.09.2017 00:00:00
6801749	78612...	<<	Входящий звонок	7961...	7861202...	01.09.2017 00:29:17	01.09.2017 00:29:39		00:00:22 [22]	0.00	01.09.2017 00:00:00
6801753	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 00:36:56	01.09.2017 00:37:26		00:00:30 [30]	0.00	01.09.2017 00:00:00
6801755	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 00:37:47	01.09.2017 00:39:19		00:01:32 [92]	0.00	01.09.2017 00:00:00
6801767	78612...	<<	Входящий звонок	7988...	7861202...	01.09.2017 01:04:16	01.09.2017 01:05:23		00:01:07 [67]	0.00	01.09.2017 00:00:00
6801771	78612...	<<	Входящий звонок	7962...	7861202...	01.09.2017 01:18:20	01.09.2017 01:19:09		00:00:49 [49]	0.00	01.09.2017 00:00:00
6801793	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 01:40:40	01.09.2017 01:42:08		00:01:28 [88]	0.00	01.09.2017 00:00:00
6801797	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 01:43:54	01.09.2017 01:44:41		00:00:47 [47]	0.00	01.09.2017 00:00:00
6801795	78612...	>>	Краснодар	7861...	7861224...	01.09.2017 01:44:15	01.09.2017 01:44:15		0.00	0.00	01.09.2017 00:00:00
6801796	78612...	>>	Краснодар	7861...	7861279...	01.09.2017 01:44:28	01.09.2017 01:44:28		0.00	0.00	01.09.2017 00:00:00
6801801	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 02:18:33	01.09.2017 02:18:52		00:00:19 [19]	0.00	01.09.2017 00:00:00
6801806	78612...	<<	Входящий звонок	7928...	7861202...	01.09.2017 02:32:43	01.09.2017 02:33:49		00:01:06 [66]	0.00	01.09.2017 00:00:00
6801818	78612...	>>	Краснодар	7861...	7861279...	01.09.2017 02:46:00	01.09.2017 02:46:00		0.00	0.00	01.09.2017 00:00:00
6801826	78612...	>>	Краснодар	7861...	7861279...	01.09.2017 03:36:12	01.09.2017 03:36:12		0.00	0.00	01.09.2017 00:00:00
6801844	78612...	<<	Входящий звонок	7961...	7861202...	01.09.2017 04:43:09	01.09.2017 04:43:34		00:00:25 [25]	0.00	01.09.2017 00:00:00
6801849	78612...	>>	Входящий звонок	7961...	7861202...	01.09.2017 04:43:50	01.09.2017 04:46:31		00:02:41 [161]	0.00	01.09.2017 00:00:00
6801857	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 05:07:43	01.09.2017 05:07:43		0.00	0.00	01.09.2017 00:00:00
6801865	78612...	>>	Краснодар	7861...	7861279...	01.09.2017 05:35:09	01.09.2017 05:35:12		00:00:03 [3]	0.00	01.09.2017 00:00:00
6801878	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 05:41:36	01.09.2017 05:45:36		00:04:00 [240]	0.00	01.09.2017 00:00:00
6801880	78612...	<<	Входящий звонок	7918...	7861202...	01.09.2017 06:12:52	01.09.2017 06:12:56		00:00:04 [4]	0.00	01.09.2017 00:00:00
6801893	78612...	<<	Входящий звонок	7989...	7861202...	01.09.2017 06:30:56	01.09.2017 06:33:25		00:02:29 [149]	0.00	01.09.2017 00:00:00
6801898	78612...	>>	Краснодар	7861...	7861224...	01.09.2017 06:31:36	01.09.2017 06:31:36		0.00	0.00	01.09.2017 00:00:00

Отчет показывается данные в 2-х режимах:

- Завершенные сессии
- Текущие сессии

Режим переключается соответствующей кнопкой сверху.

Отчет можно отфильтровать по месяцу(плюс дни) и аккаунту. Так же можно включить дополнительные столбцы с наработкой по услуге.

Отчет по аккаунтам

Отчет доступен на вкладке **Аккаунты**. Выводит данные с группировкой по аккаунту.

Аккаунт	Количество	Длительность	Стоимость
786120...	27	00:29:19 [1759]	0.00
786120...	103	01:39:08 [5948]	0.00
786120...	216	04:13:46 [15226]	0.00
786120...	85	01:00:41 [3641]	0.00
786120...	909	09:55:48 [35748]	0.00
786120...	2150	56:46:25 [204385]	0.00
786120...	28356	786:35:48 [2831748]	0.00
786120...	10	00:04:48 [288]	0.00
786120...	117	0.00	0.00

Отчет по направлениям

Отчет доступен на вкладке **Направления**. Выводит данные с группировкой по аккаунту и направлению.

Таблица	Аккаунты	Направления	Услуги			
Аккаунт	Направление			Количество	Длительность	Стоимость
78612...	Краснодар			2		0 0.00
78612...	Входящий звонок			25	00:29:19 [1759]	0.00
78612...	Краснодар			61	00:49:21 [2961]	0.00
78612...	Входящий звонок			37	00:39:21 [2361]	0.00
78612...	г. Краснодар Юг-линк			5	00:10:26 [626]	0.00
78612...	Краснодар			113	02:08:26 [7706]	0.00
78612...	Входящий звонок			92	01:54:48 [6888]	0.00
78612...	г. Краснодар Юг-линк			10	00:09:43 [583]	0.00
78612...	Узел ДО (Открытие системы: ЦУЭС0038)			1	00:00:49 [49]	0.00
78612...	Краснодар			43	00:30:15 [1815]	0.00
78612...	Входящий звонок			40	00:28:44 [1724]	0.00
78612...	г. Краснодар Юг-линк			2	00:01:42 [102]	0.00
78612...	Краснодар			4	00:01:03 [63]	0.00
78612...	Входящий звонок			903	09:54:43 [35683]	0.00
78612...	г. Краснодар Юг-линк			2	00:00:02 [2]	0.00
78612...	Краснодар			14	00:14:51 [891]	0.00
78612...	Входящий звонок			2130	56:17:39 [202659]	0.00
78612...	г. Краснодар Юг-линк			6	00:13:55 [835]	0.00
78612...	Краснодар			1140	11:42:06 [42126]	0.00
78612...	Входящий звонок			27179	773:50:04 [2785804]	0.00
78612...	Бесплатные вызовы			6	00:02:47 [167]	0.00
78612...	г. Краснодар Юг-линк			26	00:56:25 [3385]	0.00
78612...	Узел ДО (Мегафо: ЦУЭС0006)			5	00:04:26 [266]	0.00
78612...	Входящий звонок			10	00:04:48 [288]	0.00
78612...	Входящий звонок			117		0.00

Отчет по услугам

Отчет доступен на вкладке **Услуги**. Выводит данные с группировкой по аккаунту и услуге.

Таблица	Аккаунты	Направления	Услуги			
Аккаунт	Услуга			Количество	Длительность	Стоимость
7861202...				2		0 0.00
7861202...	Местная связь Юг-линк			25	00:29:19 [1759]	0.00
7861202...				23		0.00
7861202...	Местная связь Юг-линк			80	01:39:08 [5948]	0.00
7861202...				51		0.00
7861202...	Местная связь Юг-линк			165	04:13:46 [15226]	0.00
7861202...				21		0.00
7861202...	Местная связь Юг-линк			64	01:00:41 [3641]	0.00
7861202...				56		0.00
7861202...	Местная связь Юг-линк			853	09:55:48 [35748]	0.00
7861202...				11		0.00
7861202...	Местная связь Юг-линк			2139	56:46:25 [204385]	0.00
7861202...				673		0.00
7861202...	Местная связь Юг-линк			27683	786:35:48 [2831748]	0.00
7861202...				7		0.00
7861202...	Местная связь Юг-линк			3	00:04:48 [288]	0.00
7861202...				117		0.00

Монитор соединений модуля Voice

В мониторе вы можете смотреть состояние текущих сессий, лог сессии в прошлом и ошибки авторизации.

SessID	Договор	Аккаунт	Устройство	Тип	Направление	С номера/на номер	Начало	Окончание	Активность	Длительно...	Стоимость
5770532			Freewitch: [192.168.103.98]	<<	Входящий звонок		25.08.2017 11:18:...			00:00:00 [0]	0.00
6031554			Freewitch: [192.168.103.98]	<<	Входящий звонок		29.08.2017 09:54:...			00:00:00 [0]	0.00
6244253			Freewitch: [192.168.103.98]	>>	Краснодар		31.08.2017 09:59:...			00:00:00 [0]	0.00
6244254			Freewitch: [192.168.103.98]	>>	Краснодарский край		31.08.2017 09:59:...			00:00:00 [0]	0.00

Монитор нужен для работы с сессиями, которые создаются по Radius-запросам. Тут вы можете смотреть в следующих режимах:

- **Текущие сессии** - Это сессии которые еще не завершились.
- **Логи сессий** - Это завершенные сессии.
- **Ошибки** - Ошибка авторизации.

Каждый режим включается соответствующей кнопкой сверху.

В режиме логов и ошибок появляется дополнительная фильтрация по часам и дате.

Режим логов:

SessID	Договор	Аккаунт	Устройство	Тип	Направление	С номера/на номер	Начало	Окончание	Активность	Длительно...	Стоимость
87425932			Freewitch: [192.168.103.98]	>>	Краснодарский край		12.09.2017 15:03:...	12.09.2017 15:04:...	12.09.2017 15:04:...	00:00:56 [5]	3.73333
87425952			Freewitch: [192.168.103.98]	>>	Краснодарский край		12.09.2017 15:03:...	12.09.2017 15:04:...	12.09.2017 15:04:...	00:00:56 [5]	3.73333
87425940			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:03:...	12.09.2017 15:06:...	12.09.2017 15:06:...	00:02:10 [1]	0.00
87426010			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:03:...	12.09.2017 15:06:...	12.09.2017 15:06:...	00:02:10 [1]	0.00
87425962			Freewitch: [192.168.103.98]	>>	г. Краснодар Юг-линик		12.09.2017 15:04:...	12.09.2017 15:04:...	12.09.2017 15:04:...	00:00:16 [1]	0.00
87425972			Freewitch: [192.168.103.98]	>>	г. Краснодар Юг-линик		12.09.2017 15:04:...	12.09.2017 15:04:...	12.09.2017 15:04:...	00:00:16 [1]	0.00
87425982			Freewitch: [192.168.103.98]	>>	г. Краснодар Юг-линик		12.09.2017 15:04:...	12.09.2017 15:06:...	12.09.2017 15:06:...	00:01:33 [9]	0.00
87425992			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:05:...	12.09.2017 15:06:...	12.09.2017 15:06:...	00:00:00 [0]	0.00
87426002			Freewitch: [192.168.103.98]	>>	г. Москва		12.09.2017 15:06:...	12.09.2017 15:06:...	12.09.2017 15:06:...	00:00:47 [4]	1.95833
87426032			Freewitch: [192.168.103.98]	>>	Free (fw) - Аудиотеле		12.09.2017 15:06:...	12.09.2017 15:06:...	12.09.2017 15:06:...	00:00:00 [0]	0.00
87426042			Freewitch: [192.168.103.98]	>>	Краснодар		12.09.2017 15:06:...	12.09.2017 15:07:...	12.09.2017 15:07:...	00:00:00 [0]	0.00
87426052			Freewitch: [192.168.103.98]	>>	г. Краснодар Юг-линик		12.09.2017 15:07:...	12.09.2017 15:08:...	12.09.2017 15:08:...	00:01:24 [8]	0.00
87426062			Freewitch: [192.168.103.98]	>>	Краснодар		12.09.2017 15:07:...	12.09.2017 15:07:...	12.09.2017 15:07:...	00:00:00 [0]	0.00
87426072			Freewitch: [192.168.103.98]	>>	Краснодарский край		12.09.2017 15:09:...	12.09.2017 15:09:...	12.09.2017 15:09:...	00:00:07 [7]	0.46667
87426080			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:09:...	12.09.2017 15:10:...	12.09.2017 15:10:...	00:00:48 [4]	0.00
87426092			Freewitch: [192.168.103.98]	>>	Краснодарский край		12.09.2017 15:09:...	12.09.2017 15:09:...	12.09.2017 15:09:...	00:00:07 [7]	0.46667
87426101			Freewitch: [192.168.103.98]	>>	РОССИЯ МОБ Мегафон		12.09.2017 15:09:...	12.09.2017 15:11:...	12.09.2017 15:11:...	00:01:15 [7]	5.00
87426112			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:10:...	12.09.2017 15:10:...	00:00:00 [0]	0.00
87426122			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:16:...	12.09.2017 15:16:...	00:06:05 [3]	0.00
87426132			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:10:...	12.09.2017 15:10:...	00:00:00 [0]	0.00
87426140			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:12:...	12.09.2017 15:12:...	00:01:32 [9]	0.00
87426152			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:11:...	12.09.2017 15:11:...	00:00:54 [5]	0.00
87426192			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:11:...	12.09.2017 15:11:...	00:00:54 [5]	0.00
87426230			Freewitch: [192.168.103.98]	<<	Входящий звонок		12.09.2017 15:10:...	12.09.2017 15:12:...	12.09.2017 15:12:...	00:01:32 [9]	0.00

Режим ошибок:

The screenshot shows a web-based interface for network management. The main window is titled 'Режим' (Mode) and is set to 'Ошибки' (Errors). The interface includes a navigation menu on the left with options like 'Монитор', 'Поиск', 'Устройства', 'Ресурсы номеров', 'Лог/пересчет', 'Типы аккаунтов', 'Справочники', and 'Конфигурация модуля'. Below the menu, there are sections for 'Параметры' (Parameters) with a 'Выберите договор(ы)...' (Select contract(s)...) dropdown, and 'Логин:' (Login:). The main area displays a table with columns: 'Договор' (Contract), 'Устройство' (Device), 'Аккаунт' (Account), 'Время' (Time), and 'Ошибка' (Error). The table contains three rows of data, all showing 'Аккаунт не найден' (Account not found) errors. At the bottom left, there is a 'Сброс' (Reset) button and a taskbar with 'Планировщик заданий' (Task Scheduler) and 'Voice' icons.

Договор	Устройство	Аккаунт	Время	Ошибка
	Freewitch: [192.168.103.98]		12.09.2017 15:04:59	[1] Аккаунт не найден
	Freewitch: [192.168.103.98]		12.09.2017 15:04:56	[1] Аккаунт не найден
	Freewitch: [192.168.103.98]		12.09.2017 15:00:43	[1] Аккаунт не найден

Во всех режимах возможна фильтрация результата по

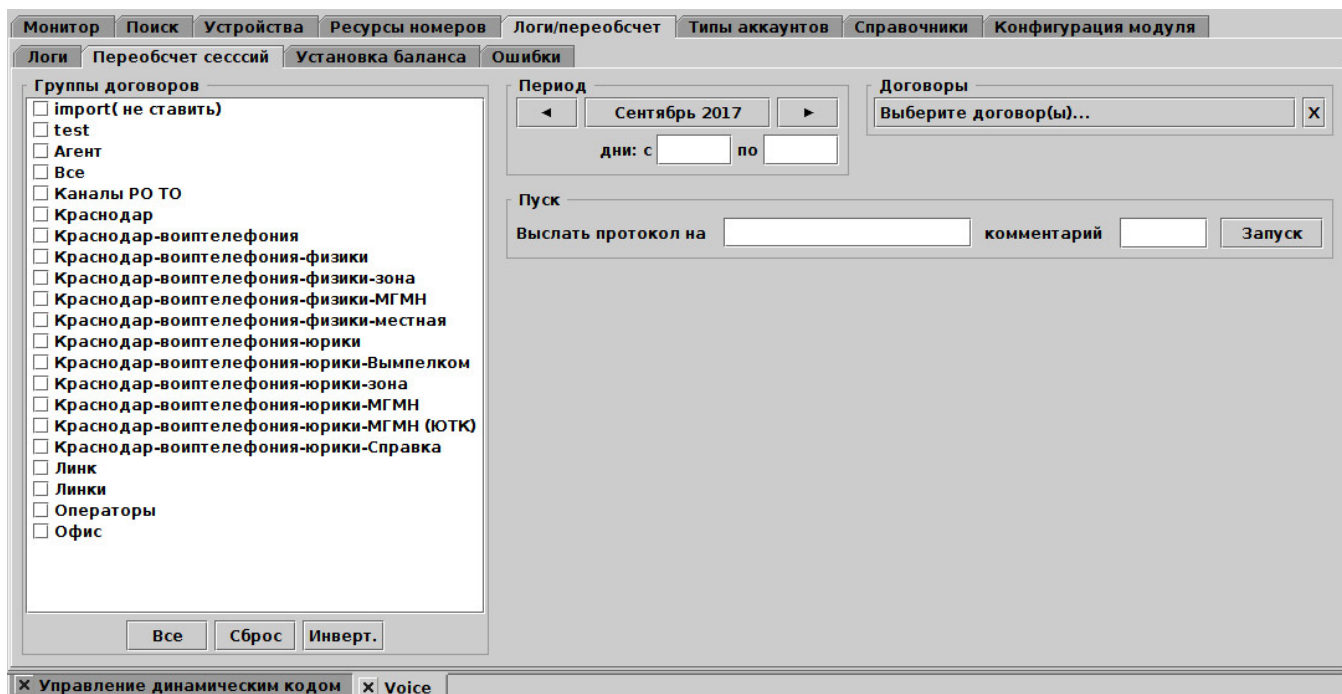
- **Договору** - можно выбрать как конкретный договор, так и задать маску номера договора
- **Логину** - задается маска логина
- **Устройству**

При нажатии правой кнопкой на любой строке доступны такие пункты

- **Получить RADIUS лог**
- **Открыть договор**
- **Завершить (зависшею) сессию** (только в режиме отображения текущих сессий).

Переобсчет сессий.

Переобсчет сессий можно осуществить на следующей вкладке:



Переобсчет сессии может быть необходим когда что-то исправляется в тарифе (например цена, или направление) и нужно исправить это в сессиях. Для работы переобсчета необходим запущенный планировщик биллинга.

Учет операторского трафика модуля Voice

Учет операторского трафика для CDR.

Для учета операторского трафика по портам и префиксам создается отдельный договор для оператора и нем создается [аккаунт оператора](#).

id	Имя	Устройство	Период	...
19064	Исходящие С 5-6-й порт на МР24	М200: [192.168.100.70]	01.07.2017-...	▲
19067	Входящие С 5-6-й порт на МР24	М200: [192.168.100.70]	01.07.2017-...	

Аккаунты клиента | Аккаунты оператора

Редактор операторского аккаунта

Тип аккаунта: | Наименование: | Тип звонка: | Период: с по

Устройство: <<<

С портов: | На порты:

С портов Regexp: | На порты Regexp:

Regexp A: | No Regexp A:

Regexp B: | No Regexp B:

На договор вешается тарифный план.

Пример тарифного плана с картой цен:

Гарифный план 'Операторы ТТК'

Voice

Параметры тарификации

- Входя...
- Направление взять из справочника геогр. кодов
- Стоимость 1 минуты = [redacted]
- => Пропуск телефонного трафика : ТТК Входящий

Исходя...

- Период: 01.07.2017
- Использовать карту цен: Сотовый краснодарские КТТК закупк.
- Стоимость минуты взять из карты цен
- => Пропуск телефонного трафика : КТТК Исходящий
- Использовать карту цен: Операторы ТТК Исходящий
- Стоимость минуты взять из карты цен
- => Пропуск телефонного трафика : ТТК Исходящий

В результате на этом договоре мы получаем отчет с сессиями оператора:

Завершенные		Текущие									
Сессия	Аккаунт	Тип	Направлен...	Ава164	Всв164	Старт	Стоп	Активность	Длительность	Стоимость	
6801743	Исходящи...	>>	Краснодар...	79182...	79892721...	01.09.2017 00:13:23	01.09.2017 00:18:17		00:04:54 [294]	7.35	01.09.2
6801741	Исходящи...	>>	Краснодар...	78612...	79183830...	01.09.2017 00:15:53	01.09.2017 00:16:04		00:00:11 [11]	0.275	01.09.2
6801748	Исходящи...	>>	Краснодар...	79189...	79892721...	01.09.2017 00:23:44	01.09.2017 00:26:59		00:03:15 [195]	4.875	01.09.2
6801745	Исходящи...	>>	Краснодар...	78612...	79676517...	01.09.2017 00:25:29	01.09.2017 00:29:57		00:04:28 [268]	6.70	01.09.2
6801751	Исходящи...	>>	Краснодар...	79384...	79892721...	01.09.2017 00:30:23	01.09.2017 00:31:00		00:00:37 [37]	0.925	01.09.2
6801758	Исходящи...	>>	Краснодар...	78612...	79083815...	01.09.2017 00:39:29	01.09.2017 00:39:45		00:00:16 [16]	0.40	01.09.2
6801760	Исходящи...	>>	Краснодар...	79604...	79892721...	01.09.2017 00:44:11	01.09.2017 00:45:14		00:01:03 [63]	1.575	01.09.2
6801765	Исходящи...	>>	Краснодар...	78612...	79180699...	01.09.2017 00:56:03	01.09.2017 00:56:14		00:00:11 [11]	0.275	01.09.2
6801774	Исходящи...	>>	Краснодар...	78612...	79034511...	01.09.2017 01:31:45	01.09.2017 01:31:52		00:00:07 [7]	0.175	01.09.2
6801780	Исходящи...	>>	Краснодар...	78612...	79897743...	01.09.2017 01:33:50	01.09.2017 01:33:58		00:00:08 [8]	0.20	01.09.2
6801782	Исходящи...	>>	Краснодар...	78612...	79528258...	01.09.2017 01:34:20	01.09.2017 01:34:30		00:00:10 [10]	0.25	01.09.2
6801784	Исходящи...	>>	Краснодар...	78612...	79180699...	01.09.2017 01:34:54	01.09.2017 01:34:54			0.00	01.09.2
6801786	Исходящи...	>>	Краснодар...	78612...	79086703...	01.09.2017 01:35:45	01.09.2017 01:35:55		00:00:10 [10]	0.25	01.09.2
6801788	Исходящи...	>>	Краснодар...	78612...	79898078...	01.09.2017 01:36:23	01.09.2017 01:36:31		00:00:08 [8]	0.20	01.09.2
6801790	Исходящи...	>>	Краснодар...	78612...	79530805...	01.09.2017 01:36:58	01.09.2017 01:37:07		00:00:09 [9]	0.225	01.09.2
6801792	Исходящи...	>>	Краснодар...	78612...	79034511...	01.09.2017 01:37:35	01.09.2017 01:38:58		00:01:23 [83]	2.075	01.09.2
6801799	Исходящи...	>>	Краснодар...	78612...	79628773...	01.09.2017 01:56:32	01.09.2017 01:57:37		00:01:05 [65]	1.625	01.09.2
6801800	Исходящи...	>>	Краснодар...	78612...	79628773...	01.09.2017 02:01:31	01.09.2017 02:02:03		00:00:32 [32]	0.80	01.09.2
6801808	Исходящи...	>>	РОССИЯ МО...	78612...	79284649...	01.09.2017 02:34:10	01.09.2017 02:34:10			0.00	01.09.2
6801810	Исходящи...	>>	РОССИЯ МО...	78612...	79284647...	01.09.2017 02:34:48	01.09.2017 02:34:48			0.00	01.09.2
6801812	Исходящи...	>>	РОССИЯ МО...	78612...	79284647...	01.09.2017 02:35:17	01.09.2017 02:35:17			0.00	01.09.2
6801816	Исходящи...	>>	РОССИЯ МО...	78612...	79284647...	01.09.2017 02:36:00	01.09.2017 02:36:17		00:00:17 [17]	0.425	01.09.2
6801823	Исходящи...	>>	Краснодар...	78612...	79180699...	01.09.2017 02:59:33	01.09.2017 02:59:45		00:00:12 [12]	0.30	01.09.2
6801825	Исходящи...	>>	Краснодар...	78612...	79180699...	01.09.2017 03:32:01	01.09.2017 03:35:00		00:02:58 [179]	4.475	01.09.2
6801828	Исходящи...	>>	Краснодар...	78612...	79034511...	01.09.2017 03:35:40	01.09.2017 03:36:42		00:01:02 [62]	1.55	01.09.2

Учет операторского трафика для Radius

Тарификация оператора по отдельном Radius-запросу

Для учета операторского трафика по протоколу Radius создается отдельный договор для оператора и нем создается [аккаунт](#).

Аккаунт	Устройство	Тип	Номер	Период	Статус	Состояние	Комментарий	id	...
test_operator	First: ID3 [local... операторский	операторский		0 13.02.2017-...	открыт	включен		100	

Аккаунты клиента | Аккаунты оператора

Редактор | Тарифы | Новые абонплаты

Тип аккаунта: операторский | Период: с 13.02.2017 по | Статус: Активен

Устройство: First: ID3 [localhost:8080]

Логин: test_operator

Комментарий:

Сохранить аккаунт | Отмена

ce_oper_radius

Тут возможно 2 варианта поиска акаунта:

- Поиск с оператора как клиентского аккаунта
- Поиск оператора с помощью скрипта

А первом случае случае оператор это обычный клиентский account, поиск его может быть настроен например по логину.

Во втором случае в настройках NAS-должны быть настроен **обработчик процессора протокола** в котором мы выставляем опцию **VoiceNas. OPERATOR_ACCOUNT** - id аккаунта.

Пример обработчика

```
package ru.bitel;
import java.util.Iterator;
import java.util.List;
import bitel.billing.server.radius.CiscoVendor;
import bitel.billing.server.radius.Vendors;
import ru.bitel.bgbilling.kernel.network.radius.RadiusAttribute;
import ru.bitel.bgbilling.kernel.network.radius.RadiusAttribute.RadiusAttributeString;
import ru.bitel.bgbilling.kernel.network.radius.RadiusPacket;
import ru.bitel.bgbilling.modules.voice.access.om.ProtocolHandler;
import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDevice;
import ru.bitel.bgbilling.modules.voice.api.common.bean.VoiceDeviceType;
import ru.bitel.bgbilling.modules.voice.radius.VoiceNas;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.ParameterMap;
import ru.bitel.common.sql.ConnectionSet;
public class Prot3
    implements ProtocolHandler
{
    @Override
    public void preprocessAccessRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
        throws Exception
    {
        String prefix = "remote-media-address";

        List<RadiusAttribute<?>> attributes = request.getAttributes( Vendors.CISCO, CiscoVendor.Cisco_AVPair );
        if( attributes != null )
            for( Iterator<RadiusAttribute<?>> it = attributes.iterator(); it.hasNext(); )
            {
                RadiusAttributeString ra = (RadiusAttributeString) it.next();
                String value = ra.getValue();

                if( value.startsWith( prefix ) )
                {
                    if( value.endsWith( "77.82.17.33" ) )
                    {
                        request.setOption( VoiceNas.OPERATOR_ACCOUNT, 1 );
                        break;
                    }
                    else if( value.endsWith( "172.36.104.61" ) )
                    {
                        request.setOption( VoiceNas.OPERATOR_ACCOUNT, 2 );
                        break;
                    }
                }
            }
    }

    @Override
    public void init( Setup setup1, int int2, VoiceDevice voiceDevice3, VoiceDeviceType voiceDeviceType4,
ParameterMap parameterMap5 )
        throws Exception
    {
    }

    @Override
    public void postprocessAccountingRequest( RadiusPacket radiusPacket1, RadiusPacket radiusPacket2,
ConnectionSet connectionSet3 )
        throws Exception
    {
    }

    @Override
    public void postprocessAccessRequest( RadiusPacket radiusPacket1, RadiusPacket radiusPacket2,
ConnectionSet connectionSet3 )
        throws Exception
    {
    }

    @Override

```

```
public void preprocessAccountingRequest( RadiusPacket request, RadiusPacket response, ConnectionSet
connectionSet )
    throws Exception
    {
    }
}
```

Остальные настройки(тариф и т.п) как при тарификации абонента.

В результате на этом договоре мы получаем [отчет с сессиями](#) оператора.

Тарификация оператора по Radius-запросу абонента

По одному Radius-запросу можно тарифицировать как абонента, так и оператора. В этом случае access запрос проверяется только для абонента(баланс, статус и т.п). При обработке accounting-а если мы нашли звонок как клиентский и скрипт поставил оператора, то мы создаем дополнительную сессию на операторе.

В этом случае настройка происходит аналогично предыдущему пункту, но на самом аккаунте не указывается никаких данных(нет логина), которые бы помогли его найти и скриптом в [обработчике процессора протокола](#) аналогично ставится **VoiceNas.OPERATOR_ACCOUNT**.

Тарификация абонента по агентской схеме

Агентская схема предполагает, что клиент, совершая звонки, может потреблять услуги нескольких операторов, с которыми у оператора местной связи, обслуживающего клиента, заключены агентские договора. Нарботка одного и того же телефонного номера должна быть отнесена к различным договорам, по каждому из которых, в общем случае, ведется свой баланс и работа с задолженностью.

Для поддержки агентской схемы в модуле Voice используются независимые субдоговоры. Номер абонента добавляется на супердоговор, одновременно являющимся договором на услуги местной связи. В супердоговор заносятся тарифные планы на местную связь.

Далее к супердоговору соотносят один или несколько независимых субдоговоров с тарифами конкретных операторов. Номера супердоговора "наследуются" субдоговором, период субдоговора определяет период отношений данного оператора с клиентом.

При тарификации последовательно просматриваются тариф основного договора и тарифы всех независимых субдоговоров, исходя из чего звонок соотносится либо к супердоговору (местный звонок, отработал тариф местной связи), либо к одному из субдоговоров. При этом последовательность просмотра субдоговоров задается переменной:

```
# - id  
radius.agent.sub.contract.order=4,5
```

В данном случае 4 и 5 это id групп договоров. Например все субдоговоры на зоновую связь должны быть в 4-ой группе, а субдоговоры на МГМН-связь в 5-ой группе, тогда вначале будет просматриваться тарифы зоновой связи, потом МГМН.

Вот пример супердоговора с двумя добавленными субдоговорами: на зоновую и МГМН связь.

Параметры Отчет Карточки

Договор

26

Период с 20.06.2017 по

Договор является супердоговором для

- Параметры договора
- Объекты договора 0/0
- Иерархия договоров супер: 0/2
- Статус Активен
- Лимит -1 000,00
- Режим дебет
- Лицо Физическое
- Баланс (сентябрь 2017)
 - Входящий остаток 0,00
 - Приход 0,00
 - Нарботка 0,00
 - Расход 0,00
 - Возврат 0,00
 - Исходящий остаток 0,00
 - Резерв 0,00
 - Доступная сумма 0,00
- Тарифные планы
 - Телефония: Местный 0 р
- Модули
 - Voice
 - Абонплаты
- Web
 - Пароль статистики
 - Управление лимитом
 - Web-меню

Открыть субдоговор

Освободить субдоговор

Выберите договор(ы)... X зависимый баланс Добавить субдоговор

Поиск договоров x FS-OFIS x MGMN-OFIS-ZONA x MGMN-OFIS-MGMN

Вот аккаунт на супердоговоре:

Параметры Отчет Карточки

Договор [KRD-FS-OFFIS []]

Период с 20.06.2017 по

Аккаунты клиента Аккаунты оператора

Аккаунт	Устройство	Тип	Номер	Период	Статус	Состояние	Комментарий	id
3@192.168.1.1(Freeswitch: [192...				20.06.2017...	открыт	включен		38

Параметры договора: 0/0
 Объекты договора: супер: 0/2
 Иерархия договоров: Активен
 Статус: Активен
 Лимит: -1 000,00
 Режим: дебет
 Лицо: Физическое

Баланс (сентябрь 2017)
 Входящий остаток: 0,00
 Приход: 0,00
 Нарботка: 0,00
 Расход: 0,00
 Возврат: 0,00
 Исходящий остаток: 0,00
 Резерв: 0,00
 Доступная сумма: 0,00

Тарифные планы
 Телефония: Местный 0

Модули
 Voice
 Абонплаты

Web
 Пароль статистики
 Управление лимитом
 Web-меню

Группы
 Все
 Краснодар
 Краснодар-воиптелефо
 Краснодар-воиптелефо
 Краснодар-воиптелефо
 Краснодар-воиптелефо
 Офис

Скрипт поведения
 Radius

Скрытый

Поиск договоров: KRD-FS-OFFIS OFIS-ZONA OFIS-MGMN

Договор на зонную связь с тарифом:

Параметры Отчет Карточки

Договор [KRD-FS-OFFIS-ZONA []]

Период с 21.06.2017 по

Глобальные тарифы Персональные тарифы Тарифные опции Группа тарифов

ID	TPID	Поз.	Наименование	Период действия	Комментарий
76	4	0	Зона	21.06.2017...	

Параметры договора: 0/0
 Объекты договора: н.суб.
 Иерархия договоров: Активен
 Статус: Активен
 Лимит: -1 000,00
 Режим: дебет
 Лицо: Физическое

Баланс (август 2017)
 Входящий остаток: -483,00
 Приход: 0,00
 Нарботка: 0,00
 Расход: 0,00
 Возврат: 0,00
 Исходящий остаток: -483,00
 Резерв: 0,00
 Доступная сумма: -483,00

Тарифные планы
 Зона

Модули
 Voice
 Абонплаты

Web
 Пароль статистики
 Управление лимитом
 Web-меню

Группы

Скрытый

Поиск договоров: KRD-FS-OFFIS OFIS-ZONA OFIS-MGMN

Звонки в отчете на зонном субдоговоре:

Завершённые Текущие

1 из 1 [25]

SessionId	Аккаунт	Направления	Услуги	Старт	Стоп	Активность	Длительность	Стоимость
1124587786121...	>>	Краснодарский к...	786121... 79183106...	10.08.2017 10:14:55	10.08.2017 10:17:24	10.08.2017 10:17...	00:02:21 [141]	7.05
6272392786121...	>>	Краснодарский к...	786121... 79183225...	31.08.2017 17:53:35	31.08.2017 17:53:47	31.08.2017 17:53...		0.00

Договор на МГМН-связь с тарифом:

Параметры | Отчет | Карточки
 Договор: МГМН-00008-KRD OFIS-MGMN [] 113

Период: с 23.06.2017 по

Глобальные тарифы			Персональные тарифы		Тарифные опции	Группа тарифов	Комментарий
ID	TRID	Поз.	Наименование			Период действия	
117	5	0	МГМН			23.06.2017-...	

Параметры догов
 Объекты договор 0/0
 Иерархия договорн.суб.
 Статус: Активен
 Лимит: -1 000,00
 Режим: дебет
 Лицо: Физическое

Баланс (август 20)

Входящий остаток	-17,87
Приход	0,00
Наработка	0,00
Расход	0,00
Возврат	0,00
Исходящий остаток	-17,87
Резерв	0,00
Доступная сумма	-17,87

Тарифные планы: МГМН

Модули: Voice, Web
 Web: Пароль статичес, Управление ли, Web-меню
 Группы: Все

Скрытый

Поиск договоров: -FS-OFIS, -OFIS-ZONA, -OFIS-MGMN

Звонки в отчете на МГМН-договоре:

Завершённые		Текущие											
Таблица	Аккаунты	Направления	Услуги										
SessionId	Аккаунт	Тип	Направление	A#e164	B#e164	Старт	Стоп	Активность	Длительность	Стоимость			
7274747861217...	>>		г. Санкт-Петербу...	786121...	78123364...	07.08.2017 14:31:40	07.08.2017 14:32:18	07.08.2017 14:32:...	00:00:33 [33]	2.20			
11247047861217...	>>		г. Ростов-на-Дон...	786121...	78632036...	10.08.2017 13:46:01	10.08.2017 13:46:55	10.08.2017 13:46:...	00:00:52 [52]	3.46667			

1 из 1 [25]

Управление доступом абонентов

Личный кабинет модуля Voice

Поддержка модуля Voice в Личном кабинете находится в разработке, за подробностями обращайтесь в Телеграмм или на E-mail.

Коды ошибок авторизации при Radius запросах.

При авторизации иногда возникают ошибки, которые увидеть в ошибках авторизации в [монитор соединений](#). Но код ошибки так же виден в логах log4j для BGVoiceRadius. Ниже представлена расшифровка кодов ошибок :

Таблица. Коды ошибок

Код ошибки	Описание
0	Авторизация успешна.
1	Аккаунт не найден.
2	Неверный пароль.
3	Количество активных сессий превышено.
4	Учётный период не активирован.
10	Аккаунт заблокирован.
11	Договор заблокирован.
12	Недостаточно средств.
44	Доступ запрещен узлом в тарифе.
62	Не определён тарифный план.
63	Не определена цена в тарифном плане.
64	Тип звонка не определен.
65	Направление не найдено.
66	Ошибка тарификации.

Переобработка Radius-логов

Иногда возникает необходимость пересоздать Radius-сессии по логам Radius.

Это можно сделать на вкладке **Логие/переобсчет->Radius**.

Там нужно выбрать месяц, устройство и добавить нужные дни в переобработку. Интерфейс тут практически такой же для обработки логов CDR.

При переобработке логов учитываются только stop-пакеты и результат может различаться от результата полученного при runtime.

Mango Office

Назначение модуля

Модуль Mango Office предназначен для интеграции биллинга с ООО «Манго Телеком» для перепродажи их услуг.

Настройка модуля

Конфигурация модуля

```
# ,
defaultService=302

# API : , , ( Mango )
apiDomenMango=http://billing-api-test.ru
apiLogin=
apiPassword=

# FTP: ip, , , ssh ( Mango )
ftpIP=
ftpLogin=
ftpPassword=
ftpSshPathKey=
# , ftp csv
#ftpPath=

# , - <1>=<1>;<2>=<2>
tariffs=400000088= ;400000089= ;400000090 -

# /, Mango
#
f.creditLimit=100
# -
f.personName=
# -
f.psptseries=
# -
f.psptnumber=
# -
f.psptissuedby=
# -
f.psptissueddt=
# -
f.psptissuedcode=
# -
f.contactPhone=
# - email.
f.email=

# /, Mango
# , u.inn=23 ,
# bill.u.inn=65:inn, 65- , inn -
#
u.creditLimit=1000
# -
u.personName=
# - .-
u.legalName=
# -
u.inn=
# - ( - )
u.contactName=
# -
u.contactPhone=
# - ( , , )
u.legalAddress=
```

```

#          - ( , , )
u.invoiceAddress=
#          - E-mail
u.email=

#          /
#          -
u.kpp=
#          -
u.ogrn=
#          -
u.directorTitle=
#          - .
u.directorTitleGenitive=
#          -
u.directorFullName=
#          -
u.directorFullNameGenitive=
#          -
u.bankName=
#          - (          u.bankName          )
u.bankCity=
#          -
u.bankBic=
#          -
u.bankCurrentAccount=
#          - .
u.bankCorrAccount=
#          - ( , , )
u.actualAddress=

```

После установки модуля в клиенте биллинга перейдите в "Модули -> Редактор модулей и услуг" и добавьте модуль Mango Office. Далее в модуле добавьте как минимум одну услугу и ее код внесите в конфигурацию модулю в параметр **defaultService**.

После создания конфигурации в модуле не забудьте установить её активной.

Регистрация клиента

После того как вы произвели интеграцию с Манго(вас зарегистрировали как агента, вы загрузили тарифы и получили данные для доступа) и заполнили все требуемые параметры в конфигурации, можно приступать к регистрации клиента - вашего абонента. Для этого, в первую очередь, добавьте ранее созданный экземпляр модуля в договор абонента.

После вам будет доступны два варианта:

- Создание новых учетных данных
- Привязка существующих учетных данных к договору(*вводится accountId*)

После одной из этих операций, вам будет доступен логин, пароль, accountId(идентификатор клиента Манго), а так же отображен статус клиента и его баланс.

Начисления

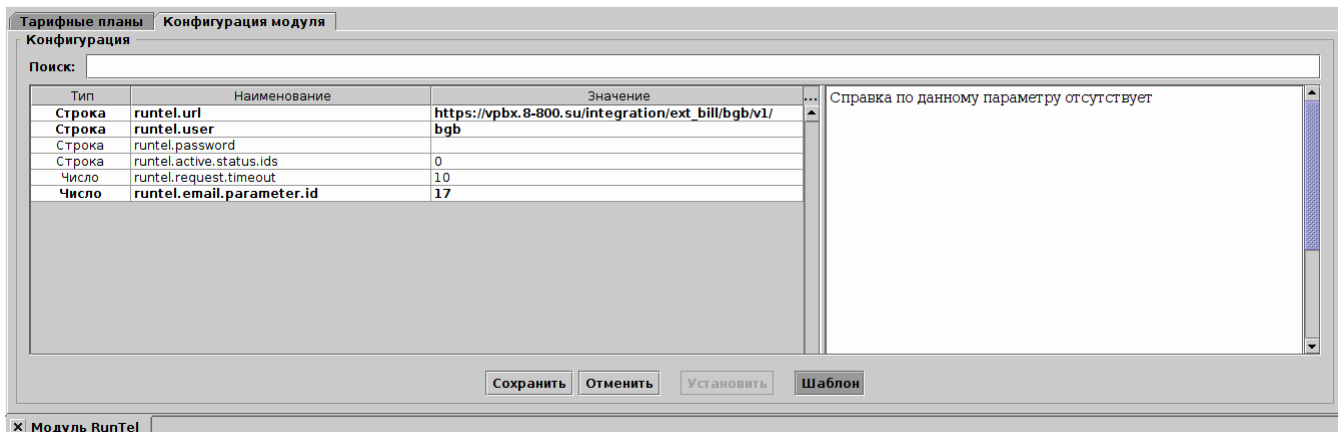
Начисления производятся задачей планировщика "**Загрузка начислений с FTP**". Данную задачу нужно настроить на запуск каждый день, она будет производить обращения на указанный в конфигурации ftp и считывать файл формата "fees_<YYYY-MM-dd>.csv" за предыдущие сутки.

Так же данную операцию можно запустить в ручную в модуле, указав дату обработки. Перед каждой наработкой производится удаление наработки за обрабатываемые сутки, по этому задачу можно запускать несколько раз без риска задвоения начислений.

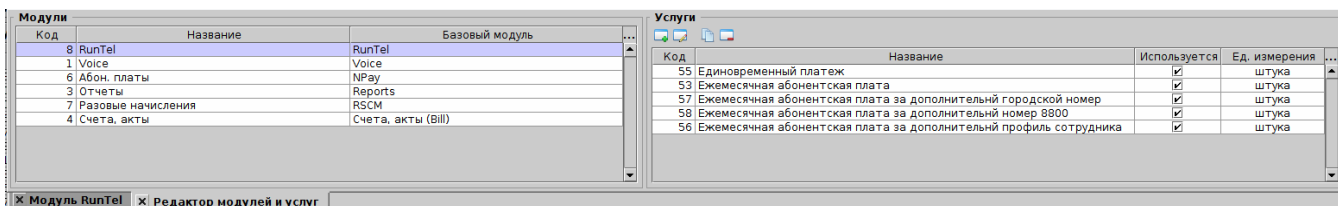
Модуль RunTel

Модуль предназначен для интеграции Коммуникационной платформы RunTel (<https://runtel.ru/ru/>) и ACP BGBilling. Модуль позволяет управлять услугами BATS RunTel из клиента биллинга, начислять наработку за оказываемые услуги, производить сверку с поставщиком платформы, формировать отчеты.

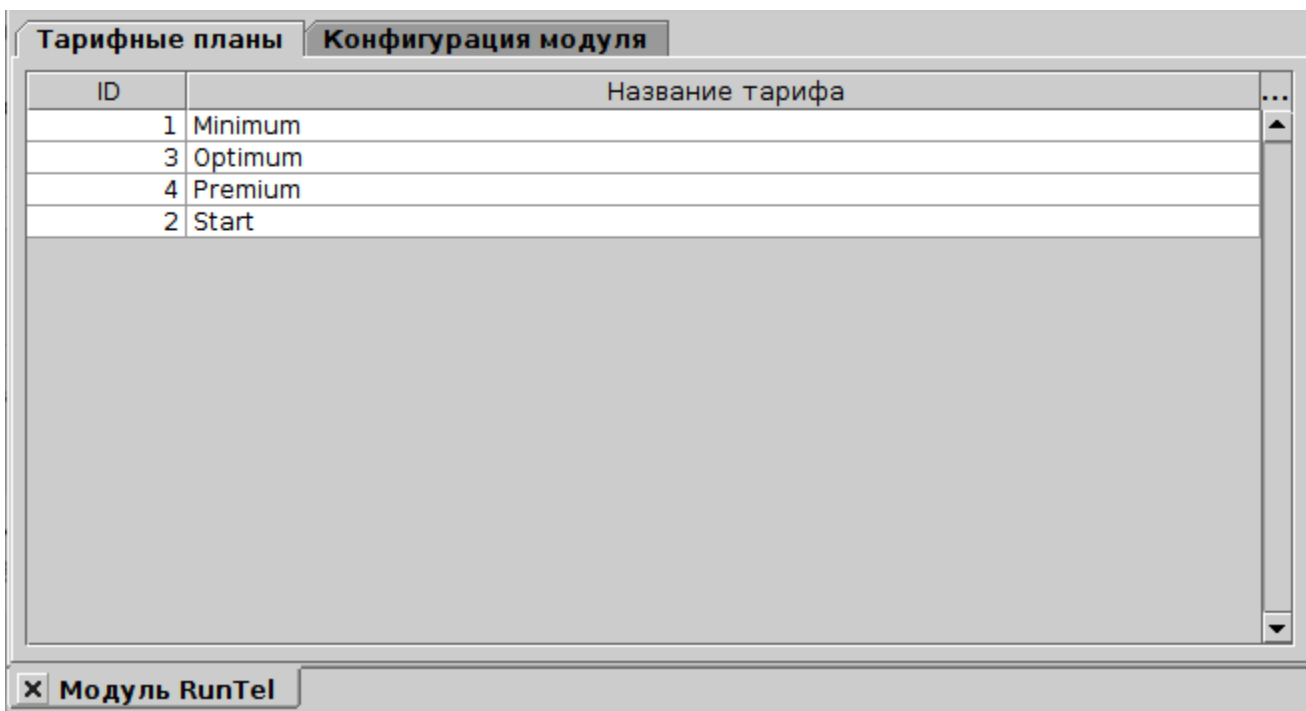
Для работы с модулем необходимо установить модуль, создать экземпляр модуля, создать и отредактировать конфигурацию модуля, сделать ее активной.



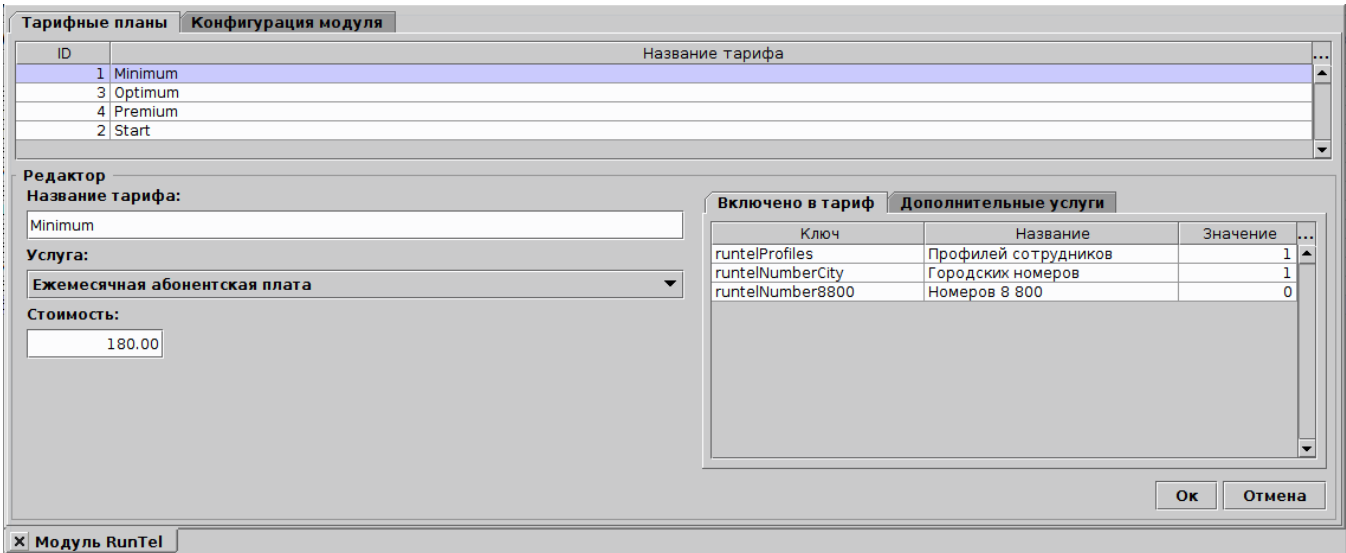
В редакторе модулей и услуг для модуля заводятся необходимый набор услуг, например такой:



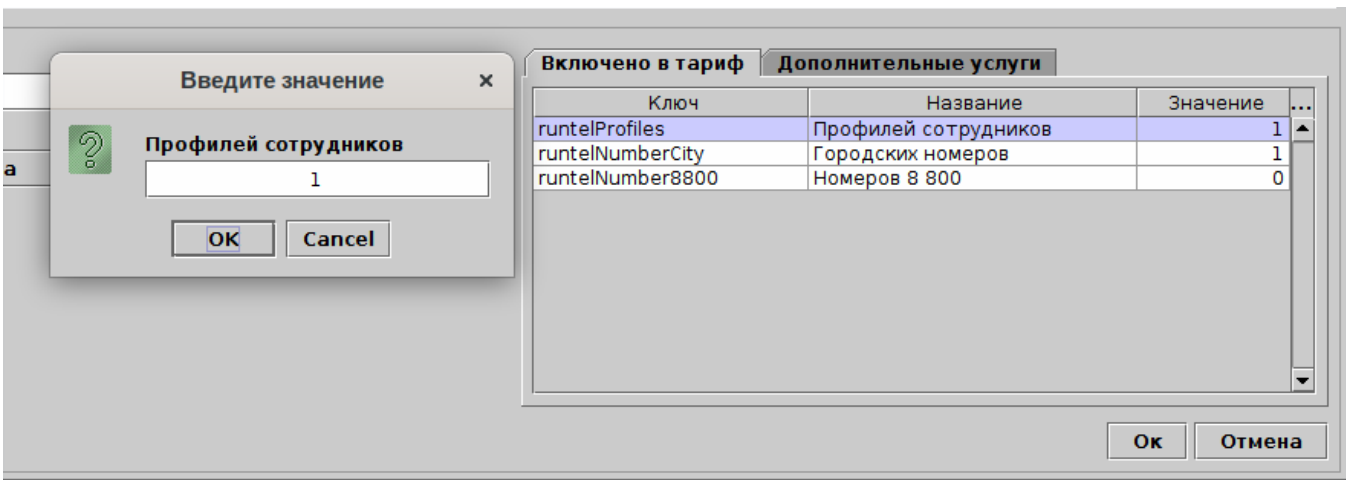
Далее в редакторе тарифных планов модуля, необходимо завести тарифные планы.



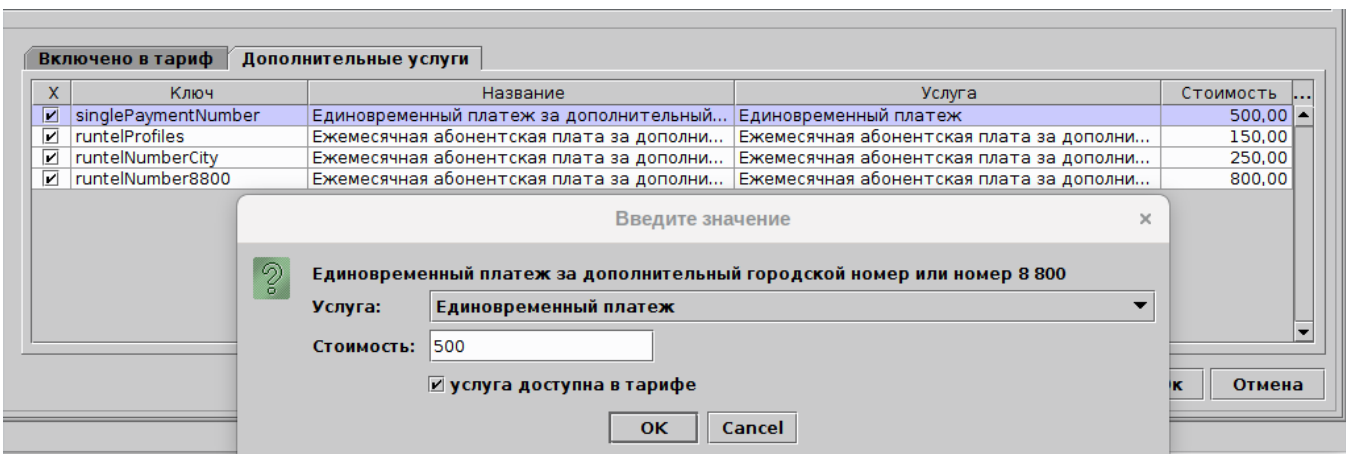
В редакторе тарифа указывается название тарифа, услуга "Абонентской платы" и ее сумма за месяц (глобальные тарифные планы для начисления за услуги модуля не используются).



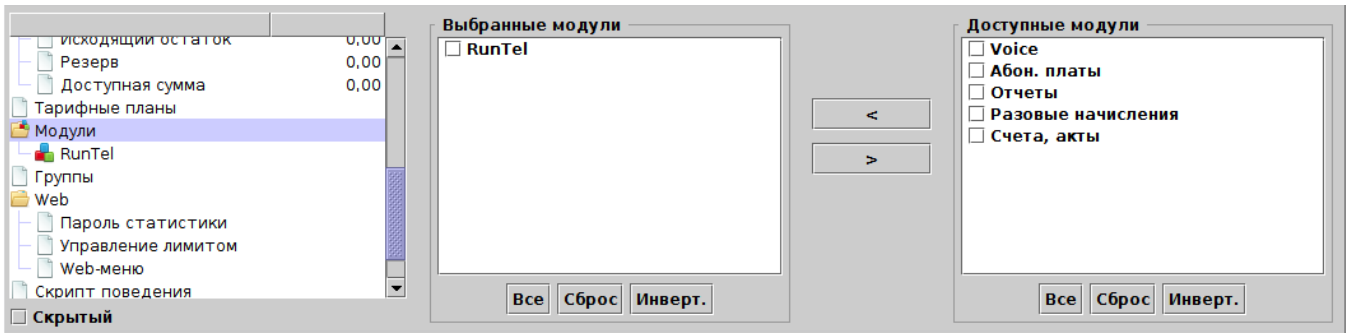
Также в редакторе задаются услуги включенные в ТП.



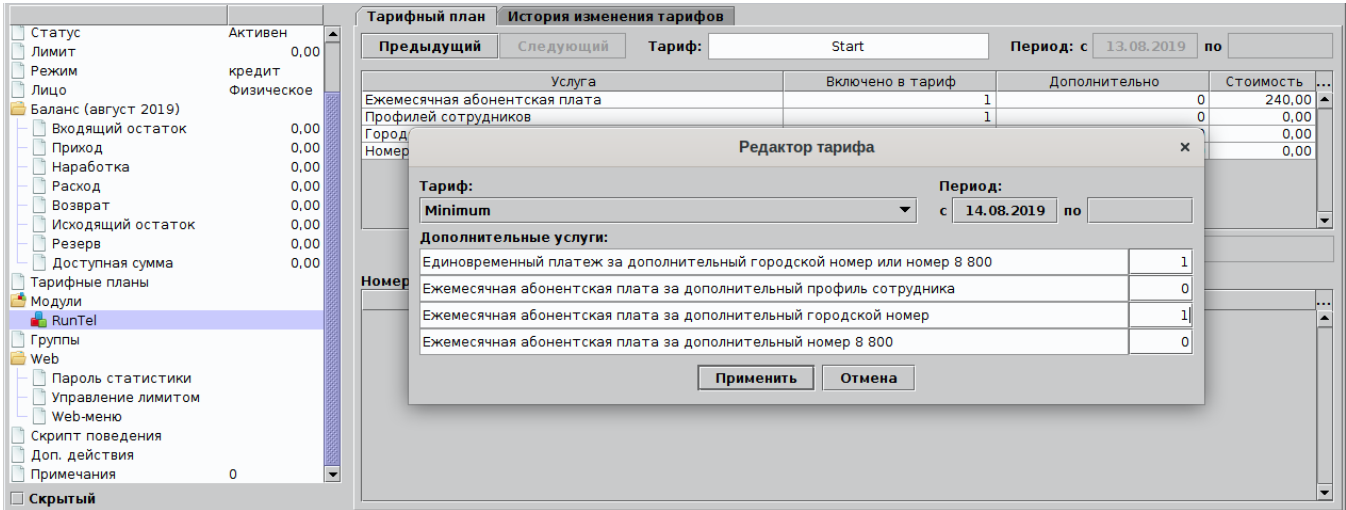
и дополнительные услуги которые могут включатся в ТП при добавлении ТП в договор. Список дополнительных услуг задается в коде, в редакторе ТП для для каждой услуги можно сопоставить услугу, стоимость и доступность услуги в данном ТП.



Далее на договоре добавляем Модуль

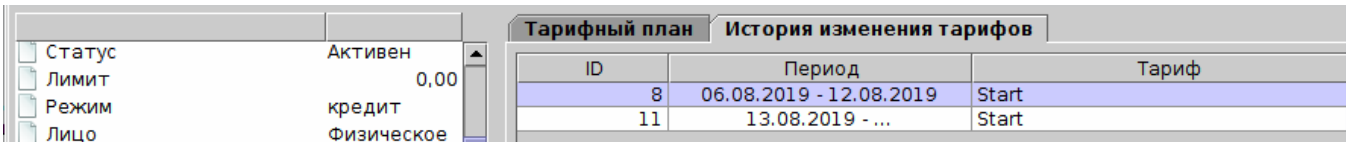


После чего можно на договор подключить ТП и указать количество дополнительных услуг. В виду особенностей взаимодействия с платформой RunTel подключить новый ТП можно только со следующего дня (добавление с текущей даты и предыдущих дней запрещено)

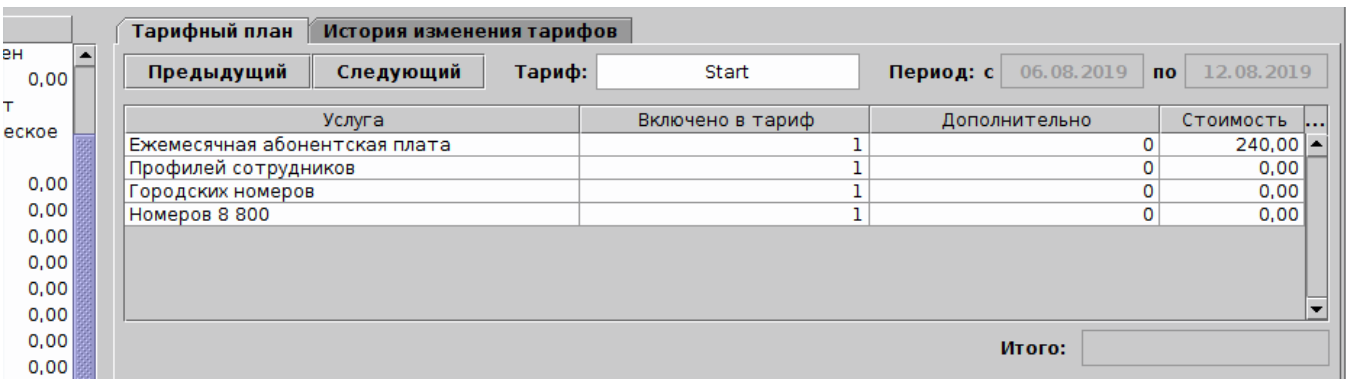


Если есть необходимость отредактировать ТП, то для этого необходимо создать новый ТП с новыми параметрами и указать дату начала (завтра или позднее). Менять услуги в действующем ТП (задним числом) запрещено. Для действующего ТП можно только установить дату окончания периода (сегодня или позднее)

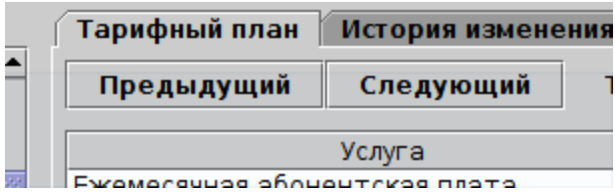
История изменений ТП на договоре доступна закладке "История изменения тарифов"



По двойному клику можно из истории перейти к просмотру параметров ТП



также переключатся между ТП планами можно по кнопкам



Данные о текущих ТП, дополнительных услугах и клиентах синхронизируются с платформой RunTel раз в сутки при помощи задачи планировщика. Начисление за услуги производится другой задачей.

12	Модуль RunTel. Синхронизация данных в биллинге и ВАС.	01.07.2019-...	RunTel	включена	нормальный	*	*	*	0	1
13	Модуль RunTel. Начисление наработки и установка статуса договора.	01.07.2019-...	RunTel	включена	нормальный	*	*	*	0	0

Модули ТВ

Модуль CerberCrypt

Назначение модуля

Модуль предназначен для управления системой цифрового вещания. Поддерживаются несколько систем: CerberCrypt ставропольского завода "Сигнал", система условного доступа Irdeto P1sys, система STI/NordE, система conax и другие (полный список в разделе настройки протоколов). Поддерживается ведение каналов (для протоколов поддерживающих управление каналами), их пакетирование, обсчёт и автоматическое управление подпиской абонентов средствами биллинга, отсылка сообщений и т.п., в зависимости от используемой системы условного доступа.

Далее в тексте все эти системы обозначены как «система условного доступа», система УД, УД.

Настройка модуля CerberCrypt

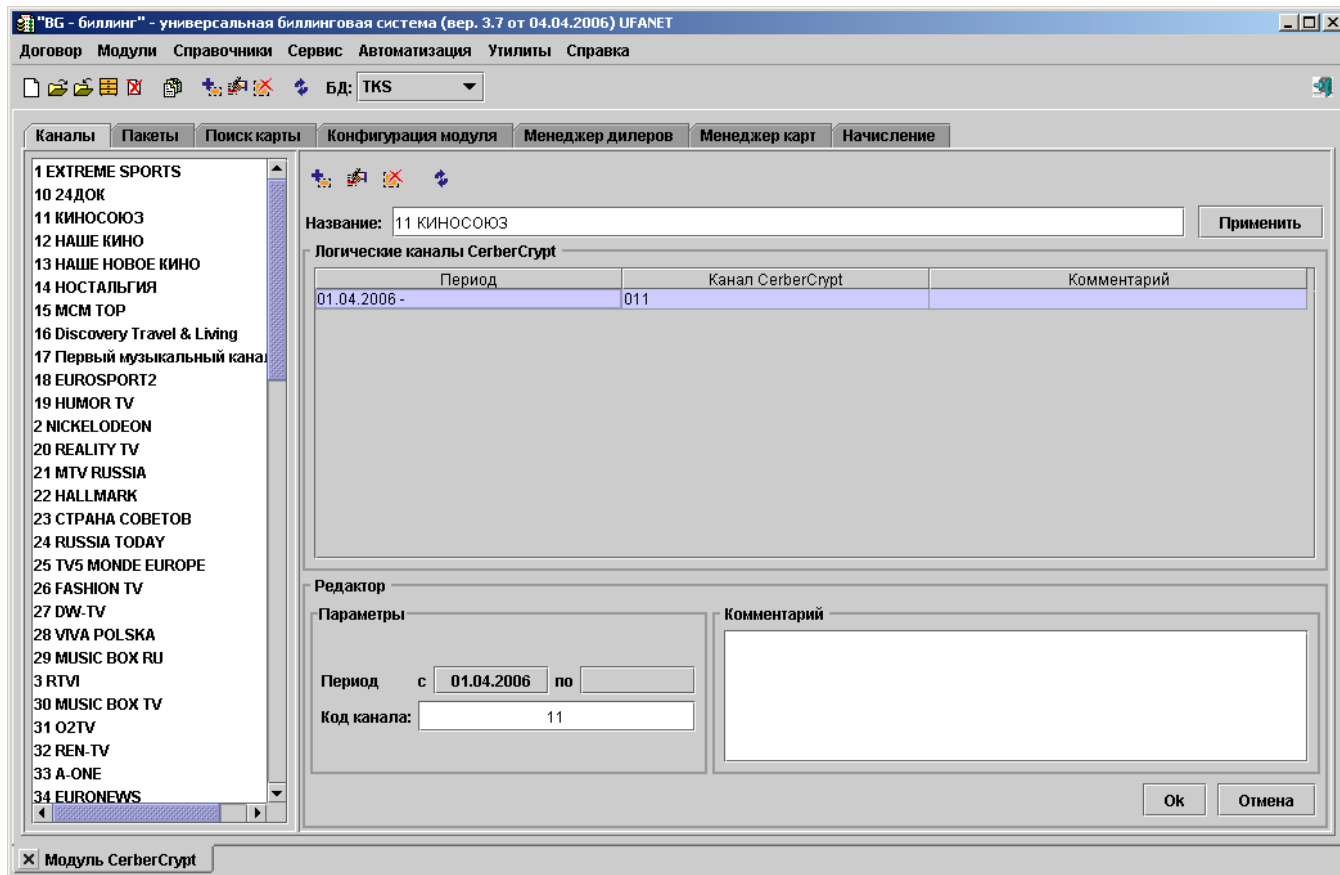
Предполагается, что у вас уже установлена и настроена система условного доступа, произведена настройка каналов и вам известны уникальные коды каналов/пакетов.

```
#
contract.status.active.codes=0

# Web
web.menuItem1= CerberCrypt
web.menuItem2= CerberCrypt
#
#0 -
#1 -
#2 - -
#3 - , -
cerbercrypt.cardpacket.addcheck=1
#
max.periodic.errors=30
#-----
#
#
#closed.date.disabled.ActionRecalculate=1
#
#closed.date.disabled.ActionUpdateUserCard=1
#
#closed.date.disabled.ActionUpdateCardPacket=1
#-----
# id , ( , ..)
not.synchronized.dealers=6,7
# " ( , )
#calculate.locked.cardpackets=1
# "
#calculate.suspend.cardpackets=1
```

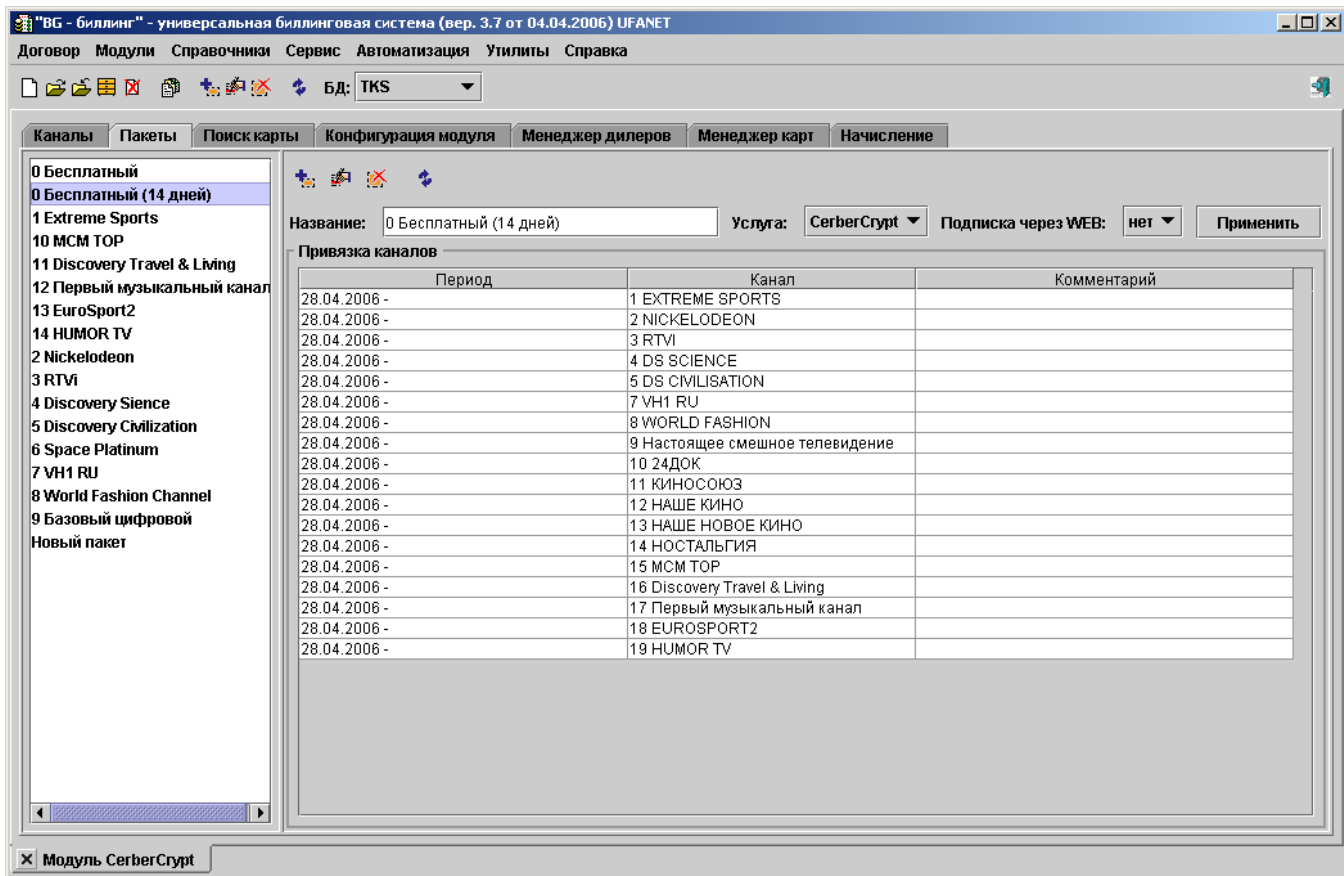
Настройка каналов/пакетов

Далее производится заведение каналов в биллинге, к каждому каналу должен быть привязан код канала системы УД.



Пакет - это логическая единица, включающая в себя несколько каналов. Пакеты используются для более простого управления подпиской абонентов. Все пакеты необходимо завести в справочнике пакетов с указанием привязанных к ним каналов.

К каждому пакету должна быть указана услуга, на которую будет начислена наработка за пользование этим пакетом. Если необходима возможность включения/выключения пакета через Web, то следует установить флаги "Добавление через WEB" и "Закрытие через WEB". Если пакет должен быть всегда доступен пользователям ("социальный пакет"), то необходимо установить галочку "Не блокируемый". Такой пакет не будет блокироваться задачей "Блокировка CerberCrypt".

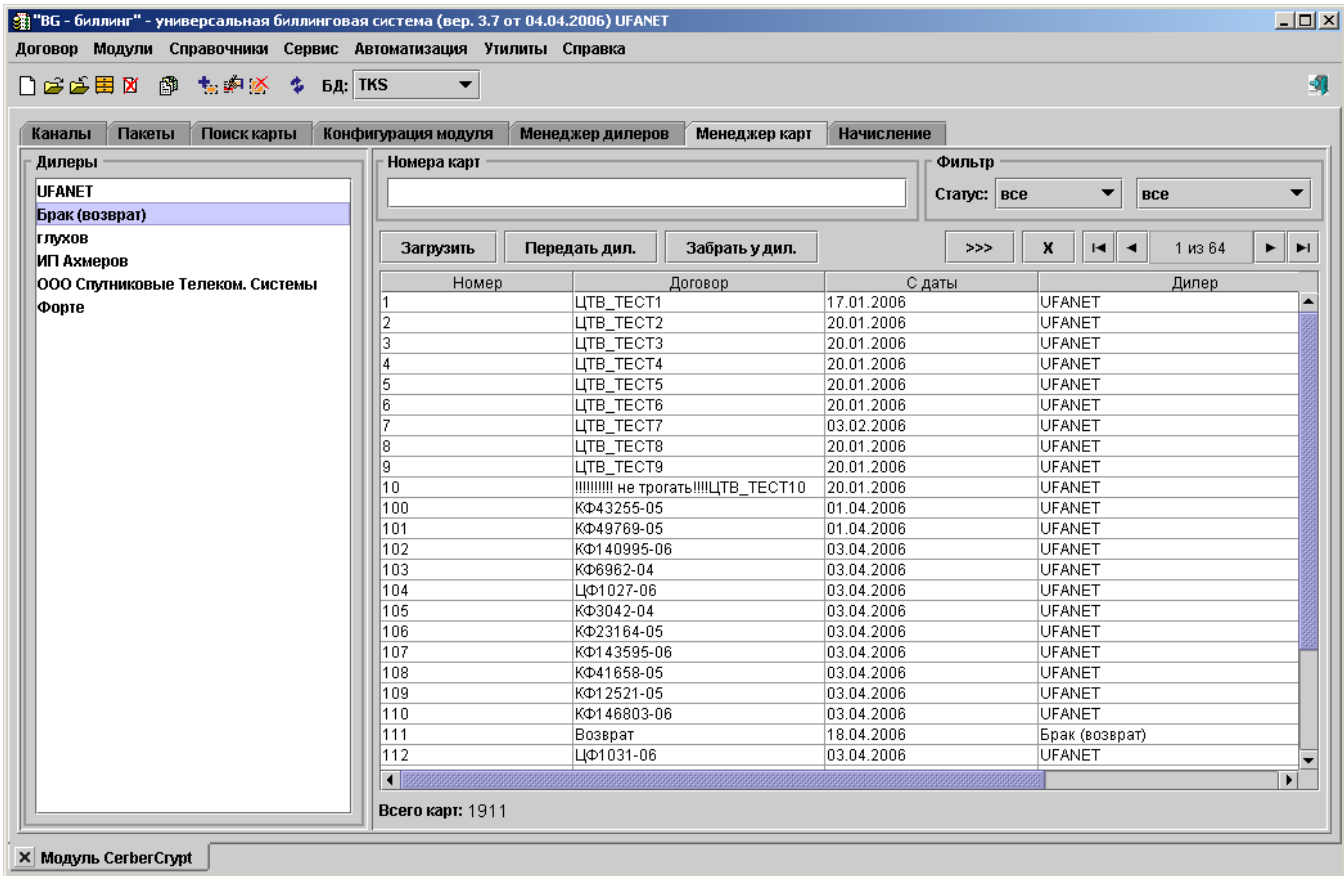


Также каждому пакету должен быть присвоен код, которым он нумеруется в системе УД, если используемый протокол предполагает пакетное управление.

Все используемые карты условного доступа должны быть загружены в модуль. Загрузка происходит в менеджере карт из файла формата:

```
< >\t< >
```

Пароль карты - уникальная последовательность для активации карты через Интернет, что позволяет провайдеру организовывать продажу карт с приемниками вне офиса.



Для загрузки карт в систему следует нажать кнопку **Загрузить** и выбрать файл.

Номера карт

Фильтр

Статус:

Загрузить

Передать дил.

Забрать у дил.

>>>

X

◀

1 из 64

▶▶

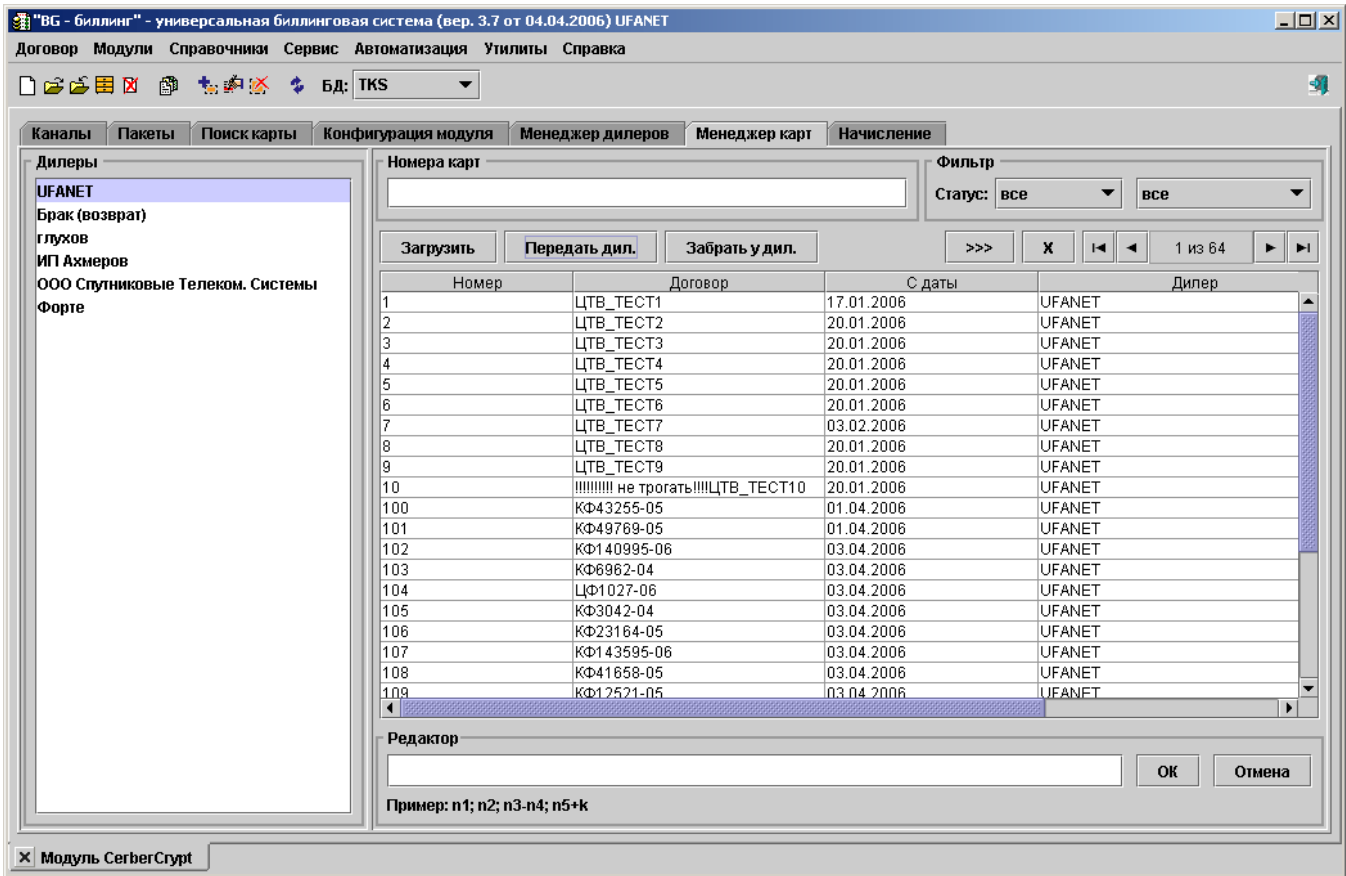
Номер	Договор	С даты	Дилер
1	ЦТВ_ТЕСТ1	17.01.2006	UFANET
2	ЦТВ_ТЕСТ2	20.01.2006	UFANET
3	ЦТВ_ТЕСТ3	20.01.2006	UFANET
4	ЦТВ_ТЕСТ4	20.01.2006	UFANET
5	ЦТВ_ТЕСТ5	20.01.2006	UFANET
6	ЦТВ_ТЕСТ6	20.01.2006	UFANET
7	ЦТВ_ТЕСТ7	03.02.2006	UFANET
8	ЦТВ_ТЕСТ8	20.01.2006	UFANET
9	ЦТВ_ТЕСТ9	20.01.2006	UFANET
10	!!!!!!!!!! не трогать!!!!ЦТВ_ТЕСТ10	20.01.2006	UFANET
100	КФ43255-05	01.04.2006	UFANET
101	КФ49769-05	01.04.2006	UFANET
102	КФ140995-06	03.04.2006	UFANET
103	КФ6962-04	03.04.2006	UFANET
104	ЦФ1027-06	03.04.2006	UFANET
105	КФ3042-04	03.04.2006	UFANET
106	КФ23164-05	03.04.2006	UFANET
107	КФ143595-06	03.04.2006	UFANET
108	КФ41658-05	03.04.2006	UFANET
109	КФ12521-05	03.04.2006	UFANET
110	КФ146803-06	03.04.2006	UFANET

Редактор

OK

Отмена

После загрузки карта должна быть передана дилеру. Дилер должен быть заведён на вкладке **Дилеры**. Для осуществления передачи в менеджере карт выбирается дилер и нажимается кнопка **Передать дил.**



Формат перечисления карт приведён под окошком. Формат "n+k" означает k карт начиная с номера n. Отбор карты у дилера производится аналогично. Дилеры представляют из себя как реальных агентов по продаже карт, так и фиктивные контейнеры для учёта повреждённых карт, утерянных и т.п.

Также менеджер карт позволяет производить простейший учёт карт. Так, можно определить количество свободных карт, либо количество карт, выданных определённому дилеру.

Тестовый

Тестовый активатор, просто выводит всё в лог с уровнем DEBUG:

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.TestServiceActivator
```

CDCAS 3.0

Реализованный протокол: CDCAS3.0_SMS Interface Specification <V1.0>. Режим работы — с пакетами. Подразумевается один сервер; коннект берётся везде нулевой.

```
# CDCAS3.0_SMS
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.CDCAS30ServiceActivator
#
server.0.host=localhost
server.0.port=7364
server.0.timeout=5000
#   : ID, SMS ID
server.0.OPE_ID=666
server.0.SMS_ID=1
# rootkey (16 24 )
#server.0.rootkey=ABCDEFGHIJKLMNPO
server.0.rootkey=ABCDEFGHIJKLMNPOABCDEFGH
# OSD (1, 2)
server.0.osdStyle=2
# OSD
server.0.osdDuration=60
# ( )
server.0.emailCharset=utf-8
server.0.osdCharset=utf-8
```

CerberCrypt

Для системы CerberCrypt укажите адрес, порт и логин с паролем для обращения к серверу CerberCrypt:

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.CerbercryptServiceActivator
server.0.host=192.168.168.10
server.0.port=8100
server.0.login=admin
server.0.pswd=xxxxx
# , CAS
#server.0.timeout=1000
```

Поддерживается всё железо аналогичное: CerberCrypt, DVCrypt, Televue и прочие, в том числе настраивающиеся согласно документации CerberCrypt. Обратите внимание на права пользователя CerberCrypt, под которым происходит авторизация на удалённом сервере CerberCrypt (в примере выше — admin), ему нужно дать необходимые права для того, чтобы использовать все функции протокола (установление, чтение подписки и прочие).

CTI CAS 4.0

```
# CTI CAS 4.0
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.CtiCas4ServiceActivator
server.0.host=localhost
server.0.port=7000
# SO-
server.0.timeout=5000
server.0.smsNumber=1
server.0.emailDay=1
server.0.emailFrom=
server.0.osdDay=1
server.0.osdAccount=1
server.0.osdFrom=
server.0.emailEnable=1
server.0.osdEnable=1
#
# Region No (CCA Number)
server.0.regionID=1
# Control District (GCA Number)
server.0.gcano=1
# Control CUSTWD
server.0.ctrcustwd=1

# ( )
server.0.emailCharset=utf-8
server.0.osdCharset=utf-8
# UA
server.0.sendUAonly=1

# , 19 "Smart Card not Exist"
server.0.autoActivate=1

# ( , , )
#server.0.commandTimeout=100
```

Управление ведётся пакетами.

Протокол игнорирует начальную дату открытия пакета. Использование `gradually.subscription` бесполезно, т.к. в протоколе не используется дат (сейчас для подписки используется команда 15, а не 103).

Параметры типа "CCA Number (Region ID)" может быть удобно задавать через конфиг пользовательских устройств.

Conax

Система conax (beta-версия) требует подобную конфигурацию:

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.ConaxServiceActivator
# (.. ftp-)
server.0.host=localhost
server.0.port=21
# /
server.0.username=anonymous
server.0.password=
# ----,
#      autreq/req, autreq/ok, autreq/err
# ( )
# "." (, , .. )
server.0.pwdroot=pub/conax
```

Conax 4.0

Система Conax Contego 4.0, протокол с веб-сервисами (Conax Contego Web Services итд)

```
# Conax 4.0
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.Conax4ServiceActivator
# auth-
server.0.user=***
server.0.password=***
# -
server.0.subscription=https://***:44302/ca-server/webservices/caclient/subscription
server.0.messaging=https://***:44302/ca-server/webservices/caclient/user-text-messaging
server.0.administration=https://***:44302/ca-server/webservices/caclient/ca-client-administration
server.0.pairing=https://***:44302/ca-server/webservices/caclient/pairing
# osd(bark)
server.0.osdDuration=58
# : OVER_AIR, SYNCHRONOUS_RETURN
server.0.emmDeliveryMethod=OVER_AIR
```


Gospell

Система Gospell. Режим работы — с пакетами. Везде подразумевается один сервер; коннект берётся везде нулевой.

```
# Gospell
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.GospellServiceActivator

#
server.0.host=localhost
server.0.port=2888
server.0.timeout=5000

# email
server.0.emailTitle=email

# osd ( . )
server.0.osdShowTimeLength=30
server.0.osdShowTimes=0
server.0.osdPriority=0
# expired time, (1 = 24*60*60)
server.0.osdExpiredTimeSecond=86400
```

Griffin

Система CAS Griffin. Режим работы — с пакетами. Везде подразумевается один сервер; коннект берётся везде нулевой.

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.GriffinServiceActivator
#  (.. ftp-)
server.0.host=10.10.10.16
server.0.port=21
server.0.username=anonymous
server.0.password=sms@sms
#  ----
#  "." (, , .. )
server.0.pwdroot=sms
#
server.0.cardcount=666
```

В редакторе пакетов используются коды серверов для пакетов и коды самих пакетов (от 1 до 16). Их надо корректно прописать.

При наличии файла `__new_db__` на ftp (сервер занят? не работает?) он сначала удаляется, чтобы сервер не стал внезапно обрабатывать новую базу. После этого в обычном порядке перетираются `grif0-sub16.dat` итд, затем снова кладётся `__new_db__`. Не нужна в любом случае старая подписка, всё равно новая должна её полностью перетирать при каждом обновлении.

NordE/CTI

Для системы NordE/CTI примерная конфигурация такова:

```
Код

sa=ru.bitel.bgbilling.modules.cerbercrypt.server.NordE2ServiceActivator

server.0.host=localhost
server.0.port=7000
server.0.timeout=5000

server.0.smsNumber=1
server.0.providerId=1
server.0.charset=ISO8859-1
server.0.providerName=Test Provider

server.0.emailEnable=1
server.0.emailDurationTime=1
server.0.emailInterval=1
server.0.emailTitle=
server.0.emailFrom=

server.0.osdEnable=1
server.0.osdTimes=1
server.0.osdInterval=1
server.0.osdTitle=
server.0.osdFrom=

# , ( ).
# , ( ) : NordE2, .
# ", 100500 , ( - - ).
# , " ", - .
period.gradually.subscription=30
```

В данном случае режим работы с пакетами: заводим в биллинге пакеты и управляем пакетами (из удалённой системы), каналы настраиваем извне. Т.е. каналы не синхронизируются из биллинга. Параметры типа "emailEnable" или "charset" больше относятся к оборудованию и, возможно, будет удобнее помимо размещения их в конфигурации модуля разместить их в конфигурации устройств (см. ниже), которые затем привяжутся к картам пользователей (в карточке договора). Также здесь удобно использование возможности "автопродление подписки" (см. ниже и см. настройку в днях кол-ва дней продления активации).

Вкладка «каналы» остаётся на месте, всеми игнорируется, при обмене с сервером, учёте и т. д. не используется. Обмен идёт только через пакеты, для этого используются коды пакетов, которые задаются на соответствующей вкладке.

Номера пакета — вводится вручную при добавлении или редактировании пакета в поле «Номер пакета (код)».

Пример конфигурации оборудования, с которым тестировался данный протокол: сремблеры двух компаний родные Communicate, и компании Gospell, любые современные железки от них совместимы с CAS. Oracle можно использовать любой. Версии модулей софта Conditional Access System : CAS Encryption Control System V2.02, CAS ECMG V2.545, CAS EMMG V2.5.4, CA Manager V2.0.2, SMS Console V2.00 2007.09. Для обмена данными между системами биллинга различных разработчиков служит ПО SMS Console, версия консоли зависит от версии системы кодирования.

Для OSD сообщений задаётся количество повторов (сколько раз пробежит строчка) и интервал в минутах (хотя в описании протокола сказано про секунды). В данный момент реализовано только из конфигурации.

Pisys Irdeto

Для Pisys Irdeto укажите адрес, порт и код оператора:

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.PisysServiceActivator
server.0.host=192.168.168.10
server.0.port=80
server.0.operatorTag=
server.0.nationality=RUS
# (MO, BA, SA)
server.0.regionTag=
```

Реализация более нового протокола, с поддержкой авторизации портала:

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.Pisys2ServiceActivator

# http://<host>:<port>/pisys
server.0.url=

server.0.operatorTag=

# smartyAuth
server.0.portal.username=
server.0.portal.password=

# createSector/overwriteSector
server.0.nationality=RUS
server.0.regionTag=MO
server.0.refreshSectorCreate=
server.0.locationTag=

# sendMail/sendMessage, : ANSI, Unicode, Raw
server.0.textMode=ANSI

# tune
server.0.emergency.networkId=
server.0.emergency.transportId=
server.0.emergency.serviceId=
```

enigo CAS comteza

```
# enigo CAS
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.EnigoCasServiceActivator
# IP CAS , http://.../api/
server.0.host=http://.../api/
#
server.0.apikey=666
# (1 <= repetition <= 255)
server.0.osdRepetition=1
# (, 0 <= duration <= 255)
server.0.osdDuration=30
# (, interval >= 60, interval > duration)
server.0.osdInterval=60
server.0.emailEnable=1
server.0.osdEnable=1
# type 2 " ", = 1, = 2, = 0.
server.0.cmd2type=1
# pid
# name - pid "contract_comment", ;
# cn - contract_title;
# address - pid, "object_title" , object:pid pid - ;
# info - pid "usercard_comment", .
parameter.name=contract_comment
parameter.cn=contract_title
parameter.address=666
parameter.info=usercard_comment
```

XCrypt

Реализация протокола: SAS Interface Specification For SMS V1.3 Xcrypt, Inc. 2010-02-18

```
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.XcryptServiceActivator

server.0.host=127.0.0.1
server.0.port=9950
server.0.timeout=5000

server.0.client_id=666
server.0.zip_code=00000000

server.0.emailEnable=1
server.0.emailGroupEnable=1

server.0.emailCharset=utf-8
server.0.emailBroadcastPeriod=1
# priority 0:normal, 1:high, 2:emergency
server.0.emailPriority=0
server.0.subscribeBroadcastPeriod=1
#
server.0.infinite.date=31.12.2049
```

DRECrypt

СУД от «Цифра».

```
# DreCrypt
sa=ru.bitel.bgbilling.modules.cerbercrypt.server.DreCryptServiceActivator

# oracle
server.0.driver=oracle.jdbc.OracleDriver
server.0.url=jdbc:oracle:thin:@//*.*.*.*:1521/xe
server.0.login=billing
server.0.password=*

#
#
server.0.INFOCAS_PERIOD=20
#
server.0.INFOCAS_INTERVAL=5
# CAS (CAS2 - 2; CAS3, CAS4 - 3)
server.0.CAS_VERSION=3
#
server.0.DEVICE_MODEL_ID=2

#
server.0.wait.result.timeout=60

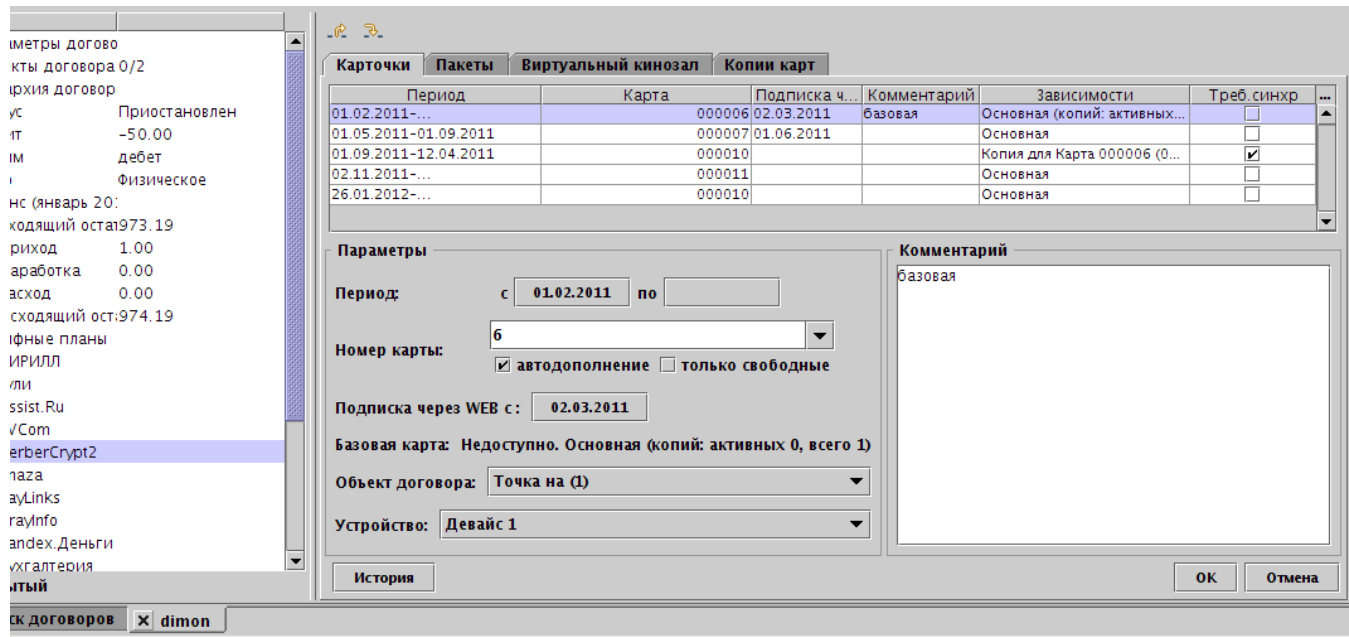
#
server.0.session.tz=+5:0

#
server.0.infinite.date=31.12.2025
```

По лицензионным причинам надо вручную скачать JDBC-драйвер Oracle (Oracle Database JDBC Driver) соответствующей версии и положить в lib на сервере (например: ojdbc8.jar, проще всего загрузить по имени и желательно взять по ссылке с сайта www.oracle.com, придётся зарегистрироваться).

Настройка клиентов

Каждому клиенту, использующему услуги модуля должны быть сопоставлены карты.



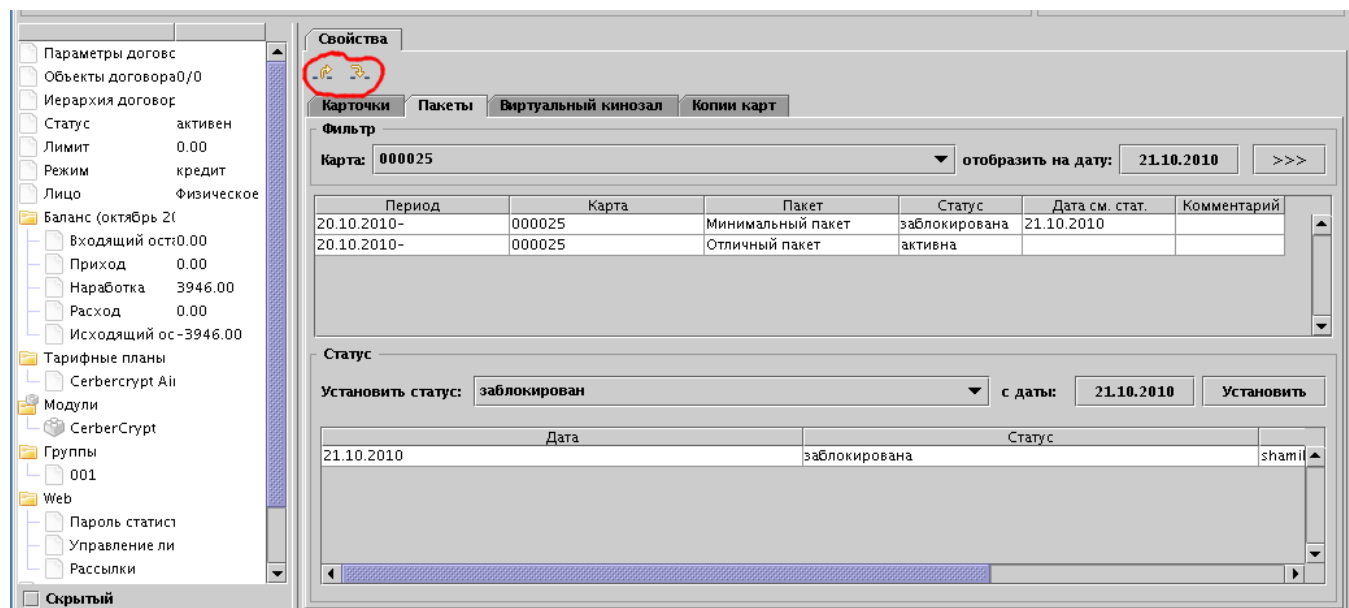
Карта должна быть заведена в системе, передана дилеру и не присутствовать у другого договора. В случае если необходимо предоставить клиенту возможность изменения подписки через Web необходимо установить дату, с которой эта подписка возможна. При включении этой возможности клиенту будет разрешено включать только те пакеты, которые разрешены к смене через Web.

Номер карты можно не вводить вручную, а выбирать из списка (с быстрым поиском при вводе). Карты и так уже вбиты в менеджере карт и назначаться могут только те карты, которые уже зарегистрированы в системе.

Также реализовать выбор «Базовой карты». При выборе оной, текущая карта становится дочерней. Информация о базовой карте наглядно отображается в табличке на вкладке. Информация о базовой карте отображается, как отдельная колонка. Для каждой карты приписывается её родитель, если есть. А для родителей - количество дочерних. Все они в порядке как есть (будто связей нету).

Ещё параметр «Тип оборудования». Выбирается из справочника. Можно, например, использовать для определения кодировки для отправки сообщений, а также для возможности отправки по Mail и OSD и т.п. См. "Типы оборудования" ниже.

На вкладке **Пакеты** каждой карте сопоставляются пакеты с определённым периодом.

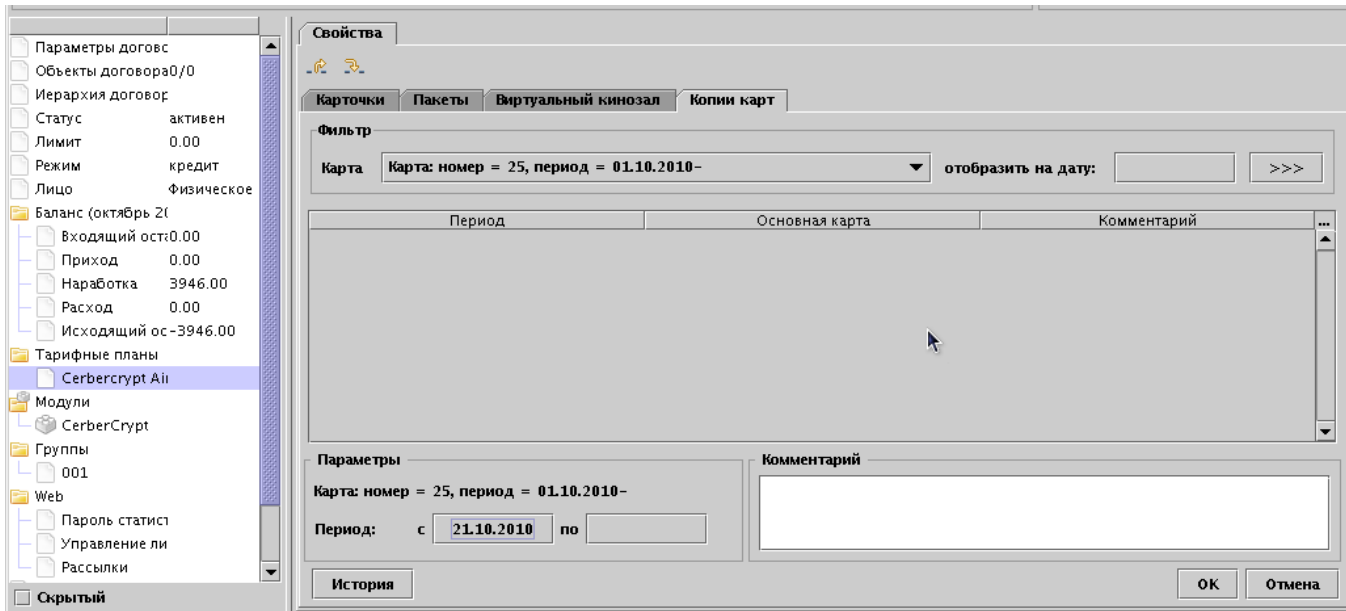


Пакеты можно добавлять и закрывать как по одному, так и группами, используя кнопки **Открыть пакеты** и **Закрывать пакеты** (выделены красным). В течение периода жизни каждый пакет на карте может быть заблокирован (в случае окончания средств), приостановлен (по просьбе клиента) и активен. Заблокированные пакеты автоматически разблокируются при пополнении договора.

С версии 4.3 возможно добавить проверку на наличие в добавляемом пакете канала, который уже есть в открытых пакетах в карте. Для этого в конфигурации модуля указать:

```
cerbercrypt.cardpacket.addcheck=1
# :
#0 -
#1 -
#2 - -
#3 - , -
#
```

Возможно контролировать копии карт. Копии карт реализуют логику наличия нескольких одинаковых карт у клиента. Контроль за их количеством позволяет правильно тарифицировать наличие нескольких карт. Если у клиента есть карта и копия, то они будут тарифицироваться как 2 карты и наработка будет удваиваться. На вкладке "Копии карт" отражаются карты в выпадающем списке и копии к ним в таблице. Для добавления, редактирования и удаления используются стандартные кнопки. Для добавления новой копии, необходимо выбрать основную карту в выпадающем списке и нажать кнопку **Добавить**. Срок действия копии ограничен сроком действия основной карты. Тарифицируются копии с даты начала по дату окончания действия включительно.



Дополнительные возможности для Irdeto Pisyс, STI/NorDE ИТД

В менеджере карт, а также на закладке Карточки в договоре в параметрах модуля доступно контекстное меню. Irdeto Pisyс позволяет отправлять сообщения/письма, устанавливать/сбрасывать PIN(родительский) код,

Отправить сообщение - отправка сообщения на клиентский STB (SetTopBox), привязанный к выбранной (первой из выбранных) карточке. **Отправить сообщение найденным** - отправка сообщения на STB, привязанным к карточкам, найденным по указанному фильтру, т.е. карточкам с первой по последнюю страницу. **Отправить сообщение всем** - отправка сообщений всем карточкам.

Отправить письмо - аналогично отправке сообщения.

Установить/сбросить PIN код - установка/сброс на по умолчанию PIN (родительского) кода.

Загрузка прошивки - указание STB, привязанной к выбранной карточке, что необходимо обновить программное обеспечение.

Привязать/отвязать карту к/от STB - привязка/отвязка карты к/от SetTopBox.

Автонастройка STB - автонастройка SetTopBox, привязанной к карте с IRD.

Активировать карту - повторная активация карты.

Запрос подписки — отображение актуальной подписки с сервера.

Номера карт		Фильтр		
<input type="text"/>		Статус: все	все	
Загрузить		Передать дил.	Забрать у дил.	
>>>		X	1 из 2 [25]	
Номер	Договор	С даты	Дилер	Дата перед
1	x		vbgfv	18.09.2007
2				
3				
4				
5	x		vbgfv	21.04.2006
6			vbgfv	21.04.2006
7			vbgfv	21.04.2006
8				
9				
10	x		21ae	26.04.2006
11	x		21ae	24.05.2006
12	x		vbgfv	21.04.2006
13			vbgfv	21.04.2006
14			vbgfv	21.04.2006
15			vbgfv	21.04.2006
16	p		vbgfv	21.04.2006
17	p		vbgfv	21.04.2006

Отправить сообщение

Отправить сообщение найденным

Отправить сообщение всем

Отправить письмо

Отправить письмо найденным

Отправить письмо всем

Установить PIN код

Сбросить PIN код

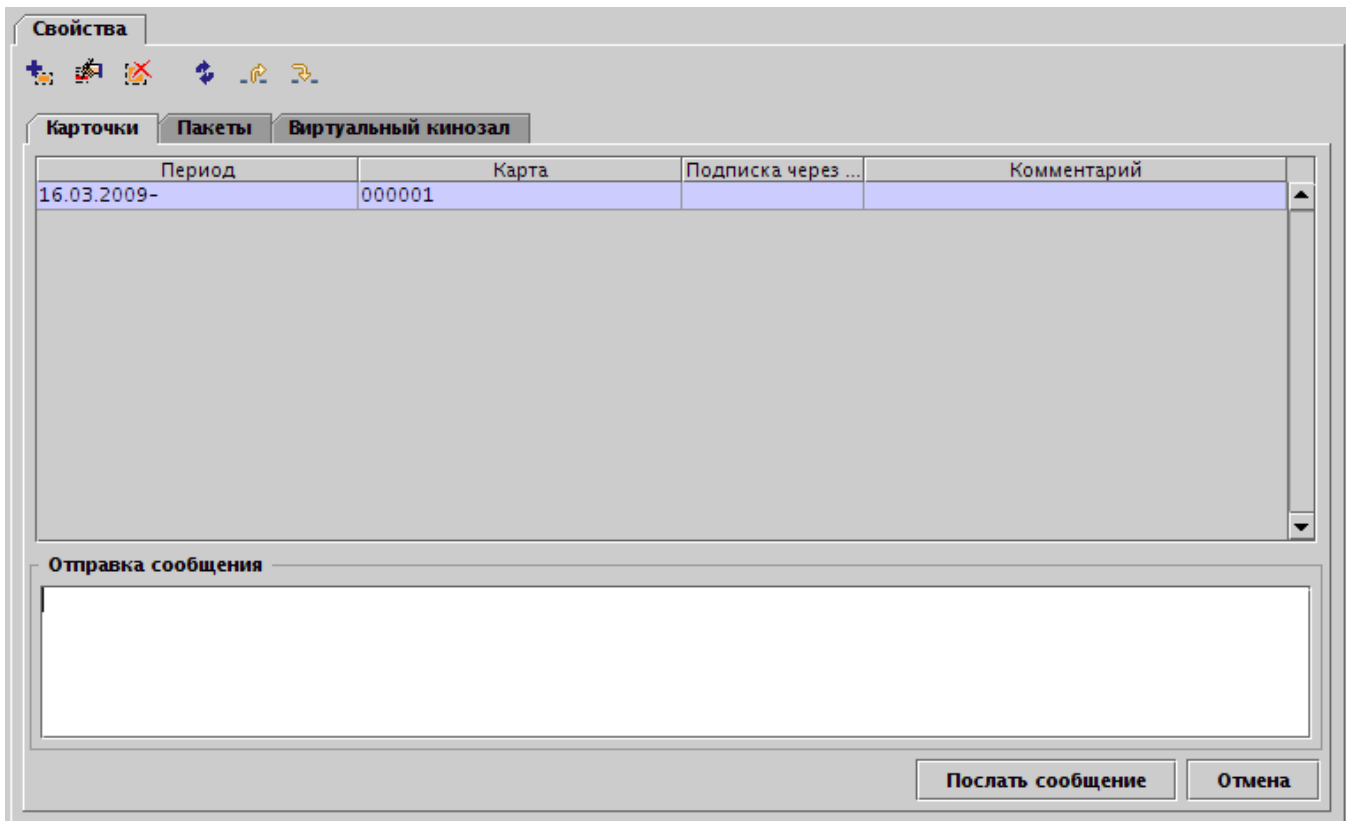
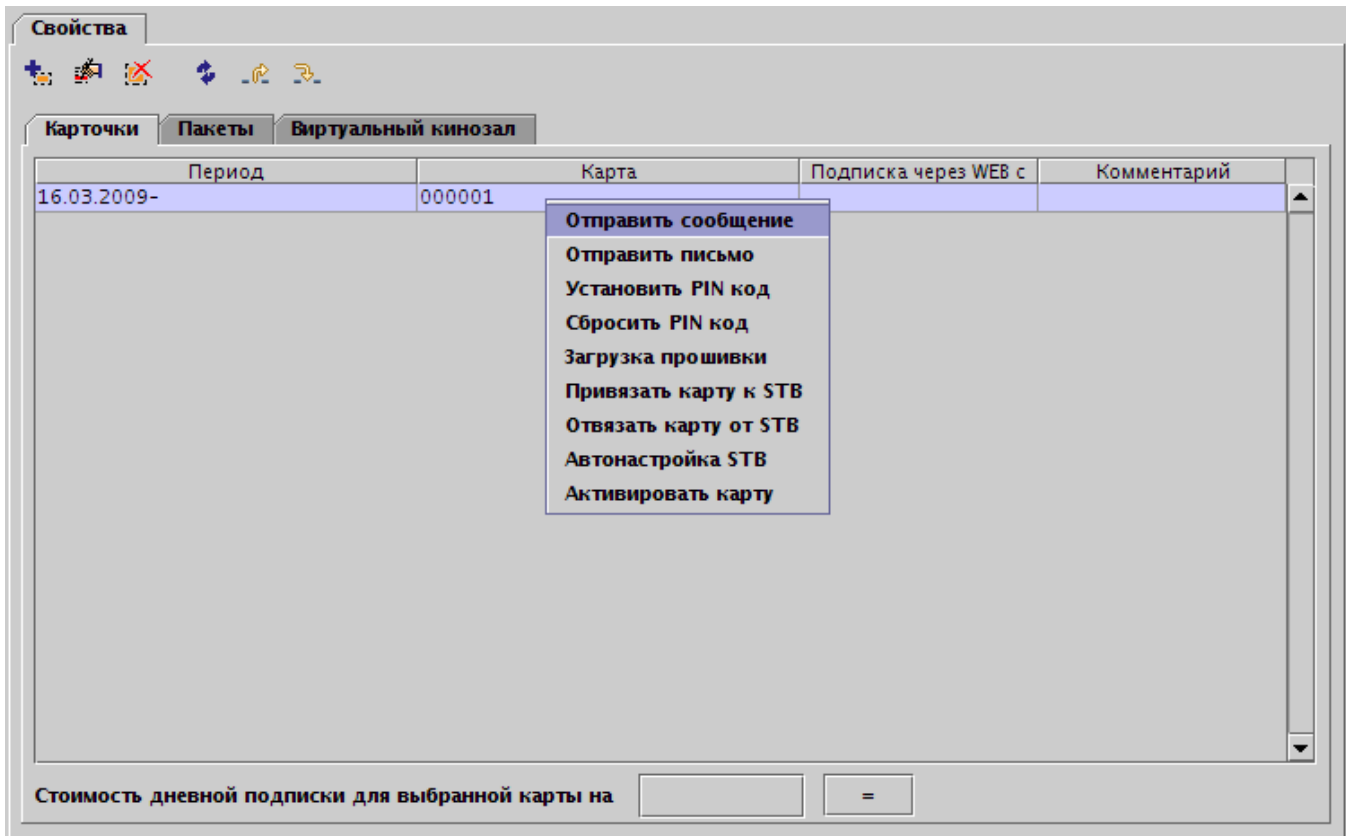
Загрузка прошивки

Привязать карту к STB

Отвязать карту от STB

Автонастройка STB

Активировать карту



Все дополнительные возможности зависят от их поддержки протоколом и возможности их использования при этом в биллинге.

Имеется возможность насильно запустить по карте синхронизацию, при этом ставится в очередь синхронизатор по этой карте и всем зависимым от неё.

Имеется возможность изменения PIN-кода карты из интерфейса модуля CerberCrypt-Менеджер карт (правой кнопкой) и возможность изменения PIN-кода карты из интерфейса модуля CerberCrypt в договоре (правой кнопкой). Фактически это тот же PIN-код, который вводится при создании договора из карты пользователем. Возможность сменить PIN-код дана для возможности использования его как пароля доступа к карте в некоторых системах. Также этот PIN-код можно видеть через интерфейс формирования печатной формы карты регистрации. Есть возможность изменения PIN-кода карты через интерфейс Web-статистики клиента (с указанием старого кода).

При назначении карты договору возможно автоматически отправлять на неё команду открытия карты (команда поддерживается/требуется, например, в протоколе STI/NordE). При снятии карты с договора автоматически отправлять команду отмены регистрации карты. Делается это через скрипты. См. события, возникающие при добавлении (из клиента биллинга) карты на договор и при удалении карты с договора. В скриптах, повешенных на эти события? возможно, например, посылать команды авторизации и отмены авторизации. Активация/деактивация карты также доступна через контекстное меню.

Типы оборудования

Существует справочник «Типы оборудования» для указания типов оборудования, поддерживаемых кодировок (при формировании отправки SMS /OSD/MAIL) и т.д. Это справочник внутри модуля на отдельной вкладке в админке. Справочник: название оборудования — конфигурация — примечание. Где в конфигурации в каком-то нативном для каждого активатора виде расположена конфигурация, которая понятна только этому активатору. Например, для протокола STI/NordE это будет что-то типа: Поле название: железка1. Поле примечание: какая-то железка. Поле конфигурация: osd=1 email=0 encode=utf-8. Потом в каждой вкладке в карточке можно привязать к каждой карточке устройство из комбобокса. При работе с устройством эта конфигурация передаётся в активатор и активатор будет его учитывать согласно карте. Т.е. возможно просто перекрывать поля из конфигурации модуля для каждой конкретной карты пользователя.

«Мультирум»

Основная настройка в конфигурации модуля:

```
# - (" ") ( 0, .. !)  
multiroom.max.slavecards=2
```

При редактировании карты в редакторе список:

- 1) только основные;
- 2) исключая саму редактируемую карту;
- 3) исключая все основные, которые уже имеют максимально разрешённое кол-во копий.

Все действия по удалению базовой карты (мультирум) всё происходит в периодическом синхронизере во вкладках в договоре "пакеты", "виртуальный кинозал" и "копии карт" рисуются только основные карты.

Мультирум в данной реализации работает как полностью копируемая подписка одной карты на другую. Тарификации мультирумных карт в данный момент нет.

Настройка задач планировщика

Для корректной работы модуля должны быть настроены следующие задачи планировщика. Подробности в соответствующих подразделах. Рекомендуемое (ориентировочное) время запуска задач и необходимая (меняйте только если знаете что делаете) последовательность:

- **Установка договоров картам CerberCrypt** - 00:01
- **Установка статусов пакетам карт CerberCrypt** - 00:01
- **Начисление CerberCrypt** - 00:40
- **Блокировка Cerber Crypt** - 01:50
- **Синхронизация CerberCrypt** - 02:00
- **Постепенное продление подписок** - необязательная задача, для некоторых протоколов, выполняется до синхронизации, например, 01:50.
- **Контроль синхронизации** - необязательная задача, период по желанию, например, «каждый час».

Обратите внимание, что задачи начисления (бывает долгая), блокировки и синхронизации (бывает очень долгая) не должны пересекаться в выполнении. Ориентируйтесь на логи и статистику планировщика по времени выполнения задач.

Сопоставление договоров картам

В менеджере задач каждой загруженной карте указывается сопоставленный в настоящий момент договор. В случае, если карта заносится на договор текущей датой, то привязка договора к карте обновляется сразу, иначе это делает задача планировщика **Установка договоров картам CerberCrypt**. Также она снимает эту привязку карта-договор, если период карты на договоре закончился и т.п..

Задача должна быть установлена на 0 часов 0 минут каждого дня.

В конфигурации задачи указывается код модуля.

```
mid=< >
```


Установка актуальных статусов пакетов в картах договоров

Задача **Установка статусов пакетам карт CerberCrypt** устанавливает пакетам на картах договора актуальные статусы, основываясь на заданиях по смене статуса. Данная задача работает для случаев, когда статусы пакетов карты изменяются будущим числом. Обычно это делает **Блокировщик** в случае "блокирования со следующих суток" - блокирует не сразу, а вешает задание.

Должна запускаться в 0 часов 0 минут каждых суток, в параметрах указывается код экземпляра модуля.

```
mid=< >
```

Начисление CerberCrypt

Осуществляется задачей планировщика **Начисление CerberCrypt**. Рекомендуемая частота запуска - один раз в сутки. Целесообразно запускать задачу в первом часе каждого суток (например, в 0 часов 30 минут каждые сутки), тарификация производится включая текущие сутки с учётом актуальных статусов пакетов, установленных задачей **Установка статусов пакетам карт CerberCrypt**.

В конфигурации задачи указывается код модуля.

```
mid=< >
```

Блокировка должников

Осуществляется задачей **Блокировка CerberCrypt**. Минимальная частота запуска - один раз в сутки после задачи начисления. Если в договоре присутствуют иные виды списаний кроме начисления за использование цифрового ТВ, то задачу можно установить несколько раз в сутки.

В конфигурации задачи указывается код модуля. При блокировке пакеты всех договоров, баланс которых менее лимита, переводятся в статус **заблокировано**.

Параметр **forgive.hours** определяет сколько часов от суток "прощать" клиенту. Под "прощением" понимается, что если задача запускается ранее указанного часа суток, то блокировка производится мгновенно и прошедшие часы достаются клиенту даром. Если задача запускается после этого часа, до добавляется задание на блокировку с 0 часов последующих суток, клиент может смотреть канал до конца суток (это задание обрабатывается задачей **установка актуальных статусов пакетов CerberCrypt**). "Прощение" определённого числа часов предотвращает недовольство клиентов в связи с отключением каналов в 0 часов, как это должно следовать из требования корректности начислений (при снятии денег посуточно блокировка должна происходить на границе суток).

```
mid=< >
#      " "
forgive.hours=2
```

В данном примере если блокировка производится после 2х часов ночи, статус **заблокировано** устанавливается со следующего дня, позволяя абоненту доработать день (и он будет протарифицирован).

Если у пакета установлен параметр "Не блокируемый", то в этой задаче он обрабатываться не будет.

Обновление подписок

Осуществляется задачей планировщика **Синхронизация CerberCrypt**. Для всех активных в текущий день карт открываются указанные на текущий момент пакеты со статусом **активен**. Данная задача производит синхронизацию подписок по результатам работ задач **Установка статусов пакетам карт CerberCrypt** и **Блокировка Cerber Crypt**. Обновление подписок при их смене оператором, либо клиентом через Web интерфейс происходит мгновенно.

В параметрах задачи указывается код модуля и адрес E-Mail на который высылаются сообщения о недоступности сервера CerberCrypt.

Также можно указать игнорируемые карты или группы договоров (к которым в данный момент привязаны карты, которые нужно игнорировать)

```
mid=< >
error.mail=bill@bill.ru
#
#ignore.cards=
#
#ignore.contract.groups=
```

Также можно игнорировать по дилерам, см. not.synchronized.dealers в настройках модуля.

Постепенное продление подписок

Специфика некоторых систем (CTI/NordE, сопах и т.д.) такова, что при указании подписки они требуют период активации, в отличие от протокола SerberSgurt, например, где подписка ставится/снимается сиюминутная. В данном случае существует примерно такое стандартное поведение:

* открываем пакет с закрытой датой закрытия — так и отсылаем;

* открываем пакет с бесконечностью — ставим большой правый период.

Встаёт задача: как-то решить недостаток от продления пакета в бесконечность и последующим отключением пользователем самого себя от системы с целью пользоваться картой вечно. Сократить подобное пользование до разумного периода. Т.е. в том числе это защита от "хитрых клиентов".

Как написано в конфигурации для активатора CTI/NordE при активации на бесконечный срок можно сделать период активации, например, 30 дней от начала периода, после, если есть деньги, перепосылать активацию (т.е. каждый день, выходит, ибо карты в разные дни активируются).

Это делается через задачу планировщика **Постепенное продление подписок пакетов**. При этом в БД создается отдельная таблица «продление пакетов/каналов», поля: entityId — идентификатор картпакета, date — дата последнего продления «сущности». Таблица не критичная, её содержимое можно удалить, тогда просто считается, что последний раз для закрытых периодов картпакетов уже было отослано, а для открытых было отослано в начале открытия на месяц, так что при следующем запуске этой задачи все они снова продлятся. В любом случае всё продлится и заново начнётся отсчёт при любом редактировании картпакета.

Сам алгоритм имеет такую логику: запускается раз в сутки, перебирает все картпакеты, активные в текущий момент. Активные картпакеты могут быть:

а) либо с закрытым периодом, тогда ничего не делаем, ибо полагаем, что уже было всё отослано один раз;

б) либо с открытым, тогда берётся для каждого картпакета из указанной таблички одна дата — дата последнего успешного продления (точнее, до какого числа РЕАЛЬНО было продлено).

Если истекает эта дата (а системный открыт), тогда он заново отсылает команду с датой начала из картпакета (т.е. дата начала всегда одинакова будет, а конец каждый месяц отодвигаться) и датой конца из этой таблички+месяц. После этого новая реальная дата снова записывается в БД и через месяц снова будет использоваться.

Период (например, как выше названо «месяц») настраивается в задаче самой конфигурации модуля (т.е. в конфигурации данного активатора).

```
period.gradually.subscription=30
```

Если в конфигурации указан небесконечный период, то это обязует использовать эту задачу для периодического продления подписки. Этот период используется как в самом модуле (для информации о том, на сколько конкретно продлить при открытии на бесконечный период), так и в этой задаче для информации о том, на сколько сделать последующее продление.

Пользователь открывает пакет в биллинге, указывает дату начала активации и конца активации: на карту шлются эти периоды, в таблице реальный период и системный совпадают.

Пользователь открывает пакет в биллинге, указывает только дату начала активации пакета: на карту шлётся период ДАТА_НАЧАЛА — ДАТА_КОНЦА, где ДАТА_КОНЦА = ДАТА_НАЧАЛА + 30 дней; в таблицу записывается реальный период и системный (реальный — ДАТА_НАЧАЛА — ДАТА_КОНЦА, а системный ДАТА_НАЧАЛА — (открытая_дата))

Пользователь меняет период активации пакета в биллинге, даты активации перепосылаются и перезаписываются в базу с учетом вышестоящих двух пунктов.

Контроль синхронизации

Имеется специальная задача контроля синхронизации, которая отвечает за корректную отправку синхронизаций по картам. То есть, если в момент синхронизаций сервер УД был недоступен, то подписка уйдёт только в лучшем случае при следующем изменении карты или на границе суток, когда запустится задача общей синхронизации (если она будет успешна). Задача контроля синхронизации следит за картами с флагом "требуется синхронизации" и при наличии карт с ним ставит синхронизатор по ним снова в очередь. Задача необязательна, период настраивается также по желанию нужной оперативности при возможных сбоях сервера УД.

У каждой карты пользователя для этого есть флажок "требуется синхронизации". При любом изменении карты он ставится, т.е. это значит "надо синхронизовать", после успешной синхронизации (в синхронизаторе) оно сбрасывается и это значит "карту уже не надо синхронизовать", т.е. "подписка актуальна". Это же поле выводится в табличке для карт пользователя (см. выше).

При нормальной работе эта задача будет просто проверять, что ошибок синхронизации нет (отсутствуют "грязные" несинхронизованные карты) и завершаться. Использовать с осторожностью в плане периодов. Если система подразумевает таймауты обращения и синхронизаторы не просто завершаются с ошибкой при ошибке доступа, а висят и ждут, то при использовании этой задачи их может скопиться слишком много в запущенном состоянии.

Настройка тарифных планов

В тарифном плане должны быть определены цены за каждый пакет в день, либо месяц. Также дополнительно могут быть определены скидки в зависимости от числа пакетов, на которые подписан пользователь. Логика поиска тарифа соответствует [Алгоритму 1](#).

Тарифный план 'Стандартный'

- СерберСпут
 - Услуга: CerberСпут
 - Скидка 15.0 % если пакетов = 3
 - Скидка 24.0 % если пакетов = 4
 - Скидка 29.0 % если пакетов = 5
 - Скидка 32.0 % если пакетов = 6
 - Скидка 35.0 % если пакетов = 7
 - Скидка 38.0 % если пакетов = 8
 - Скидка 40.0 % если пакетов >= 9
 - Период: 01.02.2006
 - Пакет '1 Extreme Sports' цена 26.0 за календарный месяц
 - Период: 01.02.2006
 - Пакет '2 Nickelodeon' цена 26.0 за календарный месяц
 - Период: 01.02.2006
 - Пакет '3 RTVi' цена 17.0 за календарный месяц
 - Период: 01.02.2006
 - Пакет '4 Discovery Science' цена 39.0 за календарный месяц
 - Период: 01.02.2006
 - Пакет '5 Discovery Civilization' цена 39.0 за календарный месяц
 - Период: 01.02.2006
 - Период: 01.04.2006
 - Период: 01.04.2006
 - Период: 01.04.2006

Также возможен формат дерева следующего вида:

СерберСпут

- Услуга: CerberСпут
 - Не участвуют в скидке пакеты:
 - Пакет 1
 - Пакет 2-3
 - Пакет 4-5
 - Скидка 50.0 % если пакетов >= 1
 - Пакет Пакет 1
 - 1.0 за день
 - Пакет Пакет 2-3
 - 50.0 за календарный месяц

Модуль предоставляет возможность выставления скидки в зависимости от количества пакетов, при этом можно указать пакеты, не участвующие в расчёте скидки. Т.е. они не учитываются при определении количества пакетов, в зависимости от которого выдаётся скидка и на них не применяется скидка.

Есть скидка по количеству копий карт, данная скидка применяется после начисления всех скидок по пакетам.

Возможности Web-интерфейса модуля

Через Web интерфейс пользователю предоставляется текущая подписка его карт. При разрешении администратором пользователь может устанавливать задания на смену подписки с определённой даты.

Договор N x0000

Управление подпиской карт SerberCrypt

Номер карты:

Текущая подписка:

Пакет	Период	Текущее состояние
Пакет 2-3	08.02.2006 -	активна
Пакет 4-5	21.04.2006 -	активна
Пакет 1	20.05.2006 -	активна

Дневная стоимость текущей подписки: 1.97

Планы по смене подписки:

Пакет	Период	Дневная стоимость	Удалить
<input type="button" value="Открыть"/>	<input type="text" value="Пакет 1"/>	с <input type="text"/>	<input type="text" value="январь"/> <input type="text" value="2006"/> <input type="button" value="Выполнить"/>

С помощью скриптов поведения можно формировать вид и поведение смены подписки через web. Можно формировать (при желании) списки дат открытия и закрытия, а также обрабатывать события "перед открытием" и "перед закрытием".

Итак, для смены подписки через web действуют следующие события:

- 1) Получение списка пакетов, которые можно открыть через Web;
- 2) Получение списка пакетов, которые можно закрыть через Web;
- 3) Перед открытием/закрытием пакета через Web. С возможностью передать ошибку и прервать изменение подписки;
- 4) Подписка изменена.

События и скрипты для них аналогичны действиям при "смене тарифного плана через веб". Например, можно прервать смену, при условии малого баланса.

```
//      - ,
if( balance.compareTo( changeCost ) < 0 )
{
    //
    event.setProcessed( true );
    event.setError( " " );
    return;
}
```


Закрытые карты отображаются в Web для истории, но сделать с ними ничего нельзя. По старым подпискам пользователь может посмотреть историю пакетов как для активных, так и для закрытых карт.

Создание договоров пользователем через Web

Модуль поддерживает создание договоров самим пользователем через специальную страницу. Таким образом возможно распространение карт и приемников через агентскую сеть.

URL страницы активации:

http://<адрес сервера биллинга>:8080/bgbilling/pubexecuter?action=CreateContract&module=cerbercrypt&mid=<код модуля>

Шаблон страницы - файл **cerbercrypt_create_contract.xml** - в нем необходимо подправить параметр формы mid на правильный код модуля.

Дополнительно необходимо указать в конфигурации модуля:

```
#
#
web.create.pattern=< >
#
web.create.subscr.offset=< >
#
web.create.address.param=< >
#
web.create.fio.param=< >
#
web.create.name.pattern=#####
#
web.create.packet.< 1>=<- >
web.create.packet.< 2>=<- >
```

Для того, чтобы создать договор по карте вам необходимо:

Введите номер карты:

Введите пароль карты:

привязать договор к адресу:

Улица: дом / кв. комн.

привязать ФИО

разрешить смену подписки через сайт

Виртуальный кинозал

Виртуальный кинозал - это тип пакета, который может быть активирован на сутки. При активации с Web-статистики - с текущего момента времени, при активации с клиента биллинга - либо с текущего момента, либо с текущего момента, либо с текущего момента в будущем. Если кинозал активирован текущим числом, то деньги снимаются со счета в момент активации, иначе - при тарификации в дату активации.

В модуле должна быть дополнительно создана услуга "Виртуальный кинозал". Активация кинозала через Web-интерфейс доступна договору только при наличии в нем данной услуги. Код услуги указывается в конфигурации модуля следующим образом:

```
# ,  
cerbercrypt.virtual_cinema.serviceId=<sid>
```

В пакете необходимо указать, что это Виртуальный кинозал, а также включить подписку через Web, если нужно дать возможность клиенту самому активировать кинозал через Web-статистику.

Период	Канал
01.08.2007 -	Канал 05

При активации кинозала через клиент биллинга можно указать дату и час активации, если час активации пуст, а дата текущая, то кинозал активируется с текущей минуты, если дата активации в будущем, то активируется с 0 часов. В клиенте отображаются кинозалы, активные на указанную дату.

Кинозал	С часа	По час	Стоимость
Виртуальный кинозал	19.09.2007 15:53	20.09.2007 15:53	12.00

Activation section: Activate cinema hall: Virtual Cinema, from: 27.09.2007, 2 hours, Activate

При активации с Web-статистики сначала проверяется наличие уже активированного в данный момент времени такого же кинозала, период карточки и наличие средств на счете для активации.

Также через Web-статистику возможно удаление кинозалов, активированных будущим числом.

Виртуальный кинотеатр CerberCrypt

Номер карты:

Текущая подписка:

Кинозал	С часа	По час	Стоимость
Виртуальный кинозал	19.09.2007 15:53	20.09.2007 15:53	12.00

Будущая активация:

Кинозал	С часа	По час	Стоимость	Удалить
Виртуальный кинозал	27.09.2007 02:00	28.09.2007 02:00	12.00	<input type="button" value="Удалить"/>

Активировать кинозал:

Лог синхронизаций

Имеется возможность ввести низкоуровневый лог синхронизаций с удалённым сервером, где в дополнение к полям: юзер_карта, время, общий результат, пишется произвольной строкой некоторая полезная информация. Конфигурация модуля:

```
# . .
# - .
# .
# , . 0-9223372036854775807. * - .
logging.card.range=1-10,58
# , , :
# create, modify, cancel, sendMessages, parentalPin, downloadCode, marrySmartcard,
# tune, patchSmartCard, recvrSectDtcode, activate, deactivate, getsubscribeinfo
logging.commands=create,modify,cancel
```

Лог синхронизаций видится из общей вкладки настройки модуля. При клике на комментарий его можно развернуть. Для логов ведётся периодическая таблица в БД.

Например, для CerberCrypt помимо битовой маски выводятся также номера каналов, которые были отправлены.

При нажатии на карту пользователя в договоре можно из всплывающего меню выбрать "Лог соединений по карте" и автоматически открыть модуль с логом соединений, уже отфильтрованным по номеру карты.

Модуль TV

Назначение модуля

Модуль предназначен для интерактивной интеграции с TV/IPTV Middleware и CAS-системами, организует доступ к услугам, пакетам и каналам, их подключение/отключение из личного кабинета и приставок, тарификацию в реальном времени с точностью до секунды и минимальным периодом тарификации - 1 минута.

На данный момент поддерживаются системы [Ministra TV platform](#) (infomir, ex Stalker Middleware), [FrontStage Middleware](#) (Mediastage.tv, TelecomTV, ВСС), [IPTV Портал](#), [СТТ TVEngine](#), [NextTV](#) (Наука-Связь), [Смотрёшка](#) (LifeStream API, lfstrm.tv, smotreshka.tv), [Commpass](#) (ТИТВ), [24 часаТВ](#), [TVIPmedia](#), [MOOVI](#) (Медиаоператор).

Базовые сведения о модуле

Базовые понятия модуля:

- **Продукт** - абстракция, которая может содержать в себе один или несколько сервисов TV или представлять собой пакет каналов, услугу или тариф MW/CAS. Именно на продукт осуществляется подписка;
- **Сервис** - абстракция, которая может содержать в себе один или несколько каналов TV или представлять собой пакет каналов, услугу или тариф MW/CAS;
- **Канал** - канал MW/CAS;
- **Подписка** - период, когда продукт подключен у аккаунта;
- **Аккаунт** - отражение аккаунта в MW/CAS, дочерний аккаунт - STB (на дочерний аккаунт невозможно активировать подписку);
- **Тип аккаунта** - определяет параметры, которые должны быть указаны у аккаунта;
- **Устройство** - в дереве устройств определяется иерархия устройств разного типа, имеющих значение для модуля. Обычно это устройство Access+Accounting, отражающее приложения BGTVAcess и BGTVAccounting, и дочернее по отношению к нему устройство, отражающее систему управления MW;
- **Тип устройства** - определяет поведение устройства, механизм управления аккаунтами и подписками на продукты на устройствах данного типа;

Не путайте понятия тип устройства и устройство.

Приложения модуля:

- **BGTVAcess** - выполняет синхронизацию аккаунтов в MW/CAS, управляет доступом аккаунтов к подписанным услугам/пакетам /каналам;
- **BGTVAccounting** - выполняет тарификацию подписок.

Связь между приложениями осуществляется посредством базы данных и MQ-сообщений.

После очередного обновления модуля необходимо в **Автоматизация->Управление динамическим кодом** скомпилировать все классы, т.к. перекompиляция после обновления автоматически не происходит, а классы, входящие в сборку, могли обновиться.

Настройка модуля

Установите модуль на сервер, создайте экземпляр. Определите в **Редакторе модулей и услуг** услуги, обчисляемые этим модулем. Например: "Подписка IPTV". Услуги используются для разделения наработки по типам в балансе договора. Установка услуги происходит в тарифе, в зависимости от продукта, на который осуществляется подписка и других параметров.

В конфигурации модуля укажите:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

#
# ( ):
# , " , ,
```

```
#
# ru.bitel.bgbilling.modules.tv.api.server.bean.TvAccountGenerate
#account.login.class=
#
#account.login.min=1
# (. . . 1,2,3 10000000,
# 4, 10000001)
#account.login.min=9999999
#
#account.login.format=0000000

# . , ,
# ( ):
# , " , ,
#
# ru.bitel.bgbilling.modules.tv.api.server.bean.TvAccountGenerate
# ru.bitel.bgbilling.modules.tv.server.bean.TvAccountGenerate ( 8.2101+)
#account.password.class=
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMN0PQRSTUVWXYZabcdefghijklmnopqrstuvwxy
# ,
account.password.chars.description= .
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=
```

Продукты, Сервисы, Каналы

Продукт - сущность на которую осуществляется подписка (активация продукта/активация подписки). Начисление наработки согласно тарифу может происходить как при активации продукта, так и периодически, согласно указанному в тарифе периоду тарификации.

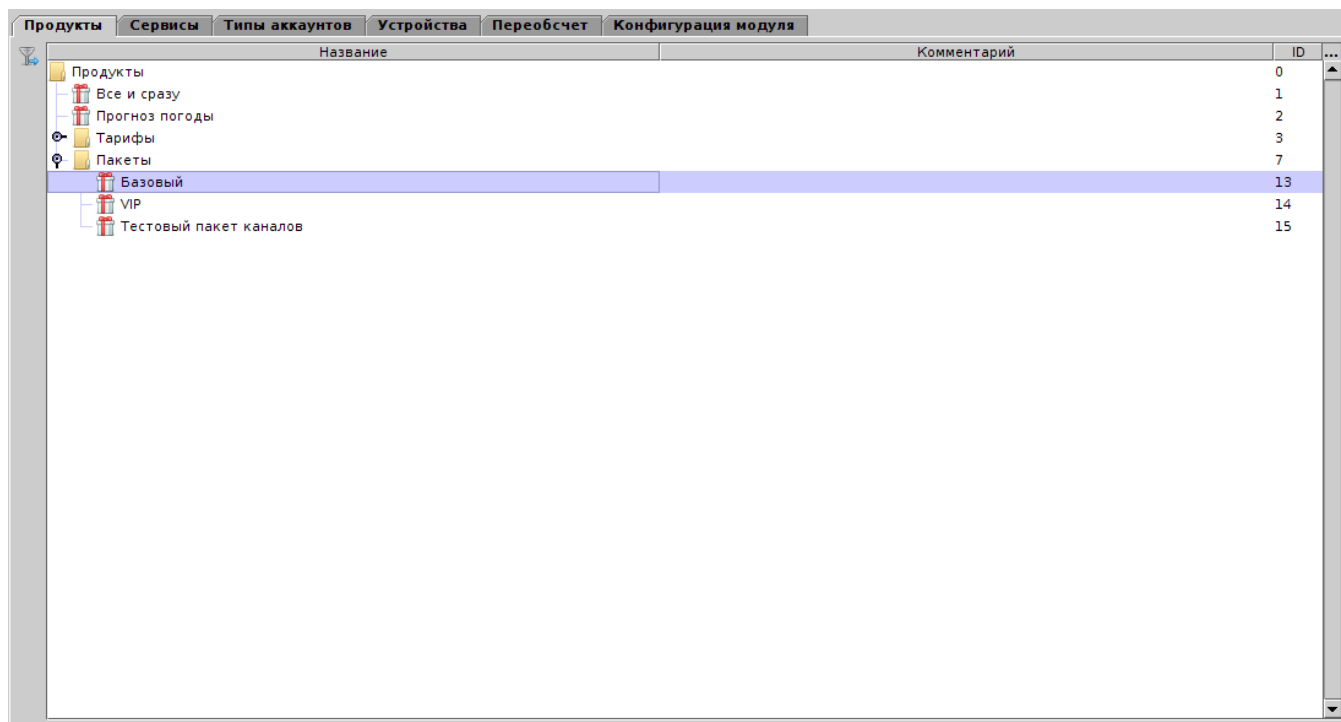
Продукт может отражать пакет каналов, услугу или пакет услуг из MW/CAS. Например, в MW/CAS заведены тарифы, на которых, внутри MW/CAS прописаны каналы и услуги, и при этом подписка будет осуществляться только на эти тарифы. Тогда достаточно будет завести продукты-тарифы MW/CAS и указать в продуктах соответствующие идентификаторы тарифов MW/CAS. В этом случае интеграция с MW/CAS будет на уровне продуктов.

В случае, если группировка в пакеты в MW/CAS отсутствует, такую группировку можно сделать средствами модуля, т.к. продукт может содержать в себе сервисы. Когда активен продукт - активны все сервисы, указанные в нем. Например, можно создать продукт "Все включено", в котором будут добавлены сервисы "Базовый пакет" и "VIP-пакет", отражающие соответствующие пакеты в MW/CAS (и с прописанными идентификаторами пакетов MW/CAS). В данном случае интеграция будет на уровне сервисов.

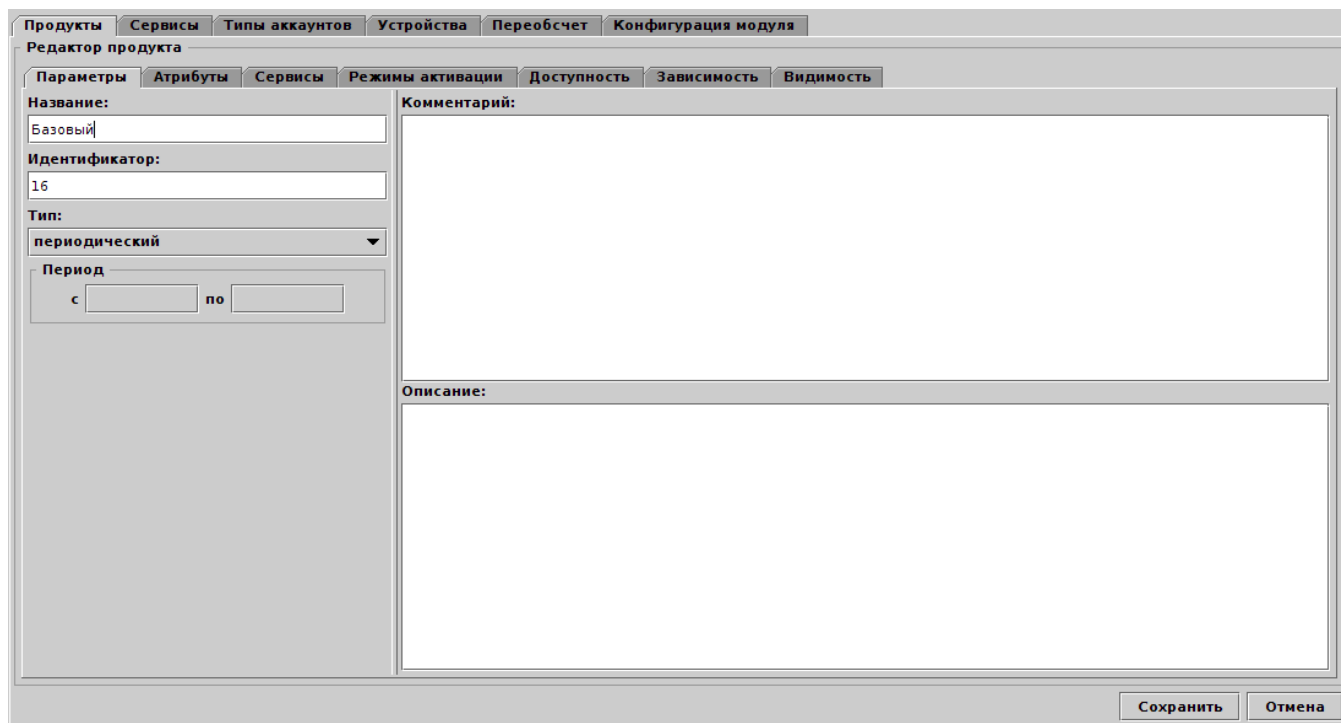
В случае, когда группировки каналов в пакеты нет, или же есть возможность подключать каналы без пакетов, а группировка внутри MW/CAS не желательна, можно интегрировать модуль на уровне каналов, т.к. сервис может содержать в себе каналы. В этом случае потребуются полная связка Продукт - Сервисы - Каналы.

Продукты TV

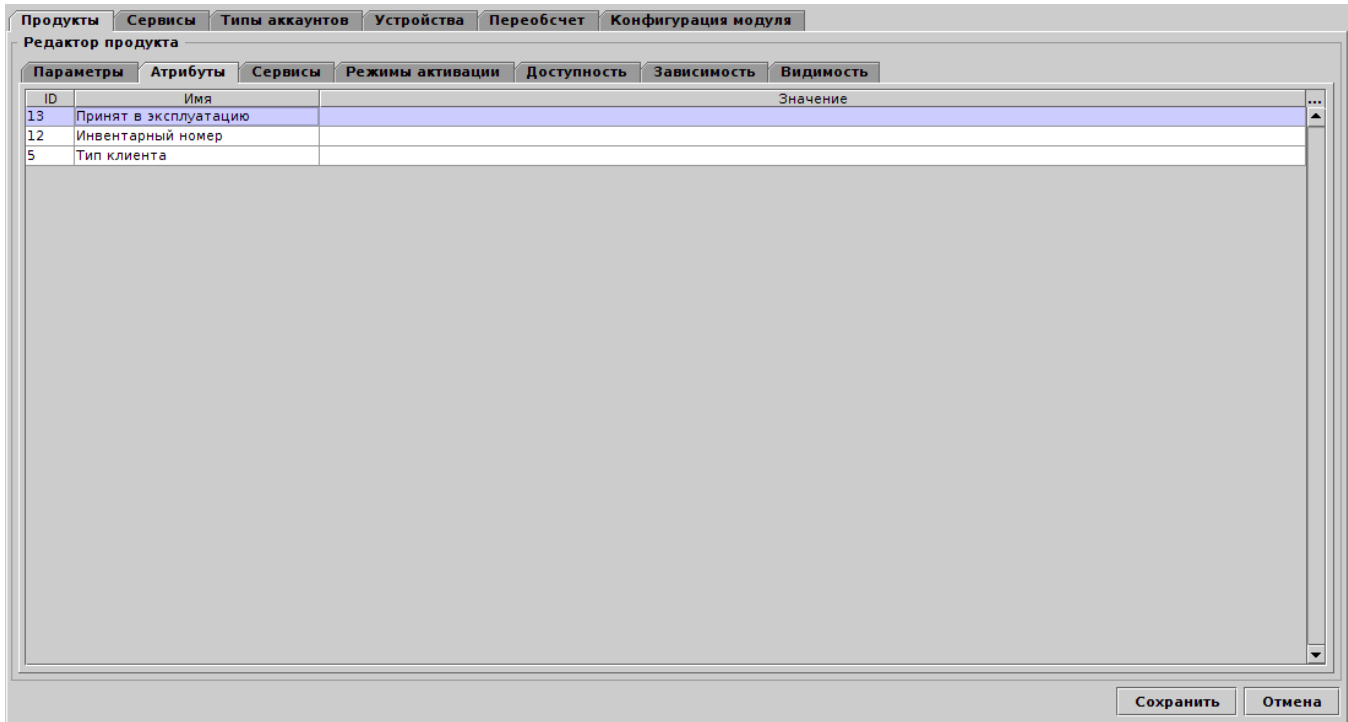
Для удобства продукты (а точнее - их спецификации/описание) можно добавлять в виде дерева.



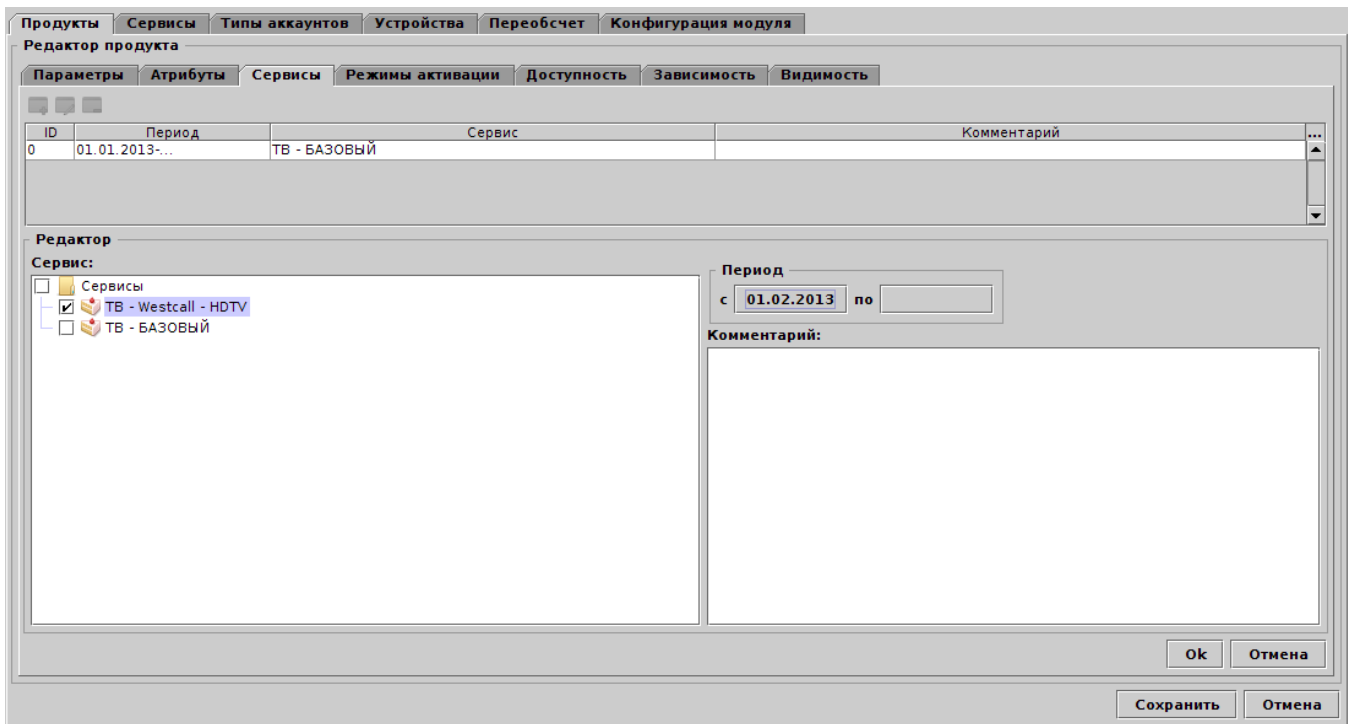
На вкладке Параметры необходимо указать название, тип, период действия спецификации продукта, а также идентификатор сущности в MW/CAS системе, если данный продукт будет отражать какую-то сущность MW/CAS.



Атрибуты сущности используются опционально и позволяют указывать произвольные параметры.



Если необходимо (например, когда интеграция с MW/CAS создается на уровне сервисов или каналов), в спецификации продукта указывается список привязанных к нему спецификаций сервисов. Таким образом, когда у аккаунта будет активен продукт, будут и активны все указанные здесь сервисы. Для того чтобы привязать спецификацию сервиса к спецификации продукта, сначала ее нужно создать.



У каждой спецификации продукта, на который будет осуществляться подписка, должен быть добавлен хотя бы один режим активации. В режиме активации указываются длительность (0, если действует бесконечно или до деактивации), время начала действия, возможна ли деактивация (для бесконечных подписок) или реактивация (для деактивированных подписок, период действия которых еще не закончился).

Для большинства продуктов-пакетов длительность (поле **Активация на**) следует установить как 0 (минут/дней/месяцев) - в этом случае у подписки не будет даты окончания и подписка будет действовать до деактивации. Следует отличать общий период продукта-пакета и периодов пролонгации, за которые производится начисление. Последние вместе с ценой указываются в тарифе.

Следует также учитывать, что при выбранном режиме "с текущего", "со следующего", появляется зависимость от выбранного календарного периода. Например, "0 месяцев с текущего" - означает, что пакет активируется от 1-ого числа текущего месяца в 00:00, "0 дней с текущего" - пакет активируется от 00:00 текущего дня.

ID	Период	Название	Деактивация	Реактивация
13	01.07.2013-...	10 минут		
14	01.07.2013-...	подписка	до конца оплаченного периода	запрещена

Редактор
 Название: Период с: по:
 Активация на: минут с: ровно: Деактивация: Реактивация:
 [Ok] [Отмена] [Сохранить] [Отмена]

На вкладке Доступность указывается тарифные планы и группы договоров, для которых подписка на данный продукт будет доступна. Если ничего не указано - доступно всем.

Тарифные планы:

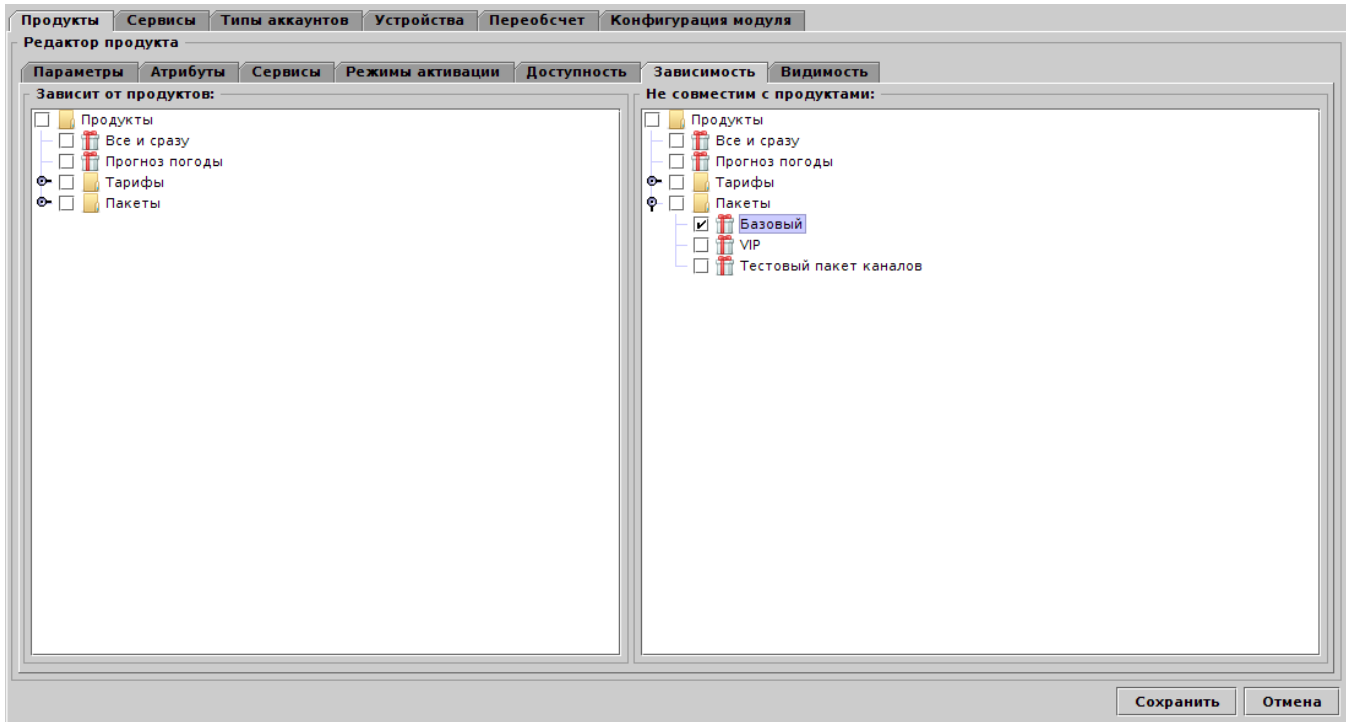
- A5
- И: Test_Abon
- Карточки в подарок
- Карточки на реализацию
- квартиры
- комбинированный трафик для асвт-нн
- коммерческие предприятия
- Новый
- повременный трафик для асвт-нн
- Схема 1 (без НДС)
- Схема 2 (без НДС)
- тариф VoIP
- Тариф2
- Тариф3
- Телефония
- Телефония: Оператор
- Тест (подписки)
- Тест-трафик
- Тестовый
- Тестовый тарифные опции
- TP1
- TP2
- Учет. период.
- физические лица
- эфирный

Группы договоров:

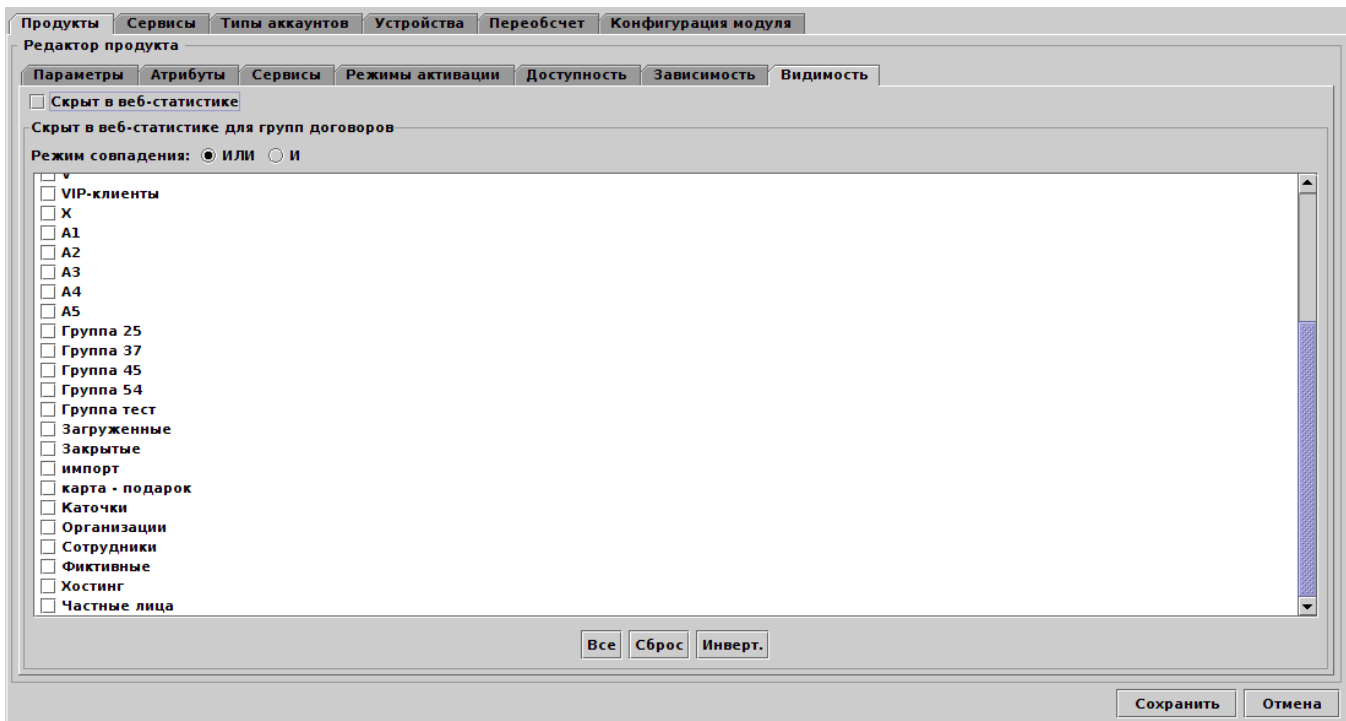
- M
- S
- V
- VIP-клиенты
- X
- A1
- A2
- A3
- A4
- A5
- Группа 25
- Группа 37
- Группа 45
- Группа 54
- Группа тест
- Загруженные
- Закрытые
- импорт
- карта - подарок
- Каточки
- Организации
- Сотрудники
- Фиктивные
- Хостинг
- Частные лица

[Все] [Сброс] [Инверт.] [Все] [Сброс] [Инверт.] [Сохранить] [Отмена]

На вкладке Зависимость указывается от каких продуктов он зависит и с какими не совместим. Т.е. если происходит попытка активации продукта, зависящего от другого, то период подписки другого продукта должен включать в себя период подписки нового продукта. И наоборот для несовместимости - период подписок данных продуктов не должен пересекаться. Для того, чтобы один продукт нельзя было активировать несколько раз, в продукте нужно указать, что он несовместим сам с собой.

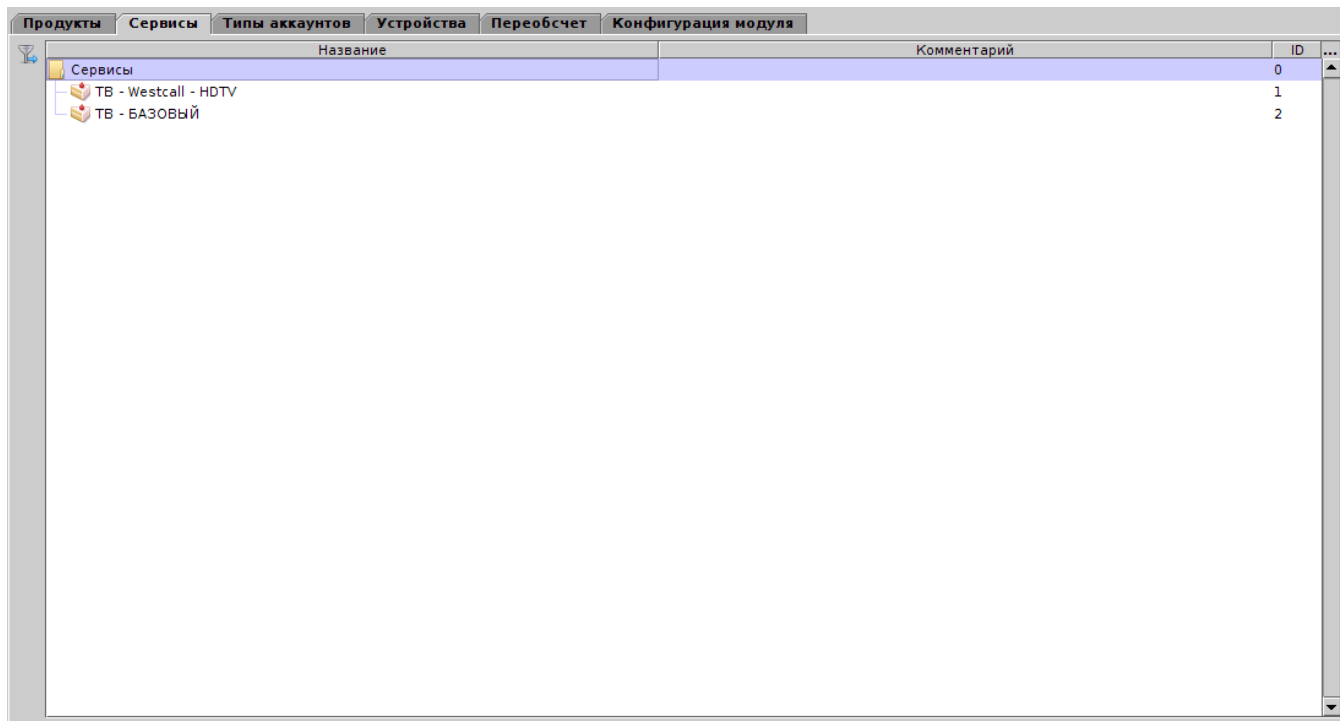


Если продукт необходимо не показывать в личном кабинете абонента для определенных групп договоров или же не показывать всем, то это нужно указать на вкладке Видимость.



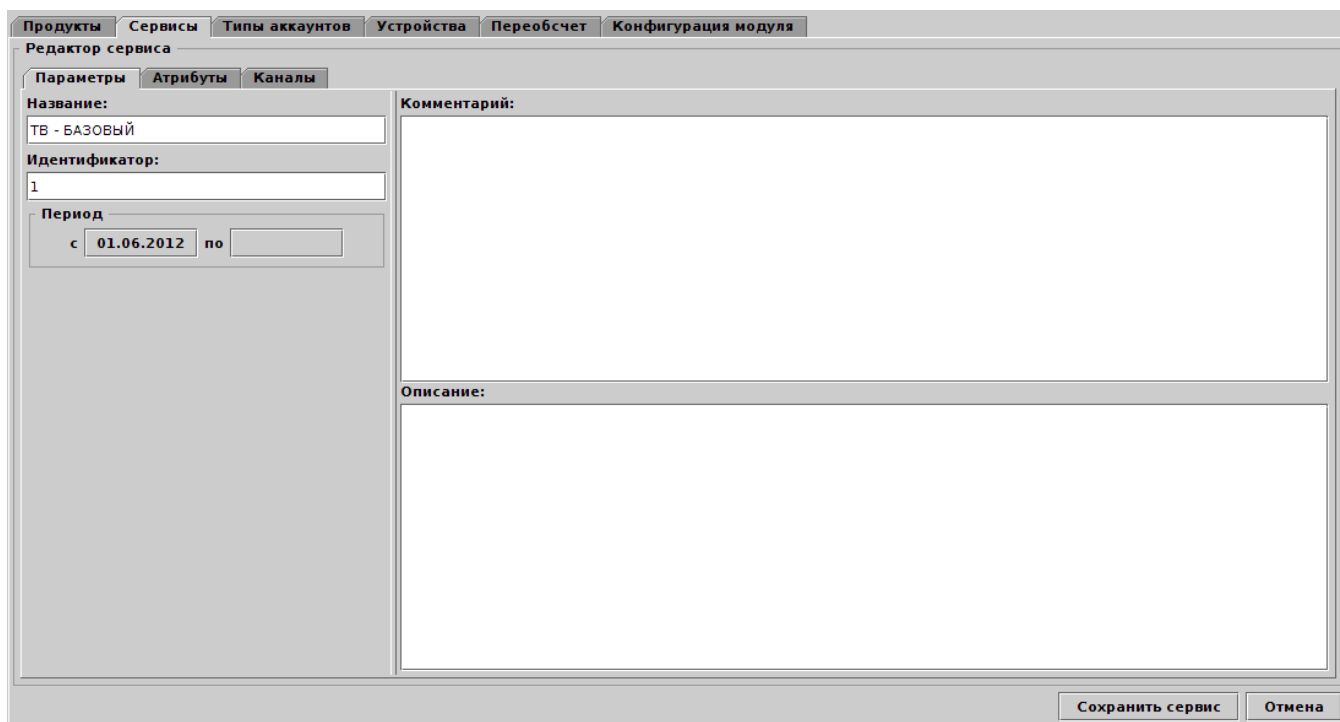
Сервисы TV

Для того, чтобы привязать спецификации сервисов к спецификации продукта, сначала их нужно создать. Для удобства спецификации сервисов можно создавать в виде дерева.



Название	Комментарий	ID
Сервисы		0
ТВ - Westcall - HDTV		1
ТВ - БАЗОВЫЙ		2

Для спецификации сервиса необходимо указать название, период действия и идентификатор сущности в MW/CAS, если данный сервис отражает какую-то сущность (например, пакет каналов) в MW/CAS.



Редактор сервиса

Параметры | Атрибуты | Каналы

Название: ТВ - БАЗОВЫЙ

Идентификатор: 1

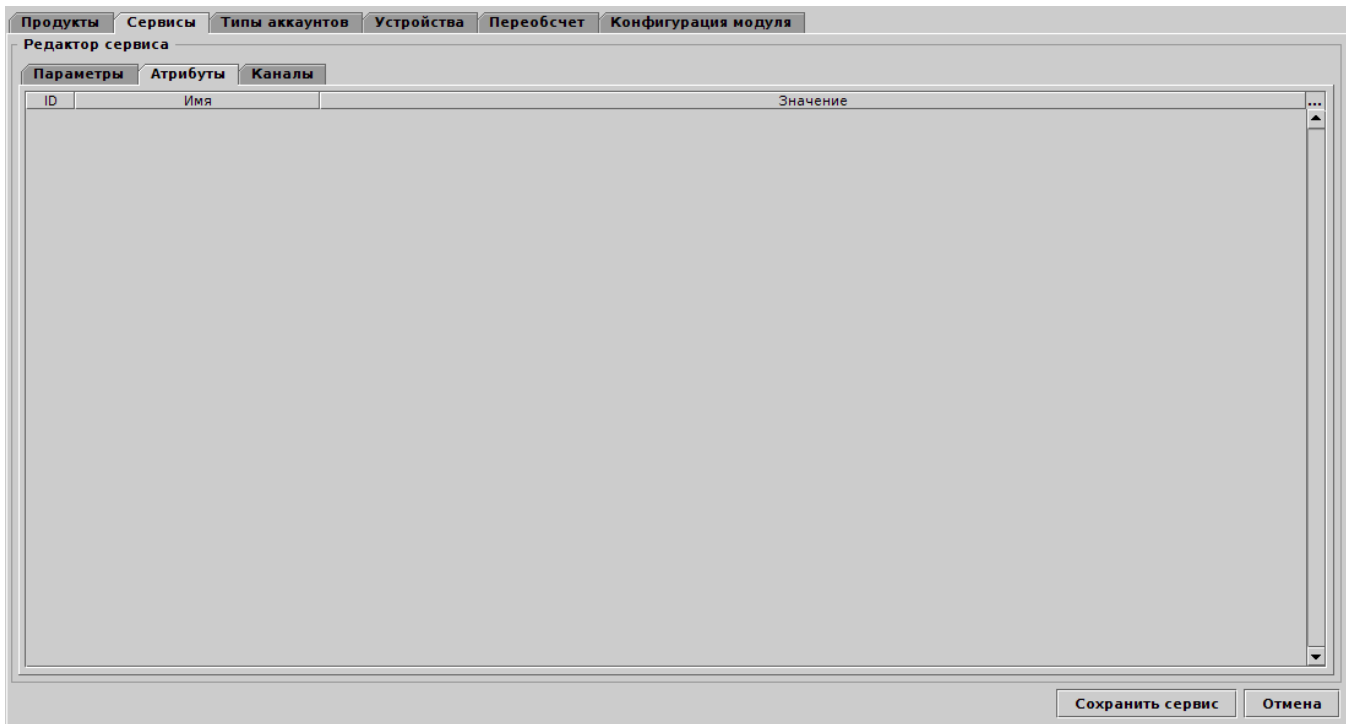
Период: с 01.06.2012 по

Комментарий:

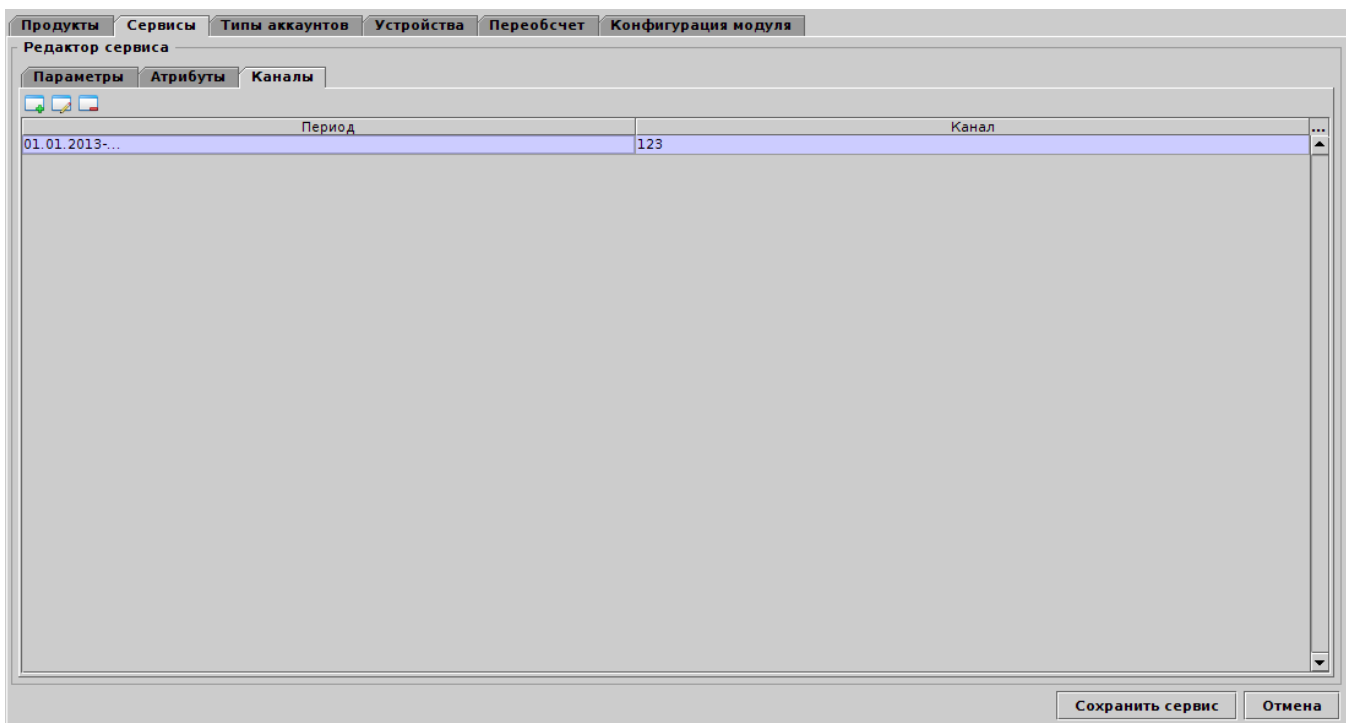
Описание:

Сохранить сервис | Отмена

Атрибуты сущности используются опционально и позволяют указывать произвольные параметры.



Если интеграция с MW/CAS осуществляется на уровне каналов, то в спецификации сервиса необходимо указать список каналов, которые будут доступны при активности данного сервиса. Таким образом сервис может являться пакетом каналов, настроенным в биллинге.



Каналы TV

В канале указывается название, период действия и идентификатор канала в MW/CAS.

Продукты Сервисы Типы аккаунтов Устройства Переобсчет Конфигурация модуля

Редактор сервиса

Параметры Атрибуты Каналы

Редактор канала

Параметры Атрибуты

Название:
Discovery Channel

Идентификатор:
123

Период
с 01.01.2013 по

Комментарий:

Сохранить канал Отмена

Сохранить сервис Отмена

Атрибуты сущности используются опционально и позволяют указывать произвольные параметры.

Продукты Сервисы Типы аккаунтов Устройства Переобсчет Конфигурация модуля

Редактор сервиса

Параметры Атрибуты Каналы

Редактор канала

Параметры Атрибуты

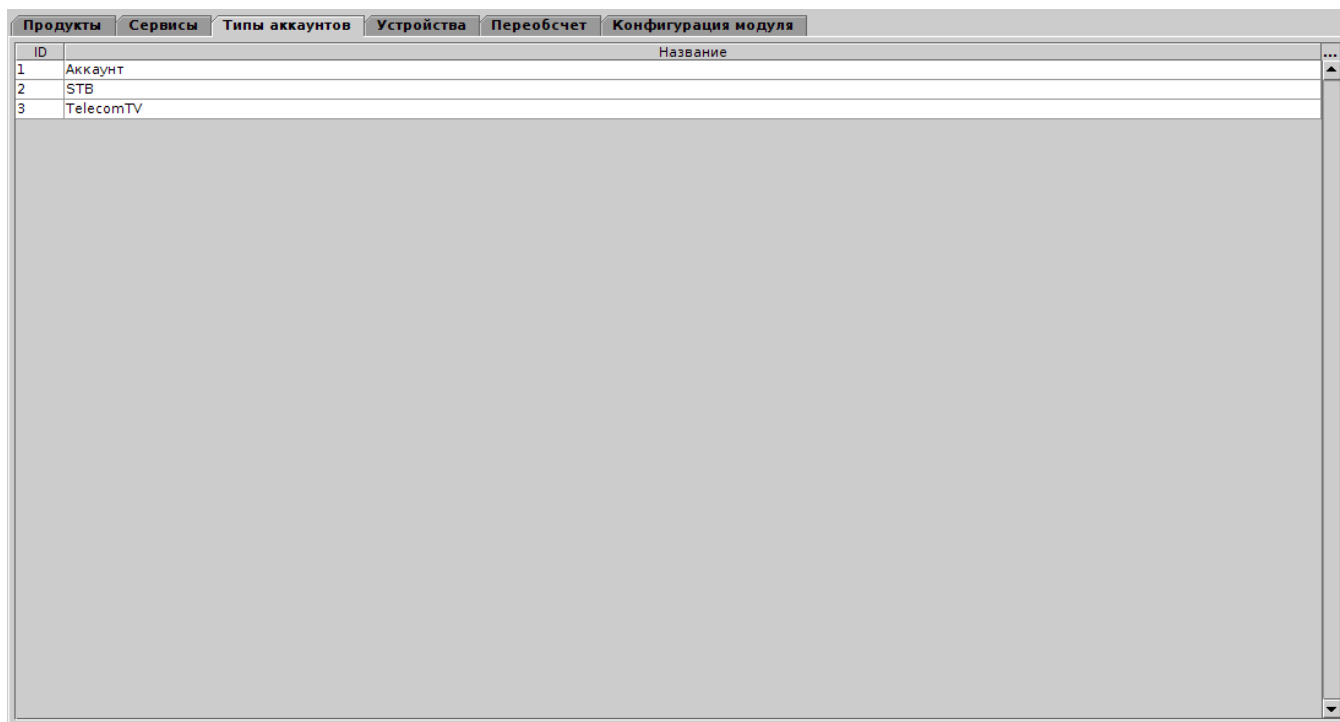
ID	Имя	Значение
----	-----	----------

Сохранить канал Отмена

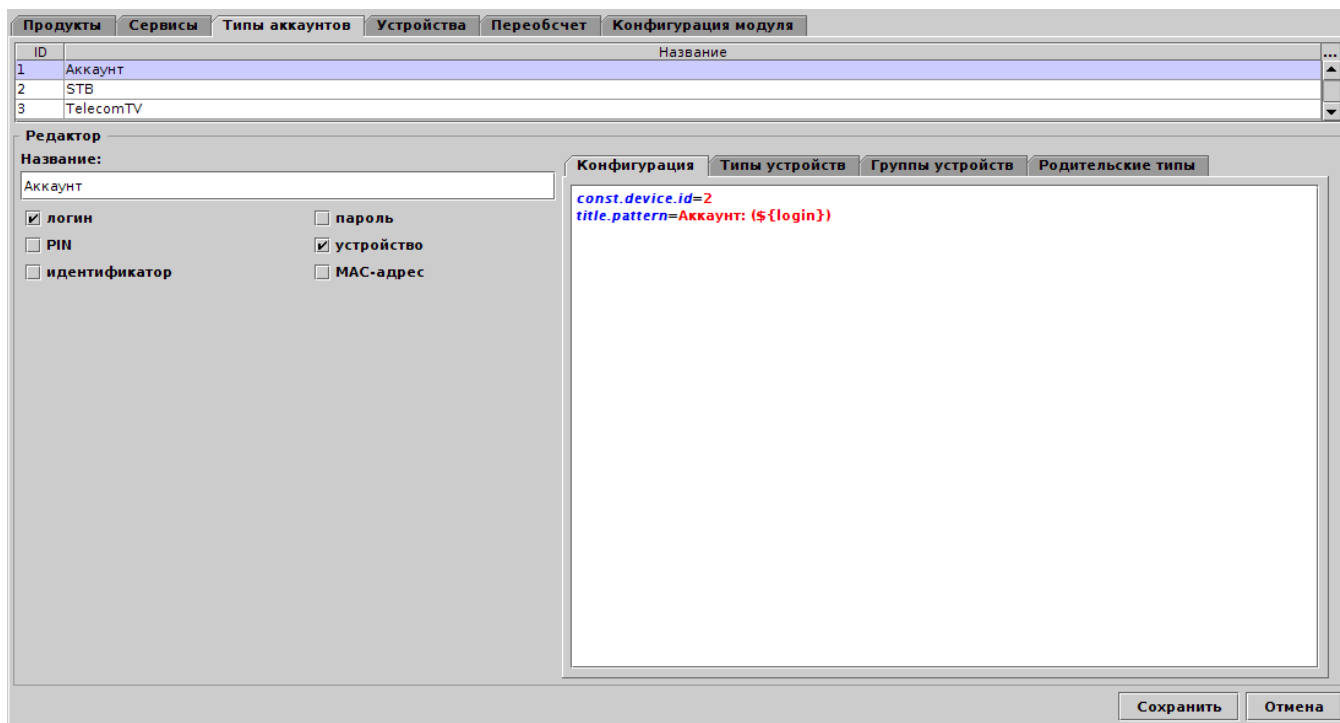
Сохранить сервис Отмена

Типы аккаунтов TV

Для того, чтобы добавить аккаунт на договор, нужно сначала создать тип аккаунта.



Тип аккаунта представляет собой описание того, что должно быть в аккаунте, конфигурацию (с указанием шаблона названия аккаунта) и указание родительских типов, если необходимо.



Флаги логин, пароль, PIN, устройство, идентификатор и MAC-адрес указывают на наличие данного поля при редактировании аккаунта данного типа.

В конфигурации аккаунта в параметре **const.device.id** необходимо указать Id устройства, к которому будет привязан добавляемый аккаунт (устройство-MW/CAS) и шаблон названия аккаунта:


```
# Id ,
const.device.id=
#
title.pattern=: (${login})
```

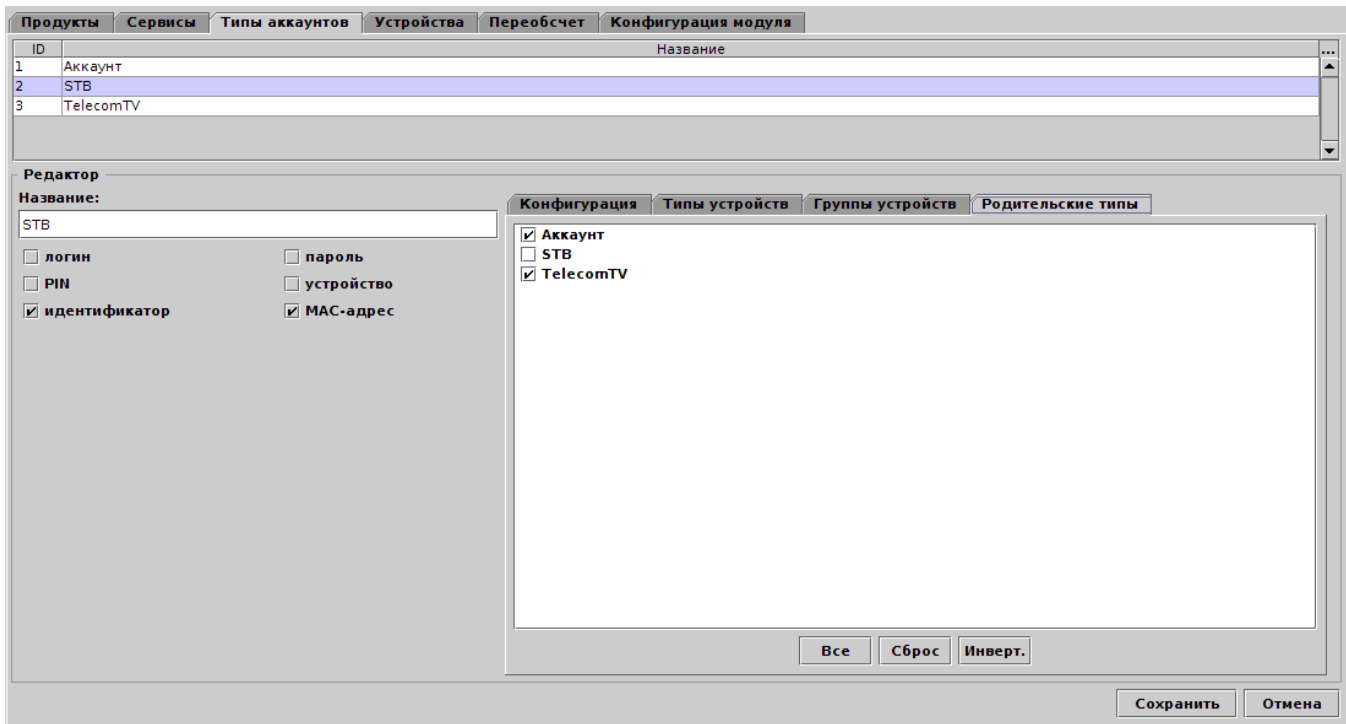
В параметре **title.pattern** можно использовать макросы вида (`${macros}`), где вместо `macros` можно указать:

- **deviceIdentifier** - идентификатор устройства, к которому привязан аккаунт,
- **deviceTitle** - название устройства, к которому привязан аккаунт,
- **login** - логин аккаунта,
- **id** - Id аккаунта,
- **interfaceId** - интерфейс,
- **vlan** - VLAN,
- **identifier** - идентификатор аккаунта,
- **macAddress** - MAC-адрес аккаунта.

Для каждого типа STB может быть свой тип аккаунта, например, с различной конфигурацией. Пример типа аккаунта описывающего STB-приставку:

The screenshot shows a software interface with a menu bar at the top containing: **Продукты**, **Сервисы**, **Типы аккаунтов**, **Устройства**, **Переобсчет**, and **Конфигурация модуля**. Below the menu is a table with columns **ID** and **Название**. The table contains three rows: 1 | Аккаунт, 2 | STB (highlighted), and 3 | TelecomTV. Below the table is a **Редактор** section with a **Название:** field containing 'STB'. To the left of the editor are several checkboxes: логин, PIN, идентификатор, пароль, устройство, and MAC-адрес. To the right of the editor is a **Конфигурация** tab with a text area containing the following configuration: `title.pattern=STB: (${identifier}) (${macAddress})`, `terminal.typeId=15`, and `terminal.bandwidth=3`. At the bottom right of the editor are **Сохранить** and **Отмена** buttons.

Аккаунт-приставка должен быть дочерним к обычному аккаунту - для этого в нем нужно указать родительские типы аккаунтов.



Пример конфигурации типа аккаунта:

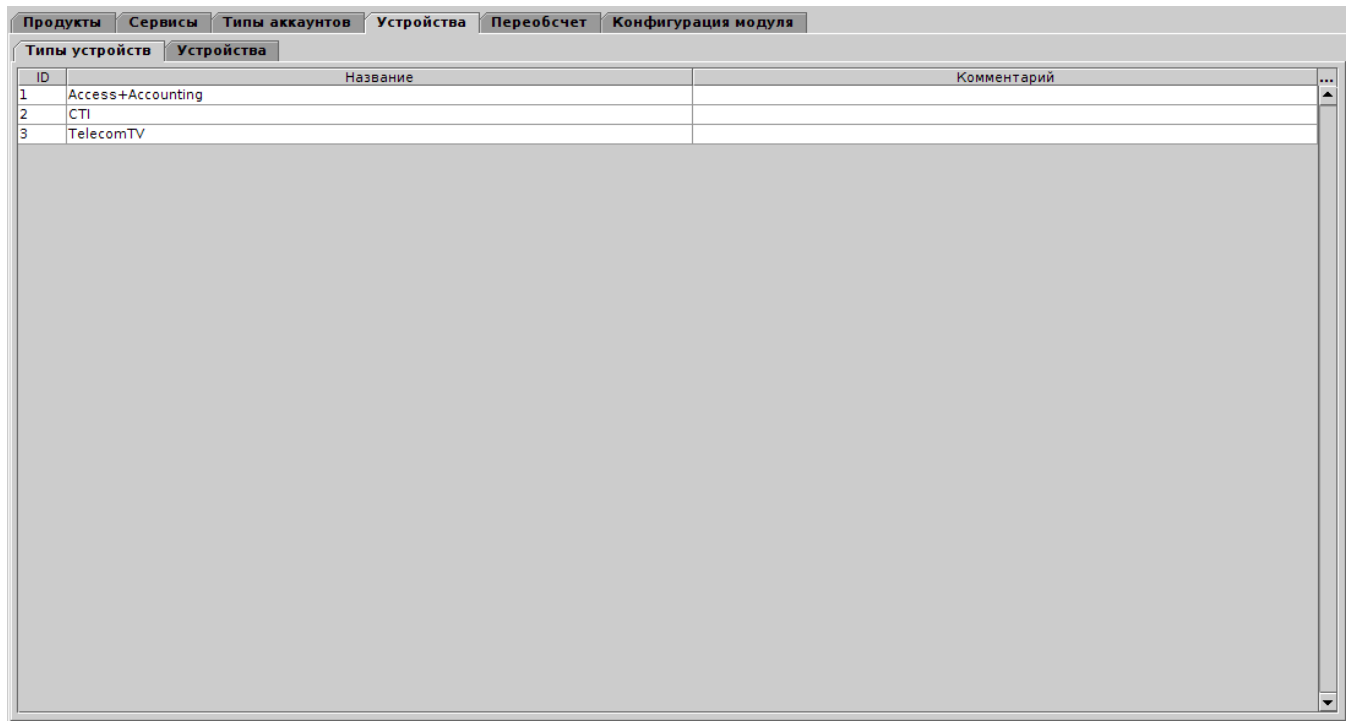
```
#
title.pattern=${login})

#
#
#const.device.id=

#
# ( ):
#
#account.password.length.min=5
#
#account.password.length.max=16
# ( )
#account.password.chars=1234567890ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyz
#
#account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars )
#account.password.chars.auto=<account.password.chars>
```

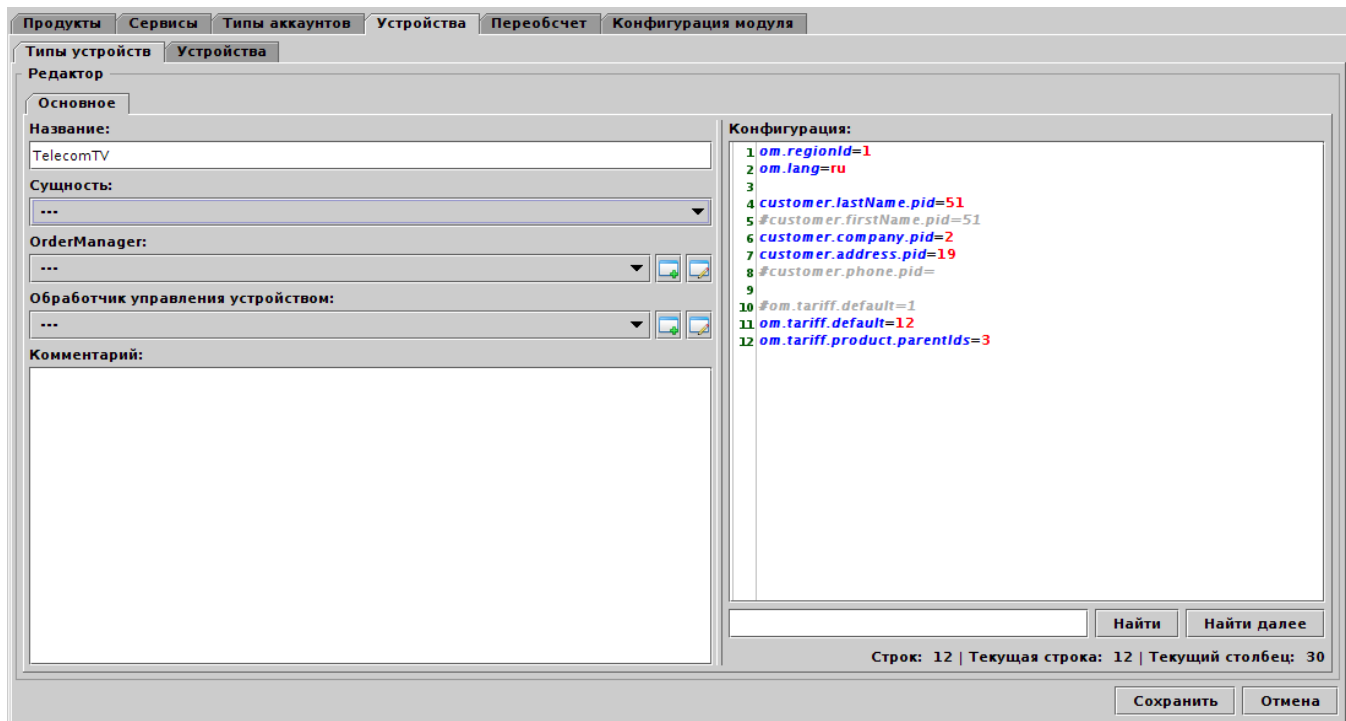
Типы устройств TV

Тип устройства описывает поведение и функции устройства.



ID	Название	Комментарий
1	Access+Accounting	
2	СТП	
3	TelecomTV	

Обычно достаточно создать два типа устройства - Access+Accounting, отражающий приложения TvAccess и TvAccounting, и тип устройства, отражающий систему MW/CAS. На скриншоте тип устройства для FrontStage Middleware.



Редактор

Основное

Название: TelecomTV

Сущность: ...

OrderManager: ...

Обработчик управления устройством: ...

Комментарий:

Конфигурация:

```
1 om.regionId=1
2 om.lang=ru
3
4 customer.lastName.pid=51
5 #customer.firstName.pid=51
6 customer.company.pid=2
7 customer.address.pid=19
8 #customer.phone.pid=
9
10 #om.tariff.default=1
11 om.tariff.default=12
12 om.tariff.product.parentIds=3
```

Строк: 12 | Текущая строка: 12 | Текущий столбец: 30

Сохранить Отмена

Сущность - сущность из Справочники - Атрибуты. Определяет опциональный набор атрибутов, доступных при редактировании устройства,

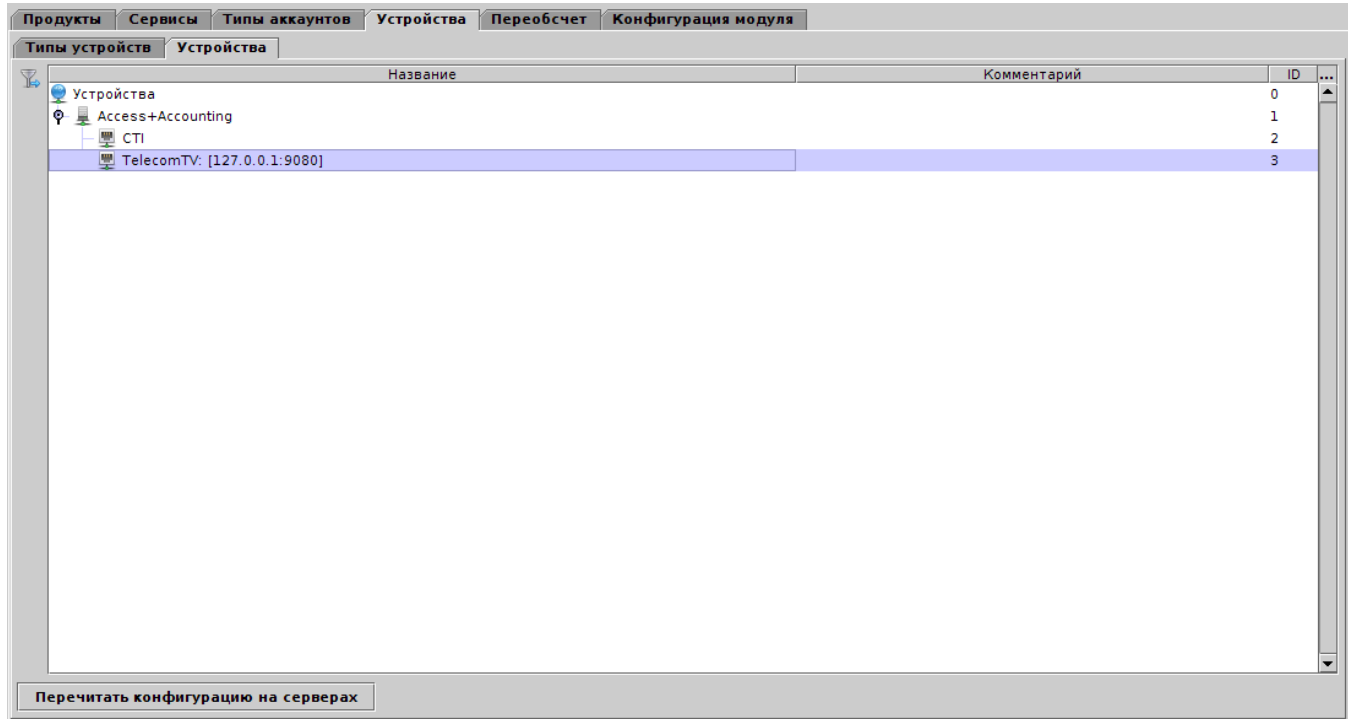
OrderManager - динамический класс, осуществляющий большую часть интеграции с MW/CAS системой со стороны биллинга.

В конфигурации типа устройства указываются параметры, специфичные для выбранного OrderManager.

Устройства TV

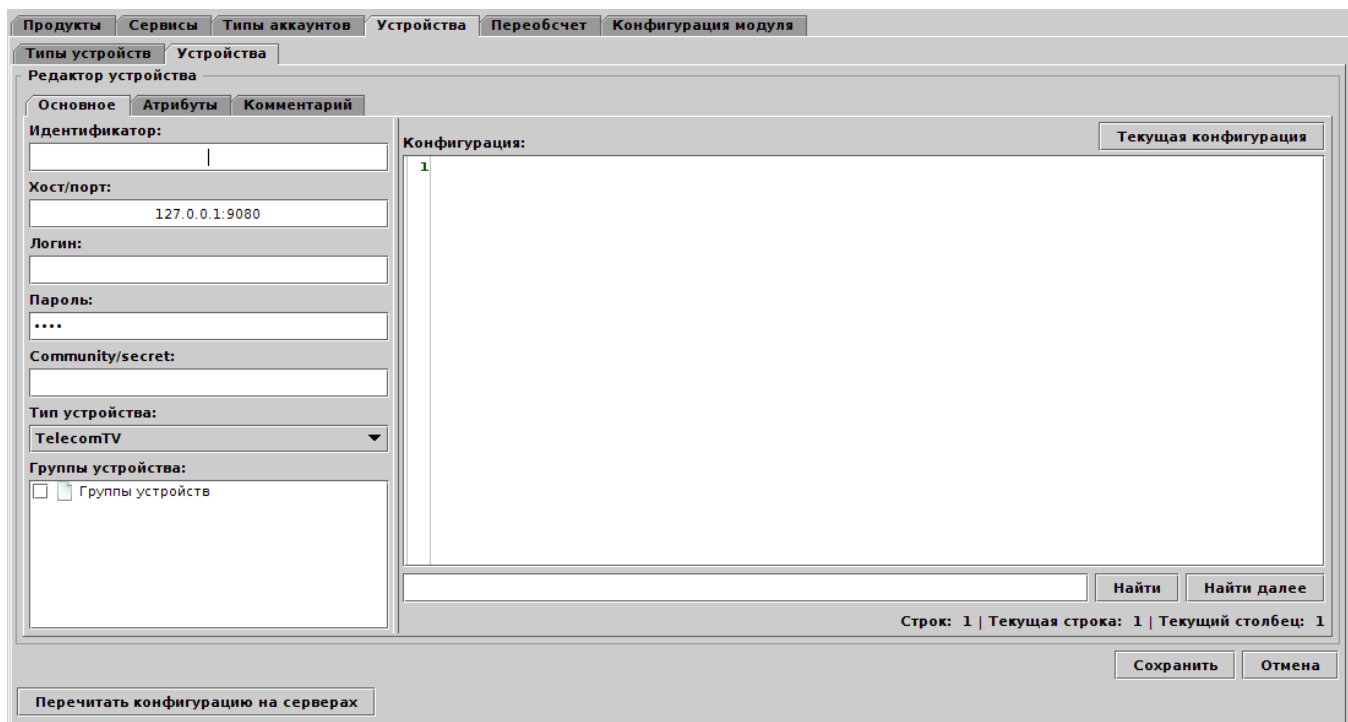
Описание

Дерево устройств состоит из корневого устройства Access+Accounting и дочернего к нему устройства-MW/CAS. Корневое устройство отражает приложения BGTVAcess и BGTVAccounting.



В параметрах устройства-MW/CAS системе указываются параметры, с помощью которых OrderManager будет подключаться к системе и выполнять синхронизацию. Обычно это хост/порт и логин/пароль для подключения к системе.

Аккаунты необходимо привязывать именно к этому устройству.



На вкладке Атрибуты можно опционально указать дополнительные параметры, набор которых определяется сущностью, назначенной в пункте Сущность типа устройства.

Корневое устройство

Корневое устройство в древовидной схеме представляет собой приложения TvAccess и TvAccounting. Соответственно конфигурация корневого устройства предназначена для этих приложений.

Пример конфигурации:

Конфигурация

```
# /
# -
accounting.worker.1.thread.count=1
# , -
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
# , -
#
#accounting.worker.1.tracking.account.1.deactivateProductsWithoutPrice=1
#
# , -
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400
```

Аккаунты TV

Аккаунты привязываются к договору на соответствующей вкладке модуля в договоре.

Сервис	Тип	Период	Статус	Состояние	ID
Аккаунты					
Аккаунт: aaaa	Аккаунт	01.08.2012-...	открыт	отключен	1
Аккаунт: test	TelecomTV	01.07.2013-...	открыт	включен	10
STB: 123456 [001CC05EC110]	STB	02.07.2013-...	открыт	включен	14
Аккаунт: bbb	TelecomTV	01.07.2013-...	открыт	включен	11
Аккаунт: 123457	Аккаунт	04.07.2013-...	открыт	удален	15
Аккаунт: 123458	Аккаунт	04.07.2013-...	открыт	удален	16

При редактировании аккаунта доступны параметры, указанные в выбранном типе аккаунта.

Аккаунты

Параметры **Продукты**

Тип
TelecomTV

Период
с 01.07.2013 по

Статус
Открыт

Устройство
TelecomTV: [127.0.0.1:9080] <<<

Логин
test авто

Пароль
..... авто

Комментарий

Состояние
Включен

Сохранить Отмена

В дочернем аккаунте недоступна активация продуктов (подписки). В данном случае дочерний аккаунт является SetTopBox'ом:

Аккаунты

Параметры

Тип
STB

Период
с 02.07.2013 по

Статус
Открыт

MAC-адрес
00:1C:C0:5E:C1:10

Идентификатор
123456

Комментарий

Сохранить Отмена

К аккаунту привязываются активированные продукты (подписки).

Аккаунты

Параметры **Продукты**

Текущие **История**

ID	Продукт	Описание	Начало	Окончание	Состояние	...
89	Базовый		04.07.2013 15:57:01		включен	▲

Сохранить Отмена

Именно для аккаунта производится активация продукта (т.е. подписка на продукт).

Аккаунты

Параметры **Продукты**

Текущие **История**

ID	Продукт	Описание	Начало	Окончание	Состояние	...
89	Базовый		04.07.2013 15:57:01		включен	▲

Активация продукта

Продукт
VIP ▼

Режим активации
подписка ▼

Период
с [] [] по [] []

Описание
[]

Комментарий
[]

OK Отмена

Сохранить Отмена

Тарифы TV

В модуле TV начисление может производиться как единовременное списание при активации продукта, как периодическое списание, так и оба вместе, в зависимости от тарифа.

Важно знать, что период продукта не означает, что в этот период абоненту будет доступен продукт. Период продукта не изменяется при периодическом начислении, а изменяется период состояния "включен" продукта.

Также важно понимать разницу и взаимосвязь параметров режима активации и параметров продления продукта, указанной в тарифе в ветке стоимость. От режима активации зависит с какого времени продукт начнется и в какое закончится; от тарифа зависит, как будет продлеваться начиная со времени начала продукта, и как - в случае появления периодов неоплаты.

В тарифе необходимо обязательно указать стоимость подписки и услугу. Дополнительно можно указать стоимость активации продукта.

При указании стоимости активации данная сумма будет списана со счета абонента при активации. Если баланса на договоре будет не хватать, т.е. баланс минус стоимость активации будет меньше лимита, то абонент не сможет активировать продукт.

При указании стоимости подписки 0.0 за определенный период и удачной активации, продукт будет активен на весь период подписки.

При указании стоимости подписки больше 0.0 за определенный период сразу после активации продукта произойдет попытка списания данной суммы. Списание произойдет только в том случае, если баланс минус сумма стоимость подписки за период больше или равно чем лимит. В случае списания продукт поменяет состояние с выключен на включен, т.е. станет активным. По истечении периода из тарифа произойдет новая попытка начисления. В случае, если денег на счету абонента не хватает, сумма не будет списана, а продукт перейдет в состояние выключен.

Не следует для бесплатных продуктов без необходимости указывать маленький период (например, 0 руб. за 1 мин.), т.к. это может приводить к ненужной лишней нагрузке и лишней синхронизации с MW.

INET

TV

Продукт:
- Все и сразу

Продукт:
- Базовый

==>Подписка

10.0 за 2 минут выровнено

Продукт:
- VIP

==>Подписка

10.0 за 2 минут выровнено

Если необходимо единовременное списание за активацию, нужно указать режим активации и стоимость активации, а стоимость подписки указать 0 за какой-либо не нулевой период:

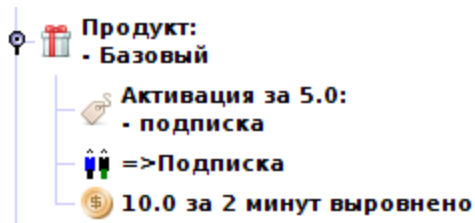
Продукт:
- Базовый

Активация за 5.0:
- подписка

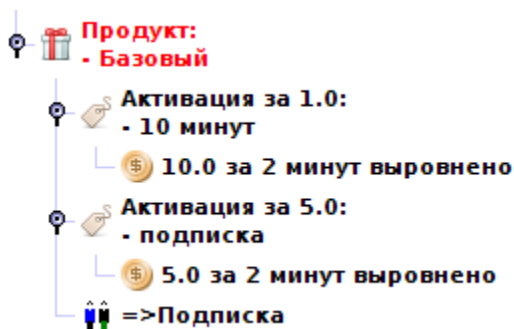
==>Подписка

0.0 за 1 месяцев выровнено

При необходимости можно совместить списание за активацию и периодическое списание:



Или же, в зависимости от режима активации указать разную стоимость периодического начисления:



Выровнено

Флаг **выровнено** означает, что периоды когда продукт включен или выключен всегда будут кратны периоду, указанному в тарифе и зависеть от времени начала активированного продукта. Т.е., предположим, что в тарифе указана стоимость подписки 10 руб за 30 минут. У абонента на счету 15 рублей, лимит 0 рублей, он активирует продукт в 12:46, происходит списание 10 рублей и продукт становится активным. В 13:16 происходит попытка продления периода, когда продукт включен, но денег на счету не хватает, продукт переходит в состояние выключен (и происходит отключение соответствующего, например, пакета каналов). В 13:30 абонент вносит платеж 5 рублей - при установленном флаге **выровнено**, произойдет продление с 13:16 до 13:46, когда произойдет новая попытка начисления; при неустановленном - с 13:13 до 14:00, когда произойдет новая попытка начисления. Если же абонент вносит платеж не в 13:30, а в 14:10, при включенном флаге - с 13:46 до 14:16, а при выключенном - с 14:10 до 14:40.

Флаг **выровнено** подойдет в случае подневного начисления - необходимо в режиме активации указать 0 дней с текущего, в тарифе - X за 1 день. Теперь, если абонент заплатит в течение дня, следующее начисление все равно произойдет в 00:00. Тоже самое справедливо и для помесечного начисления.

Зависимый/пропорциональный период

Флаг **зависимый/пропорциональный период** указывает, что период может быть пропорциональным, а не обязательно такой длины, которая указана в ветке стоимость.

Если есть активный продукт, от которого зависит данный, то конец периода данного продукта не будет позже конца периода главного продукта (стоимость данного продукта будет пропорционально уменьшена, если произошла коррекция времени окончания).

При установленном флаге **выровнено** и пополнении счета после нехватки средств конец периода будет выровнен согласно условию флага **выровнено**, но начало периода будет от времени платежа, а стоимость пропорционально меньше.

Начисление "1 месяц (подневное продление)"

В этом режиме начисление происходит каждый день, при этом цена за день - указанная стоимость за 1 месяц, делённая на количество дней в текущем месяце.

Начисление "1 месяц (подневное до конца месяца)"

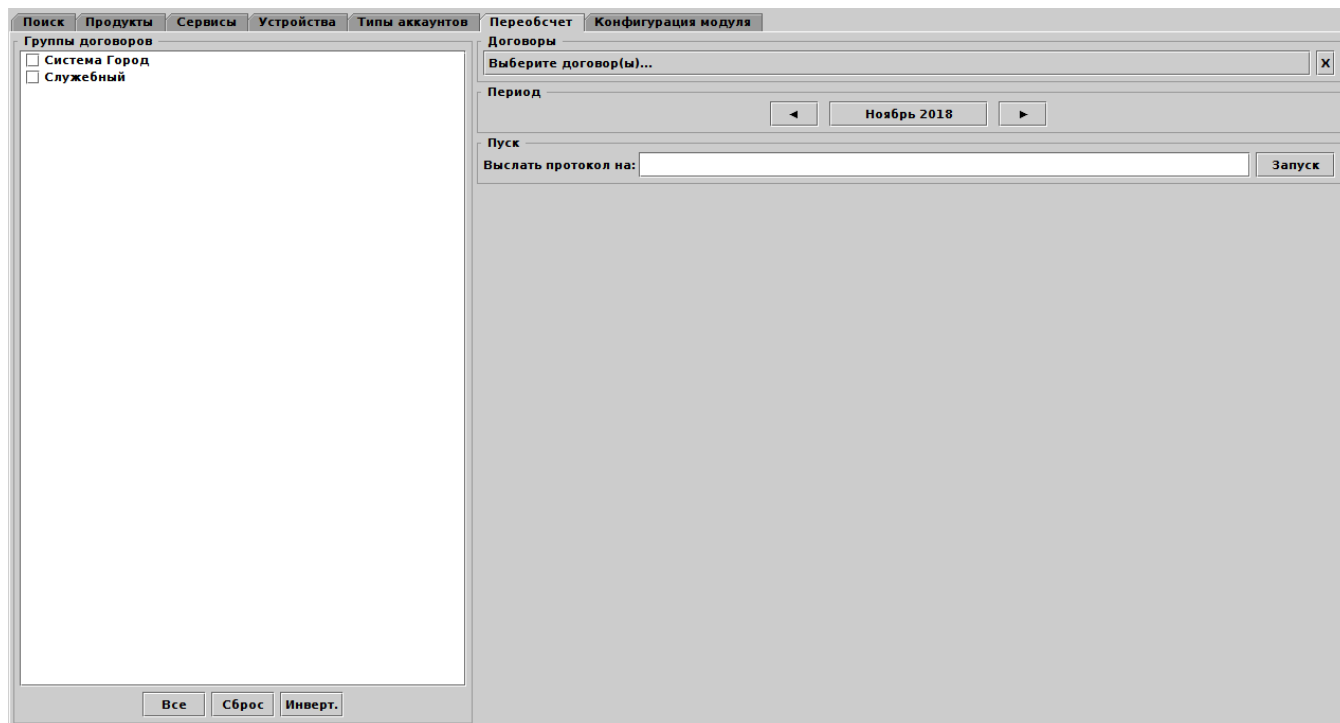
В этом режиме начисление происходит помесечно, при этом, если продукт был активирован в середине месяца - то за этот месяц начисляется сумма, пропорциональная количеству активных дней. После отключения по нехватке денег, если абонент пополнил счет в середине месяца, за данный месяц начисление также будет пропорционально.

Отложенная тарификация

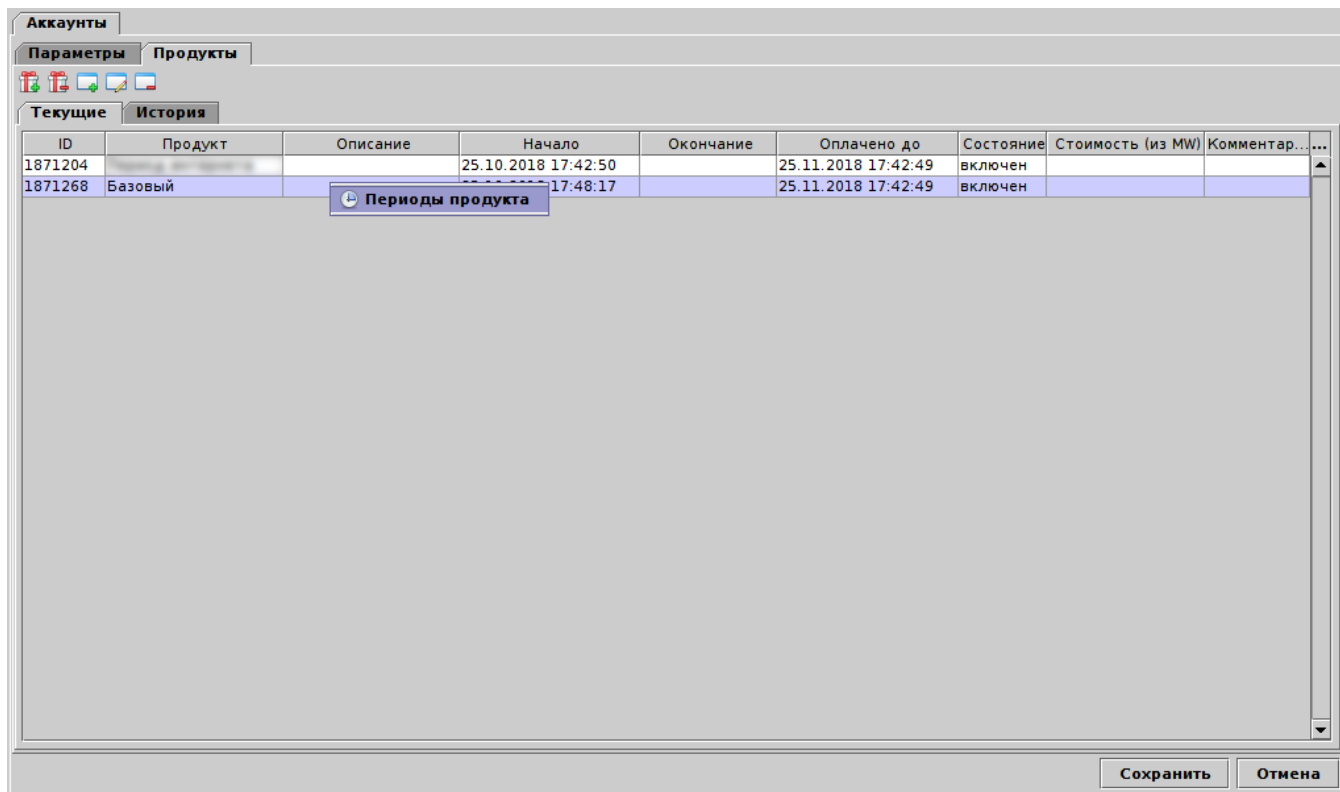
По умолчанию тарификация происходит в реальном времени, при достижении времени окончания предыдущего тарифицированного продления. Например, если период продления привязан к календарным суткам, то тарификация и попытка продления произойдет в 00:00 нового дня. При наличии задачи тарификации, например, модуля абонплат возможна ситуация, когда желательно, чтобы тарификация модуля абонплат сработала раньше, чем тарификация модуля TV. В этом случае в Продукте в параметре "Продление" можно указать вместо "в реальном времени" значение "по планировщику" и настроить соответствующую задачу планировщика.

Переобсчет TV

При необходимости переобсчета месяца после изменения цены в тарифе откройте сервисную вкладку модуля, перейдите на вкладку **Переобсчет**. На данной вкладке можно переобсчитать как отдельный договор, так и группу договоров или все договора биллинга.



При необходимости внести изменения в отдельный договор (например, подписка должна была быть отключена на договоре ранее), перейдите на вкладку модуля в договоре, откройте редактор аккаунта и перейдите на вкладку **Продукты**:



В модуле TV продукты состоят из периодов активности, именно по ним биллинг осуществляет тарификацию. Поэтому для переобсчета нужно редактировать периоды активности, а не только период самого продукта.

Аккаунты

Параметры | Продукты

Периоды продукта

ID	Время активации	Начало	Окончание	Время пролонгации	По	Зк	Пп	Зв	...
6491596	25.10.2018 17:48:17	25.10.2018 17:48:17	25.11.2018 17:42:49	25.10.2018 17:48:17	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	▲

Редактирование периода продукта

Период

с по

Флаги

приостановлен закрыт

пропорциональный завершающий

OK Отмена

Закреть панель "Периоды продукта"

Сохранить Отмена

При редактировании периодов нужно учитывать значения флагов. Если на периоде не стоит флаг **закрыт** или флаг **завершен**, и если дата окончания периода меньше чем дата завершения самого продукта, то биллинг попытается продлить данный период при наступлении времени его окончания. Периоды длиной более одного месяца не являются корректными для биллинга (т.е. время окончания периода должно быть в месяце начала периода или в следующем, но не дальше). После окончания редактирования периодов необходимо запустить переобсчет на сервисной вкладке модуля.

При удалении продукта также удаляются его периоды и происходит автоматический переобсчет.

Интеграция

Установка серверных приложений TV

Сервера Access и Accounting модуля TV предназначены, соответственно, для управления и доступом и тарификации. Для корректной работы модуля они должны быть установлены и запущены. Установка Access и Accounting-серверов происходит одинаково. Разница только в названиях папок, служб и системных переменных.

BGTVAccess и BGTVAccounting обновляются как обычные [серверные приложения биллинга](#). Необходимо обновить каждое из приложений перед первым запуском.

Установка Access-сервера

Установка на платформу Linux

- 1) Извлеките **BGTVAccess** из архива и скопируйте в каталог **/opt/bgbilling**;
- 2) Перейдите в каталог **/opt/bgbilling/BGTVAccess**;
- 3) Удалите все .ini, .bat и .exe файлы:

```
rm -f /*.bat & rm -f /*.exe & rm -f /*.ini
```

- 4) Откройте для редактирования файл **setenv.sh** и пропишите в нем путь к Java-машине, например так:

```
...  
cd ${0%${0##*/}}.  
  
JAVA_HOME=/opt/java/jdk8  
  
if [ -z "$JAVA_HOME" ]; then  
    echo "The JAVA_HOME environment variable is not defined"  
    echo "This environment variable is needed to run this program"  
    exit 1  
fi  
...
```

- 5) Установите права запуска для всех *.sh файлов:

```
chmod 744 *.sh
```

- 6) Возьмите из каталога **BGTVAccess/script** скрипт запуска **bgtv_access** и скопируйте его в каталог **/etc/init.d**, установите права на исполнение (см. выше). Если вы изменили каталог установки или переименовывали BGTVAccess, скорректируйте скрипт.

- 7) Выясните текущий уровень запуска системы командой:

```
[root@gate init.d]# runlevel  
N 3
```

- 8) Создайте линк для автоматического запуска Access-сервера:

```
ln -s /etc/init.d/bgtv_access /etc/rcN.d/S99bgtv_access
```

где N - требуемый уровень запуска.

- 9) Произведите настройку **tv-access.xml**;
- 10) Обновите как обычные [серверные приложения биллинга](#);
- 11) Для запуска и останова сервера **BGTVAccess** используйте скрипты **access_start.sh** и **access_stop.sh**.

При необходимости установки нескольких BGTVAccess-серверов на одной машине конечный каталог может быть переименован, например, в **SmotrshkaTVAccess**. Также требуется переименование и корректировка скрипта запуска, указание разных **app.id** в **tv-access.xml**.

Установка на платформу Windows

Для установки BGTVAcess на платформу Windows на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить BGTVAcess стоит Java-машина. Если её нет, установите версию не меньше 1.6.20. Загрузить можете с нашего сайта;
- 2) Загрузите с сервера BGTVAcess;
- 3) Распакуйте архив на диск C:;
- 4) Установите переменную окружения **BGTV_ACCESS_HOME =C:\BGTVAcess**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;
- 5) Установите службу BGTVAcess, для чего запустите файл **access_install.bat**;
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **access_uninstall.bat**;
- 7) Обновите как обычные [серверные приложения биллинга](#) ;
- 8) Для запуска и останова сервера BGTVAcess используйте консоль запуска и управления службами, служба BGTVAcess.

Установка Accounting-сервера

Установка на платформу Linux

- 1) Извлеките **BGTVAccounting** из архива и скопируйте в каталог **/opt/bgbilling**;
- 2) Перейдите в каталог **/opt/bgbilling/BGTVAccounting**;
- 3) Удалите все .ini, .bat и .exe файлы:

```
rm -f /*.bat & rm -f /*.exe & rm -f /*.ini
```

- 4) Откройте для редактирования файл **setenv.sh** и пропишите в нем путь к Java-машине, например так:

```
...
    cd ${0%${0##*/}}.

    JAVA_HOME=/opt/java/jdk8

    if [ -z "$JAVA_HOME" ]; then
        echo "The JAVA_HOME environment variable is not defined"
        echo "This environment variable is needed to run this program"
        exit 1
    fi
    ...
```

- 5) Проверьте .sh файлы на наличие символов ^M, если символы присутствуют их можно удалить вручную, либо воспользоваться утилитой:

```
dos2unix *.sh
```

- 6) Установите права запуска для всех *.sh файлов:

```
chmod 744 *.sh
```

- 7) Возьмите из каталога **BGTVAccounting/script** скрипт запуска **bgtv_accounting** и скопируйте его в каталог **/etc/init.d**, установите права на исполнение (см. выше). Если вы изменили каталог установки или переименовывали BGTVAccounting, скорректируйте скрипт.

- 8) Выясните текущий уровень запуска системы командой:

```
[root@gate init.d]# runlevel
    N 3
```

9) Создайте линк для автоматического запуска Accounting-сервера:

```
ln -s /etc/init.d/bgtv_accounting /etc/rcN.d/S99bgtv_accounting
```

где N - требуемый уровень запуска.

10) Произведите настройку **tv-accounting.xml**;

11) Обновите как обычные [серверные приложения биллинга](#).

11) Для запуска и останова сервера **BGTVAccounting** используйте скрипты `accounting_start.sh` и `accounting_stop.sh`.

При необходимости установки нескольких BGTVAccounting-серверов на одной машине конечный каталог может быть переименован, например, в **ShotreshkaTVAccounting**. Также требуется переименование и корректировка скрипта запуска, разнесение портов в **tv-accounting.xml**.

Установка на платформу Windows

Для установки BGTVAccounting на платформу Windows на диск C:.

- 1) Убедитесь, что на машине, где вы собрались ставить BGTVAccounting стоит Java-машина. Если её нет, установите версию не меньше 1.6.20. Загрузить можете с нашего сайта;
- 2) Загрузите с сервера BGTVAccounting;
- 3) Распакуйте архив на диск C:;
- 4) Установите переменную окружения **BGTVAccounting_HOME=C:\BGTVAccounting**. Как устанавливать переменные окружения можете посмотреть в инструкции по установке сервера и клиента биллинга;
- 5) Установите службу BGTVAccounting, для чего запустите файл **accounting_install.bat**;
- 6) Убедитесь, что служба появилась в списке служб Windows. В дальнейшем, можете удалить эту службу, используя **accounting_uninstall.bat**;
- 7) Обновите как обычные [серверные приложения биллинга](#);
- 8) Для запуска и останова сервера BGTVAccounting используйте консоль запуска и управления службами, служба BGTVAccounting.

Настройка tv-access.xml и tv-accounting.xml

С версии 8.2107+ данные файлы переименованы в **application.xml**.

В указанных файлах (первый находится в BGTVAccess, второй - в BGTVAccounting) необходимо указать значения параметров:

- **app.name** определяет имя приложения, оно используется, например в системе алармов;
- **app.id** - уникальный числовой идентификатор приложения среди всех приложений биллинга с данным параметром в XML-конфигурации, значение его не должно меняться всё время жизни системы;
- **moduleId** - код экземпляра модуля TV, к которому относится сервер.

Далее указать значения стандартных параметры настройки доступа к серверу БД и к MQ-серверу (серверам).

Каждый аккаунт привязан к своему устройству. В конфигурации каждого из серверов Access и Accounting указывается корневое устройство, от которого, включительно, начинается загрузка в память устройств и сервисов. Код этого устройства указывается в параметре **rootDeviceId**. Данный параметр также не должен быть пустым.

Пример файла настройки:

Код

```
<?xml version="1.0" encoding="UTF-8"?>
<application context="access">
  <!-- -->
  <param name="app.name" value="BGTVAccess" />
  <!-- id -->
  <param name="app.id" value="80101" />

  <!-- -->
  <param name="db.driver" value="com.mysql.jdbc.Driver" />
```



```
<param name="db.url" value="jdbc:mysql://127.0.0.1/bgbilling?useUnicode=true&characterEncoding=UTF-8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull&jdbcCompliantTruncation=false&queryTimeoutKillsConnection=true"/>
<param name="db.user" value="bill"/>
<param name="db.pswd" value="bgbilling"/>

<!-- MQ -->
<param name="mq.url" value="failover:(tcp://localhost:61616)"/>
<param name="mq.user" value="bill"/>
<param name="mq.pswd" value="bgbilling"/>

<!-- id -->
<param name="moduleId" value="8"/>
<!-- id -->
<param name="rootDeviceId" value="1"/>

....
```

24часаТВ

Интеграция с 24часаТВ (24h.tv) представлена в виде открытого кода (динамические классы).

При интеграции с данной системой продукты модуля (или, в зависимости от конфигурации, сервисы модуля) являются подключаемыми пакетами MW.

- [Настройка](#)
- [Настройка Webhooks](#)
- [Биллинг на стороне оператора \(в разработке\)](#)

Настройка

Тарификация выполняется на стороне 24часаТВ - к биллингу просто идут запросы о необходимости снятия суммы со счета и информация о подключении/отключении пакетов.

Добавьте конфигурацию модуля и установите ее активной:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

# Web -
web.menuItem1=TV

#
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

#
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890
# ,
account.password.chars.description= .
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=

# webhook-
tv.rest.handler.tv24h.class=ru.bitel.bgbilling.modules.tv.dyn.tv24h.Tv24hWebhookHandler
# ( )
tv.lifecycle.handler.tv24h.class=ru.bitel.bgbilling.modules.tv.dyn.tv24h.Tv24hEventWorker

# - 24hTV
om.tv24h.deviceTypeIds=
# ID
om.tv24h.contractPatternId=
```

```

# ID
om.tv24h.tvAccountSpecId=
# ID
om.tv24h.chargeTypeId=
#om.tv24h.paymentTypeId=

# ID -
#om.tv24h.param.name.id=2
# ID -
om.tv24h.param.lastName.id=
# ID -
om.tv24h.param.firstName.id=
# ID - Email
om.tv24h.param.email.id=9
# ID -
om.tv24h.param.phone.id=7

#   webhook.   ,   .
# 1 - ( ), 0 -   24 . (   ( ),   )
om.tv24h.needCreateNewContract=1
#   24.   9.2207+
# 1 - ( ), 0 -
om.tv24h.needUpdatePhone=1

```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.

Укажите в конфигурации:

Конфигурация

```

# /
# -
accounting.worker.1.thread.count=1
# , -
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
# ,

```

```
# -
#accounting.worker.1.tracking.account.1.deactivateProductsWithoutPrice=1
#
# , -
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400
```

Далее создайте новый тип устройства, назовите его 24часаТВ и укажите в поле **OrderManager** класс **ru.bitel.bgbilling.modules.tv.dyn.tv24h.Tv24hOrderManager**:

Поиск | Продукты | Сервисы | **Устройства** | Типы аккаунтов | Переобсчет | Конфигурация модуля

Устройства | Типы устройств

Редактор

Основное

Название:
24часаТВ

является источником данных

Сущность:

OrderManager:
ru.bitel.b.m.tv.dyn.tv24h.Tv24hOrderMa...

Обработчик процессора протокола:

Обработчик управления устройством:

Комментарий:

Конфигурация:

1

Найти | Найти далее

Строк: 1 | Текущая строка: 1 | Текущий столбец: 1

Сохранить | Отмена

У нас получилось два типа устройства:

ID	Название	Комментарий
1	Access+Accounting	
2	24часаТВ	

Заметьте, что у типа устройства 24часаТВ ID получился равным 2. Именно это значение и прописано в конфигурации модуля, в параметре `om.tv24h.deviceTypeIds`

В дереве устройств создайте устройство типа Access+Accounting и укажите конфигурацию:

Создайте дочернее к Access+Accounting устройство типа 24часаТВ, в поле **Community/secret** укажите ключ доступа к API, созданный в панели администратора 24часаТВ. Поле **Хост/порт** оставьте пустым.

Поиск	Продукты	Сервисы	Устройства	Типы аккаунтов	Переобсчет	Конфигурация модуля		
<div style="display: flex; justify-content: space-between;"> Устройства Типы устройств </div> <div style="border: 1px solid gray; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> Основное Атрибуты Комментарий </div> <div style="display: flex;"> <div style="width: 30%; padding-right: 10px;"> <p>Идентификатор: <input type="text"/></p> <p>Хост/порт: <input type="text"/></p> <p>Логин: <input type="text"/></p> <p>Пароль: <input type="password"/></p> <p>Community/secret: <input type="password"/></p> <p>Тип устройства: 24часаТВ</p> <p><input type="checkbox"/> OrderManager отключен</p> <p>Период с <input type="text"/> по <input type="text"/></p> <p>Группы устройства: <input type="checkbox"/> Группы устройств</p> </div> <div style="width: 70%;"> <p>Конфигурация: Текущая конфигурация</p> <table border="1" style="width: 100%; height: 150px;"> <tr> <td style="width: 20px; text-align: center;">1</td> <td></td> </tr> </table> <p style="text-align: right;">Строк: 1 Текущая строка: 1 Текущий столбец: 1</p> </div> </div> </div>							1	
1								

Получившееся дерево устройств:

Поиск				Продукты	Сервисы	Устройства	Типы аккаунтов	Переобсчет	Конфигурация модуля
Устройства				Типы устройств					
	Устройства	Название	Комментарий	ID					
	Устройства			0					
	Access+Accounting			1					
	24часаТВ			2					

Перечитать конфигурацию на серверах

Добавьте тип аккаунта Аккаунт, укажите галочки **логин** и **пароль**, в конфигурации укажите:

Конфигурация

```
# ID 24
const.device.id=2
#
title.pattern= ({login})
```

Поиск				Продукты	Сервисы	Устройства	Типы аккаунтов	Переобсчет	Конфигурация модуля
ID	Название								...
1	24часаТВ								▲
▼									
Редактор									
Название:									
24часаТВ									
<input checked="" type="checkbox"/> логин	<input checked="" type="checkbox"/> пароль								
<input type="checkbox"/> PIN	<input type="checkbox"/> устройство								
<input type="checkbox"/> идентификатор	<input type="checkbox"/> MAC-адрес								
				<div style="border: 1px solid gray; padding: 5px;"> <p>Конфигурация</p> <p># ID устройства 24часаТВ</p> <p>const.device.id=2</p> <p># Шаблон имени аккаунта</p> <p>title.pattern=Аккаунт: ({login})</p> </div>					
								Сохранить	Отмена

Теперь необходимо создать Продукты - они будут отражать пакеты системы 24часаТВ. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор пакета системы 24часаТВ. Поле **Продление** укажите **по планировщику** (т.к. продлением и тарификацией занимается сам 24часаТВ).

Чтобы была возможность активировать пакет из биллинга (тем самым активируя подписку в 24часаТВ) необходимо, чтобы в продукте был задан хотя бы один режим активации.

ID	Период	Название	Деактивация	Реактивация	...
1	01.09.2017-...		моментально	запрещена	▲

Для удобства в дереве продуктов можно копировать и вставлять копию для последующего изменения (например, чтобы не создавать каждый раз режим активации). Получившееся дерево пакетов-продуктов:

Поиск	Продукты	Сервисы	Устройства	Типы аккаунтов	Переобсчет	Конфигурация модуля
Название		Идентификатор		Комментарий		ID
Продукты						0
24часа						1
Стартовый	9			Общероссийские обязательные и популярные т...		2
Базовый	1			Обширный пакет, включающий все основные и...		3
VIP	8			Обширный пакет, включающий премиальные т...		4
Увлекательный	28					5
Эротический	10					6
МАТЧ! ФУТБОЛ	4			Весь футбол в пакете премиальных HD-телека...		7
НАСТРОЙ КИНО!	6			Золотой фонд кинохитов и самые громкие пре...		8
НАСТРОЙ КИНО + МАТЧ! ФУТБОЛ	14			5 превосходных каналов кино для всей семьи с...		9
Amedia Premium HD	13			Лучшие телесериалы мира, без рекламы и в вы...		10
Амедiateка	15					11
Дождь	16			Телеканал Дождь - независимый круглосуточн...		12

Настройка Webhooks

В панели администратора 24часаТВ необходимо добавить webhook'и, чтобы 24часаТВ оповещало биллинг о событиях. URL во всех webhook'ах одинаков, вида: https://<domain>/bgbilling/tv-ws/rest/<module_id>/tv24h/webhook. Webhook'и необходимо добавить на события: On create user, On user update, On user login, On create subscription, On update subscription, On create transaction, On update transaction. В логах webhook'a можно увидеть, дошел ли запрос до биллинга, и что биллинг отправил в ответ.

Биллинг на стороне оператора (в разработке)

По умолчанию **24часаТВ** сам осуществляет тарификацию абонентов и запрашивает снятие денег с баланса биллинга, биллинг при этом работает в "агентском режиме", в его тарифах указана нулевая стоимость подписки. Однако можно осуществлять тарификацию и управление средствами биллинга (так называемый "Расширенный метод интеграции" в документации 24часаТВ).

Для включения расширенного режима интеграции в конфигурации модуля добавьте параметр **om.tv24h.agentMode=0**:

Конфигурация

```
# 0 - ( )
# 1 ( ) -
om.tv24h.agentMode=0
```

В **24часаТВ** также необходимо указать интеграционный URL, по которому **24часаТВ** будет обращаться к биллингу, когда абонент хочет подключить или отключить пакет: https://<domain>/bgbilling/tv-ws/rest/<module_id>/tv24h.

Для этого режима в поле **Продление** продукта должно быть указано **в реальном времени** или настроена задача планировщика для продления подписок по планировщику.

Compass (compass.tv, ТИТВ)

Интеграция с Compass API представлена в виде открытого кода (динамические классы).

При интеграции с системой Compass используется единый тариф MW, продукты модуля являются подключаемыми подписками MW.

Создайте конфигурацию модуля и сделайте ее активной:

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

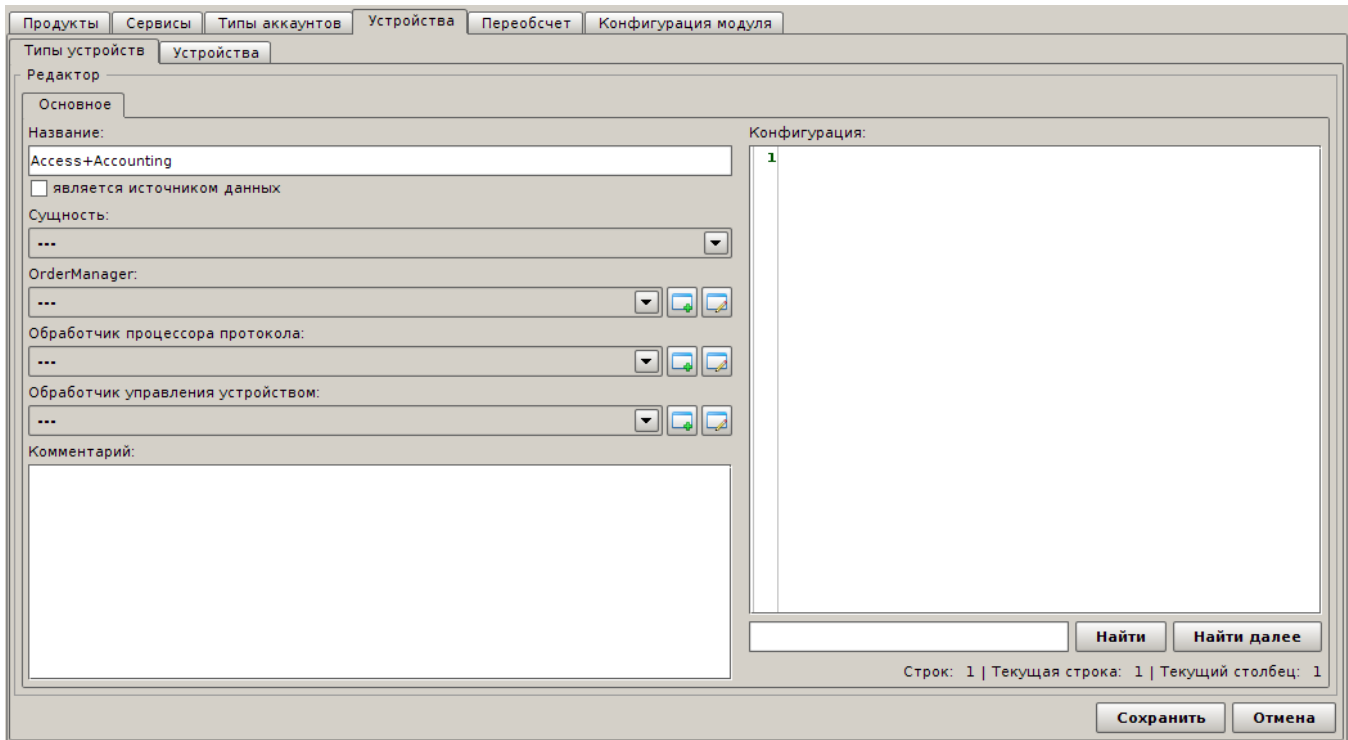
# Web -
web.menuItem1=TV

#
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

#
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMNopQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
# ,
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=

# ID - Compass
om.compass.deviceTypeIds=
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.



Далее создайте новый тип устройства, назовите его, например, Compass и добавьте конфигурацию:

Конфигурация

```
#      Middleware
#customer.lastName.pid=
#customer.firstName.pid=
#customer.middleName.pid=

#      ( 0 - , 1 - , )
om.product.syncMode=1
#      , 0 - , 1 -
#om.product.serviceMode=0

# ID - STB
om.terminal.tvAccountSpecId=
```

Выберите в поле OrderManager **динамический класс** `ru.bitel.bgbilling.modules.tv.dyn.compass.tv.CompassOrderManager` (чтобы его можно было выбрать, данный дин. класс должен быть скомпилирован через вкладку [Управление динамическим кодом](#)).

Укажите ID только что созданного типа устройства в конфигурации модуля, в параметре **om.commpass.deviceTypeIds**.

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```
# /
# -
accounting.worker.1.thread.count=1
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100
```

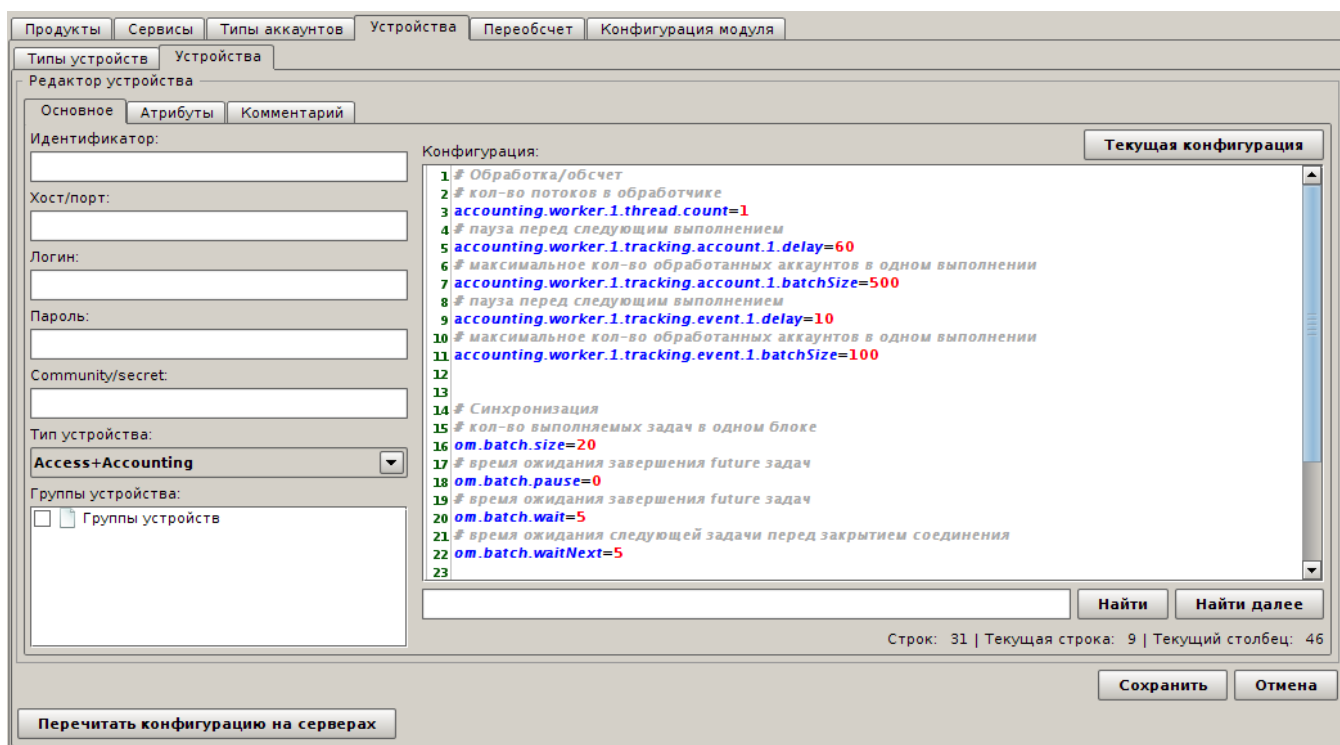
```

#
# -
om.batch.size=20
# future
om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400

```

Данное устройство будет отражать приложения TVAccess и TVAccounting.



Создайте дочернее к предыдущему устройству устройство типа Commpass, укажите хост подключения, логин и пароль.

Добавьте тип аккаунта "Аккаунт", укажите в его конфигурации ID только что добавленного устройства Commpass:

Конфигурация

```

# Commpass
const.device.id=
#
title.pattern= (${login})

```

Добавьте тип аккаунта "STB", установите галочку MAC-адрес, на вкладке "Родительские типы" установите галочку на типе аккаунта "Аккаунт", в конфигурации укажите:

Конфигурация

```
# Compass
const.device.id=
#
title.pattern=STB: (${macAddress})
```

Укажите ID только что добавленного типа аккаунта в конфигурации типа устройства Commpass в параметре **om.terminal.tvAccountSpecId**.

Теперь необходимо создать Продукты - они будут отражать пакеты системы Commpass. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор пакета системы Commpass (поле "id"). Список и id доступных пакетов можно посмотреть запросом <https://api.commpass.tv/rest/package>.

Название: М Прайм 1

Идентификатор: 105

Тип: периодический

Продление: по планировщику

Период с 01.03.2015 по

Сохранить Отмена

Чтобы абонент мог активировать Продукт (тем самым активируя подписку в Commpass) необходимо, чтобы в продукте был задан хотя бы один режим активации. Далее от выбранного режима активации может зависеть режим тарификации.

ID	Период	Название	Деактивация	Реактивация	...
10	01.03.2015-...	подневной	до конца оплаченного периода	запрещена	▲

Редактор

Название: подневная подписка

Период с 01.03.2015 по

Активация на 0 дней с текущего момента ровно Деактивация до конца оплаченного периода Реактивация запрещена

Сохранить Отмена

При регистрации приставки в системе Commpass не приходит какое-либо уведомление в биллинг. Поэтому для синхронизации информации о терминалах нужно создать отдельную задачу. Для этого нужно добавить глобальный скрипт поведения (Сервис - Автоматизация - Глобальные скрипты поведения), выбрав класс `ru.bitel.bgbilling.modules.tv.dyn.commpasstv.CommpassTerminalSynchronizingTask`, далее добавить задачу в планировщик "Выполнение глобальных скриптов по таймеру".

CTI TVEngine

Интеграция с CTI TVEngine представлена в виде открытого кода (динамические классы).

При интеграции с данной системой продукты модуля являются подключаемыми сервисами TVEngine.

После установки TVAccess и TVAccounting скопируйте папку BGBillingServer/lib/endorsed в TVAccess/lib и TVAccounting/lib, если она там отсутствует. Скопируйте BGBillingServer/lib/ext/serializer.jar в TVAccess/lib/ext и TVAccounting/lib/ext. Удостоверьтесь, что в access.sh и accounting.sh есть строка -Djava.endorsed.dirs=\${APP_HOME}/lib/endorsed:\${JAVA_HOME}/lib/endorsed" именно в таком виде. После этого необходимо перезапустить TVAccess и TVAccounting.

Добавьте конфигурацию модуля, укажите в ней веб-сервисы, к которым будет обращаться TVEngine и установите конфигурацию активной:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

# Web -
web.menuItem1=TV

#
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

#
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMNQRSTUWXYZabcdefghijklmnopqrstuvwxyz
# ,
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=

# - , TVEngine
tv.ws.billing.class=ru.bitel.bgbilling.modules.tv.dyn.cti.tve.ws.billing.impl.BillingServiceWSImpl
tv.ws.pricing.class=ru.bitel.bgbilling.modules.tv.dyn.cti.tve.ws.pricing.impl.PricingServiceWSImpl
tv.ws.order.class=ru.bitel.bgbilling.modules.tv.dyn.cti.tve.ws.order.impl.OrderManagementServiceWSImpl
```

Обратите внимание, что для этих сервисов нет авторизации, т.е. они должны быть закрыты от внешнего доступа (контекст tv-ws, т.е. <http://billing:8080/tv-ws/...>).

В конфигурации TVEngine iptvmw-config.properties должны быть указаны пути к этим веб-сервисам биллинга (10 - это код модуля, значение нужно будет заменить):

Конфигурация

```
#-----
# 1.2 PUSH (EXTERNAL) MODE
```

```

#-----
# type of billing system:
# db - internal billing
# ws - external billing through web-services
# cti - CTI-BILLING
# fastcom - FASTCOM-billing
tve.billing.type=ws
# getting customer status from billing during stb authentication process (default - FALSE)
tve.billing.needGetDataFromBillingDuringStbAuthentication=false
# allows to select OrderManagementService implementation type (available values: db , ws, cti, fastcom)
tve.billing.external.orderManagementService.type=ws
# allows to select BillingManagementService implementation type for 'queryCustomerAccountInfo' method only
(available values: db , ws, cti, fastcom)
tve.billing.external.billingManagementService.queryCustomerAccountInfo.type=ws
# external billing realization for "askPermissionToActivateOnBilling" method
# (using during set top box registration operation)
tve.billing.external.customerManagementService.askPermissionToActivateOnBilling.type=ws
tve.billing.external.billingManagementService.type=db
tve.billing.external.billingManagementService.queryCustomerAccountInfo.type=ws
tve.billing.external.billingManagementService.queryBillingEntries.type=ws
tve.billing.external.billingManagementService.querySummByService.type=ws

#-----
# 1.3.1.1 SINGLE BILLING SYSTEM - setting for single billing PULL mode
#-----
billadapter.pull.pricingservice=http://billing:8080/bgbilling/tv-ws/ru.bitel.bgbilling.modules.tv.dyn.cti.tve.
ws.pricing/10/PricingServiceWS?wsdl
billadapter.pull.orderservice=http://billing:8080/bgbilling/tv-ws/ru.bitel.bgbilling.modules.tv.dyn.cti.tve.ws.
order/10/OrderManagementServiceWS?wsdl
billadapter.pull.billingservice=http://billing:8080/bgbilling/tv-ws/ru.bitel.bgbilling.modules.tv.dyn.cti.tve.
ws.billing/10/BillingServiceWS?wsdl

```

После изменения конфигурации TVEngine необходимо его перезапустить (service jetty restart).

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.

Далее создайте новый тип устройства, назовите его, например, TVEngine и добавьте конфигурацию:

```
#      CTI TVEngine
#      ( )
customer.lastName.pid=
#      ( )
#customer.firstName.pid=
#      ( )
#customer.middleName.pid=
# - ( . )
customer.company.pid=
# -
#customer.gender.pid=
# -
customer.birthDate.pid=
```

Укажите в нем OrderManager - ru.bitel.bgbilling.modules.tv.dyn.cti.tve.TveOrderManager.

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```
# /
# -
accounting.worker.1.thread.count=1
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=30
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400
```

Данное устройство будет отражать приложения TVAccess и TVAccounting.

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Типы устройств Устройства

Редактор устройства

Основное Атрибуты Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства: **Access+Accounting**

Группы устройства: Группы устройств

Конфигурация:

```

1 # Обработка/обсчет
2 # кол-во потоков в обработке
3 accounting.worker.1.thread.count=1
4 # пауза перед следующим выполнением
5 accounting.worker.1.tracking.account.1.delay=60
6 # максимальное кол-во обработанных аккаунтов в одном выполнении
7 accounting.worker.1.tracking.account.1.batchSize=500
8 # пауза перед следующим выполнением
9 accounting.worker.1.tracking.event.1.delay=10
10 # максимальное кол-во обработанных аккаунтов в одном выполнении
11 accounting.worker.1.tracking.event.1.batchSize=100
12
13
14 # Синхронизация
15 # кол-во выполняемых задач в одном блоке
16 om.batch.size=20
17 # время ожидания завершения future задач
18 om.batch.pause=0
19 # время ожидания завершения future задач
20 om.batch.wait=5
21 # время ожидания следующей задачи перед закрытием соединения
22 om.batch.waitNext=5
23

```

Текущая конфигурация

Найти Найти далее

Строк: 31 | Текущая строка: 9 | Текущий столбец: 46

Сохранить Отмена

Перечитать конфигурацию на серверах

Создайте дочернее к предыдущему устройству устройство типа TVEngine, укажите хост подключения, логин и пароль.

Добавьте тип аккаунта Аккаунт, установите галочки на "логин" и "пароль", укажите конфигурацию:

```

# ID TVEngine
const.device.id=
#
title.pattern=: (${login})

```

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

ID	Название
1	Аккаунт
2	Терминал

Редактор

Название: Аккаунт

логин пароль

PIN устройство

идентификатор MAC-адрес

Конфигурация Типы устройств Группы устройств Родительские типы

```

# автоматически привязываем к устройству 2 (IPTVPorta)
const.device.id=2
# шаблон названия
title.pattern=Аккаунт: (${login})

```

Сохранить Отмена

Добавьте тип аккаунта для приставки. Конфигурацию укажите в зависимости от типа приставки:

```
#
title.pattern=SagemHD85: (${identifier}) ([$macAddress])

# MW CTI TVEngine
stb.type=sagemHD85
```

В поле Родительские типы поставьте галочку на типе "Аккаунт", который создали только что. В договоре он будет дочерним по отношению к аккаунту типа "Аккаунт".

IP TV Портал (iptvportal.ru)

Интеграция с Middleware "IP TV Портал" представлена в виде открытого кода (динамические классы).

При интеграции с данной системой продукты модуля (или, в зависимости от конфигурации, сервисы модуля) являются подключаемыми пакетами MW.

Добавьте конфигурацию модуля и установите ее активной:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

# Web -
web.menuItem1=TV

#
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

#
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMN0PQRSTUVWXYZabcdefghijklmnopqrstuvwxy
#
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=

# - IPTVPortal ( MW)
om.deviceTypeIds=2
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.

Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Типы устройств | **Устройства**

Редактор

Основное

Название: Access+Accounting

является источником данных

Сущность: ...

OrderManager: ...

Обработчик процессора протокола: ...

Обработчик управления устройством: ...

Комментарий:

Конфигурация:

1

Строк: 1 | Текущая строка: 1 | Текущий столбец: 1

Найти | Найти далее

Сохранить | Отмена

Далее создайте новый тип устройства, назовите его, например, IPTVPortal и добавьте конфигурацию:

```
# MW- ( - )
# ( )
#customer.lastName.pid=
# ( )
#customer.firstName.pid=
# - ( . )
#customer.company.pid=

# (0 - , 1 - , )
om.product.syncMode=1
# , 0 - , 1 -
om.product.serviceMode=0

# , , /
om.defaultPackages=

#
om.terminal.registered=1

# ID - (STB)
# IPTVPortal
om.terminal.tvAccountSpecId=
```

Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Типы устройств | **Устройства**

Редактор

Основное

Название: IPTVPortal

является источником данных

Сущность: ...

OrderManager: ru.bitel.bgbilling.modules.tv.dyn.ares.iptvportal.IptvPortalOrderManager

Обработчик процессора протокола: ...

Обработчик управления устройством: ...

Комментарий:

Конфигурация:

```

1 # полная (1) или атомарная (0) синхронизация продуктов
2 om.product.syncMode=1
3 # синхронизация на уровне продуктов (0)
4 om.product.serviceMode=0
5
6 # ФИО/название вытаскиваем из комментария договора
7 #customer.lastName.pid=
8 #customer.firstName.pid=
9 #customer.company.pid=
10
11
12 # тип аккаунта - терминала (для синхронизации терминалов из
13 # MW)
13 om.terminal.tvAccountSpecId=2

```

Строк: 13 | Текущая строка: 13 | Текущий столбец: 30

Найти | Найти далее

Сохранить | Отмена

Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Типы устройств | **Устройства**

ID	Название	Комментарий
1	Access+Accounting	
2	IPTVPortal	

Заметьте, что у типа устройства IPTVPortal ID получился равным 2. Именно это значение и прописано в конфигурации модуля, в параметре om.deviceTypeIds.

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```

# /
# -
accounting.worker.1.thread.count=1
#

```

```

accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400

```

Данное устройство будет отражать приложения TVAccess и TVAccounting.

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Типы устройств **Устройства**

Редактор устройства

Основное **Атрибуты** Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства:
Access+Accounting

Группы устройства:
 Группы устройств

Конфигурация:

```

1 # Обработка/обсчет
2 # кол-во потоков в обработчике
3 accounting.worker.1.thread.count=1
4 # пауза перед следующим выполнением
5 accounting.worker.1.tracking.account.1.delay=60
6 # максимальное кол-во обработанных аккаунтов в одном выполнении
7 accounting.worker.1.tracking.account.1.batchSize=500
8 # пауза перед следующим выполнением
9 accounting.worker.1.tracking.event.1.delay=10
10 # максимальное кол-во обработанных аккаунтов в одном выполнении
11 accounting.worker.1.tracking.event.1.batchSize=100
12
13
14 # Синхронизация
15 # кол-во выполняемых задач в одном блоке
16 om.batch.size=20
17 # время ожидания завершения future задач
18 om.batch.pause=0
19 # время ожидания завершения future задач
20 om.batch.wait=5
21 # время ожидания следующей задачи перед закрытием соединения
22 om.batch.waitNext=5
23

```

Текущая конфигурация

Строк: 31 | Текущая строка: 9 | Текущий столбец: 46

Найти Найти далее

Сохранить Отмена

Перечитать конфигурацию на серверах

Создайте дочернее к предыдущему устройству устройство типа IPTVPortal, укажите хост подключения, логин и пароль.

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Типы устройств **Устройства**

Редактор устройства

Основное Атрибуты Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства: **IPTVPortal**

Группы устройств: Группы устройств

Конфигурация:

Текущая конфигурация

Найти Найти далее

Строк: 1 | Текущая строка: 1 | Текущий столбец: 1

Сохранить Отмена

Перечитать конфигурацию на серверах

Получившееся дерево устройств:

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Типы устройств **Устройства**

Имя	Название	Комментарий	ID
Устройства			0
Access+Accounting			1
IPTVPortal: [admin, iptvportal.ru]			2

Перечитать конфигурацию на серверах

Добавьте тип аккаунта Аккаунт.

ID	Название
1	Аккаунт
2	Терминал

Редактор

Название: Аккаунт

логин пароль
 PIN устройство
 идентификатор MAC-адрес

Конфигурация Типы устройств Группы устройств Родительские типы

```

# автоматически привязываем к устройству 2 (IPTVPortal)
const.device.id=2
# шаблон названия
title.pattern=Аккаунт: (${login})
  
```

Сохранить Отмена

А также тип устройства - Терминал. В поле Родительские типы поставьте галочку на типе "Аккаунт", который создали только что. Установите галочку на "MAC-адрес". Также при необходимости можно установить галочку на поле "идентификатор", но следует учитывать, что в текущей версии поле Идентификатор сохраняется в поле inet_addr IPTVPortal'a, и потому должно быть либо пустым, либо правильным IP-адресом.

ID	Название
1	Аккаунт
2	Терминал

Редактор

Название: Терминал

логин пароль
 PIN устройство
 идентификатор MAC-адрес

Конфигурация Типы устройств Группы устройств Родительские типы

```

# шаблон названия
title.pattern=Терминал: (${macAddress})
  
```

Сохранить Отмена

Теперь необходимо создать Продукты - они будут отражать пакеты системы IPTVPortal. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор пакета системы IPTVPortal (по умолчанию обычно free, pkg0, pkg1, pkg2...).

Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

Название: free

Идентификатор: free

Тип: **периодический**

Период с [] по []

Комментарий:

Описание:

Сохранить | Отмена

Чтобы абонент мог активировать Продукт (тем самым активируя подписку в IPTVPortal) необходимо, чтобы в продукте был задан хотя бы один режим активации. Далее от выбранного режима активации может зависеть режим тарификации.

Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | **Режимы активации** | Доступность | Зависимость | Видимость

ID	Период	Название	Деактивация	Реактивация	...
2	12.03.2014-...	поминутная подписка	до конца оплаченного периода	запрещена	▲
3	12.03.2014-...	10 минут			
4	12.03.2014-...	подневная подписка	до конца оплаченного периода	запрещена	
5	12.03.2014-...	помесячная подписка	до конца оплаченного периода	запрещена	

Редактор

Название: подневная подписка

Период с 12.03.2014 по []

Активация на 0 дней с текущего(ей) ровно Деактивация до конца оплаченного периода Реактивация запрещена

Ok | Отмена

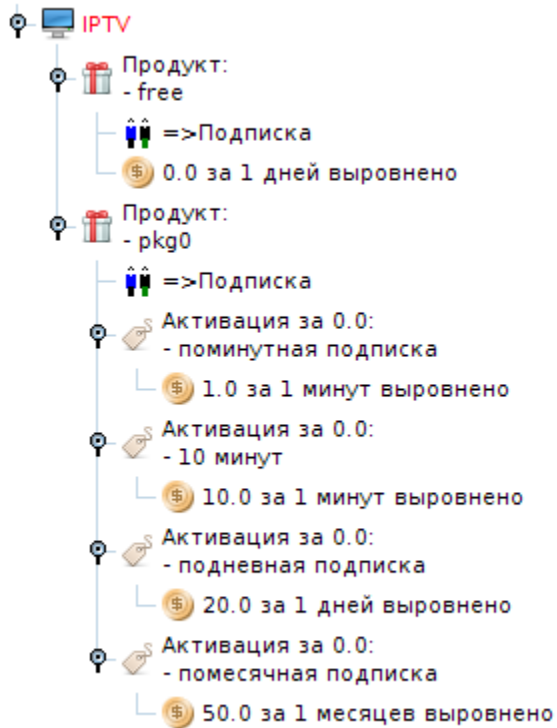
Сохранить | Отмена

Получившееся дерево Продуктов:

Продукты		
Сервисы	Типы аккаунтов	Устройства
Переобсчет	Конфигурация модуля	
Название	Комментарий	ID
Продукты		0
Пакеты		1
free		2
pkg0		3

Далее нужно создать тарифный план с веткой модуля. Пример тарифа:

Тарифный план 'IPTVTest'



На договоре нужно добавить модуль TV и создать аккаунт, который будет связан с аккаунтом из системы IPTVPortal

Аккаунты

Параметры **Продукты**

Тип: Аккаунт

Период: с 12.03.2014 по

Статус: Открыт

Логин: 12345 авто

Пароль: авто

Комментарий

Состояние: Включен

Сохранить Отмена

Можно (не обязательно) добавить терминал, как дочерний к аккаунту.

Аккаунты

Параметры

Тип: Терминал

Период: с 12.03.2014 по

Статус: Открыт

MAC-адрес: 08:00:21:D1:81:01

Идентификатор:

Комментарий:

Сохранить Отмена

Получившееся дерево аккаунтов:

Аккаунты						
Аккаунт	Период	Статус	Состояние	ExtID	ID	...
Аккаунты						
Аккаунт: 12345	12.03.2014-...	открыт	включен	1	1	
Терминал: 08002	C	12.03.2014-...	включен	572831	2	
Терминал: 001A7	A	12.03.2014-...	включен	557074	3	
Терминал: 94DE8	6	12.03.2014-...	включен	556459	4	
Терминал: 00218	3	12.03.2014-...	включен	574806	5	

При регистрации терминала в системе IPTVPortal не приходит какое-либо уведомление в биллинг. Поэтому для синхронизации информации о терминалах нужно создать отдельную задачу. Для этого сначала нужно добавить глобальный скрипт поведения:

Скрипты		
Код	Название	Класс
1	Синхронизация терминалов из IPTVPortal	ru.bitel.bgbilling.modules.tv.dyn.ares.iptvportal.IptvPortalTerminalSynch...

Редактор

Название: Синхронизация терминалов из IPTVPortal

Класс: `ru.bitel.bgbilling.modules.tv.dyn.ares.iptvportal.IptvPortalTerminalSynchronizingTask`

Показать результат выполнения **Выполнить сейчас**

OK Отмена

Taskbar: X IPTV X Редактор тарифных планов X Поиск договоров X testIPTV X Глобальные скрипты поведения

Далее добавить задачу в планировщик "Выполнение глобальных скриптов по таймеру". Для корректной работы не забудьте указать правильный ID типа аккаунта-терминала в поле **om.terminal.tvAccountSpecId** (чтобы добавленные дочерние аккаунты-терминалы были указанного типа) в конфигурации типа устройства IPTVPortal:

```
# ID - (STB)
# IPTVPortal
om.terminal.tvAccountSpecId=
```

FrontStage Middleware (Mediastage.tv, TelecomTV, BCC)

При интеграции с системой FrontStage Middleware используется единый тариф MW, продукты модуля являются подключаемыми услугами MW.

Создайте конфигурацию модуля и сделайте ее активной:

```
#
#
account.login.min=100000
#
account.login.max=999999
#
account.login.format=000000
#
account.password.length.auto=4
#
account.password.chars=0123456789

# - TelecomTV
# MW
om.deviceTypeIds=

# Web- MW /bgbilling/tv-ws/ru.bitel.bgbilling.modules.tv.integration.bcc.telecomtv.ws.billing/<mid>
/BillingIptv
tv.ws.billing.class=ru.bitel.bgbilling.modules.tv.integration.bcc.telecomtv.ws.billing.impl.BillingIptvImpl

# orderType MW ( TV1)
om.orderType.TV1.productSpecId=
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.

Далее создайте новый тип устройства, назовите его, например, TelecomTV и добавьте конфигурацию:

Конфигурация

```
# MW-
om.regionId=1
# MW-
om.lang=ru

# FrontStage Middleware
# ( )
customer.lastName.pid=
# ( )
#customer.firstName.pid=
# - ( . )
customer.company.pid=
# -
customer.address.pid=
# -
#customer.phone.pid=

# FrontStage Middleware
om.tariff.default=

# FrontStage Middleware
#om.integrationMode=Prepaid 2.0

# ( 0 - , 1 - , )
om.product.syncMode=1
# , 0 - , 1 -
#om.product.serviceMode=0
```

Создайте устройство типа Access+Accounting и дочернее к нему устройство типа TelecomTV, укажите хост и порт подключения, логин и пароль.

Добавьте тип аккаунта Аккаунт, установите галочки на "логин" и "пароль", укажите конфигурацию:

```
# ID TelecomTV
const.device.id=
#
title.pattern=: (${login})
```

Далее добавьте тип аккаунта для STB, установите галочки на "идентификатор" и "MAC-адрес", укажите конфигурацию, в зависимости от типа приставки:

```
#
title.pattern=Yuxing IPTV: (${identifier}) ([${macAddress}])

# FrontStage Middleware
terminal.typeId=18
terminal.bandwidth=99
```

В поле Родительские типы поставьте галочку на типе "Аккаунт", который создали только что.

LifeStream (Смотрёшка, lfstrm.tv)

Интеграция с LifeStream API представлена в виде открытого кода (динамические классы).

При интеграции с системой LifeStream используется единый тариф MW, продукты модуля являются подключаемыми подписками MW.

Создайте конфигурацию модуля и сделайте ее активной:

Обратите внимание, что логины назначаются со стороны биллинга, а не MW, но при этом они сквозные для всех провайдеров, поэтому во избежание ошибок "Пользователь с таким username уже существует" необходимо указать параметр **account.login.prefix**. Данный префикс будет использоваться при синхронизации с MW, однако в биллинге логины будут без него.

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

# Web -
web.menuItem1=TV

#
# ( ):
# , " , ,
#
# ru.bitel.bgbilling.modules.tv.api.server.bean.TvAccountGenerate
#account.login.class=
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

# LifeStream
account.login.prefix=

#
account.operator.name=

#
# ( ):
# , " , ,
#
# ru.bitel.bgbilling.modules.tv.api.server.bean.TvAccountGenerate
#account.password.class=
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyz
#
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Типы устройств **Устройства**

Редактор

Основное

Название: Access+Accounting

является источником данных

Сущность: ...

OrderManager: ...

Обработчик процессора протокола: ...

Обработчик управления устройством: ...

Комментарий:

Конфигурация:

1

Строк: 1 | Текущая строка: 1 | Текущий столбец: 1

Найти Найти далее

Сохранить Отмена

Далее создайте новый тип устройства, назовите его, например, LifeStreamTV и добавьте конфигурацию:

Конфигурация

```
#      Middleware

# ID      (      )
customer.fio.pid=0
#
# ID
#customer.lastName.pid=0
# ID
#customer.firstName.pid=0
# ID
#customer.middleName.pid=0

# email,      (1 -      , 2 -      , 3 -      )
customer.email.sources=1
# email (      )
customer.email.pid=
#      ,      email,      email
#customer.email.domain=

#      (0 -      , 1 -      , )
om.product.syncMode=1
#      , 0 -      , 1 -
#om.product.serviceMode=0
```

В случае, если Вы хотите указывать фиктивный email в MW (например, у вас нет данных о email абонентов), то укажите такие параметры:

Конфигурация

```
#      Middleware
# email,      (1 -      , 2 -      , 3 -      )
customer.email.sources=1
# email (      )
customer.email.pid=0
#      ,      email,      email
#customer.email.domain=
```

При указании логина идентичным email абонента (т.е. при использовании customer.email.sources=2) можно добавить проверку на то, чтобы в поле логин вводили корректный email. Для этого в скрипт поведения, привязанный к договору, нужно на событие "Изменяется аккаунт договора" добавить дин. класс ru.bitel.bgbilling.modules.tv.dyn.lfstrm.LifeStreamEventListener, а также в конфигурации модуля прописать account.login.email.check=1.

Выберите в поле OrderManager **динамический класс** ru.bitel.bgbilling.modules.tv.dyn.lfstrm.LifeStreamV2OrderManager (чтобы его можно было выбрать, данный дин. класс должен быть скомпилирован через вкладку [Управление динамическим кодом](#)).

Если не передавать в Смотрёшку email при обновлении аккаунта, то она возвратит код ошибки 400 с сообщением {"error": "" is not a 'email'}.

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```
# /
# -
accounting.worker.1.thread.count=1
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
#   future
om.batch.pause=0
#   future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400
```

Данное устройство будет отражать приложения TVAccess и TVAccounting.

Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Типы устройств Устройства

Редактор устройства

Основное Атрибуты Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства: **Access+Accounting**

Группы устройства: Группы устройств

Конфигурация:

```

1 # Обработка/обсчет
2 # кол-во потоков в обработке
3 accounting.worker.1.thread.count=1
4 # пауза перед следующим выполнением
5 accounting.worker.1.tracking.account.1.delay=60
6 # максимальное кол-во обработанных аккаунтов в одном выполнении
7 accounting.worker.1.tracking.account.1.batchSize=500
8 # пауза перед следующим выполнением
9 accounting.worker.1.tracking.event.1.delay=10
10 # максимальное кол-во обработанных аккаунтов в одном выполнении
11 accounting.worker.1.tracking.event.1.batchSize=100
12
13
14 # Синхронизация
15 # кол-во выполняемых задач в одном блоке
16 om.batch.size=20
17 # время ожидания завершения future задач
18 om.batch.pause=0
19 # время ожидания завершения future задач
20 om.batch.wait=5
21 # время ожидания следующей задачи перед закрытием соединения
22 om.batch.waitNext=5
23

```

Текущая конфигурация

Найти Найти далее

Строк: 31 | Текущая строка: 9 | Текущий столбец: 46

Сохранить Отмена

Перечитать конфигурацию на серверах

Создайте дочернее к предыдущему устройству устройство типа LifeStreamTV, укажите хост подключения (например, provider.test.lfstrm.tv), логин и пароль. Также URL подключения можно указать в конфигурации устройства, тогда поле Хост/порт не будет использоваться:

Конфигурация

```
om.url=http://provider.test.lfstrm.tv
```

Добавьте тип аккаунта Аккаунт, укажите в его конфигурации ID только что добавленного устройства LifeStreamTV:

Конфигурация

```
# LifeStreamTV
const.device.id=
#
title.pattern= (${login})
```

Установите галочки на полях **логин** и **пароль**.

Теперь необходимо создать Продукты - они будут отражать пакеты системы LifeStreamTV. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор пакета системы LifeStreamTV (поле "id").

Обратная интеграция

в конфиге модуля прописать

Конфигурация

```
tv.rest.handler.lfstrm.class=ru.bitel.bgbilling.modules.tv.dyn.lfstrm.LifeStreamHandler
```

Адрес обращения со стороны сервера LifeStreamTV : [http\[s\]://host:port/bgbilling/tv/rest/<mid>/lfstrm/upsale/v2](http[s]://host:port/bgbilling/tv/rest/<mid>/lfstrm/upsale/v2)

Ministra TV platform (ex Stalker Middleware)

Интеграция с Ministra TV platform (Middleware Stalker) от Infomir LLC представлена в виде открытого кода (динамические классы).

При интеграции с данной системой продукты модуля (или, в зависимости от конфигурации, сервисы модуля) являются подключаемыми пакетами MW.

Добавьте конфигурацию модуля и установите ее активной:

Конфигурация

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

# Web -
web.menuItem1=TV

#
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

#
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMN0PQRSTUVWXYZabcdefghijklmnopqrstuvwxy
#
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.

Поиск Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Устройства Типы устройств

Редактор

Основное

Название: Access+Accounting

Сущность:

OrderManager: ---

Обработчик управления устройством:

Комментарий:

Конфигурация:

1

Строк: 1 | Текущая строка: 1 | Текущий столбец: 1

Найти Найти далее

Сохранить Отмена

Далее создайте новый тип устройства, назовите его, например, Ministra и добавьте конфигурацию:

```
# account_number :
# 0 - ; , ID ,
# om.accountNumber.format
# 1 - ID , om.accountNumber.format
# 2 - ; , ID ,
# om.accountNumber.format
om.accountNumber.mode=2

# account_number, ID
#om.accountNumber.format=

# , ID
#om.account.loginFormat=

#
om.tariff.default=1

# ID -, -
om.tariff.product.parentIds=

# (0 - , 1 - , )
om.product.syncMode=1
```

В качестве OrderManager выберите **MinistraOrderManager**.

Поиск | Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Устройства | Типы устройств

Редактор

Основное

Название: Ministra

Сущность: ...

OrderManager: ru.bitel.b.m.tv.dyn.infomir.ministra....

Обработчик процессора протокола: ...

Обработчик управления устройством: ...

Комментарий:

Конфигурация:

```

1# В качестве account_number используется:
2# 0 - поле Логин корневого аккаунта; если не указано, то ID корневого аккаунта,
3# отформатированный с помощью om.accountNumber.format
4# 1 - ID корневого аккаунта, отформатированный с помощью om.accountNumber.format
5# 2 - поле Идентификатор корневого аккаунта; если не указано, то ID корневого аккаунта,
6# отформатированный с помощью om.accountNumber.format
7 om.accountNumber.mode=2
8
9# Формат account_number, если в качестве него используется ID корневого аккаунта
10 #om.accountNumber.format=
11|
12# Формат логина, если в качестве него используется ID аккаунта
13 #om.account.loginFormat=
14
15# Тариф по умолчанию
16 om.tariff.default=1
17
18# ID продукта-папки, содержащей в себе папки-тарифы
19 om.tariff.product.parentIds=1
20
21# Режим синхронизации продуктов (0 - по событию, 1 - по событию, полная)
22 om.product.syncMode=1

```

Строк: 22 | Текущая строка: 11 | Текущий столбец: 1

Найти | Найти далее

Сохранить | Отмена

Поиск | Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Устройства | Типы устройств

ID	Название	Комментарий
1	Access+Accounting	
2	Ministra	

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```

# /
# -
accounting.worker.1.thread.count=1
#
accounting.worker.1.tracking.account.1.delay=60
# -

```

```

accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400

```

Данное устройство будет отражать приложения TVAccess и TVAccounting.

Создайте дочернее к предыдущему устройству устройство типа **Ministra**, укажите хост подключения, логин и пароль.

Поиск | Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Устройства | Типы устройств

Редактор устройства

Основное | Атрибуты | Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства:

OrderManager отключен

Период с по

Группы устройства: Группы устройств

Конфигурация: Текущая конфигурация

```
1
```

Строк: 1 | Текущая строка: 1 | Текущий столбец: 1

Найти | Найти далее

Сохранить | Отмена

Перечитать конфигурацию на серверах

Получившееся дерево устройств:

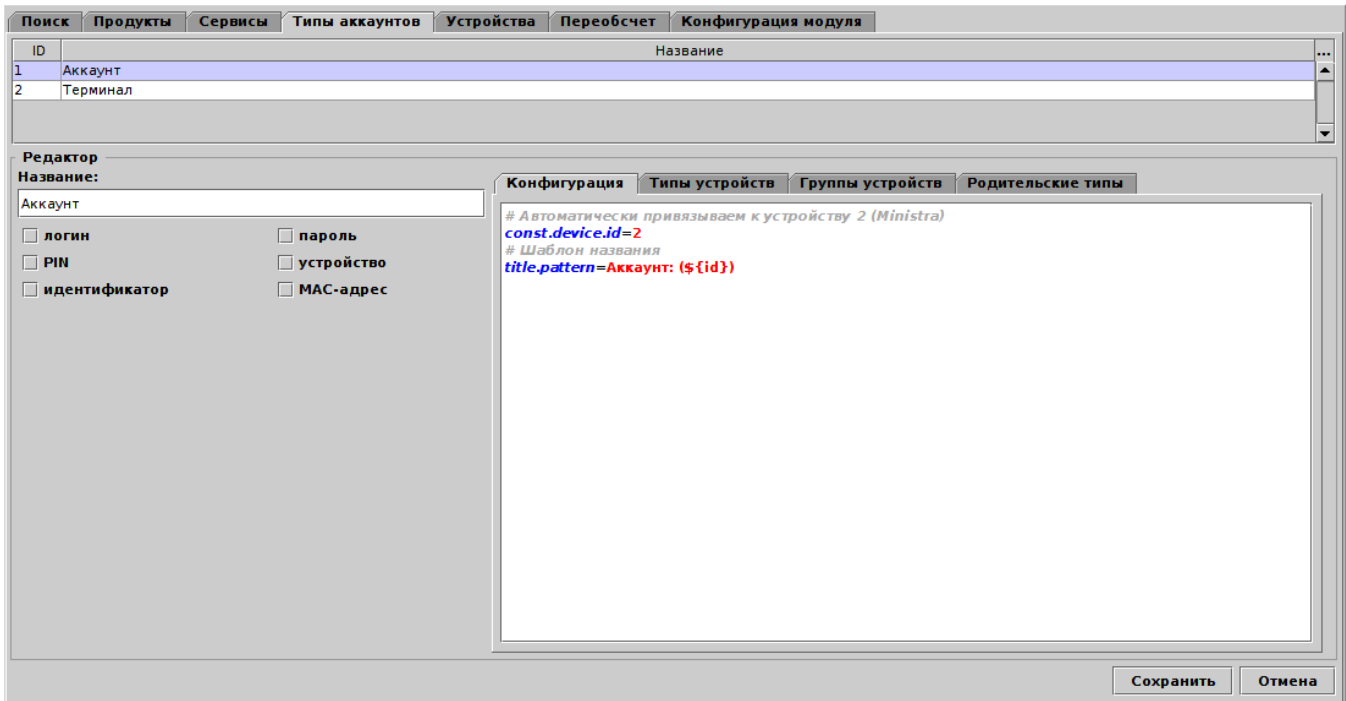
Поиск | Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Устройства | Типы устройств

Имя	Название	Комментарий	ID
Устройства			0
Access+Accounting			1
Ministra: [192.168.10.3]			2

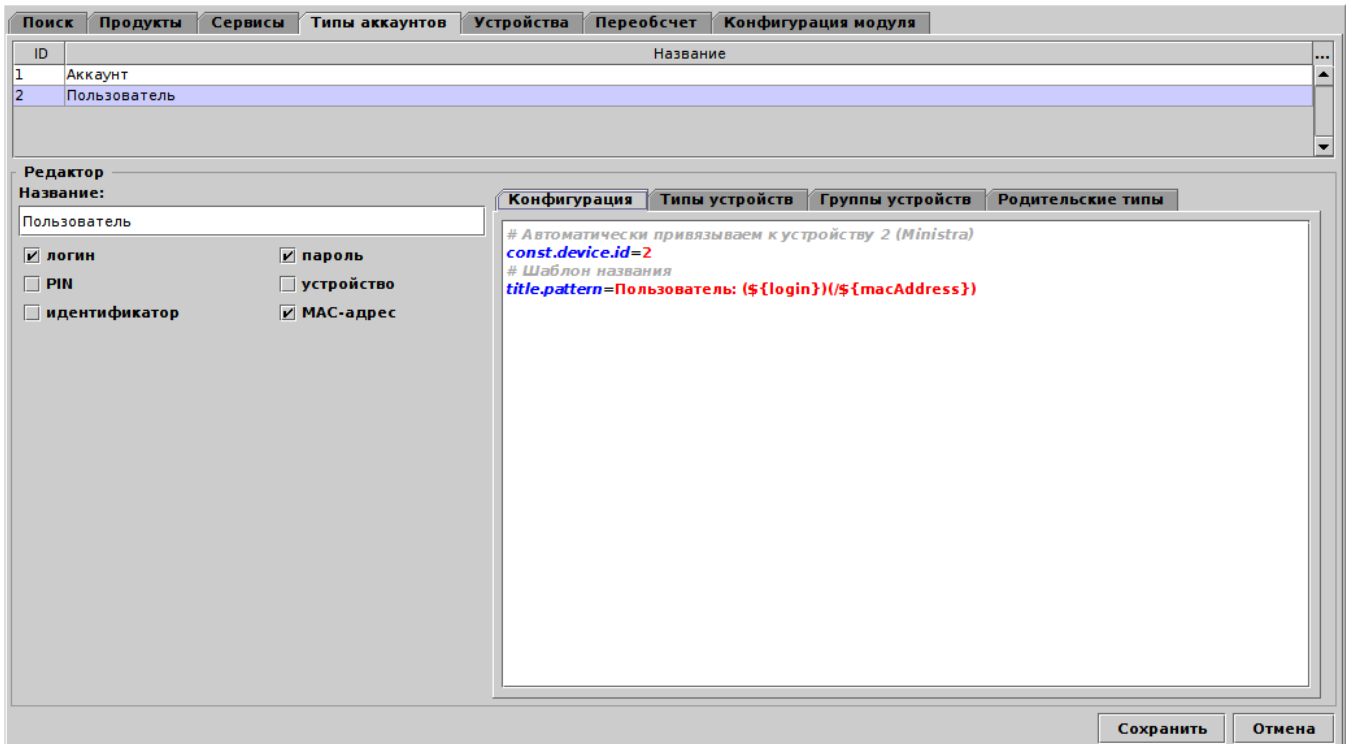
Перечитать конфигурацию на серверах

Добавьте тип аккаунта Аккаунт.



```
#      2 (Ministra)
const.device.id=2
#
title.pattern=: ({id})
#      Ministra
user=1
```

А также тип аккаунта - Пользователь. В поле Родительские типы поставьте галочку на типе **Аккаунт**, который создали только что. Установите галочку на полях "логин", "пароль" и "MAC-адрес".



Укажите в конфигурации типа аккаунта:

```
#      2 (Ministra)
const.device.id=2
#
title.pattern=: (${login})/${macAddress}
#      Ministra
user=1
```

Аккаунт биллинга (дочерний или нет) считается пользователем в Ministra, если в нем (в аккаунте) указано не пустое значение в поле Логин, а также в конфигурации типа аккаунта указано **user=1** или **stb_type=<тип STB>**.

В таком виде на договор нужно будет добавлять сначала Аккаунт ("лицевой счет" в Ministra), а затем дочерний к нему Пользователь, и уже в нем указывать логин, пароль и при необходимости MAC-адрес. Однако в типе аккаунта **Аккаунт** можно указать галочки в полях "логин", "пароль" и "MAC-адрес", тогда к этому аккаунту будет сразу привязываться не только "лицевой счет", но и "пользователь" Ministra, а дочерний аккаунт Пользователь использовать для добавления дополнительных пользователей этого лицевого счета.

Теперь необходимо создать Продукты - они будут отражать тарифы или пакеты системы Ministra. Для разделения тарифов от пакетов создайте две продукты-папки с названиями **Тарифы** и **Продукты** без режимов активации. Далее создайте продукты-тарифы и продукты-пакеты. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор пакета системы Ministra (поле "Внешний ID" в Ministra).

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

Название: Тариф "Социальный"

Идентификатор: 1

Тип: периодический

Продление: в реальном времени

Приоритет: 0

Период с по

Комментарий:

Описание:

Сохранить | Отмена

Чтобы абонент мог активировать Продукт (тем самым активируя подписку в Ministra) необходимо, чтобы в продукте был задан хотя бы один режим активации. Далее от выбранного режима активации может зависеть режим тарификации.

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

ID	Период	Название	Деактивация	Реактивация
1	28.11.2018-...		до конца оплаченного периода	запрещена

Редактор

Название:

Период с: 28.11.2018 по:

Активация на: 0 минут с текущего момента ровно Деактивация: до конца оплаченного периода Реактивация: запрещена

Ok Отмена

Сохранить Отмена

В продуктах-тарифах на вкладке **Зависимость** укажите, что они не совместимы с другими продуктами-тарифами (чтобы нельзя было активировать два продукта-тарифа одновременно).

Получившееся дерево Продуктов:

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Название	Комментарий	ID
Продукты		0
Тарифы		1
Тариф "Социальный"		2
Тариф "Социальный плюс"		3
Тариф "Полный"		4
Пакеты		5
Социальный		6
Социальный плюс		7
Полный		8

Обратите внимание, что у продукта-папки **Тарифы** ID равен 1 - именно это значение указано в параметре конфигурации типа устройства **Ministr** в параметре **om.tariff.product.parentIds**.

На договоре нужно добавить модуль TV, настроенный тариф и создать аккаунт, который будет связан с аккаунтом из системы Ministra.

MOOVI

Интеграция с MOOVI API представлена в виде открытого кода (динамические классы).

При интеграции с системой MOOVI продукты модуля являются подключаемыми тарифами MW.

Создайте конфигурацию модуля и сделайте ее активной:

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

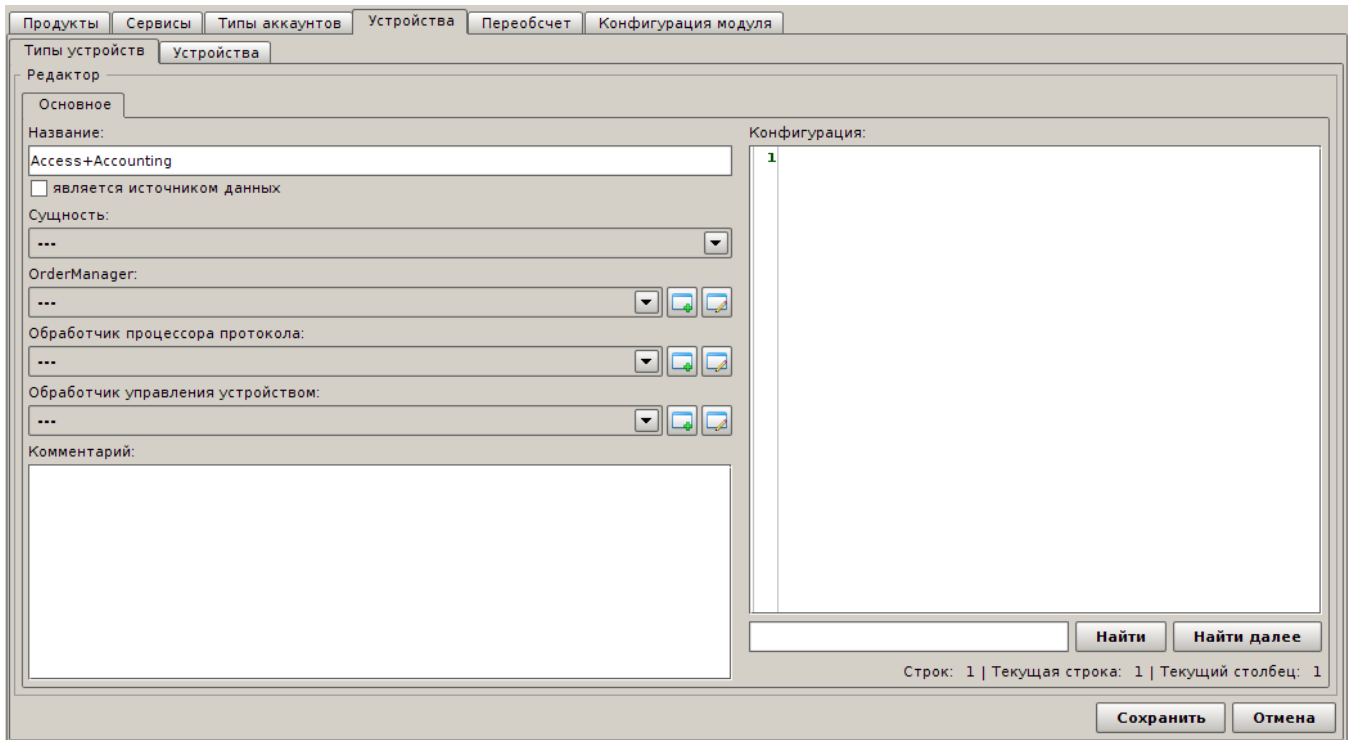
# Web- ( )
web.menuItem1=TV

#
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

# MW ( , )
account.login.prefix=
# . , MW.
# , - ID ( )
account.login.format=

#
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMN0PQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
#
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.



Далее создайте новый тип устройства, назовите его, например, MOOVI и добавьте конфигурацию:

Конфигурация

```
#      Middleware
#      ,
# /
customer.name.pid=
#
customer.lastName.pid=
#
customer.firstName.pid=
#
customer.middleName.pid=
```

Выберите в поле OrderManager [динамический класс](#) `ru.bitel.bgbilling.modules.tv.dyn.moovi.MooviOrderManager` (чтобы его можно было выбрать, данный дин. класс должен быть скомпилирован через вкладку [Управление динамическим кодом](#)).

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```
# /
# -
accounting.worker.1.thread.count=1
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
```

```

om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400

```

Данное устройство будет отражать приложения TVAccess и TVAccounting.

Создайте дочернее к предыдущему устройству устройство типа MOOVI, укажите логин и пароль. Также логин/пароль подключения можно указать в конфигурации устройства, в этом случае поля Логин/Пароль не будут использоваться:

Конфигурация

```

# API
om.login=
om.password=

```

Добавьте тип аккаунта Аккаунт, укажите в его конфигурации ID только что добавленного устройства MOOVI:

Конфигурация

```

# MOOVI
const.device.id=
#
title.pattern= ({login})

```

Установите галочки на полях **логин** и **пароль**.

Теперь необходимо создать Продукты - они будут отражать тарифы системы MOOVI. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор тарифа системы MOOVI (поле "ID").

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

Название: Тариф "Социальный"

Идентификатор: 1

Тип: периодический

Продление: в реальном времени

Приоритет: 0

Период с по

Комментарий:

Описание:

Сохранить | Отмена

Чтобы абонент мог активировать Продукт (тем самым активируя подписку в MOOVI) необходимо, чтобы в продукте был задан хотя бы один режим активации. Далее от выбранного режима активации может зависеть режим тарификации.

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

ID	Период	Название	Деактивация	Реактивация	...
1	28.11.2018-...		до конца оплаченного периода	запрещена	▲

Редактор

Название: [] Период с 28.11.2018 по []

Активация на 0 минут с текущего момента ровно Деактивация до конца оплаченного периода Реактивация запрещена

Ok | Отмена

Сохранить | Отмена

В продуктах-тарифах на вкладке **Зависимость** укажите, что они не совместимы с другими продуктами-тарифами, если есть необходимость, чтобы нельзя было активировать два продукта-тарифа одновременно.

Установите серверные приложения модуля согласно [инструкции](#).

Далее на договоре нужно добавить модуль TV, настроенный тариф и создать аккаунт, который будет связан с аккаунтом из системы MOOVI.

TVIP Media (TVIP TMS)

Интеграция с TVIP Media (<https://tvip.media>) (а так же с базовой платформой TVIP TMS (<http://tms.tviplabs.com>)) представлена в виде открытого кода (динамические классы).

При интеграции с системой TVIP продукты модуля являются подключаемыми тарифами MW.

Создайте конфигурацию модуля и сделайте ее активной:

```
#
contract.status.active.codes=0
contract.status.suspend.codes=3,4

# Id ( - ) , ,
#
#productSpec.entitySpecId=
#serviceSpec.entitySpecId=
#tvChannelSpec.entitySpecId=

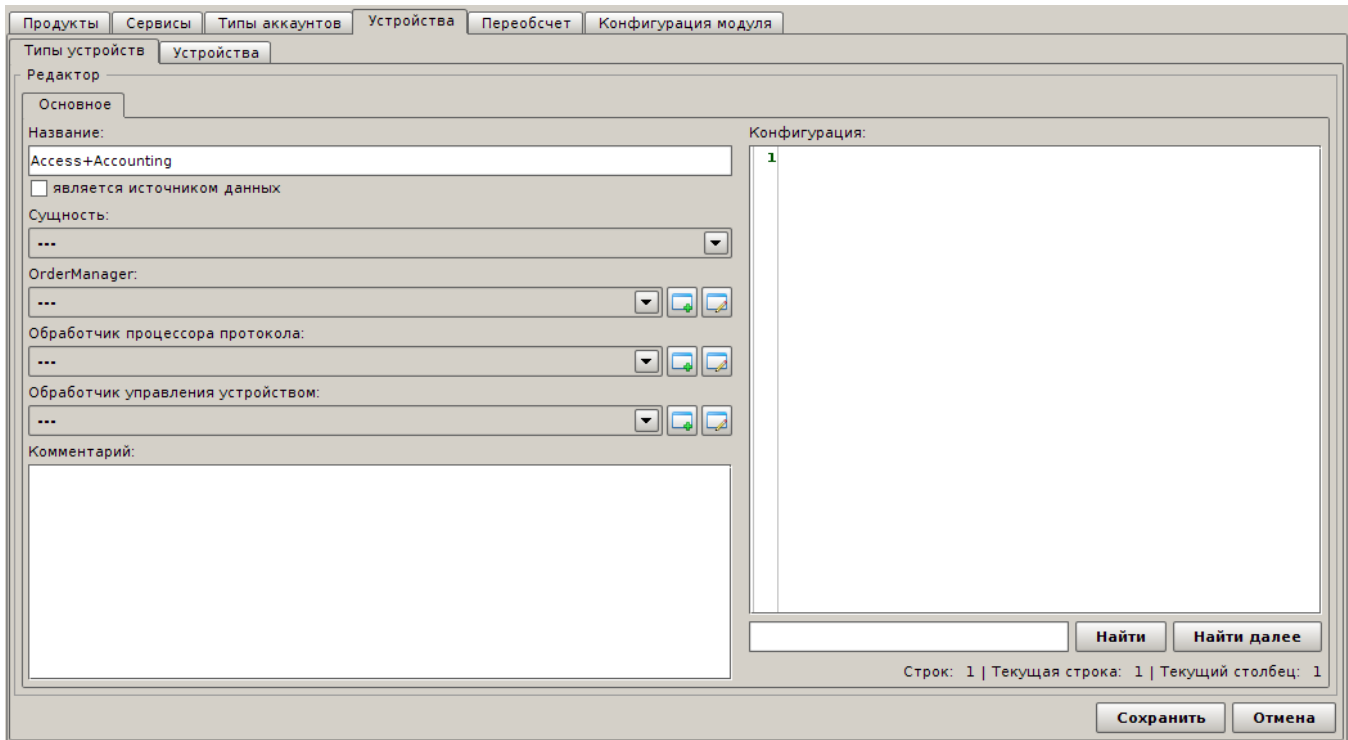
# Web- ( )
web.menuItem1=TV

# . , ,
# ( ):
#
account.login.min=10000001
# (.. 1,2,3 10000000,
# 4, 10000001)
account.login.max=99999999
#
#account.login.format=0000000

# . , - MW.
# , - ID ( )
account.login.format=

# . , ,
# ( ):
#
account.password.length.min=4
#
account.password.length.max=8
# ( )
account.password.chars=1234567890ABCDEFGHIJKLMNQRSTUUVWXYZabcdefghijklmnopqrstuvwxyz
# ,
account.password.chars.description=
#
account.password.length.auto=6
# ( account.password.chars)
#account.password.chars.auto=
```

Необходимо создать новый тип устройства, назвать его, например, Access+Accounting - данный тип устройства будет отражать приложения TvAccess и TvAccounting.



Далее создайте новый тип устройства, назовите его, например, TVIPmedia и добавьте конфигурацию:

Конфигурация

```
#      Middleware
#      ,
# /
customer.name.pid=
#
customer.lastName.pid=
#
customer.firstName.pid=
#
customer.middleName.pid=
```

Выберите в поле OrderManager [динамический класс](#) `ru.bitel.bgbilling.modules.tv.dyn.tvip.TVIPOrderManager` (чтобы его можно было выбрать, данный дин. класс должен быть скомпилирован через вкладку [Управление динамическим кодом](#)).

Создайте устройство типа Access+Accounting и укажите конфигурацию:

Конфигурация

```
# /
# -
accounting.worker.1.thread.count=1
#
accounting.worker.1.tracking.account.1.delay=60
# -
accounting.worker.1.tracking.account.1.batchSize=500
#
accounting.worker.1.tracking.event.1.delay=10
# -
accounting.worker.1.tracking.event.1.batchSize=100

#
# -
om.batch.size=20
# future
```

```

om.batch.pause=0
# future
om.batch.wait=5
#
om.batch.waitNext=5

#
om.error.pause=60
# -
om.error.redelivery.count=5
# -
om.error.alarm.count=20
# ,
om.error.redelivery.timeout=86400

```

Данное устройство будет отражать приложения TVAccess и TVAccounting.

Продукты | Сервисы | Типы аккаунтов | **Устройства** | Переобсчет | Конфигурация модуля

Типы устройств | **Устройства**

Редактор устройства

Основное | Атрибуты | Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства: **Access+Accounting**

Группы устройства: Группы устройств

Конфигурация:

```

1 # Обработка/обсчет
2 # кол-во потоков в обработке
3 accounting.worker.1.thread.count=1
4 # пауза перед следующим выполнением
5 accounting.worker.1.tracking.account.1.delay=60
6 # максимальное кол-во обработанных аккаунтов в одном выполнении
7 accounting.worker.1.tracking.account.1.batchSize=500
8 # пауза перед следующим выполнением
9 accounting.worker.1.tracking.event.1.delay=10
10 # максимальное кол-во обработанных аккаунтов в одном выполнении
11 accounting.worker.1.tracking.event.1.batchSize=100
12
13
14 # Синхронизация
15 # кол-во выполняемых задач в одном блоке
16 om.batch.size=20
17 # время ожидания завершения future задач
18 om.batch.pause=0
19 # время ожидания завершения future задач
20 om.batch.wait=5
21 # время ожидания следующей задачи перед закрытием соединения
22 om.batch.waitNext=5
23

```

Текущая конфигурация

Найти | Найти далее

Строк: 31 | Текущая строка: 9 | Текущий столбец: 46

Сохранить | Отмена

Перечитать конфигурацию на серверах

Создайте дочернее к предыдущему устройству устройство типа TVIPmedia, укажите логин и пароль. Также логин/пароль подключения можно указать в конфигурации устройства, в этом случае поля Логин/Пароль не будут использоваться:

Конфигурация

```

# API
#om.url=https://my.tvip.media/api/provider/
om.login=
om.password=

```

Добавьте тип аккаунта Аккаунт, укажите в его конфигурации ID только что добавленного устройства TVIPmedia:

Конфигурация

```

# TVIP
const.device.id=
#
title.pattern= (${login})

```

Установите галочки на полях **логин** и **пароль**.

Теперь необходимо создать Продукты - они будут отражать тарифы системы TVIPmedia. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор тарифа системы TVIPmedia (поле "ID").

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

Название: Тариф "Социальный"

Идентификатор: 1

Тип: периодический

Продление: в реальном времени

Приоритет: 0

Период с по

Комментарий:

Описание:

Сохранить | Отмена

Чтобы абонент мог активировать Продукт (тем самым активируя подписку в TVIP) необходимо, чтобы в продукте был задан хотя бы один режим активации. Далее от выбранного режима активации может зависеть режим тарификации.

Поиск | Продукты | Сервисы | Типы аккаунтов | Устройства | Переобсчет | Конфигурация модуля

Редактор продукта

Параметры | Атрибуты | Сервисы | Режимы активации | Доступность | Зависимость | Видимость

ID	Период	Название	Деактивация	Реактивация	...
1	28.11.2018-...		до конца оплаченного периода	запрещена	▲

Редактор

Название

Период с 28.11.2018 по

Активация на 0 минут с текущего момента ровно Деактивация до конца оплаченного периода Реактивация запрещена

Сохранить | Отмена

В продуктах-тарифах на вкладке **Зависимость** укажите, что они не совместимы с другими продуктами-тарифами, если есть необходимость, чтобы нельзя было активировать два продукта-тарифа одновременно.

Установите серверные приложения модуля согласно [инструкции](#).

Далее на договоре нужно добавить модуль TV, настроенный **тариф биллинга** и создать аккаунт, который будет связан с аккаунтом из системы TVIP.

Smarty (microimpulse.ru)

Документация по АПИ - <https://microimpuls.github.io/smarty-billing-api-docs/>

Интеграция с MW Smarty (microimpulse.ru) представлена в виде открытого кода (динамические классы).

При интеграции с данной системой продукты модуля (или, в зависимости от конфигурации, сервисы модуля) являются подключаемыми тарифами MW Smarty.

Настройка

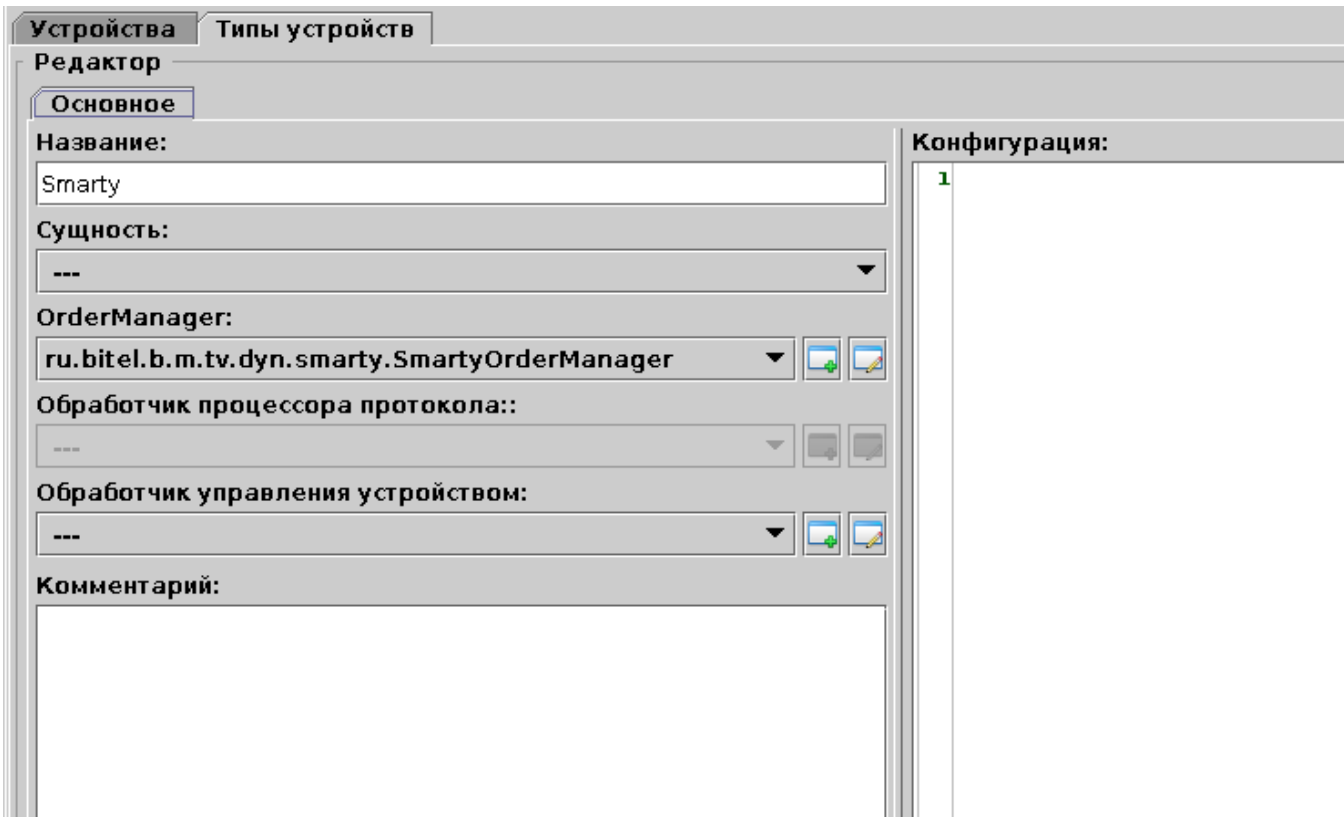
Добавьте конфигурацию модуля и установите ее активной:

Конфигурация

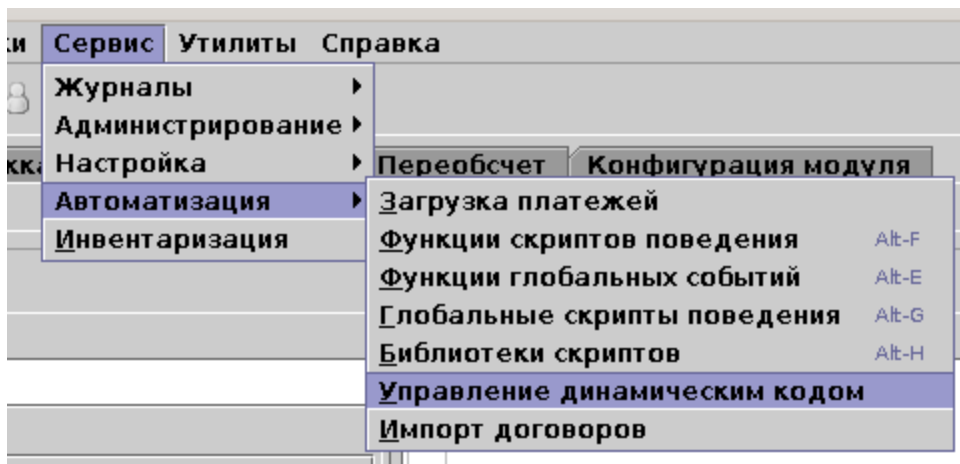
```
#
# ID MW
om.smarty.client.id=
# , ( TV) MW
#
#om.smarty.module.prefix=bg;
# MW
om.smarty.secret.key=
# MW
om.smarty.server.url=
# MW
om.smarty.read.timeout=1000
# MW
om.smarty.connect.timeout=1000
# ID " "
#om.smarty.firstname.id=0
# ID " "
#om.smarty.middlename.id=0
# ID " "
om.smarty.lastname.id=0
```

```
 , , , Access+Accounting - TvAccess TvAccounting.
( . . 24)
```

Далее создайте новый тип устройства, назовите его **Smarty** и укажите в поле **OrderManager** класс ru.bitel.bgbilling.modules.tv.dyn.smarty.
SmartyOrderManager:

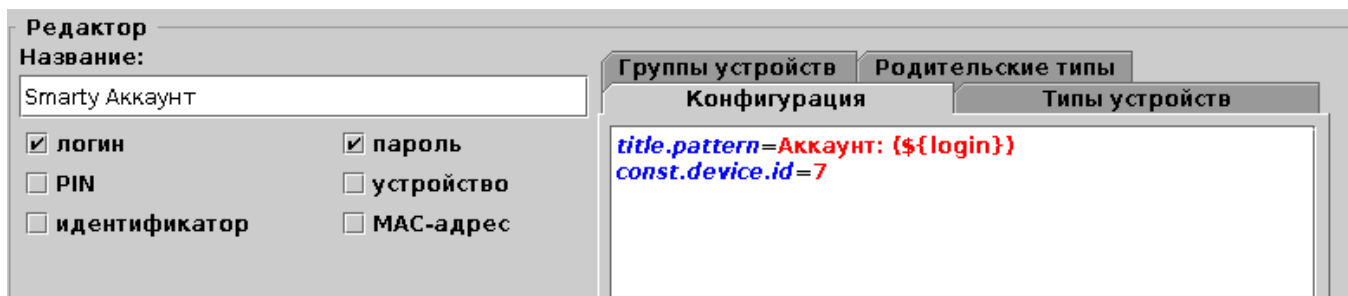


Если выпадающий список **OrderManager** пустой, попробуйте перекомпилировать классы Динамического кода



На закладке **Устройства** добавьте Корневое устройство с Типом **Access+Accounting** и дочернее к нему устройство с типом **Smarty**. Далее на закладке **Типы аккаунтов** добавьте **Тип аккаунта - Smarty**

в конфигурации в параметре **const.device.id=** укажите код устройства **Smarty** с закладки **Устройства**



Редактор

Название:

Smarty Аккаунт

логин пароль

PIN устройство

идентификатор MAC-адрес

Группы устройств Родительские типы

Конфигурация Типы устройств

Access+Accounting

TVEngine (CTI)

qqq

Smarty

Moovi

Теперь необходимо создать Продукты - они будут отражать пакеты тарифы MW Smarty. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор тарифа системы Smarty. Поле **Продление** укажите **в реальном времени**.

Чтобы была возможность активировать тариф из биллинга (тем самым активируя подписку в Smarty) необходимо, чтобы в продукте был задан хотя бы один режим активации.

Для того, чтобы связка биллинг MW Smarty заработала, необходимо что бы были запущены и настроены следующие приложения БД, MQ сервер, сервер биллинга, TV Access сервер, TV Accounting сервер. Также необходимо убедиться, что с данных серверов есть доступ к серверу MW Smarty.

Если есть проблемы с синхронизацией аккаунтов в биллинге и MW Smarty см. в сторону TV Access сервер и MQ сервер

Если проблемы с тарификацией и продлением подписок см. в сторону TV Accounting сервер, TV Access сервер и MQ сервер

Wink

В конфиг модуля добавляем

```
tv.rest.handler.wink.class=ru.bitel.bgbilling.modules.tv.dyn.wink.WinkHandler
tv.product.incompatible.items=0: ;1:;-1: ;-2: 1

#wink.base.id=25
#wink.ext.id=31
#wink.subs.id=36
#wink.serv.id=48

# Wink
om.wink.provider.url=http://host[:port]
om.wink.provider.url.prefix=/TEST/WinkProviderService-web/facade
om.wink.provider.username=provider_login
om.wink.provider.password=passwd
om.wink.partner.region.id=nw
om.wink.test.mode=test
om.wink.ott.tv.code.pattern=rt[23456789abcdefghijklmnopqrstuvwxyz]{5}

# ID
#om.wink.charge.type.id=
```

Добавляем **Тип устройств**

Поиск Продукты Сервисы **Устройства** Типы аккаунтов Переобсчет Конфи

Устройства Типы устройств

Редактор

Основное

Название: Wink Конфигурация 1

Сущность: ---

OrderManager: ru.bitel.b.m.tv.dyn.wink.TeilaWinkOrderManager

Обработчик процессора протокола: ---

Обработчик управления устройством: ---

Комментарий:

Добавляем **Устройства**

Поиск | Продукты | Сервисы | **Устройства** | Типы аккаунтов | Переобсч

Устройства | Типы устройств

Редактор устройства

Основное | Атрибуты | Комментарий

Идентификатор:

Хост/порт:

Логин:

Пароль:

Community/secret:

Тип устройства:

OrderManager отключен

Период с по

Группы устройства:
 Группы устройств

Конфигура
 1

Добавляем Тип Аккаунта

Поиск | Продукты | Сервисы | **Устройства** | Типы аккаунтов | Переобсчет | **Конфигурация модуля**

ID	Название
7	Wink Аккаунт

Редактор

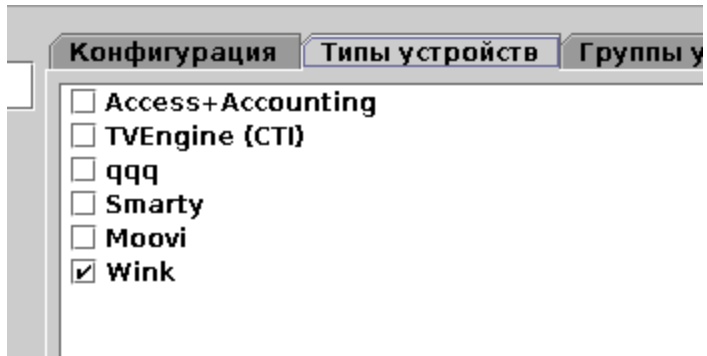
Название:

логин пароль
 PIN устройство
 идентификатор MAC-адрес

Конфигурация | Типы устройств | Групп

title.pattern=Аккаунт: ({login})

Указываем Тип устройства



Добавляем **Продукты**

Wink			16
Основные пакеты			25
Трансформер	81356		26
Трансформер Акция	81353		27
Максимум	81383		28
Премиум	81386		29
КиноVIP	81328		30
Дополнительные пакеты			31
Взрослый	81032		32
МАТЧ! ФУТБОЛ	81119		33
МАТЧ ПРЕМЬЕР	81033		34
НАСТРОЙ КИНО!	81298		35
Подписка			36
Подписка AMEDIATEKA Ноi	81250		37
Подписка Детский клуб	81122		38
Подписка English Club	81325		39
Подписка Суперхиты	81439		40
Подписка Лицей	81401		41
Подписка more.TV	81359		42
Подписка START	81393		43
Подписка START+more.TV	81394		44
Абонемент Киноман	81446		45
Абонемент Киноман спеi	81447		46
Подписка START. Сезон ки	81452		47
Услуги			48
Караоке	81419		49

Для каждого **Продукта** указываем **Идентификатор**

Поиск | **Продукты** | Сервисы | Устройства | Типы аккаунтов

Редактор продукта

Продукт: Трансформер

Параметры | Атрибуты | Сервисы | **Режимы активации** | Д

Идентификатор:

Тип:

Продление:

Приоритет:

Период с по

У Продукта должен быть один Режим активации

Редактор

Название

Активация на месяцев с ровно Деактивация Реактивация

Указываем Тарифные планы для Продукта

Редактор продукта

Продукт: Максимум

Параметры | Атрибуты | Сервисы | **Режимы активации** | **Доступность**

Тарифные планы:

- 111111:test<~>
- abon
- abonka
- airat_plan
- amir
- artur_tariff
- boris
- boris copy
- cerbercrypt test dimon
- den_tariff
- den1
- den4
- dialup
- DialUp - Test delayed
- inet
- IPN
- IPN_RULE

И настраиваем Зависимости

не зависит		Основные пакеты
не зависит	взаимоисключение в момент заказа	Трансформер
не зависит	взаимоисключение в момент заказа	Трансформер Акция
не зависит	не совместим	Максимум
не зависит	взаимоисключение с 1 числа	Премиум
не зависит	взаимоисключение с 1 числа	КиноVIP
не зависит		Дополнительные пакеты
не зависит	доступен	Взрослый
не зависит	доступен	МАТЧ! ФУТБОЛ
не зависит	доступен	МАТЧ ПРЕМЬЕР
не зависит	доступен	НАСТРОЙ КИНО!
не зависит		Подписка
не зависит	доступен	Подписка AMEDIATEKA Home of HBO
не зависит	взаимоисключение в момент заказа	Подписка Детский клуб
не зависит	доступен	Подписка English Club
не зависит	доступен	Подписка Суперхиты
не зависит	доступен	Подписка Лицей
не зависит	взаимоисключение в момент заказа	Подписка more.TV
не зависит	доступен	Подписка START
не зависит	доступен	Подписка START+more.TV
не зависит	доступен	Абонемент Киноман
не зависит	доступен	Абонемент Киноман спец.цена
не зависит	доступен	Подписка START. Сезон кино и сериалов
не зависит		Услуги
не зависит	доступен	Караоке

Далее настраиваем **Тарифные планы**

Продукт:
- Трансформер [81356]

Период:
320.0 за 1 месяц (подневное до конца месяца)

Продукт:
- Трансформер Акция [81353]

Период:
299.0 за 1 месяц (подневное до конца месяца)

Продукт:
- Максимум [81383]

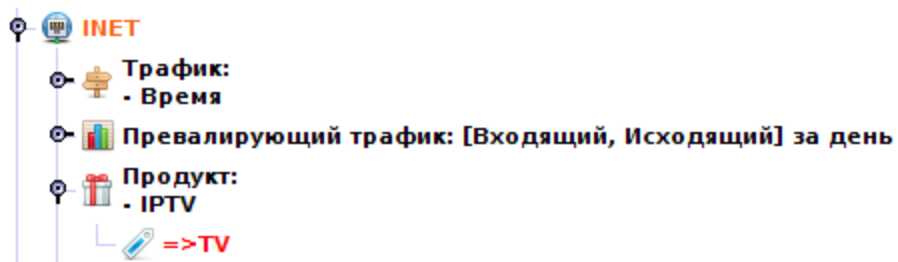
Период:
620.0 за 1 месяц (подневное до конца месяца)

Добавляем **Аккаунт** на договор

Аккаунты	
Параметры Продукты	
Тип	
Wink Аккаунт	
Устройство	
Wink: Wink	
Логин	PIN
10000017 <input type="checkbox"/> авто
Идентификатор	
120000000017	
Комментарий	

Модуль Inet (продукты TV - опции Inet)

Интеграция с модулем Inet производится на уровне активных продуктов с помощью опций Inet. В тариф необходимо добавить ветку Продукт, в нее ветку Опция. Внутри ветки Продукт запрос будет попадать только, если выбранный продукт(ы) на данный момент активен. А уже в модуле Inet, в зависимости от того активна или нет данная опция, можно производить какие-либо действия, например, менять параметры порта на коммутаторе.



NetUP IPTV

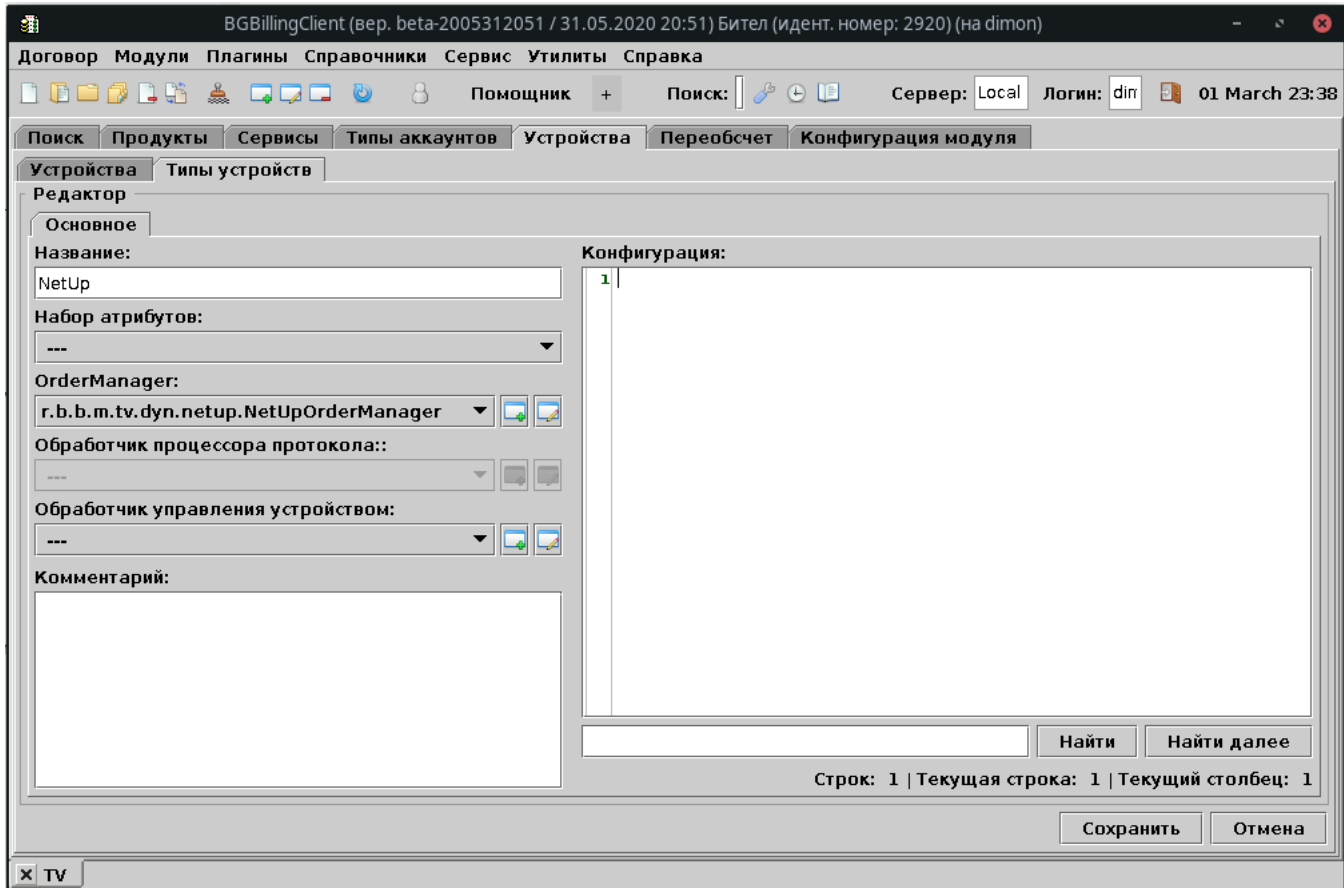
Реализация протокола "NetUP IPTV MW API for external billing systems (1.0.0)" в виде открытого кода (динамические классы).

При интеграции с системой продукты модуля являются подключаемыми media content groups MW.

Помимо стандартной конфигурации модуля используются следующие настройки:

```
#  
om.netup.providerURL=http://...nnp/***/api
```

В корневом типе устройства создайте тип устройства (например, NetUp), укажите OrderManager: **ru.bitel.bgbilling.modules.tv.dyn.netup.NetUpOrderManager** (должен быть заранее скомпилирован).



В корневом устройстве (например, Access+Accounting) создайте дочернее устройство типа NetUp, в конфиге удачное место чтобы указать например providerUrl (вместо конфига модуля)

Помощник + Поиск: Сервер: Local Логин: admin 02 March 00:00

Поиск Продукты Сервисы Типы аккаунтов **Устройства** Переобсчет Конфигурация модуля

Устройства **Типы устройств**

Редактор устройства

Основное **Атрибуты** Комментарий

id: 6
Название: NetUp: netup [4354.45.2523.75432]
Идентификатор:
netup
Хост/порт:
4354.45.2523.75432
Логин:
Пароль:
Community/secret:
Тип устройства:
NetUp
 OrderManager отключен
Период с по
Группы устройства:
 Группы устройств

Конфигурация: Текущая конфигурация

```
1 # url для нетапа
2 om.netup.providerURL=http://4354.45.2523.75432/**/api
```

Найти Найти далее

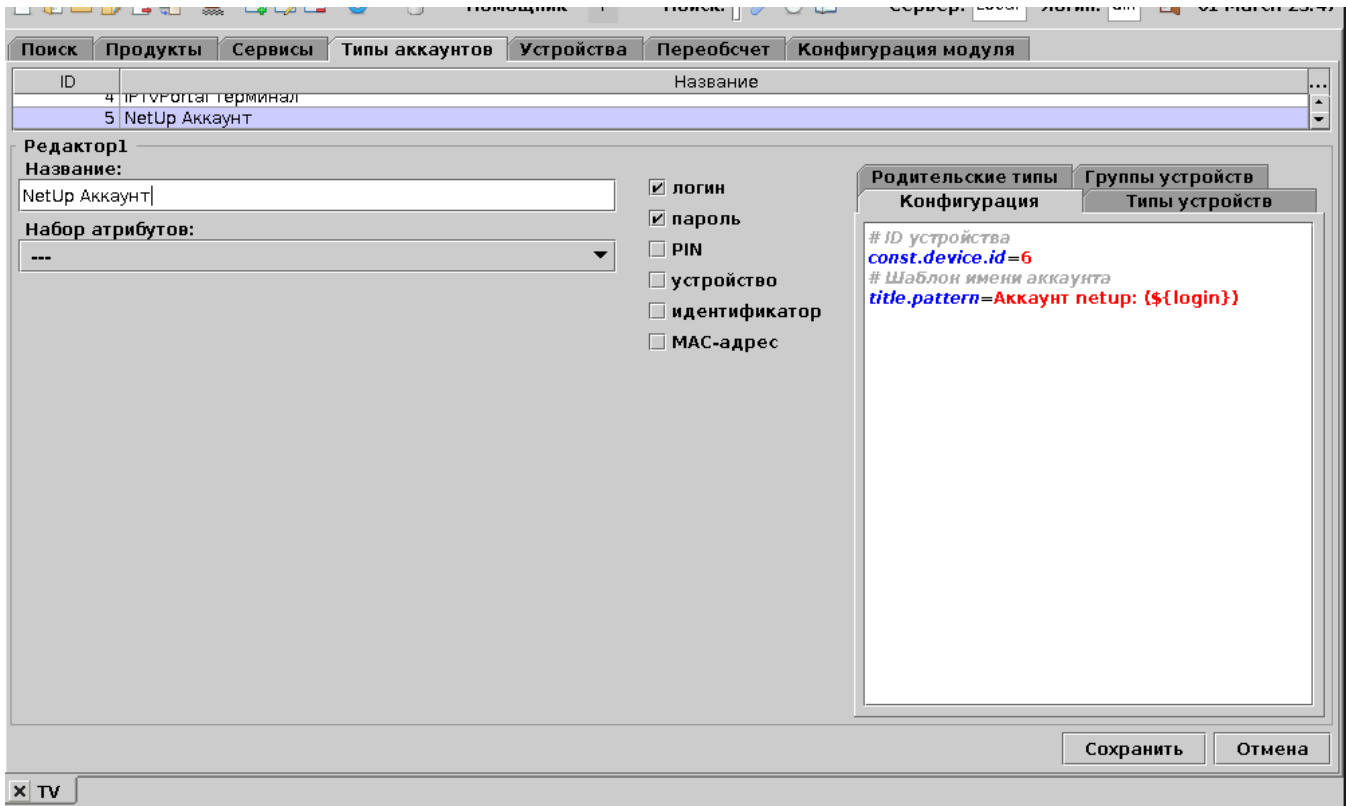
Строк: 2 | Текущая строка: 2 | Текущий столбец: 51

Сохранить Отмена

Перечитать конфигурацию на серверах

x TV

В типах аккаунтов создайте аккаунт NetUp



конфигурация

```
# ID
const.device.id=6
#
title.pattern= netup: (${login})
```

Далее создаются Продукты - они будут отражать группы/пакеты системы netup. Поле Название - это название, с которым данный продукт будет отображаться в биллинге. Поле Идентификатор - идентификатор пакета системы (поле "media_group_code" итд). Чтобы абонент мог активировать Продукт (тем самым активируя подписку) необходимо, чтобы в продукте был задан хотя бы один режим активации. Далее от выбранного режима активации может зависеть режим тарификации.

Редактор продукта

Продукт: NetUp пакет 1|

Идентификатор:

Тип:

Продление:

Приоритет:

Стоимость:

Период:
с по

Комментарий:

Описание:

Сохранить | Отмена

Модуль Vidimax

Назначение модуля

Модуль Vidimax предназначен для интеграции биллинга с поставщиком услуг video on demand Vidimax. На данный момент модуль предоставляет следующие возможности:

- автоматическая привязка ЛС Абонента в Vidimax к договору в биллинге.
- установка тарифов Vidimax на договоре в момент создания ИД абонента.
- возможность оплаты услуг Vidimax, после привязки, со счета клиента;
- просмотр платежей и активных тарифов клиента из договора.

Настройка модуля

Установите модуль на сервер, обновите клиентскую часть. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, скопируйте туда приведённый ниже текст и введите требуемые параметры. После этого сохраните конфигурацию и сделайте её активной.

Конфигурация

```
#####

# ( Vidimax- )
operator.secret=password

# Basic http ( Vidimax- )
basic.auth.login=username
basic.auth.password=password

# Basic http Vidimax ( Vidimax- )
vidimax.basic.auth.login=vmusername
vidimax.basic.auth.password=vmpassword

# url ( TV VoD ) listActiveTariffsTv listActiveTariffsVod
url.tariff.active=https://adressVidimax:1234/api/5.0/

# ,
account.default.serviceId=1

#####
# ,
contract.status.active.codes=0

# , .
tariffsIds=0,210,492
tariff.0.title= 90
tariff.210.title= 300
tariff.492.title= 500

# ( tariffsIds ):
tariffsIds.available=492,0

# .
serviceIds=1,2,3,4,5
service.1.title= SD
service.2.title= HD
service.3.title= SD
service.4.title= HD
service.5.title=.

# account.default.serviceId .
service.1.accountServiceId=41
service.2.accountServiceId=52
service.4.accountServiceId=21

#####
```

```
# ( ),
#tariffsIds.notAccounting=492

# ( )
#contract.canDeletePaired=1
```

Замечания:

1. Прежде, чем задавать **account.default.serviceId** или **service.X.accountServiceId**(где X код сервиса видимакса) необходимо создать соответствующую услугу в **Модули->Редактор модулей и услуг->"Название вашего модуля"**, а коды услуг вписать в значения параметров.
2. После заключения договора с системой Vidimax нужно будет передать им значение URL-адреса, по которому к биллингу будут приходить запросы. URL должен выглядеть следующим образом: **http://<адрес_машины_биллинга>/vidimax_api/<mid>** .

Например, если у вас биллинг находится по адресу **http://billing.example.com/bgbilling/** и модуль Vidimax имеет mid=17, то результирующее URL, которые нужно передать компании Vidimax, выглядит следующим образом: **http://billing.example.com/bgbilling/vidimax_api/17**.

3. Логин и пароль для запроса на списание средств запрашиваются при входящем запросе с сервера Vidimax на списание средств.

Принцип работы модуля

Последовательность действий при работе с модулем следующая:

1. Заводится договор;
2. Вносится номер телефона в модуль(он выступает в качестве идентификатора для Vidimax-а) и если надо выбирается тариф;
3. Пользователь подключает приставку и производится автоматическое связывание ЛС Vidimax-а и Биллинга на основании идентификатора;
4. Далее Vidimax может запросить списание средств;

Алгоритм списания средств:

1. Поиск договора с переданным от Vidimax ид. абонента;
2. Проверка является ли договор связанным;
3. Проверка является ли статус договор разрешенным для данного модуля(параметр contract.status.active.codes);
4. Проверка на достаточность средств на договоре(расход Vidimax-а будет отвергнут, если баланс станет ниже 0 в результате этого расхода);
5. Добавляется наработка на договор(если только данный тариф не входит в параметр tariffsIds.notAccounting и расход является абонентской платой);

Мониторинг платежей

В клиенте биллинга(а так-же в ЛК клиента) есть возможность отслеживать историю расходов по каждому абоненту. Для этого необходимо выбрать модуль **Vidimax** в дереве параметров договора. Платежи можно отображать с фильтрацией по периоду, в течение которого производилось списание средств.

Транзакции

Пользовательские данные

Идентификатор Сохранить Удалить

Активные тарифы

id	Название	Сумма	Основной тариф	Начало	Окончание	Завершение(отмена подписки)	Комментарий
584	SkyNet-TV Maximum TV	880.00	<input type="checkbox"/>	18.02.2015 03:00:00	09.01.1970 06:34:17		Пакет включает полный список теле...
582	SkyNet-TV Amedia Premiu...	99.00	<input type="checkbox"/>	18.02.2015 03:00:00	09.01.1970 06:34:17		Лучшие сериалы планеты по версии...
492	Безлимит.Спецтариф.6	295.00	<input checked="" type="checkbox"/>	18.02.2015 03:00:00	09.01.1970 06:34:17		Спецтариф "Безлимитный" предоста...

Фильтр

Показать за период с по OK

id	Дата	Сумма	id операции	id абонента	Тариф	Сервис	Комментарий
28	18.02.2015 17:06:50	295.00	20150218389822	7912914	Безлимит.Спецтариф.6	Абон. плата за тариф	Subscription charge for ServiceS...
29	18.02.2015 17:53:01	149.00	20150218389824	7912914	Безлимит.Спецтариф.6	Аренда HD фильма	Фильм [Легкие деньги HD]
30	18.02.2015 17:53:55	120.00	20150218389826	7912914	Безлимит.Спецтариф.6	Аренда SD фильма	Фильм [Легкие деньги]
31	18.02.2015 17:57:42	880.00	20150218389830	7912914	SkyNet-TV Maximum TV	Абон. плата за тариф	Subscription charge for ServiceS...
32	18.02.2015 18:42:18	99.00	20150218389834	7912914	SkyNet-TV Amedia Premiu...	Абон. плата за тариф	Subscription charge for ServiceS...

Модуль Megogo

Назначение модуля

Модуль Megogo предназначен для интеграции биллинга с одноименным видеосервисом Megogo.

Настройка модуля

Конфигурация модуля

```
#
# ( Megogo )
contract.status.suspend.codes=3,4
# , Megogo
megogo.partner.id=
# . Megogo prefixUser+contractId , contractId
megogo.prefix.user=
# , Megogo Megogo
megogo.salt=
# , @. - id@domain.com
credentials.domain=test.ru

#
# ( )
service.main.cost.order=293,294,295,296
# ; 1- , 2- , 3-
service.main.migration.expensive.type=1
# - , = 1(-) , = 0( ) ,
#
service.main.migration.expensive.value=0
# ; 1- , 2- , 3-
service.main.migration.cheap.type=2
# - , 1(-)
service.main.migration.cheap.value=5
#
# Megogo:
# 1 ( ) - ( . ).
# 2 - ( . )
#
user.create.mode=2
```

После установки модуля в клиенте биллинга перейдите в "Модули -> Редактор модулей и услуг" и добавьте модуль Megogo. Далее в модуле добавьте требуемые услуги для каждой подписки.

Код	Название	Используется	Ед. измерения	...
298	Доп. пакет "Наш футбол"	<input checked="" type="checkbox"/>		
297	Доп. пакет Амедиа	<input checked="" type="checkbox"/>		
293	Подписка Легкая	<input checked="" type="checkbox"/>		
295	Подписка Максимальная	<input checked="" type="checkbox"/>		
294	Подписка Оптимальная	<input checked="" type="checkbox"/>		
296	Подписка Премиум	<input type="checkbox"/>		

После перейдите в созданный модуль и настройте конфигурацию модуля. А в вкладке "Настройка подписок" для ранее созданных услуг, которые соотносятся один к одному с подпиской, установите тип, описание и ключ(выдается Megogo).

Id	Услуга	Тип	Используется	Описание	Пользователь	...
2	Подписка Легкая	Основная	<input checked="" type="checkbox"/>	Легкая	lldar	
4	Подписка Премиум	Основная	<input type="checkbox"/>	Премиум	lldar	
5	Подписка Максимальная	Основная	<input checked="" type="checkbox"/>	Макс 34/день	lldar	
6	Подписка Оптимальная	Основная	<input checked="" type="checkbox"/>	Оптимум	lldar	
1	Доп. пакет Амедиа	Дополнительная независимая	<input checked="" type="checkbox"/>	Амедиа	lldar	
3	Доп. пакет "Наш футбол"	Дополнительная независимая	<input checked="" type="checkbox"/>	Наш футбол	lldar	

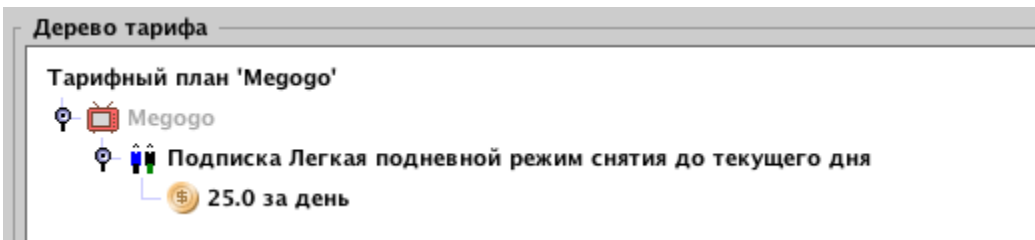
Редактор

Услуга: Подписка Оптимальная MegogoKey: testmplus

Тип: Основная Описание: Оптимум

Модуль Megogo

Создайте тариф.



Задачи модуля

Начисление Megogo - производит начисление наработки по услугам модуля Megogo по всем договорам. Необходимо установить запуск один раз в сутки(в начале суток).

Изменение подписок - производит смену подписок для подписок период которых уже наступили или закончился. Необходимо установить запуск как минимум один раз в сутки.

Договор

После добавления модуля на договор можно добавить подписку в клиенте так и в личном кабинете.

Редактор

Период:
с по

Комментарий:

Отправить запрос сразу

Услуга:

- Доп. пакет "Наш футбол"
- Доп. пакет Амедиа
- Подписка Легкая
- Подписка Максимальная
- Подписка Оптимальная

Если установить "Отправить запрос сразу", то подписка будет активирована(то есть будет отправлен запрос к Megogo) или деактивирована непосредственно в момент её добавления, при условии, что период начала или конца текущий день. Иначе это будет произведено в очередной запуск задачи "изменение подписок"

Модули платежных систем

Кратко о модулях

Модуль MPS предназначен для обработки платежей от различных платежных систем по принципу оплаты через терминал. Другие модули биллинга, описанные в этом разделе реализуют поддержку соответствующих платежных систем. Бесплатный модуль Card имеет свой протокол для приема платежей, а также интерфейс дилера.

SSLv3

SSLv3 по умолчанию отключен в JRE/JDK с билдов:



- Java 8 update 31
- Java 7 update 75
- Java 6 update 91

Некоторые платежные системы не работают через TLS, а требуют SSLv3. Т.е. при обновлении Java они перестают работать. Чтобы включить поддержку SSLv3 обратно, нужно в файле конфигурации <JRE_HOME>/lib/security/java.security убрать из параметра jdk.tls.disabledAlgorithms значение "SSLv3".

Однако следует учитывать, что SSLv3 устарел и не рекомендуется к использованию.

<http://www.oracle.com/technetwork/java/javase/documentation/cve-2014-3566-2342133.html>

SSLv3 is disabled by default

Starting with JDK 6u91 release, the SSLv3 protocol (Secure Socket Layer) has been deactivated and is not available by default. See the java.security.Security property jdk.tls.disabledAlgorithms in <JRE_HOME>/lib/security/java.security file.

If SSLv3 is absolutely required, the protocol can be reactivated by removing "SSLv3" from the jdk.tls.disabledAlgorithms property in the java.security file or by dynamically setting this Security property to "true" before JSSE is initialized.

It should be noted that SSLv3 is obsolete and should no longer be used.

Также для встроенного tomcat нужно добавить дополнительно параметр в data.properties сервера биллинга:

```
connector.https.sslEnabledProtocols=SSLv2,SSLv3,TLSv1,TLSv1.1,TLSv1.2,SSLv2Hello
```

Так как tomcat уже по умолчанию тоже не поддерживает SSLv3. Описание этого параметра есть тут :

<http://ci.apache.org/projects/tomcat/to.../http.html>

Здесь перечислены все возможные протоколы, некоторые можно убрать согласно целям и задачам.

Второй параметр определяет используемые cipher-ы:

```
connector.https.ciphers=TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_RC4_128_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA
```

Здесь перечислены подходящие параметры, например, всё отсюда: <https://www.ssllabs.com/ssltest/viewClient.html?name=Java&version=6u45> с убранными WEAK и INSECURE.

SSLv3

SSLv3 по умолчанию отключен в JRE/JDK с билдов:



- Java 8 update 31
- Java 7 update 75
- Java 6 update 91

Некоторые платежные системы не работают через TLS, а требуют SSLv3. Т.е. при обновлении Java они перестают работать. Чтобы включить поддержку SSLv3 обратно, нужно в файле конфигурации <JRE_HOME>/lib/security/java.security убрать из параметра jdk.tls.disabledAlgorithms значение "SSLv3".

Однако следует учитывать, что SSLv3 устарел и не рекомендуется к использованию.

<http://www.oracle.com/technetwork/java/javase/documentation/cve-2014-3566-2342133.html>

SSLv3 is disabled by default

Starting with JDK 6u91 release, the SSLv3 protocol (Secure Socket Layer) has been deactivated and is not available by default. See the java.security.Security property jdk.tls.disabledAlgorithms in <JRE_HOME>/lib/security/java.security file.

If SSLv3 is absolutely required, the protocol can be reactivated by removing "SSLv3" from the jdk.tls.disabledAlgorithms property in the java.security file or by dynamically setting this Security property to "true" before JSSE is initialized.

It should be noted that SSLv3 is obsolete and should no longer be used.

Также для встроенного tomcat нужно добавить дополнительно параметр в data.properties сервера биллинга:

```
connector.https.sslEnabledProtocols=SSLv2,SSLv3,TLSv1,TLSv1.1,TLSv1.2,SSLv2Hello
```

Так как tomcat уже по умолчанию тоже не поддерживает SSLv3. Описание этого параметра есть тут :

<http://ci.apache.org/projects/tomcat/to.../http.html>

Здесь перечислены все возможные протоколы, некоторые можно убрать согласно целям и задачам.

Второй параметр определяет используемые cipher-ы:

```
connector.https.ciphers=TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_RC4_128_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA
```

Здесь перечислены подходящие параметры, например, всё отсюда: <https://www.ssllabs.com/ssltest/viewClient.html?name=Java&version=6u45> с убранными WEAK и INSECURE.

Модуль Alfabank (АО "Альфа банк"

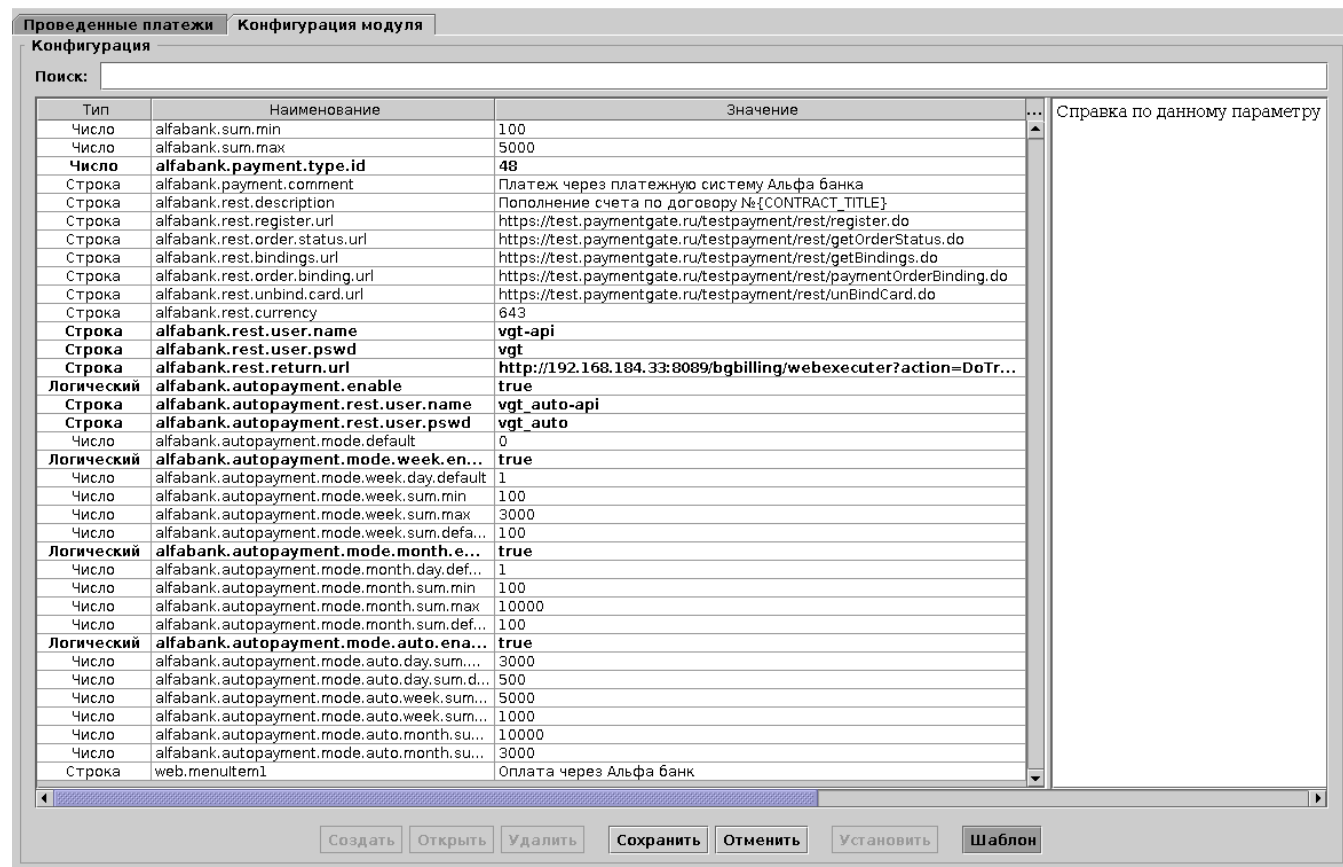
Назначение модуля

Модуль биллинга **Alfabank** предназначен для оплаты картами через процессинг АО "Альфа банка".

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.



Тип	Наименование	Значение
Число	alfabank.sum.min	100
Число	alfabank.sum.max	5000
Число	alfabank.payment.type.id	48
Строка	alfabank.payment.comment	Платеж через платежную систему Альфа банка
Строка	alfabank.rest.description	Пополнение счета по договору №{CONTRACT_TITLE}
Строка	alfabank.rest.register.url	https://test.paymentgate.ru/testpayment/rest/register.do
Строка	alfabank.rest.order.status.url	https://test.paymentgate.ru/testpayment/rest/getOrderStatus.do
Строка	alfabank.rest.bindings.url	https://test.paymentgate.ru/testpayment/rest/getBindings.do
Строка	alfabank.rest.order.binding.url	https://test.paymentgate.ru/testpayment/rest/paymentOrderBinding.do
Строка	alfabank.rest.unbind.card.url	https://test.paymentgate.ru/testpayment/rest/unBindCard.do
Строка	alfabank.rest.currency	643
Строка	alfabank.rest.user.name	vgt-api
Строка	alfabank.rest.user.pswd	vgt
Строка	alfabank.rest.return.url	http://192.168.184.33:8089/bgbilling/webexecuter?action=DoTr...
Логический	alfabank.autopayment.enable	true
Строка	alfabank.autopayment.rest.user.name	vgt_auto-api
Строка	alfabank.autopayment.rest.user.pswd	vgt_auto
Число	alfabank.autopayment.mode.default	0
Логический	alfabank.autopayment.mode.week.en...	true
Число	alfabank.autopayment.mode.week.day.default	1
Число	alfabank.autopayment.mode.week.sum.min	100
Число	alfabank.autopayment.mode.week.sum.max	3000
Число	alfabank.autopayment.mode.week.sum.defa...	100
Логический	alfabank.autopayment.mode.month.e...	true
Число	alfabank.autopayment.mode.month.day.def...	1
Число	alfabank.autopayment.mode.month.sum.min	100
Число	alfabank.autopayment.mode.month.sum.max	10000
Число	alfabank.autopayment.mode.month.sum.def...	100
Логический	alfabank.autopayment.mode.auto.ena...	true
Число	alfabank.autopayment.mode.auto.day.sum...	3000
Число	alfabank.autopayment.mode.auto.day.sum.d...	500
Число	alfabank.autopayment.mode.auto.week.sum...	5000
Число	alfabank.autopayment.mode.auto.week.sum...	1000
Число	alfabank.autopayment.mode.auto.month.su...	10000
Число	alfabank.autopayment.mode.auto.month.su...	3000
Строка	web.menuitem1	Оплата через Альфа банк

В настройках Личного кабинета на сайте АО "Альфа банка" необходимо прописать Callback URL (merchant-url) для оповещения о статусах транзакций, в формате

`http[s]://host[:port]/bgbilling/alfabank/<код_модуля>`

Оплата через Личный кабинет

Если у клиента подключен экземпляр модуля в дереве договора, то он может осуществлять оплату через шлюз АО "Альфа банк", используя личный web-интерфейс.

В личном кабинете на странице отображается история платежей, совершенных клиентом:

Пополнить баланс

Настроить Автоплатеж

Платежный сервис Альфа банка

Сумма (от 100,00 до 5 000,00):

Оплатить

Текущий баланс:
601,67 руб.

Период:

<

Сентябрь, 2016

>

Страница:

<<

<

1 из 1 [20]

>

>>

[показать допол. фильтры]

№	Дата/Время оплаты	Статус	Сумма
4	19.09.2016 16:42:21	Одобренная	100,00
5	23.09.2016 16:48:28	Отклоненная	151,00
6	23.09.2016 16:48:28	Отклоненная	152,00
7	23.09.2016 16:48:28	Отклоненная	153,00
8	19.09.2016 17:08:40	Одобренная	50,00
9	23.09.2016 13:36:24	Одобренная	108,00
10	23.09.2016 16:46:23	Одобренная	109,00
11	23.09.2016 16:48:52	Одобренная	110,00

Над таблицей с историей платежей расположена форма для совершения нового платежа. Чтобы осуществить платеж, необходимо заполнить обязательные поля **Сумма**. Далее необходимо нажать кнопку **Оплатить** и подтвердить. После подтверждения клиент попадает на страницу с формой ввода реквизитов карты на сайте Альфа банка.

После оплаты (или отмены), клиент будет перенаправлен на страницу, указанную в конфигурации модуля в параметре [alfabank.rest.return.url](#).

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль Alfabank в дереве параметров договора. Здесь присутствует фильтр по статусу платежей с указанием периода, когда производилась оплата.

Баланс (сентябрь 2016)		Транзакции		Автоплатеж		Период:		Статус:		Проверить...		Всего...	
Входящий остаток	123,67	ID	...	Автоплатеж	Карта	Имя держателя карты	ApprovalCode	AuthCode	...	Сумма
Приход	478,00	4	...	<input type="checkbox"/>	555555**5599	User USER	123456	2	...	100,00
Наработка	0,00	5	...	<input checked="" type="checkbox"/>				2	...	151,00
Расход	0,00	6	...	<input checked="" type="checkbox"/>				2	...	152,00
Возврат	0,00	7	...	<input checked="" type="checkbox"/>				2	...	153,00
Исходящий остаток	601,67	8	...	<input type="checkbox"/>	555555**5599	user user	123456	2	...	50,00
Резерв	0,00	9	...	<input type="checkbox"/>	555555**5599	USER USER	123456	2	...	108,00
Доступная сумма	601,67	10	...	<input type="checkbox"/>	555555**5599	USER USER	123456	2	...	109,00
Тарифные планы		11	...	<input checked="" type="checkbox"/>	555555**5599	USER USER	123456	2	...	110,00
Модули												Сумма итог:	933,00
Alfabank													
Скрытый													

Для просмотра ВСЕХ платежей, проведенных с использованием модуля Alfabank, существует глобальный монитор транзакций в параметрах данного модуля биллинга (**Модули -> Модуль Alfabank**). На открывшейся вкладке у Вас есть возможность просмотреть все платежи, совершенные вашими абонентами за указанный временной период. Также можно установить фильтр платежей по имени договора, по статусу.

Проведенные платежи		Конфигурация модуля		Период:		Статус:		Договор:		Проверить...		Всего записей: 8	
ID	Договор	Дата создания	Дата обработки	Orderid	Автоплатеж	Карта	Имя держателя карты	Описание ошибки	Статус	Сумма
4	kirill	19.09.2016 16:...	23.09.2016 16:4...	3ae1bb9d-3bc4-46fa-ae27-bd7b9159435b	<input type="checkbox"/>	555555**5599	User USER	Успешно	Одобренная	100,00
5	kirill	19.09.2016 16:...	23.09.2016 16:4...	665ee9ab-7a4a-486e-b4dd-2be0f8072c53	<input checked="" type="checkbox"/>			Платеж отклонен	Отклоненная	151,00
6	kirill	19.09.2016 16:...	23.09.2016 16:4...	03b9feb3-1507-466b-8f6b-b783d5325978	<input checked="" type="checkbox"/>			Платеж отклонен	Отклоненная	152,00
7	kirill	19.09.2016 16:...	23.09.2016 16:4...	271f26a1-448a-4e14-b923-31d74fe2199	<input checked="" type="checkbox"/>			Платеж отклонен	Отклоненная	153,00
8	kirill	19.09.2016 17:...	19.09.2016 17:0...	6238996a-aad8-4ea5-a6b4-8271ee26645b	<input type="checkbox"/>	555555**5599	user user	Успешно	Одобренная	50,00
9	kirill	23.09.2016 13:...	23.09.2016 13:3...	0cd0ce47-4c70-4919-856a-8985f64c9b67	<input type="checkbox"/>	555555**5599	USER USER	Успешно	Одобренная	108,00
10	kirill	23.09.2016 16:...	23.09.2016 16:4...	8717bc95-5825-4e62-9087-c914460e6c33	<input type="checkbox"/>	555555**5599	USER USER	Успешно	Одобренная	109,00
11	kirill	23.09.2016 16:...	23.09.2016 16:4...	2df5ca79-6700-4b7d-88ea-851a63631c94	<input checked="" type="checkbox"/>	555555**5599	USER USER	Успешно	Одобренная	110,00
										Сумма итог:	933,00		

Модуль Assist

Назначение модуля Assist

Модуль Assist предназначен для проведения платежей через систему электронных платежей Assist.Ru. В данный момент система позволяет принимать следующие типы платежей:

- оплата по кредитной карте;
- оплата через платёжную систему WebMoney Transfer;
- оплата через платёжную систему Яндекс.Деньги;
- оплата через QIWI;
- оплата кредитной картой с использованием Assist®ID.

Следует учесть, что модуль изначально создавался для оплаты только по кредитным картам. Для оплаты через систему WebMoney Transfer следует использовать предназначенный для этого модуль, остальные системы оплаты не тестировались.

Для работы в нетестовом режиме у вас должен быть заключён договор с системой и создан готовый интернет-магазин в системе [Assist](#).

В данный момент модуль работает только через новый протокол (конца 2011 года), работа через старый протокол больше не поддерживается.

Настройка модуля Assist

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте новую конфигурацию модуля в редакторе конфигурации и сделайте её активной. Шаблон конфигурации приведен ниже:

Конфигурация

```
# -.
web.menuItem1= Assist.Ru

# (ID), .
assist.paymenttype=1

# ASSIST.
assist.Merchant_ID=*****
# webexecuter. , .
assist.path=http://provaider.ru/bgbilling/webexecuter
# . ASSIST ( ).
assist.OrderComment= ${contract} (${contract_comment})
# ( ).
assist.payment_comment= . ${contract} (${contract_comment}) Assist ( #${billnumber})(: ${name})(
${cardnumber} ${cardtype}/${cardsubtype}, : ${cardholder}) ${date}, ${total}
# (0 - , 1 - ). - .
#assist.Delay=0

# sale
assist.login=*****
# sale
assist.password=*****

# . (1 - , 0 - ). , 0.
assist.CardPayment=1
# WebMoney Transfer. (1 - , 0 - ).
assist.WMPayment=0
# .. (1 - , 0 - ).
assist.YMPayment=0
# Assist@ID (1 - , 0 - )
assist.AssistIDPayment=0
# QIWI
assist.QIWIPayment=0
assist.QIWIMtsPayment=0
assist.QIWIMegafonPayment=0
assist.QIWIBeelinePayment=0

# :
# test - ;
# battle - ;
# battletest - , ( TestMode=1).
assist.ShopMode=test

# ( ) ( ).
assist.actionUrl=https://test.paysecure.ru/pay/order.cfm
```

После регистрации на сервисе Assist.Ru вам дадут параметры "код магазина", а также возможность настроить "логин" и "пароль" типа sale, которые нужно прописать в конфигурации. Чтобы узнать ID нужного платежа, выберите его в справочнике и нажмите Ctrl+i.

Необходимо получить у службы поддержки ООО «АССИСТ» (support@assist.ru) идентификатор предприятия merchant_id, логин и пароль администратора юридического лица для работы с Личным кабинетом (он находится по адресу <https://account.paysecure.ru/>), а также логин и пароль пользователя типа sale для работы с веб-сервисами.

Основное, на что необходимо обратить внимание при настройке магазина в личном кабинете:

- Режим. Тестовый или Рабочий. Таким образом, режим определятся двумя настройками — в самом магазине Assist и в конфигурации модуля.
- Для ручного запроса результатов платежей необходимы логин и пароль типа sale (см. личный кабинет).

- Можно также настроить автоматическое перенаправление покупателя на сайт предприятия после завершения платежа в АПК ПР. Для этого необходимо выбрать действие после авторизации «Перейти на страницу магазина» в Личном кабинете по адресу <https://account.rausecure.ru/> в разделе «Настройки мерчантов», вкладка «Настройки платежей». В этом случае после авторизации сразу загружается страница веб-статистики, на которой показывается предварительный результат этого платежа.
- Настройка получения результатов операций: в личном кабинете assist поставить галку "отправлять результаты платежей", ввести URL, примерно такой: <http://billing/bgbilling/assistexecuter?mid=<mid ассиста>>, указать тип: SOAP. URL зависит от ваших настроек, проверьте доступность извне сервлета assistexecuter.
- Не ставить (снять) галочку "отправлять только успешные", иначе неудачные платежи будут вечно висеть в статусе "в обработке".

Если у вас работала до этого старая версия протокола, то нужно **ОБЯЗАТЕЛЬНО ОТКЛЮЧИТЬ** задачу планировщика "Получение результатов операций".

Перед сменой состояния модуля с "рабочего" на "тестовый" необходимо убедиться, что в данный момент нет транзакций. Также необходимо решить вопрос с транзакциями со статусом "В обработке". После смены режима все ранее необработанные транзакции при обработке получат статус текущего режима! То есть, если транзакция начата как "тестовая", а запрос о подтверждении пришёл после смены режима на "рабочий", то она будет считаться проведённой в рабочем режиме, и наоборот. Платежи в договор в любом режиме добавляются одинаково.

Рекуррентный платёж

Поддерживается возможность периодических платежей. Пользователь при оплате может указать, что платёж периодический

```
# ( , )
#
assist.RecurringMinAmount=1.00
#
assist.RecurringMaxAmount=666.66
# (31 - magic number, - )
assist.RecurringPeriod=31
# . ( ru.bitel.bgbilling.modules.assist.server.RecurringRequest). , .
assist.recurring.request.class=ru.bitel.bgbilling.modules.assist.AssistRecurringRequest
# . ASSIST . ( )
assist.OrderCommentRecurring= ${contract} (${contract_comment})
```

Есть возможность ручной инициации платежа (см. в договоре на модуле assist внизу). Для работы должна быть настроена задача ежедневная.

Этот таск не смотрит на деньги, нехватку, подписку, абонку, блокировку. Всё это должно делаться в упомянутом дин.коде, который нужно написать под свои требования. "Делаться" только одно по сути: вычисление суммы, которую сейчас надо заплатить. По умолчанию поведение следующее: каждые N (из конфига) дней платится сумма S (заявленная юзером при подключении платежа) до достижения максимальной даты D (задаётся юзером при подключении платежа согласно сроку своей карты). С технической точки зрения: при этом платёж происходит точно также, как если бы человек сам платил из личного кабинета: сначала создаётся заказ, который сначала "в обработке", потом отправляется запрос в ассист, если он одобряется, то через некоторое время в тот же самый экзекютер приходит оповещение, заказ отмечается оплаченным, на договор добавляется платёж на эту сумму.

Оплата через систему Assist

Если в договоре клиента подключён экземпляр модуля Assist, то у него появляется возможность произвести оплату услуг через web-интерфейс (подпункт меню **Оплата через Assist.Ru**). Здесь же клиент может просмотреть все проведённые им платежи через Assist и их статусы:

Месяц: Год:

Номер	Дата/время оплаты	Карта	Владелец	Сумма	Статус операции
53	2009.02.04 00:00:00			101.00	Неуспешно
54	2009.02.11 10:49:36	411111****1111	Ivan Titov	200.00	Успешно
55	2009.02.13 00:00:00			300.00	В обработке
56	2009.02.13 13:43:48	411111****1111	Ivan Titov	400.00	Успешно
57	2009.02.13 14:09:03	411111			Успешно
58	2009.02.13 14:12:51	411111			Успешно
59	2009.02.18 17:28:04	411111			Успешно
60	2009.02.18 17:35:33	411111			Успешно
61	2009.02.19 14:04:43	411111			Успешно
62	2009.02.19 14:08:04	411111			Успешно
63	2009.02.19 14:12:39	411111****1111	Ivan Titov	102.00	Успешно
64	2009.02.19 16:14:46			103.00	В обработке
66	2009.02.19 14:21:26	411111****1111	Ivan Titov	106.00	Неуспешно

При нажатии кнопки **Пополнить счёт** клиент попадает на новую страницу с формой для ввода суммы. Там же отображается текущий баланс.

Новый платеж	
Текущий баланс	463.00
Введите сумму для оплаты	<input type="text" value="100"/>
Нажмите "Далее" для перехода к системе Assist.Ru и продолжения проведения оплаты.	
<input type="button" value="Далее"/> <input type="button" value="Отмена"/>	

При нажатии кнопки **Далее** клиент переходит к [системе авторизации Assist.Ru](#), где может выбрать удобный для себя способ оплаты из разрешённых настройками модуля. После проведения оплаты (или отказа от неё) клиент возвращается обратно к списку платежей Assist, где устанавливается предварительный статус платежа. Если статус "в обработке", то результат ещё не дошёл до биллинга.

Мониторинг и администрирование платежей

В параметрах договора можно посмотреть все платежи: проведённые, не проведённые и находящиеся в обработке.

Свойства

Дата транзакции с 01.02.2009 по 31.03.2009 Статус Проведен Не проведен В обработке

ID	Карта	Держатель	Плательщик	Сумма	Дата	#billnumber	Комментарий	Статус
53				101.00	04.02.2009 00:00:00		руками поставил статус	Не проведён
54	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	200.00	11.02.2009 10:49:36	2147483647	Оплата по договору dimon...	Проведён
55				300.00	13.02.2009 00:00:00			В обработке
56	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	400.00	13.02.2009 13:43:48	2147483647	Оплата по договору dimon...	Проведён
57	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	500.00	13.02.2009 14:09:03	2147483647	Оплата по договору dimon...	Проведён
58	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	500.00	13.02.2009 14:12:51	331323520360713	Оплата по договору dimon...	Проведён
59	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	159.00	18.02.2009 17:28:04	331323520395980	Оплата по договору dimon...	Проведён
60	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	4543.00	18.02.2009 17:35:33	331323520396042	Оплата по договору dimon...	Проведён
61	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	100.00	19.02.2009 14:04:43	331323520401585	Оплата по договору dimon...	Проведён
62	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	101.00	19.02.2009 14:08:04	331323520401606	Оплата по договору dimon...	Проведён
63	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	102.00	19.02.2009 14:12:39	331323520401643	Оплата по договору dimon...	Проведён
64				103.00	19.02.2009 16:14:46			В обработке
65				105.00	19.02.2009 16:16:20			В обработке
66	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	106.00	19.02.2009 14:21:26	331323520401710	Оплата по договору dimon...	Не проведён

В параметрах модуля есть глобальный монитор. Можно сделать выборку по многим параметрам, а также принудительно запросить статусы платежей с сервера авторизации Assist. Это можно сделать либо для конкретного платежа (нажать правой кнопкой мыши на строке таблицы и выбрать пункт "Запросить"), либо для всех платежей за период. Период не должен превышать некоторое ограничение Assist. Ручные запросы могут пригодиться при тестировании, каких-то сбоях в автоматической отправке платежей и т.д. и т.п.

Ручная проверка и установка статусов платежей

Период транзакций (промежуток не более 3-х дней) с 10.01.2010 по 12.01.2010 Запрос

Фильтр

ID Номер договора Карта Держатель/Плательщик #billnumber Статус Дата транзакции

ID	Договор	Карта	Держатель	Плательщик	Сумма	Дата	#billnumber	Комментарий	Статус
53	dimon				101.00	04.02.2009 00:00:00		руками поставил статус	Не проведён
54	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	200.00	11.02.2009 10:49:36	2147483647	Оплата по договору dimon (...)	Проведён
55	dimon				300.00	13.02.2009 00:00:00			В обработке
56	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	400.00	13.02.2009 13:43:48	2147483647	Оплата по договору dimon (...)	Проведён
57	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	500.00	13.02.2009 14:09:03	2147483647	Оплата по договору dimon (...)	Проведён
58	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	500.00	13.02.2009 14:12:51	331323520360...	Оплата по договору dimon (...)	Проведён
59	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	159.00	18.02.2009 17:28:04	331323520395...	Оплата по договору dimon (...)	Проведён
60	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	4543.00	18.02.2009 17:35:33	331323520396...	Оплата по договору dimon (...)	Проведён
61	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	100.00	19.02.2009 14:04:43	331323520401...	Оплата по договору dimon (...)	Проведён
62	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	101.00	19.02.2009 14:08:04	331323520401...	Оплата по договору dimon (...)	Проведён
63	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	102.00	19.02.2009 14:12:39	331323520401...	Оплата по договору dimon (...)	Проведён
64	dimon				103.00	19.02.2009 16:14:46			В обработке
65	dimon				105.00	19.02.2009 16:16:20			В обработке
66	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	106.00	19.02.2009 14:21:26	331323520401...	Оплата по договору dimon (...)	Не проведён
67	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	100.00	17.03.2009 10:24:37	331323520588...	Оплата по договору dimon (...)	Не проведён
68	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	200.00	17.03.2009 10:40:10	331323520588...	Оплата по договору dimon (...)	Не проведён
69	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	300.00	17.03.2009 10:44:20	331323520588...	Оплата по договору dimon (...)	Проведён
70	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	200.00	17.03.2009 10:55:46	331323520588...	Оплата по договору dimon (...)	Проведён
71	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	200.00	17.03.2009 11:04:12	331323520588...	Оплата по договору dimon (...)	Проведён
72	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	100.00	17.03.2009 12:53:18	331323520588...	Оплата по договору dimon (...)	Проведён
73	dimon	411111****1111(...)	Ivan Titov	Titov Ivan	1.00	17.03.2009 13:39:54	331323520590...	Оплата по договору dimon (...)	Проведён
74	dimon				100.00	18.03.2009 20:31:23			В обработке
75	dimon				200.00	18.03.2009 21:07:06			В обработке
76	dimon				200.00	18.03.2009 21:08:17			В обработке
77	dimon				200.00	18.03.2009 21:12:06			В обработке
78	dimon				300.00	19.03.2009 13:09:50			В обработке

Имеется возможность отредактировать некоторые параметры платежа: сумму, статус, комментарий.

Договор

Период

dimon (не распечатать)

352715

с 04.02.2009 по

- Параметры договора
- Объекты договора 0
- Иерархия договоров
- Статус активен
- Лимит 0.00
- Режим кредит
- Баланс (март 2009)
 - Входящий остаток 462.00
 - Приход 1.00
 - Наработка 0.00
 - Расход 0.00
 - Исходящий остаток 463.00
- Тарифные планы
- Модули
 - Assist.Ru
 - Бухгалтерия
- Группы
 - dimon
- Web
 - Пароль статистик
 - Управление лимит
 - Скрипт поведения
 - Доп. действия
- Примечания 0
- Скрытый

Свойства



Дата транзакции с 01.02.2009 по 31.03.2009 Статусы Проведён Не проведён В обработке

ID	Карта	Держатель	Платежщик	Сумма	Дата	#billnumber	Комментарий	Статус
53				101.00	04.02.2009 00:00:00		руками поставил статус	Не проведён
54	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	200.00	11.02.2009 10:49:36	2147483647	Оплата по договору dim...	Проведён
55				300.00	13.02.2009 00:00:00			В обработке
56	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	400.00	13.02.2009 13:43:48	2147483647	Оплата по договору dim...	Проведён
57	411111****1111(VISA/AlfaVisaC)	Ivan Titov	Titov Ivan	500.00	13.02.2009 14:09:03	2147483647	Оплата по договору dim...	Проведён

Операция Assist

ID номер заказа: 53 Assist-платеж:

Договор: dimon email:

Дата: 04.02.2009 00:00:00 IP:

Сумма: 101 Статус: Не проведён

Комментарий:

Тип средства: Подтип средства:

Номер средства: Адрес:

Держатель средства:

Платежщик:

OK Отмена

Настройка автоматической установки статусов платежей

В новом протоколе не используется.

Настройка сети, фаервола и т.д.

Между сервером Assist и модулем биллинга, как и во многих подобных системах, происходит некоторый обмен. С сервера Assist производится обращение к сервлету assistexecuter. Снаружи нужно открыть доступ. Изнутри надо лишь убедиться, что имеется выход на 443 и/или 80 порт для запроса ручного статусов платежей.

Обратите внимание на ограничение Assist. В соответствии с стандартом PCI DSS сервера Assist из продакшн среды не могут устанавливать соединения во внешнюю среду по нестандартным портам. Допустимо использование 80 и 443 (как с SSL, так и без) портов. Это означает, что ваш assistexecuter должен быть доступен извне по одному из этих портов. В противном случае оповещения об операциях не будут доходить.

Важные замечания

Обратите внимание на особенности функционирования модуля и особенности редактирования Assist-платежей.

1) Данные Assist-платежей представляют собой некий протокол проведения или не-проведения транзакции на удалённом сервере Assist. И к "платежам в биллинге" отношение имеют только в одном: в момент выставления статуса "проведён" создаётся платёж с суммой по данным транзакции с сервера авторизации Assist. Из этого следуют пункты 2 и 3:

2) Вы не можете изменить параметры Assist-платежа в протоколе, уже установленного в статусы "проведён" и "не проведён". Но в реальности возможна ситуация, когда с сервера Assist изначально приходит ненастоящий статус, а в дальнейшем выдаётся другой и, в итоге, получается ошибка (происходит крайне редко, причины не были выяснены). Или статус может по какой-либо причине зависнуть в "в обработке". Для этого существует возможность менять статус и/или сумму у платежей в реестре Assist-платежей в биллинге. Но в целом у таких платежей рекомендуется изменять комментарий. Предполагается, что этот протокол отражает реальное положение дел с транзакциями. Возможность сменить статус оставлена как запасной ход для менеджера подправить "подвисший платёж" и должна использоваться только в крайних случаях! При этом необходимо проверить выставляемую сумму.

3) Как уже сказано в пункте 1, при переводе Assist-платежа в статус "проведён" создаётся новый платёж в договоре. Это взаимодействие однонаправленное. То есть при удалении, редактировании платежа в договоре сумма и статус Assist-платежа в этом протоколе не изменится. Модуль Assist после создания платежа не отслеживает его дальнейшую судьбу. А как сказано в пункте 2, редактировать протокол вы тоже не можете, так что вопрос об обратном взаимодействии протокол->платежи договора тоже не рассматривается. Также при изменении статуса с "проведён" на "в обработке" или "не проведён" с уже добавленным платежом ничего не происходит. Нужно вручную удалить лишний платёж или выставить расход. При обратной смене статуса с любого на "проведён" платёж снова создаст с указанной суммой. Как видно, действия неоднозначные. Возможно, следует закрыть действие ActionUpdatePay (Редактирование параметров платежа) в правах, чтобы было невозможно изменять значения статуса и суммы.

4) При отмене платежа нужно также вручную разрешать несоответствия. Возможности получить автоматически информацию об этих действиях в протоколе в данный момент нет.

Модуль Buyemoney (beta-версия)

Назначение модуля Buyemoney (beta-версия)

Модуль предназначен для покупки электронной валюты из личного кабинета пользователя.

Структура и настройка модуля

Модуль содержит несколько различных "драйверов", связанных одинаковым API и реализующих несколько протоколов. Все драйверы защиты в модуль, в конфигурации можно привязать каждый из протоколов, подразумеваемых к использованию к нужной электронной валюте.

В общем виде конфигурация выглядит так:

Конфигурация

```
# :
currency.1.title=yandex RUR
currency.1.code=yandex RUR
currency.1.protocol=ru.bitel.bgbilling.modules.buyemoney.server.protocols.yd.YDProtocol
# id ,
currency.1.pursePid=35
# ,
currency.1.chargeType=55
# , ,
currency.1.chargeCommentTemplate= ${sumcur} ${cur} ${sumtotal} ( ${rateprice}, ${date}, ${id}) ${purse}
# , ( " " )
currency.1.transactCommentTemplate=${contract} (${contract_comment}), ${id}, : ${user_comment}
#
# 1 - ;
# 2 - ;
# 3 - , - ( , )
currency.1.purseMode=1
# :
# url
currency.1.url=https://bo-demo02.yamoney.ru/onlinegates/maglan.aspx
# (643 - , 10643 - (-))
currency.1.curcode=10643
# ( " " , )
currency.1.GpgPassPhrase=123456
#   gpg
currency.1.GpgFullPath=/usr/bin/gpg
# ,
currency.1.registryMail=onlinegate@yamoney.ru
# ( , )
currency.1.Agent_ID=123456
#
currency.1.Agent_name= «»
#
currency.1.ontract_number=2000.998.001
# ( )
currency.1.registryGpgKeyID=BCD3ACEF

currency.2.title=webmoney WMR
currency.2.code=WMR
currency.2.protocol=ru.bitel.bgbilling.modules.buyemoney.server.protocols.wm.WMProtocol
currency.2.pursePid=53
currency.2.chargeType=55
currency.2.chargeCommentTemplate= ${sumcur} ${cur} ${sumtotal} ( ${rateprice}, ${date}, ${id}) ${purse}
currency.2.transactCommentTemplate=${contract} (${contract_comment}), ${id}, : ${user_comment}
currency.2.purseMode=1
# WebMoneyCA.crt,
currency.2.WebMoneyCA=/home/dimon/projects/webmoney/WebMoneyCA.crt
# WMID,
currency.2.wmid=323446780013
#
currency.2.kwm=/home/dimon/projects/webmoney/323446780013.kwm
#
currency.2.kwmpass=123456
#   wmid
currency.2.purse=R198105158104

#currency.3.title=webmoney WMZ
#currency.3.code=WMZ
#currency.3.protocol=ru.bitel.bgbilling.modules.buyemoney.server.protocols.wm.WMProtocol
```

Примечание: \${user_comment} не будет виден в шаблоне комментария расхода (за исключением ситуаций, когда результат был известен сразу после транзакции, а не при выполнении периодической задачи лишь), а только в шаблоне транзакции.

Настройка gpg-подписи для Yandex.Деньги

Нам необходим ключ для подписывания наших запросов. Для работы с PGP используется нативная утилита GnuPG (gpg), установленная в системе, и её хранилище ключей. Вкратце рассмотрим настройку и работу утилиты. Инструкция приведена для версии gpg (GnuPG) 1.4.11 под linux.

Предположим, что работа с gpg только что началась и до этого никаких операций не проводилось. Иначе при необходимости надо скорректировать свои действия, возможно, пропустить некоторые этапы.

Прежде всего нужно сгенерировать свою пару открытый/закрытый ключ:

```
$gpg --gen-key
```

При этом выбирается нужный тип ключа (например, для работы Yandex.Money выбирается "(4) RSA (только для подписи)"); выбирается длина; выбирается срок действия.

Далее формируется UserID (имя, e-mail, комментарий) в таком виде: "dimon (first key) <dimon@***.ru>". Три параметра вводятся по очереди или сразу, зависит от реализации.

Наконец, вводится пароль для доступа к ключу. Предположим, мы ввели "123456". По этому паролю в некоторых реализациях и в некоторых случаях будет идентифицироваться ключ. В вышеприведённой конфигурации этот пароль указывается в параметре GpgPassPhrase для соответствующей валюты.

После некоторых дополнительных указаний сгенерируются (открытый и закрытый) ключи.

Можно посмотреть на ключ:

```
$ gpg --list-key
/home/dimon/.gnupg/pubring.gpg
-----
pub   2048R/F99D7F2B 2010-11-08
uid           dimon (first key) <dimon@***.ru>
```

Аналогично смотрятся секретные ключи:

```
$ gpg --list-secret-keys
/home/dimon/.gnupg/secring.gpg
-----
sec   2048R/F99D7F2B 2010-11-08
uid           dimon (first key) <dimon@***.ru>
```

Чтобы отослать свой ключ получателю (в нашем случае менеджерам Yandex.Деньги) его надо экспортировать в файл. Выгрузим в текстовом (не бинарном) виде:

```
$gpg --export --armor > mykey.asc
```

Получится файл mykey.asc с вашим ключом:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

mQENBEzXmH8BCADJE6LkH9i4+Q96OzuvrSnAK9u9xxmp3x05FtGLswUNn6n0Ux7X
eBLGTQn+aTlKGlgc6O8c0187A1hWk/VIE9ipCIbccY8xa1SWBcjHHTZIXYPG5ToM
q4jWuW7WJzgpzV6TuYrffzz+iX/SQOfn0gav61ZonCcLQ+Ypzz075r1Ax/S/Kfin
fiUyjjqDZi1ajfE1jWgXn/nxnWQ7DD5P2FJ4/HBR7HF7EABqo+TLk74hbYISoPww
pAnYpzVn0t1XlhWi6CgIHx8YC9xhBA5LsmZk1SBKZhviSMCFuGCQnFpXFyX8fmfU
TCfJTFMDnBFAEPRJKeG5G2/Ahe2IrlwoSXB7ABEBAAG0ImRpbW9uIChmaXJzdCBBr
ZXkpdDxkaW1vbkbIaXRlbC5ydT6JATgEEwECACIFAKzXmH8CGwMGCwkIBwMCBhUI
AgkKCQwAgMBAh4BAheAAoJEE3+EBD5nX8rWdcH/2w/VpmlgYOHApHC9zXsTv09
K4jNaLV6BwG0XqW1Af3Y49tDYpv9zaobtV/yRd0gaZo2rONTX1b4GGpBsbPBWwkg
eYb/z+7AZm0auCEYlc4Fk9J7QN0NvbGccw+I6dRIrykK+VwZUS4+nQB+EIDoR33U
zaT47IPqilHkYiRpxB3B/4axMiTjfx28gdzX3EICQtMNEDczcPCYsYCB9A2y2XBd
N40UWanrcceEvUeEhPa4rR/sYpvg3JZv44Asm7du3GprYn1FcoeSdmeJ/Bl/au+eU
```

```
8gvzDCEVBmUNRYG1cTXgfoJqcFVLULoASwfAYVfJQVRMBP60ecz4wPMxhHvV2Gc=
=Cgdy
-----END PGP PUBLIC KEY BLOCK-----
```

Далше нужно импортировать выдаваемые второй стороной ключи. Нужный в данный момент имеет имя "Yandex.Money Payment Gate.asc".

```
$ gpg --import ./Yandex.Money\ Payment\ Gate.asc
gpg: B0B2F0D8: "Yandex.Money Demo Payment Center (Yandex.Money Demo Payment Center)
<demopaymentcenter@yamoney.ru>"
gpg: : 1
gpg: : 1
```

Второй ключ - "Yandex.Money Online Registry Key.asc" - тоже пригодится в последующем.

```
$ gpg --import ./Yandex.Money\ Online\ Registry\ Key.asc
gpg: ACE74CE2: "Yandex.Money Online Registry Key <onreg@yamoney.ru>"
gpg: : 1
gpg: : 1
```

Можно убедиться, что теперь открытых ключей стало больше (наш закрытый, конечно, остался один):

```
$ gpg --list-key
/home/dimon/.gnupg/pubring.gpg
-----
pub 2048R/F99D7F2B 2010-11-08
uid dimon (first key) <dimon@***.ru>

pub 1024D/B0B2F0D8 2008-05-29
uid Yandex.Money Demo Payment Center (Yandex.Money Demo Payment Center)
<demopaymentcenter@yamoney.ru>
sub 2048g/2A9A05C6 2008-05-29

pub 1024D/ACE74CE2 2006-10-05
uid Yandex.Money Online Registry Key <onreg@yamoney.ru>
sub 2048g/BCD3ACEF 2006-10-05
```

Пример того, как это все работает (вывод на экран в бинарном виде).

Подпись файла test.txt:

```
$ gpg --armor --output - --sign --passphrase 123456 test.txt
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.10 (GNU/Linux)

owEBPgHB/pANAwACAXzF27f83iK8AawOYgR0ZXN0S6h2SXRlc3SJARwEAAECAAYF
AkuodkkACgkQfMXbt/zeIrwE6Qf/bNGgTUytFAuPaInkx49L8DQyUx72Lra9uOin
TCfaRPZJlwjt4wdsF0FYyLuFYoQ25WwfVmdZwTfvd1qV+drWBJKZx/s8t0WDrnw
0rQL9JainBkojNdmGfa3UMEAYLQgXezpGwZtQ4iURjVbChH521R3NFDGjt+fR8p
dVsdjz7owhTldC08ceVSKLDagW+XFYC9OaeWjDvLKN2dsYbeWLlg0VjnzSHHPjFG
XL8J90FpTymL/DZzsnOWUxUmEU7ymZ14NAwtOV9jyyGjYL8bGSDymJEqtXKJWbO
Z1HUbzJG23m2YK1BoW7ojNbAD8ilrIMWyjCa6id+OB291JXzza==
=IhW8
-----END PGP MESSAGE-----
```

Подпись и шифрование для отправки сверки на Яндекс:

```
$ gpg --armor --output - --encrypt -r BCD3ACEF --sign --passphrase 123456 ./test.txt
```

В данном случае файл будет зашифрован указанным ключём Яндекса и подписан нашим ключом по умолчанию (он устанавливается командой `gpg --default-key`), либо первым ключом в секретном хранилище.

Могут возникнуть проблемы:

Недоверие ключу:

```
gpg: BCD3ACEF: , , User ID
```

и запрос всё время доверять или нет. Надо установить уровень доверия ключу:

```
$ gpg --edit-key BCD3ACEF
```

И далее вводится "trust" и выбрать "доверять абсолютно", потом "quit".

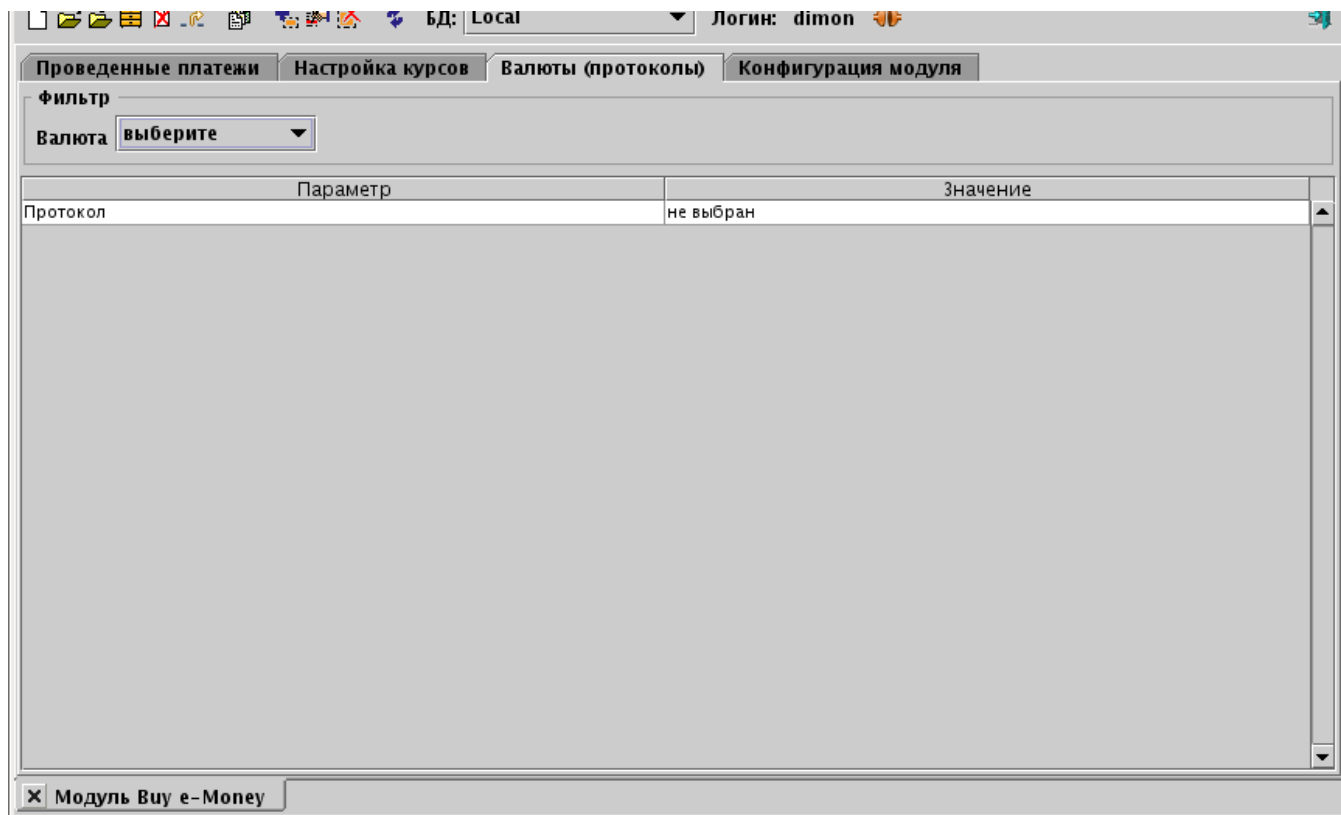
Также, как вариант, можно использовать --trust-model always или --always-trust, но не нужно.

Настройка WebMoney

В данный момент находится в alpha-тестировании.

Использование модуля Buyemoney

Настройка модуля не вызывает особых трудностей. Прежде всего есть список всех протоколов валют, настроенных в конфигурации модуля. Для каждой валюты есть набор некоторой информации, читаемой из конфигурации и с удалённого сервера, соответствующего этой валюте (в случае неправильно настроенного модуля это, конечно, не прочитается).



Далее для каждой физической валюты надо настроить её курс. Курс включает в себя цену единицы валюты (в рублях системы) для какого-то определённого промежутка времени. Таким образом, можно настроить определённый курс валюты и её активность на разные числа. Цена единицы определяется как сумма, списываемая со счёта клиента при покупке единицы соответствующей валюты, за счёт этого можно настроить комиссию, как положительную, так и отрицательную.

Проведенные платежи Настройка курсов **Валюты (протоколы)** Конфигурация модуля

Валюта	Период	Активность	Цена единицы валюты
yandex RUR	22.12.2009-	Активна	1.01
webmoney WMR	07.12.2009-27.12.2009		1.02
webmoney WMZ	25.12.2009-26.12.2009	Активна	1.05

Модуль Buy e-Money

Имеется список транзакций, как общий, так и для каждого договора.

Проведенные платежи Настройка курсов **Валюты (протоколы)** Конфигурация модуля

Группы

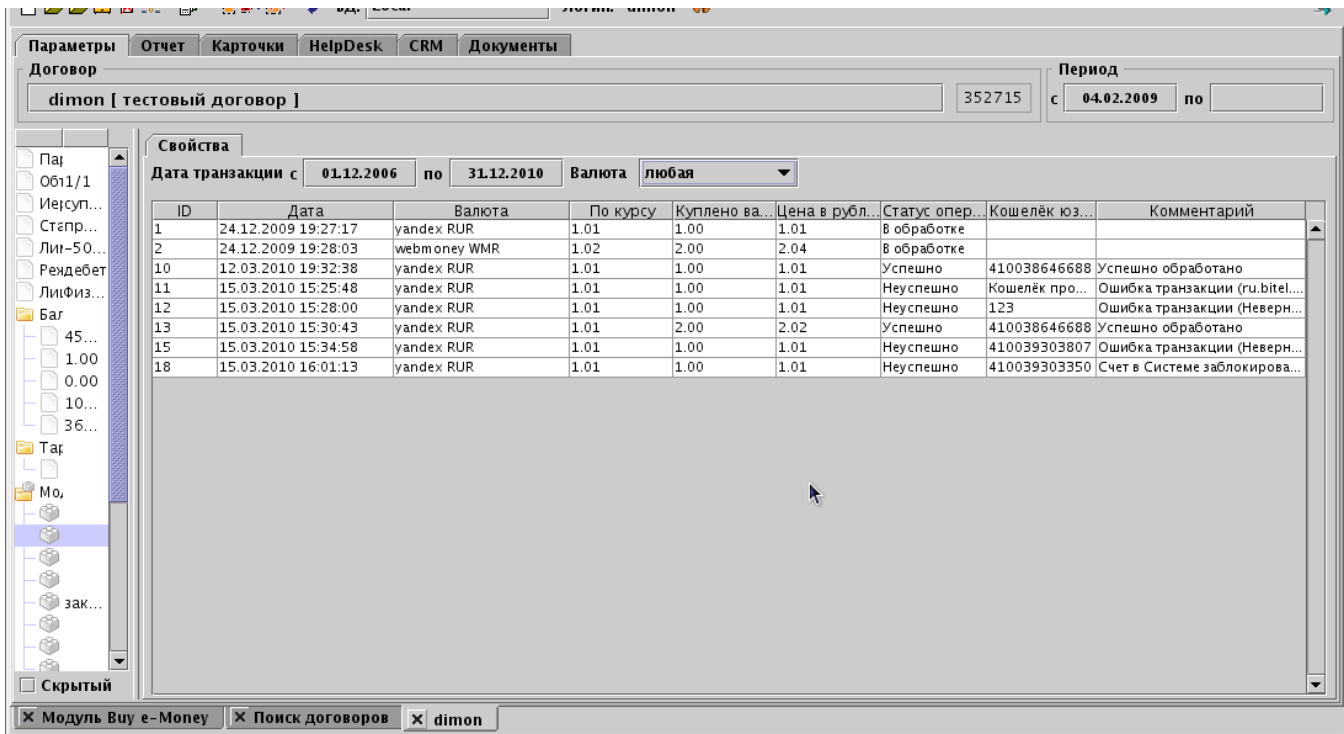
- 001
- 002
- B
- C
- dimon
- E
- IMPORT(числится! - не ст...
- L
- M
- S
- V
- VIP-клиенты
- X
- A1
- A2
- A3
- A4
- A5
- Группа 25
- Группа 37
- Группа 54
- Загруженные
- Закрытые
- импорт
- карта - подарок
- Каточки

Фильтр

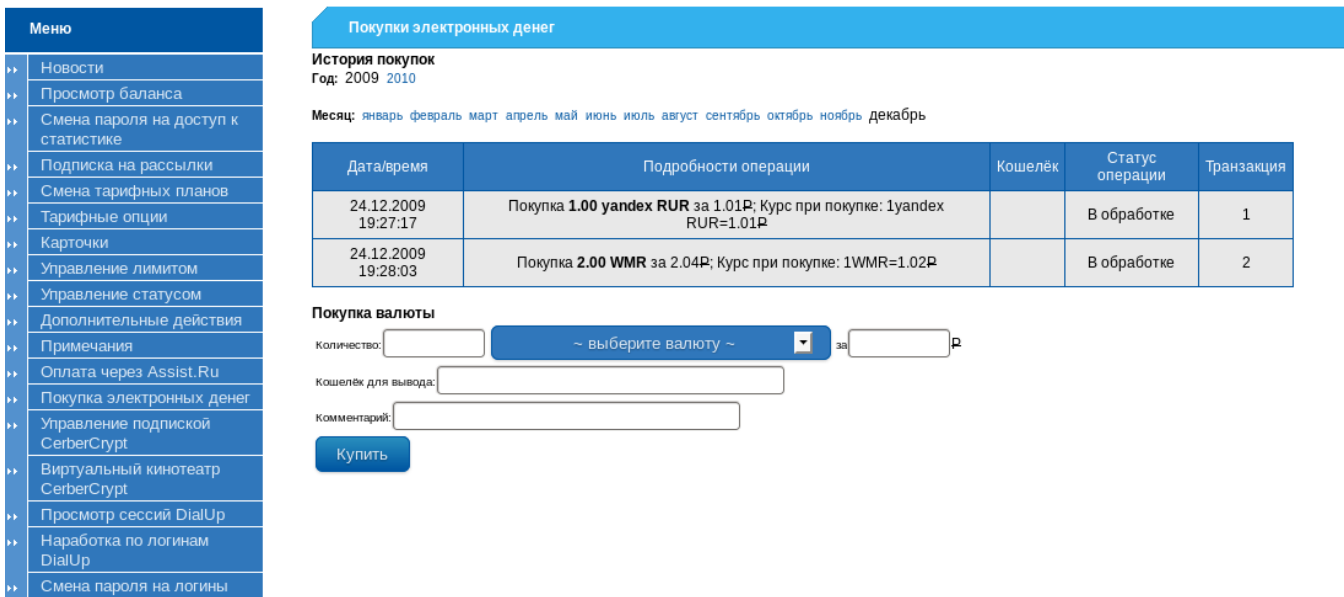
Номер договора: Валюта: **любая** Дата транзакции: **01.12.2008** по **31.12.2010**

ID	Договор	Комментарий до...	Дата	Валюта	По курсу	Куплено ва...	Цена в руб...	Статус опе...	Кошелёк юз...	Комментар
1	dimon	тестовый договор	24.12.2009 19:27:17	yandex RUR	1.01	1.00	1.01	В обработке		
2	dimon	тестовый договор	24.12.2009 19:28:03	webmoney ...	1.02	2.00	2.04	В обработке		
10	dimon	тестовый договор	12.03.2010 19:32:38	yandex RUR	1.01	1.00	1.01	Успешно	4100386466...	Успешно обработа
11	dimon	тестовый договор	15.03.2010 15:25:48	yandex RUR	1.01	1.00	1.01	Неуспешно	Кошелёк про...	Ошибка транзакци
12	dimon	тестовый договор	15.03.2010 15:28:00	yandex RUR	1.01	1.00	1.01	Неуспешно	123	Ошибка транзакци
13	dimon	тестовый договор	15.03.2010 15:30:43	yandex RUR	1.01	2.00	2.02	Успешно	4100386466...	Успешно обработа
15	dimon	тестовый договор	15.03.2010 15:34:58	yandex RUR	1.01	1.00	1.01	Неуспешно	4100393038...	Ошибка транзакци
18	dimon	тестовый договор	15.03.2010 16:01:13	yandex RUR	1.01	1.00	1.01	Неуспешно	4100393033...	Счет в Системе за...

Модуль Buy e-Money



Для пользователя в web-интерфейсе имеется список его транзакций. Для каждой операции имеется описание подробностей операции - сколько куплено какой валюты и за какую цену. Каждая операция находится в статусе "в обработке", "успешно" или "неуспешно".



Ниже находится форма, которая позволяет произвести покупку валюты. Можно вводить как необходимую сумму валюты, так и конечную сумму затраченных денег. Для каждой валюты будет использован текущий на сегодня курс. Если курса нет, то валюты в списке не будет. В зависимости от параметра конфигурации purseMode кошелёк для вывода денег будет задан жёстко, или дана возможность его ввода.

Модуль BVCom

Назначение модуля

Модуль BVCom предназначен для проведения платежей через платежный шлюз BVCom (Arius, www.ariuspay.ru) с использованием пластиковых карт. Для проведения платежей вашими клиентами у вас должен быть заключен договор с системой.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, скопируйте туда приведенный ниже текст и введите требуемые параметры. После этого сохраните конфигурацию и сделайте её активной.

Конфигурация

```
#####
# : 1 - , 2 -
bvcom.protocol.version=2
#-, BVCom/Arius. : bitel.
bvcom.client=bitel
# . : BG00000000
bvcom.trans.format=BG00000000
#
bvcom.payment.type=32
#
bvcom.charge.type=57
#
bvcom.comment= BVCom
#url-,
bvcom.url.result=http://localhost:6080/bgbilling/webexecuter
#####
web.menuItem1= BVCom
# . : secretkey
bvcom.key=]FR{EPz(
#####
# BVCom
bvcom.endpointid=16
#, BVCom
bvcom.merchant.control=4B613C8C-F5C2-4B33-B9D3-8C4D29F8EBC2
# ,
bvcom.callback.url=http://<billing_server>/bgbilling/bvcomexecuter/<mid>
# . RUR , RUB
bvcom.currency=RUR
#
bvcom.server.url=https://sandbox.ariuspay.ru/paynet/api/v2/
```

Замечания:

1. Прежде, чем задавать **bvcom.payment.type**, **bvcom.charge.type** необходимо создать соответствующие типы платежей и расходов в Справочниках (**Справочники->Другие->Типы платежей**, **Справочники->Другие->Типы расходов**).
2. Номер транзакции создается следующим образом: берется ID транзакции из таблицы **bvcom_transaction_<mid>** и соединяется с шаблоном. Например: если шаблон **"BG0000"**, а ID пусть будет 34, тогда номер транзакции, отсылаемый на платежный шлюз BVCom, будет иметь вид: **BG0034**.
3. После заключения договора с системой BVCom им нужно передать адрес скрипта на машине биллинга, который ждет результаты от платежной системы. URL скрипта выглядит следующим образом: **http://<адрес_машины_биллинга>/bvcomexecuter/<mid>**. Например, если у вас биллинг находится по адресу <http://billing.example.com/bgbilling/> и модуль BVCom имеет mid=16, то результирующий URL, который нужно дать компании BVCom, выглядит следующим образом: <http://billing.example.com/bgbilling/bvcomexecuter/16>.

Оплата через систему BVCom

Если у клиента подключен экземпляр модуля в дереве договора, то он может осуществлять оплату через платежный шлюз **BVCom**, используя личный web-интерфейс - пункт меню **Оплата через BVCom** (название пункта меню возможно изменить в настройках).

В личном кабинете на странице отражается история платежей, совершенных клиентом:

Список платежей BVCom

Месяц: Год:

Номер	Дата/время оплаты	Сумма
73	12.01.2010 15:58:58	100
74	18.01.2010 23:08:18	500

Ниже таблицы с платежами расположена кнопка **Пополнить счет**, нажав на которую клиент попадает на страничку, где отображается его текущий баланс, а также форма с запросом суммы платежа.

Пополнение счета через BVCom

Новый платеж	
Текущий баланс	-1584.00
Введите сумму для оплаты	<input type="text" value="2000"/>
<input type="button" value="Далее"/> <input type="button" value="Отмена"/>	

Щелкнув по кнопке **Далее**, клиент попадает на страницу подтверждения оплаты. На данном этапе уже сформирован запрос на платежный шлюз и от клиента требуется нажать на кнопку **Далее** для продолжения оплаты, либо **Отмена** для отмены платежа.

Список платежей BVCom

Новый платеж	
Нажмите "Далее" для перехода к системе BVCom и продолжения проведения оплаты	
<input type="button" value="Далее"/> <input type="button" value="Отмена"/>	

В случае, если клиент подтверждает оплату, то он перенаправляется на платежный шлюз, где может ввести информацию о своей пластиковой карте.

Платежный шлюз ООО Бвком

Введите информацию о карте

Номер карты:	5543863311777001
CVV2(CVC2) ₁ :	000
Имя владельца карты ₂ :	Dmitry P.
Срок действия:	05 2010
<input type="button" value="OK"/> <input type="button" value="Отмена"/>	

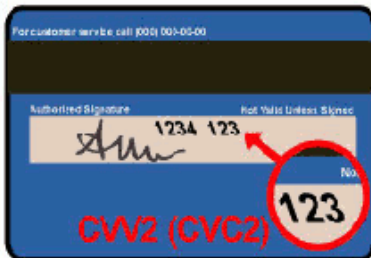


На этой странице Вы должны ввести информацию о Вашей банковской карте

Для оплаты заказа Вам необходимо последовательно ввести информацию о Вашей банковской карте и внимательно проверить введенную информацию. Эта информация не доступна посторонним лицам. Все данные передаются зашифрованными с применения протокола безопасности SSL.

Примечание

1. Код CVV2 (CVC2) - это трехзначное число, узнать которое Вы можете, посмотрев на обратную сторону банковской карты. Обычно CVV2 (CVC2) напечатано на полосе для подписи.
2. Имя владельца карты должно быть введено в латинской транскрипции именно так, как оно указано на Вашей карте. В том случае, если имя не указано на карте, то поле можно оставить пустым.



Информация о заказе

Название магазина:	БВКОМ
Валюта:	Рубли РФ
Сумма:	2 700,00
Номер платежной операции:	8004

© 2010 ООО Бвком

Правильно заполнив предложенную форму и нажав кнопку **OK**, платежный шлюз отобразит информацию о результате платежа.

Платежный шлюз ООО Бвком

Номер платежной операции	8086
Номер банковской карты	4150XXXX0032
Статус оплаты	Одобен
Сумма	4 000,00
Валюта	Рубли РФ
Дата и время совершения авторизации	24/02/2010 06:59:40
Код авторизации	557002
Результат авторизации	Одобрено

В случае положительного результата платежа биллинговая система начислит клиенту указанную им сумму на счет.

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль BVCom в дереве параметров договора. Здесь присутствует фильтр по статусу платежей (проведенные/непроведенные/возвращенные/все) с указанием периода, когда производилась оплата.

The screenshot shows the main interface of the billing system. At the top, there are navigation tabs: "Договор", "Модули", "Плагины", "Справочники", "Сервис", "Утилиты", "Справка". The current user is logged in as "admin" on "02 апреля 11:41". The main area is titled "Параметры" and contains a tree view on the left and a data table on the right.

Параметры
Договор: artur_test [Гареев Артур Маратович]
Период: с 24.11.2009 по

Фильтр по платежам
Период с 01.04.2009 по 30.04.2015 Статус: проведенные

№	Дата/Время оплаты	Сумма	Статус
163		120.00	Платеж проведен
165	27.08.2012 15:33:05	100.00	Платеж проведен
168	03.03.2014 15:12:53	10.00	Платеж проведен

Всего платежей: 3 на сумму: 230.00

Для просмотра ВСЕХ платежей, проведенных через систему BVCom, существует глобальный монитор в параметрах модуля. В открывшейся вкладке модуля **BVCom** у Вас есть возможность просмотреть все платежи, совершенные вашими абонентами за указанный временной период. Также можно установить фильтр платежей по группам договоров, по имени договора, по номеру платежа, по статусу.

The screenshot shows the "Конфигурация модуля" window for the BVCom module. It features a "Фильтр" section with search criteria and a table of payments.

Фильтр
Номер договора: artur_test
Номер платежа: []
Статус: проведенные
Период: с 01.08.2006 по 31.08.2010

ID	Договор	Время	Сумма	Статус
1	artur_test	15.12.2009 17:15:57	100.00	Платеж проведен
2	artur_test	05.12.2009 21:12:46	1000.00	Платеж проведен
67	artur_test	16.12.2009 14:44:39	1.00	Платеж проведен
68	artur_test	16.12.2009 17:15:25	100.00	Платеж проведен
69	artur_test	21.12.2009 16:47:54	15.00	Платеж проведен
73	artur_test	12.01.2010 15:58:58	100.00	Платеж проведен
74	artur_test	18.01.2010 23:08:18	500.00	Платеж проведен

Кол-во платежей: 7 на сумму: 1816.00

Возврат платежей

Начиная со второй версии протокола платежная система поддерживает возврат оплаченных средств обратно на счет абонента. Для функционирования механизма возврата необходимо настроить задачу планировщика модуля **BVCom**, которая будет периодически опрашивать платежный шлюз о результатах запроса на возврат.

Задача называется **BVCom => Проверка результатов запроса на возврат денежных средств**. Периодичность зависит от числа возвратов. Первоначально можно выставить раз в час. В настройках задачи отсутствуют параметры запуска.

Возврат платежа осуществляется только через клиент биллинга, щелкнув правой кнопкой на **ПРОВЕДЕННОМ** платеже.

Модуль Chronopay

Назначение модуля

Модуль предназначен для интеграции с платежной системой (ПС) <https://chronopay.com/>

Настройка модуля

Тип	Наименование	Значение
Число	chronopay.sum.min	100
Число	chronopay.sum.max	5000
Строка	chronopay.product.id	XXXXXX-YYYY-YYYY
Строка	chronopay.sharedsec	ONBikCfqAtZ8Rno5Ybwh
Строка	chronopay.country	RUS
Строка	chronopay.gate.url	https://payments.test.chronopay.com/
Строка	chronopay.callback.url	https://host/bgbilling/chronopay/2
Строка	chronopay.success.url	https://host/bgbilling/webexecuter?module=chronopay&mid=2&action=DoTransaction&operation=success
Строка	chronopay.decline.url	https://host/bgbilling/webexecuter?module=chronopay&mid=2&action=DoTransaction&operation=decline
Строка	chronopay.description	Пополнение счета по договору №{CONTRACT_TITLE}
Число	chronopay.payment.type.id	63
Логический	chronopay.autopayment.enable	true
Строка	chronopay.autopayment.features	
Строка	chronopay.autopayment.rest.user.name	user

Модуль Gorod

Назначение модуля

Модуль предназначен для интеграции системы с платёжной системой ГОРОД. Модуль поддерживает выгрузку реестров 3, 7 и 9.

Настройка модуля

Установите модуль на сервер, обновите клиент биллинга. Затем создайте экземпляр модуля.

Создайте в редакторе конфигурации модуля новую конфигурацию, скопируйте туда приведённый ниже текст, подправьте под ваши нужды параметры и сделайте данную конфигурацию активной.

Конфигурация

```
#
register.pattern={${fio}}[;${city}][,${street}][,${house}][${frac}][,${flat}][;${contractTitle}][;${summ}]
[;;;${account}][;${contractId}]

#, , register.pattern.<tag_id>. , !
register.pattern.1=${fio}][;${city}][,${street}][,${house}][,${flat}][; ${dateTo}][;${contractTitle}][;${summ}]
[;;;${account}][;${contractId}][;${dateFrom}]
register.pattern.2=[;${dateFrom}][;${city}][,${street}][; ${dateTo}][;${contractTitle}][;];;${account}]
[;${contractId}][;${fio}][;${summ}]
register.pattern.3=[;${city}][;${fio}][dateTo: ${dateTo}][;${contractTitle}][;${summ}][;dateFrom:${dateFrom}]
[;${account}][;${contractId}]

# ( , , cp866)
register.encoding=Cp1251

# . , .
register.encoding.1=Cp1251
register.encoding.2=cp866

# dateFrom dateTo
register.date.format=dd/MM/yyyy

#, NOTE
param.note=1

# NOTE <_> (0 - )
param.note.content.date=1

# summ ,
# 2 :
#SALDO( - - )
#IS_GREATER_ZERO - 0.0 ,
register.summ.macros=SALDO

# " " " " " "
# . " " ,
# - .
param.address=< >
param.fullname=< " ">

# . ${cid} -
default.address=,,0,${cid}

# , . : ,
# , 0. mask = 1<<gr1 | 1<<gr2...| 1<<grN
group.mask=262144

#, 1 ( )
account.1=175
```

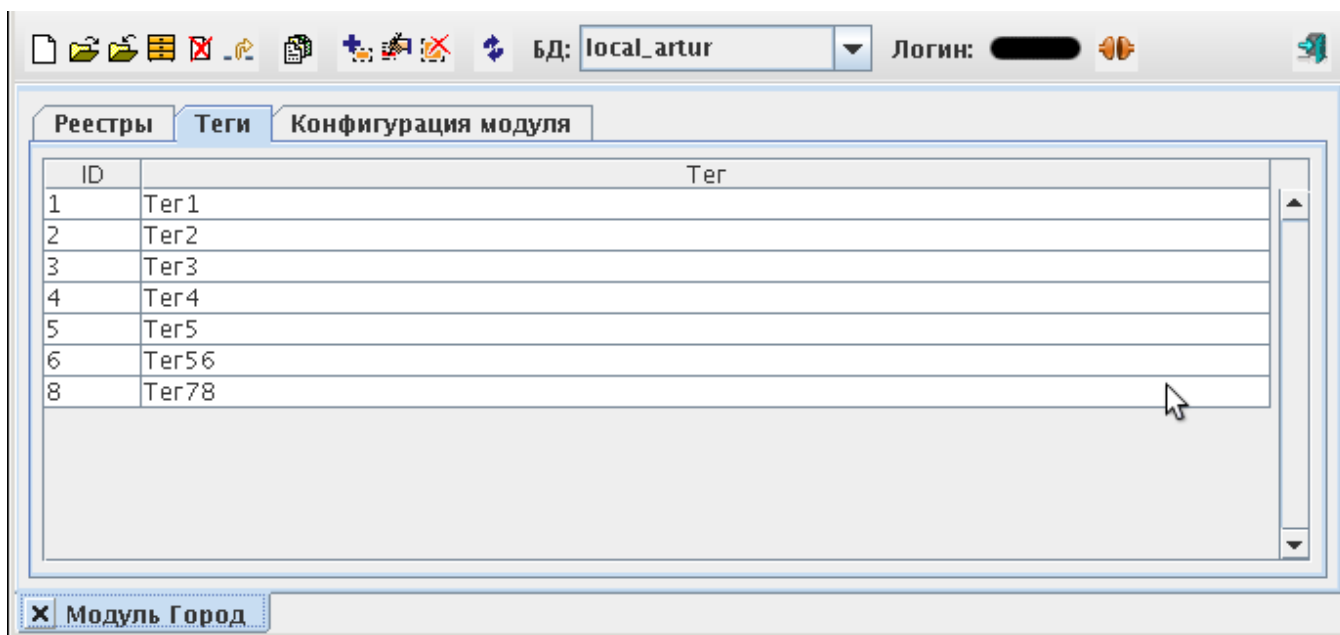
```
#,          3 ( .      (gr&3)>0)
#
account.gr.3=176
```

Для настройки шаблона реестра можно использовать следующие макроподстановки:

- [\${fio}] - ФИО клиента;
- [\${city}] - город;
- [\${street}] - улица;
- [\${house}] - дом;
- [\${frac}] - дробь дома;
- [\${flat}] - квартира;
- [\${contractTitle}] - название договора;
- [\${summ}] - сумма;
- [\${account}] - счет;
- [\${contractId}] - id контракта;
- [\${dateFrom}] - дата начала периода оплаты (ставится начало месяца);
- [\${dateTo}] - дата окончания периода оплаты (ставится дата формирования реестра).

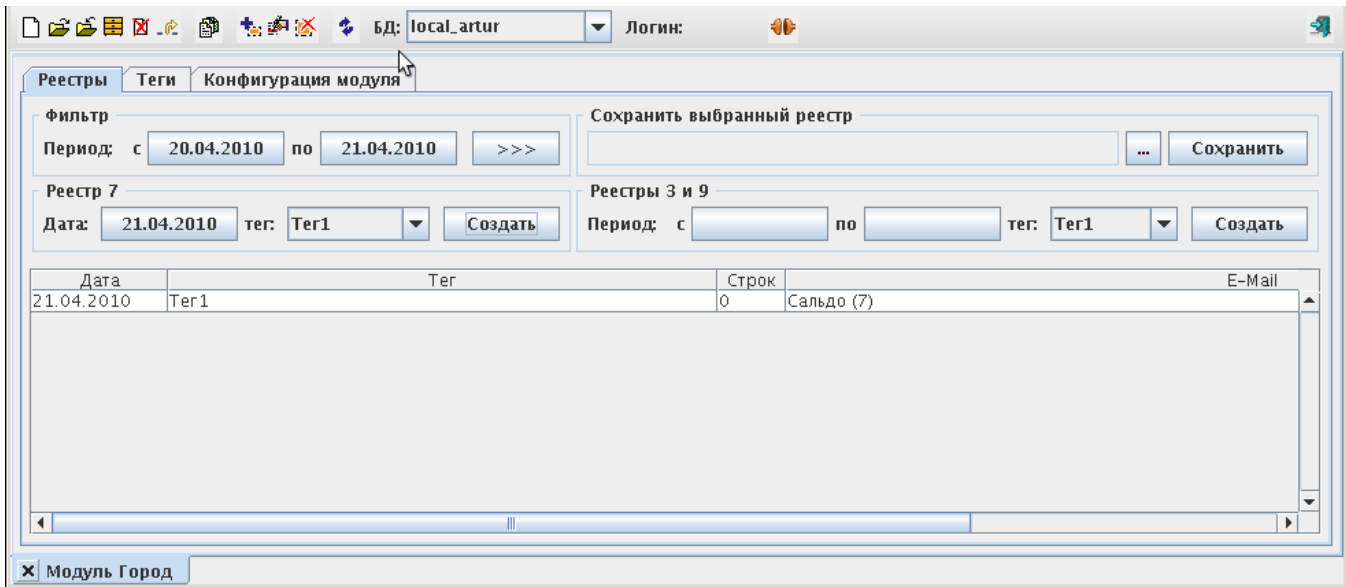
Каждая макроподстановка может встречаться в реестре только 1 раз. В шаблон не обязательно включать все поля - только те, что реально нужны. Между полями возможно вставить любой текст - он попадет в реестр в каждую строку. **Важно!** Квадратные скобки у полей обязательны!

Для того, чтобы настроить выгрузку реестров для договоров у них должен быть проставлен параметр **Ter**. Редактор тегов открывается с помощью меню **Модули=>Город=>Вкладка Теги**. Добавление, удаление и изменение тегов доступно через основную панель инструментов клиента биллинга с помощью кнопок **Добавить, Удалить, Изменить**.



Работа с реестрами

Для каждого дня должен быть зафиксирован реестр-сальдо 7, на основании 2х сальдовых реестров может быть созданы реестры 3 (изменений) и 9 (удаление).

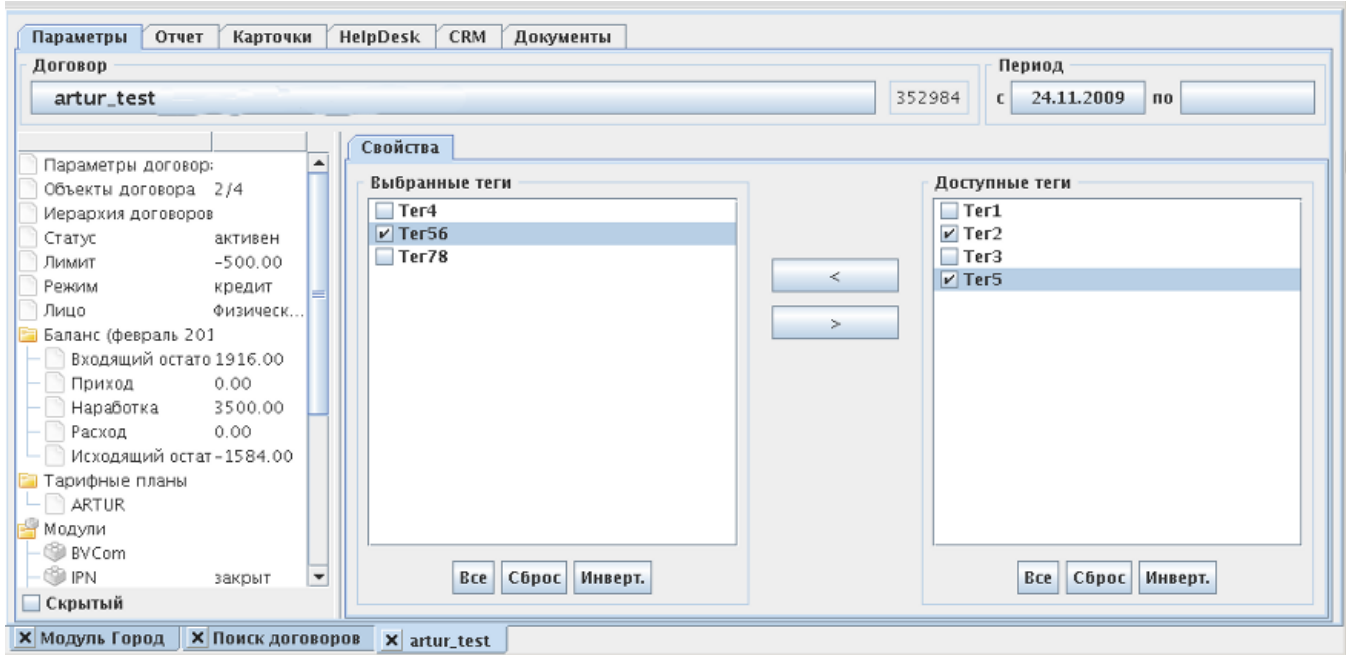


Как видно из скриншота, интерфейс очевиден и сложности не представляет: в левом верхнем углу расположен фильтр, позволяющий выбрать реестры лишь за определённый период. Выбранный реестр можно сохранить в указанный в правом верхнем углу файл.

При создании реестра 7 достаточно указать дату и тег, для создания реестров 3 и 9 - период и тег, для дат из периода должны существовать 7-ые реестры.

Использование модуля

Подключите экземпляр модуля к договору. Слева в дереве договора в разделе **Модули** появится экземпляр модуля **Город**. Выберите его. Справа появится возможность привязки тегов к договору.



Привязка не должна вызывать трудностей. Необходимо из списка **Доступные теги** выбрать те теги, которые нужно привязать к договору, затем нажать кнопку с изображением знака меньше "<". Выбранные теги появятся в списке **Выбранные теги**.

Удаление происходит аналогично добавлению тегов: из левого списка **Выбранные теги** необходимо отметить те теги, которые нужно удалить, а затем нажать кнопку с изображением знака больше ">". Удаленные теги появятся в правом списке **Доступные теги**.

При создании договора возможна установка первоначального набора тегов и автоматическое подключение модуля **Город**. Это осуществляется путем добавления модуля **Город** в шаблон создания договора. Там же устанавливаются теги, автоматически прикрепляемые к создаваемому договору.

Модуль MOBI.Деньги

Назначение модуля

Платежный модуль MOBI.Деньги предназначен для проведения платежей посредством мобильного телефона с использованием сервиса мобильных платежей [MOBI.Деньги](#).

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию.

```
# Web-
web.menuItem1= MOBI.
#
mobi.payment.comment= MOBI.
# " "
mobi.payment.type=
# MOBI.
mobi.provider.id=
# (demo|work)
mobi.mode=demo
```

Замечания:

1. Прежде, чем задавать **mobi.payment.type**, необходимо создать соответствующий тип платежа в Справочнике (**Справочники->Другие->Типы платежей**);
2. Для работы модуля необходимо передать адрес Web-сервиса на стороне биллинга, который будет осуществлять авторизацию и проведение платежа. URL выглядит следующим образом:

http(s)://<адрес_машины_биллинга>/bgbilling/mobexecuter/ru.bitel.bgbilling.modules.mobimoney/<mid>/MobiMoneyWS

где

- **<адрес_машины_биллинга>** - адрес сервера, на котором установлен биллинг;
- **<mid>** - код модуля mobi (можно посмотреть в редакторе модулей и услуг).

Например, если у вас биллинг находится по адресу **http://billing.example.com/bgbilling/** и модуль Mobi имеет mid=16, то результирующий URL, который нужно передать в сервис MOBI.Деньги, выглядит следующим образом: **http(s)://billing.example.com/bgbilling/mobexecuter/ru.bitel.bgbilling.modules.mobimoney/16/MobiMoneyWS**.

3. Для работы с системой Mobi необходимо установить сертификаты для этого необходимо сгенерировать запрос на подпись сертификата и отправить его в MOBI (процедура генерации описана на сайте [mobi](#)).

Полученный от MOBI файл с подписанным сертификатом **cert.crt** скопируйте в каталог сервера биллинга **cert/mobimoney/** (если каталогов не существует, создайте их вручную), туда же скопируйте файл с закрытым ключом **cert.key** и корневой сертификат от MobiMoney (**ca.crt**, файл нужно загрузить с сайта MobiMoney).

Если биллинг работает в связки в nginx, то файлы с ключом и сертификаты необходимо подключить к nginx (как это сделать смотрите в документации к nginx)

Настройте биллинг на работу по HTTPS протоколу. Как это сделать описано [здесь](#).

Проведение платежей

Платежи осуществляются абонентом через личный кабинет Web-статистики путем выбора соответствующего пункта меню.

Мониторинг платежей

Существует глобальный монитор всех платежей в системе. Он вызывается с помощью меню **Модули=>Модуль MOBI.Деньги**. Здесь можно отфильтровать платежи по дате, статусу, номеру договора.

Также в системе есть локальный монитор платежей для каждого договора. Посмотреть платежи договора можно на вкладке модуля MOBI.Деньги в дереве договора. Здесь также доступен фильтр по дате платежа, статусу.

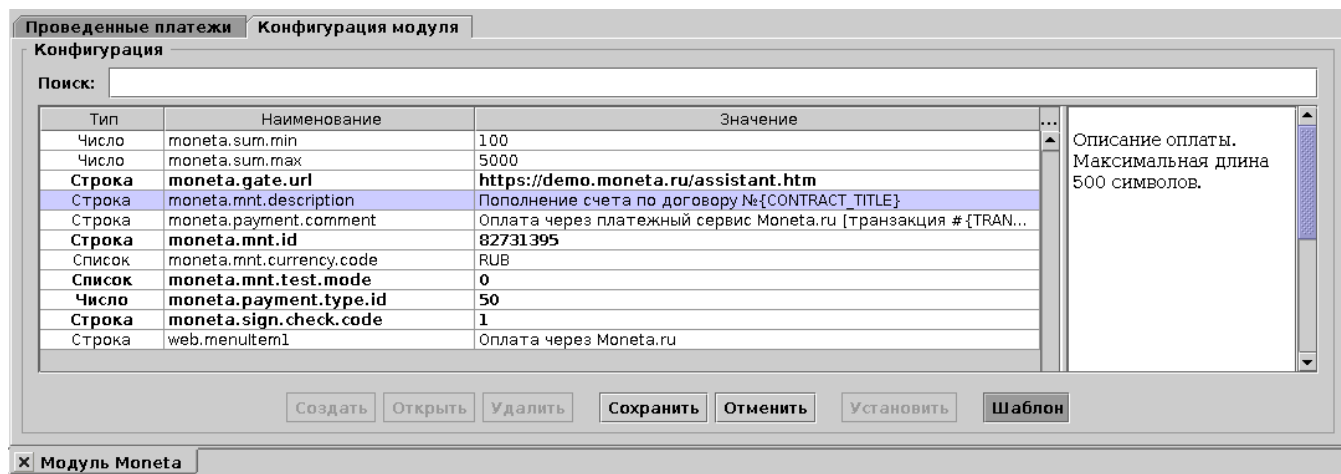
Модуль Moneta

Назначение модуля

Платежный модуль Moneta предназначен для осуществления безопасного приема платежей от абонентов. Более подробную информацию о сервисе, размере комиссии можно прочитать на сайте платежного сервиса Moneta (<http://moneta.ru>).

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.



Для оповещения о результатах оплаты, в Личном кабинете на сайте платежной системы нужно указать URL, по которому будет проходить оповещения.

Данный URL имеет следующий формат: **http(s)://host[:port]/bgbilling/moneta/<код_модуля_robokassa>**

URL успешной оплаты (SuccessURL) - **http[s]://hostname[:port]/bgbilling/webexecuter?module=moneta&mid=<код_модуля>&action=DoTransaction&operation=success**

URL ошибки (FailURL) - **http[s]://hostname[:port]/bgbilling/webexecuter?module=moneta&mid=<код_модуля>&action=DoTransaction&operation=fail**

Для возможности оплачивать через данный сервис необходимо добавить модуль на договор. После чего в личном кабинете будет доступен соответствующий пункт меню и страница оплаты.

Платежный сервис Moneta.ru

Сумма (от 100,00 до 5 000,00):

Оплатить

Текущий баланс:
99 878 754,1 руб.

Период:



Июль, 2016



Всего записей: 14

Страница:



1 из 1 [25]

[\[показать допол. фильтры\]](#)

№	Дата/Время оплаты	Статус	Сумма
18	21.07.2016 15:45:39	Завершена	118,00
17	21.07.2016 15:37:29	Завершена	117,00
16	21.07.2016 15:50:01	Завершена	116,00
15	21.07.2016 15:50:01	Завершена	115,00
14	21.07.2016 15:50:01	Завершена	114,00
13	21.07.2016 15:30:56	Новая	113,00
12	21.07.2016 15:29:32	Новая	112,00
11	21.07.2016 15:27:48	Завершена	111,00
10	21.07.2016 15:23:00	Завершена	110,00
9	21.07.2016 14:32:30	Новая	109,00
8	21.07.2016 13:10:00	Новая	108,00
7	21.07.2016 12:28:28	Новая	107,00
6	21.07.2016 12:16:46	Новая	106,00
5	21.07.2016 12:15:52	Новая	105,00

Модуль MPS

Назначение модуля

Модуль предназначен для интеграции с платёжными системами. В настоящее время поддерживается приём платежей из систем

- ОСМП/Qiwi (osmp) - <http://qiwi.ru>
- CyberPlat (cp) - <https://cyberplat.ru>
- XPlat (xplat) - <https://x-plat.ru>
- E-port (eport) - <http://qiwi.ru>
- Empray (empray)
- Rapida (rapida)
- Pegas (pegas)
- Comeraу (comeraу),
- Юникасса (unikassa) - <https://unikassa.kz>
- Quickraу (quickraу) - <http://quickpay.ru>
- Сбербанк (sbrf)
- Payfon24 (payfon24) - <http://payfon24.ru>
- (sfoura)
- (mgpe)
- (erc)
- (colomna)
- (sb)
- (optimaplus)
- (elecsnet)
- (sberbank)
- (unikassa)
- (bisys) ("центральная касса")
- (cityraу)
- (paynet)
- (click)
- (freecash)
- (elecsnetspb)
- (uepc)
- (mbank)
- (kaspi) - <https://www.kaspi.kz/>
- vtb
- и другие.

SSLv3

SSLv3 по умолчанию отключен в JRE/JDK с билдов:



- Java 8 update 31
- Java 7 update 75
- Java 6 update 91

Некоторые платёжные системы не работают через TLS, а требуют SSLv3. Т.е. при обновлении Java они перестают работать. Чтобы включить поддержку SSLv3 обратно, нужно в файле конфигурации <JRE_HOME>/lib/security/java.security убрать из параметра jdk.tls.disabledAlgorithms значение "SSLv3".

Однако следует учитывать, что SSLv3 устарел и не рекомендуется к использованию.

<http://www.oracle.com/technetwork/java/javase/documentation/cve-2014-3566-2342133.html>

SSLv3 is disabled by default

Starting with JDK 6u91 release, the SSLv3 protocol (Secure Socket Layer) has been deactivated and is not available by default. See the java.security.Security property jdk.tls.disabledAlgorithms in <JRE_HOME>/lib/security/java.security file.

If SSLv3 is absolutely required, the protocol can be reactivated by removing "SSLv3" from the jdk.tls.disabledAlgorithms property in the java.security file or by dynamically setting this Security property to "true" before JSSE is initialized.

It should be noted that SSLv3 is obsolete and should no longer be used.

Также для встроенного tomcat нужно добавить дополнительно параметр в data.properties сервера биллинга:

```
connector.https.sslEnabledProtocols=SSLv2,SSLv3,TLSv1,TLSv1.1,TLSv1.2,SSLv2Hello
```

Так как tomcat уже по умолчанию тоже не поддерживает SSLv3. Описание этого параметра есть тут :

<http://ci.apache.org/projects/tomcat/to.../http.html>

Здесь перечислены все возможные протоколы, некоторые можно убрать согласно целям и задачам.

Второй параметр определяет используемые cipher-ы:

```
connector.https.ciphers=TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,  
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_RC4_128_SHA,  
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA256,  
TLS_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA
```

Здесь перечислены подходящие параметры, например, всё отсюда: <https://www.ssllabs.com/ssltest/viewClient.html?name=Java&version=6u45> с убранными WEAK и INSECURE.

Настройка модуля MPS

Создайте не редактируемые типы платежей для систем платежей. id типов платежей можно узнать также в справочнике. id нужно будет указать в конфигурации.

Принестируйте модуль на сервер. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию по нижеприведённому примеру и сделайте ее активной.

Для всех систем платежей рекомендуется использовать только **https**-порт.

Здесь и далее "ПС" - платёжная система.

Возможные параметры конфигурации модуля:

Конфигурация

```
#      ,      ,      mps.ids=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
mps.ids=<mpsId_1>[,<mpsId_2>][,<mpsId_3>]...[,<mpsId_N>]
#      mps.<mpsId>.,      <mpsId> -
# / - 1/0 -
mps.<mpsId>.mode=1
#
mps.<mpsId>.title=CyberPlat
#
mps.<mpsId>.protocol=cp
# / .
mps.<mpsId>.login=
mps.<mpsId>.passwd=
# id      ,      ,
mps.<mpsId>.pid=
#
#      ,      /
# (      )
#mps.comment=$CONTRACT ($COMMENT)
#
#
#      (,      )
mps.<mpsId>.cert=1
#      ( )
# ( eport      , mps.1.cert=0)
#      modulus exponent:
#mps.<mpsId>.cert.mod=
#mps.<mpsId>.cert.exp=
#      encoded( 16- )
#mps.<mpsId>.cert.encoded=
#      pem (base64), header/footer (-----** PUBLIC KEY-----)
mps.<mpsId>.cert.pem=
#      , utf-8, cp1251
#mps.<mpsId>.request.encoding=
#
#      .
#      - .
#      1,      . ,      0,
# ..      mps.1.search.xxx ( mps.1.search.mode).
#      ,      mps.1.search.1.xxx ( mps.1.search.1.mode)
#      (...search.mode=),
#      (...search.pattern=)      .
#
#
#      (..      x0000 c 1,      account osmp 1_x0000)
#mps.<mpsId>.protocol.ext=osmpPrefix
#      Base-
#mps.<mpsId>.protocol.ext=noBaseAuth
#      :
#mps.<mpsId>.protocol.ext=noBaseAuth, osmpPrefix
#
#      (mps_login|contract|login|voice|email|phone|parameter|inet_login|custom)
mps.<mpsId>.search.mode=
#      ( login, phone)
#mps.<mpsId>.search.mid=
# " ",      regex
```

```

#
#mps.<mpsId>.search.multi=1
#mps.<mpsId>.search.1.mode=contract
#mps.<mpsId>.search.1.pattern=afsdfaf
#mps.<mpsId>.search.2.mode=parameter
#mps.<mpsId>.search.2.pid=4
#mps.<mpsId>.search.3.mode=phone
#mps.<mpsId>.search.3.mid=5
#   - pattern:::result,      regexp,
#   ,   -   ,   -   .
#   , \A(?:\d{5})|(?:\d{6})|(?:\d{7}))(\d{2})\z:::NK$1-$2
#       12345608,      NK123456-08,
#
#mps.<mpsId>.search.pattern=
#   , ru.bitel.bgbilling.modules.mps.server.bean.FindContract
#mps.<mpsId>.search.custom=
#   regexp ,      .      regexp,
#
#mps.<mpsId>.search.allow.contract.regexp=NK-.*
#   ,
#mps.<mpsId>.search.allow.contract.groups=4,2,5,3
#
#   .   - 0.
mps.<mpsId>.min.summ=
#   .   - 1000000
mps.<mpsId>.max.summ=
#
#   (   ,
#   +2 - 180)
#mps.<mpsId>.timeoffset=
#   ,   timeoffset.
#
#mps.<mpsId>.paytime=
#
#   mps (   )
#   mps ( web-)
#mps.mps_login.format=
#   regexp
#mps.mps_login.regexp=
#   , web-   mps regexp
#mps.mps_login.regexp.error=
#
#mps.<mpsId>.request.encoding=utf-8
#mps.<mpsId>.response.encoding=utf-8

```

В поле **mps.1.protocol.ext** можно указать расширения, если их несколько - через запятую:


osmpPrefix - означает, что тип поиска может быть передан как префикс в поле account. Расширение работает для протоколов ОСМП, Empay, Pegas, Rapida, Comeray.

noBaseAuth - означает, что в запросе не проверяется Base-аутентификация, т.е. параметры конфига mps.1.login и mps.1.passw не нужны.

payOnCheck - означает, что платеж проводится сразу при запросе на проверку.

addBalanceInfo - в ответ добавляется текущий баланс в поле account_balance (ОСМП, Empay, Pegas, Rapida, Comeray).

URL для платежных систем формируется из шаблона **https://хост[:порт]/контекст/mpsexecuter/<mid>/<mpsId>**,

 где **контекст** - по умолчанию bgbilling, **<mid>** - id модуля MPS, **<mpsId>** - код системы платежей, т.е. для вышеописанной конфигурации у CyberPlat mpsid=1.

Например, <https://server:8443/bgbilling/mpsexecuter/10/1>

Типы поиска :

- **login** - поиск по логину модулей DialUp или Voip (необходимо указать id модуля (mid)):

mps.1.search.mode=login

mps.1.search.mid=3

- **contract** - поиск по названию договора (например NK-0012; Здесь можно указать шаблон преобразования):

```
mps.1.search.mode=contract
```

```
mps.1.search.pattern=NK-$NUMBER - для поиска клиенту нужно будет ввести только 0012
```

или

```
mps.1.search.pattern=A((?:\d\d\d\d\d\d\d\d\d\d)?(?:\d\d\d\d\d\d\d\d\d\d)?(?:\d\d\d\d\d\d\d\d\d\d)?\d\d)z:::КФ$1-$2
```

(в последнем случае, если пришли 7, 8 или 9 цифр преобразовываем их к виду КФ\$1-\$2, т.е значения КФ12345-12 и 12345612 будут эквивалентны)

- **phone** - поиск по номеру телефона модуля phone (необходимо указать id модуля (mid));

- **parameter** - поиск по параметру договора (необходимо указать код типа параметра (pid)):

```
mps.1.search.mode=parameter
```

```
mps.1.search.pid=9
```

- **mps_login** - логин модуля MPS. Его выбирает себе клиент на странице статистики. Один для всех систем платежей.

- **inet_login** - логин модуля Inet (необходимо указать mid модуля Inet)

Для протоколов, не поддерживающих передачу типа поиска, реализована передача типа поиска через префикс x_идентификатор клиента (например, для ОСМП в поле account для поиска по search.1 логина 13 в запросе должно приходиться: account=1_13). Для поддержки этого режима нужно установить mps.x.protocol.ext=1.

В комментарии ответа (если такой поддерживается протоколом) можно передать номер, комментарий, параметр и/или баланс договора

```
mps.1.comment=$contract_title ($contract_comment) [$contract_param(4)] $contract_balance
```

Модификация ответа сервера биллинга на запросы от ПС

Для некоторых протоколов возможно модификация ответа сервера, например добавить дополнительные теги.

Для протокола OSMP есть возможность дополнять ответ на запрос, например тегом **<recsum>** с программным выставлением значения рекомендованной суммы платежа.

Цель:

```
<response>
  <result>0</result>
  <osmp_txn_id>122</osmp_txn_id>
  <comment>3#0034586 ( )</comment>
  <account_balance>0</account_balance>
  <recsum>1000</recsum>
</response>
```

это можно реализовать в скрипте при обработке события MPSBeforeRequestEvent

```
public class AddRecsum
    extends EventScriptBase<MPSBeforeRequestEvent>
{
    @Override
    public void onEvent( MPSOSMPRequestEvent event, Setup setup, ConnectionSet set )
        throws Exception
    {
        event.getResponse().getParameters().put( "recsum", 1000 );
        // event.setProcessed( true );
    }
}
```

С версии 7.2+ доступно событие **MPSBeforeResponseEvent** (поддерживается не всеми протоколами, есть в osmp, cp, kaspi, quickpay)


```

public class MPSResponseScript
    extends EventScriptBase<MPSBeforeResponseEvent>
{
    @Override
    public void onEvent( MPSBeforeResponseEvent mpsBeforeResponseEvent, Setup setup, ConnectionSet
connectionSet )
        throws Exception
    {
        Contract contract = mpsBeforeResponseEvent.getResponse().getContract();

        if ( contract != null )
        {
            MPSResponse response = mpsBeforeResponseEvent.getResponse();

            try( ContractParameterManager contractParameterManager = new ContractParameterManager(
connectionSet.getConnection() ); )
            {
                ContractAddressParamValue contractAddressParamValue = contractParameterManager.
getAddressParam( contract.getId(), PARAM_ADDRESS_ID );
                if ( contractAddressParamValue != null )
                {
                    address = contractAddressParamValue.getAddress();
                }
            }

            Map<String, String> responseParameters = new HashMap<>();
            if ( address != null )
            {
                responseParameters.put( "add", "address: " + address );
            }
            if ( !responseParameters.isEmpty() )
            {
                response.getParameters().put( MPSTransactionManager.KEY_RESPONSE_PARAMETERS,
responseParameters );
            }
        }
    }
}

```

ОСМП/Qiwi, Empay, Pegas, Rapida, Comerpay

В протоколе ОСМП/Qiwi (Empay, Pegas, Rapida, Comerpay) по умолчанию отсутствует параметр типа поиска, которым можно было бы разделить разные типы платежей. Однако его можно передавать в запросе дополнительным полем **pay_type** или же вложить префиксом в поле **account**, например 1_x0000, где

1 - это тип поиска, x0000 - это значение, по которому происходит поиск.

Параметр mps.x.numberPattern - регулярное выражение, которому должно удовлетворять значение поля account, иначе модуль возвращает ошибку 4 (не соответствие формата идентификатора абонента).

Аутентификация происходит по логину/паролю через BASE-AUTH протокола http и, если указана, по клиентскому сертификату, переданному при взаимодействии через протокол https.

В зависимости от типа протокола, необходимо указать соответствующий в конфигурации:

```
#
mps.<mpsId>.protocol=osmp
#Empay
mps.<mpsId>.protocol=empay
#Pegas
mps.<mpsId>.protocol=pegas
#Rapida
mps.<mpsId>.protocol=rapida
#Comerpay
mps.<mpsId>.protocol=comerpay
```

Пример конфигурации:

Код

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=
mps.<mpsId>.protocol=osmp
#
#   osmpFields noBaseAuth osmpSubBalance ( )
mps.<mpsId>.protocol.ext=osmpPrefix
mps.<mpsId>.login=
mps.<mpsId>.passwd=
mps.<mpsId>.pid=
# 0 ( )
mps.<mpsId>.search.mode=contract
# 1 (1_12345608)
mps.<mpsId>.search.1.mode=contract
#   - pattern:::result
mps.1.search.1.pattern=A((?:\d{5})|(?:\d{6})|(?:\d{7}))(\d{2})\z::NK$1-$2
#       id=12
mps.1.search.1.allow.contract.groups=12
#       account
mps.1.numberPattern=A\d{7,9}\z
#
mps.1.cert=1
mps.1.cert.pem=3bab58c...
#
# . - 0.
mps.1.min.summ=0
# . - 1000000
mps.1.max.summ=15000
#     " ",
#     (,
# ),
# ,
# mps.
mps.1.store.original.sum=true
```

В опции `mps.<mpsId>.protocol.ext=` можно указать расширение протокола `osmpFields`. Это расширение протокола позволяет добавить в ответ платежной системе (на любом этапе платежа: check, pay) специальный тег `<fields>` (см. документацию), в который с помощью динамического класса, реализующего обработчик глобального события "Перед обработкой Ответа", можно добавить произвольные параметры, которые будут отображены на экране, распечатаны на чеке, сохранены в ПС.

Пример обработчика события, который отображает фио на экране, распечатывает на чеке, показывает сумму долга и сумму для оплаты.

```
import java.math.BigDecimal;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Calendar;

import bitel.billing.server.contract.bean.Contract;
import bitel.billing.server.contract.bean.ContractManager;
import bitel.billing.server.contract.bean.ContractParameterManager;
import ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase;
import ru.bitel.bgbilling.modules.mps.server.events.MPSBeforeResponseEvent;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.Utills;
import ru.bitel.common.sql.ConnectionSet;

public class OSMPRequestHandler
extends EventScriptBase<MPSBeforeResponseEvent>
{
    //
    private static final int FIO_PID = 1;

    @Override
    public void onEvent( MPSBeforeResponseEvent event, Setup setup, ConnectionSet set )
    throws Exception
    {
        Connection con = set.getConnection();

        //
        Calendar c = Calendar.getInstance();
        Contract contract = event.getResponse().getContract();
        BigDecimal saldo = BigDecimal.ZERO;

        String query = "select summa1 + summa2 - summa3 - summa4 as saldo from contract_balance where
cid=" + contract.getId() + " and yy=" + c.get( Calendar.YEAR ) + " and mm=" + (c.get( Calendar.MONTH ) + 1);
try(Statement st = con.createStatement();ResultSet rs = st.executeQuery( query ))
{
    if(rs.next())
    {
        saldo = rs.getBigDecimal( "saldo" );
    }
}

// name1:type1:value1;name2:type2:value2;...;nameN:typeN:valueN
// , ,
// (.. disp), name:value. - ";",
// - ":"
StringBuilder sb = new StringBuilder();
sb.append( "sum_dolg:" + Utills.formatBigDecimalSumm( saldo ) + ";" );
sb.append( "oplata:" + Utills.formatBigDecimalSumm( saldo ) + ";" );
sb.append( "fio:" + cpm.getStringParam( contract.getId(), FIO_PID ) + ";" );
sb.append( "fio:ptr-data:" + cpm.getStringParam( contract.getId(), FIO_PID ) );
event.getResponse().getParameters().put( "fields", sb.toString() );
}
}
```

В некоторых случаях бывает полезно изменить сумму платежа, которая заносится в БД. Например, в ситуации, когда баланс абонента ведется в некоторой валюте, а платежная система присылает сумму в местной валюте. Для такого варианта предусмотрено событие "Перед добавлением платежа", в обработчик которого передается сумма платежа в виде 123.45. Обработчик может установить свою сумму и вернуть ее в модуль.

Событие называется **ru.bitel.bgbilling.modules.mps.server.events.MPSBeforePaymentEvent** и настраивается на вкладке **Сервис->Автоматизация->Функции глобальных событий**. Сумму платежа, пришедшую от платежной системы, можно получить с помощью метода **getPaymentSum()**, которая возвращает `java.util.BigDecimal`. Установить измененную сумму можно с помощью метода **setPaymentSum(BigDecimal newSum)**.

Пример обработчика события:

Пример обработчика события "Перед занесением платежа"

```
package ru.bitel.bgbilling.modules.mps;

import java.math.BigDecimal;
import ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase;
import ru.bitel.bgbilling.modules.mps.server.events.MPSBeforePaymentEvent;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.sql.ConnectionSet;

public class ModifyPaymentSumm
    extends EventScriptBase<MPSBeforePaymentEvent>
{
    @Override
    public void onEvent( MPSBeforePaymentEvent event1, Setup setup2, ConnectionSet connectionSet3 )
        throws Exception
    {
        BigDecimal sum = event1.getPaymentSum();
        sum = sum.multiply( new BigDecimal( 1.18 ) );
        event1.setPaymentSum( sum );
    }
}
```

CyberPlat

Типы поиска встроены в протокол.

Аутентификация происходит по логину/паролю через BASE-AUTH протокола http и, если указана, по клиентскому сертификату, переданному при взаимодействии через протокол https.

Для получения клиентского сертификата CyberPlat посылает Certificate Signing Request, при его подписи будет создан сертификат с данными и открытым ключом CyberPlat, подписанный серверным сертификатом. Сертификат следует генерировать версии 1 (V1).

Пример конфигурации:

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=
mps.<mpsId>.protocol=cp
mps.<mpsId>.login=
mps.<mpsId>.passwd=
mps.<mpsId>.pid=
# 0 ( )
mps.<mpsId>.search.mode=contract
# 1 (1_12345608)
mps.<mpsId>.search.1.mode=contract
# - pattern:::result
mps.<mpsId>.search.1.pattern=\A(?:\d{5})|(?:\d{6})|(?:\d{7})(\d{2})\z::NK$1-$2
mps.<mpsId>.search.2.mode=login
mps.<mpsId>.search.2.mid=5
mps.<mpsId>.search.3.mode=login
mps.<mpsId>.search.4.mid=8
#
mps.<mpsId>.cert=1
mps.<mpsId>.cert.pem=3bab58c...
```

XPlat

Для XPlat необходимо назначить дополнительный параметр запроса `account`, в котором передаётся номер договора, логин или номер телефона, а также можно добавить необязательный параметр `type`, в котором будет передаваться тип поиска (аналогично CyberPlat). В случае, если используется параметр `type`, то при формировании md5 системой XPlat параметры должны идти в порядке: `account`, `type`.

Аутентификация происходит по подписи. Проверку сертификата следует отключить (`mps.<x>.cert=0`).

В конфигурации необходимо указать секреты запроса и ответа для проверки/создания подписи.

Пример конфигурации:

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=XPlat
mps.<mpsId>.protocol=xplat
mps.<mpsId>.pid=
mps.<mpsId>.search.mode=contract
mps.<mpsId>.search.mid=
#512 xplat
mps.<mpsId>.secret=
# md5
mps.<mpsId>.secret.response=
```

Eport

Аутентификация EPort происходит по подписи закрытого ключа. Открытый ключ eport необходимо указать в mps.<x>.cert. Им будет производиться проверка запросов. Сервер подписывает ответы своим закрытым ключом, поэтому необходимо передать открытый ключ (сертификат с открытым ключом) ps.

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=E-port
mps.<mpsId>.protocol=eport
# (search.x)
#mps.<mpsId>.protocol.ext=0
mps.<mpsId>.pid=
mps.<mpsId>.search.mode=
mps.<mpsId>.cert=1
#
mps.<mpsId>.cert.pem=
```

SFOUR PayBox Alternative

Аутентификация запросов/ответов происходит по подписи, основанной на секретном слове.

Также как и в других системах рекомендуется использование транспорта https.

Для передачи номера договора/логина/номера телефона в запросе на проведение платежа необходимо назначить в системе дополнительное поле AccountNumber. Также возможно назначить поле AccountType, в котором будет передаваться тип поиска.

При ответе на проверку, если договор найден, в ответе идёт поле AccountComment, в котором содержится комментарий (см. опцию mps.comment в конфигурации). Подпись ответа сервера: SHA-1(<SessionID>, <ErrorCode>, <ClearingNumber>, <AccountNumber>, <MachineSecret>)

При запросе на проведение платежа сумма должна быть передана в параметре Amount.

Поле AccountComment должно быть возвращено в запросе на проведение платежа, подпись при этом: SHA-1(<SessionID>, <MachineMark>, <ClearingNumber>, <AccountNumber>, <AccountComment>, <Amount>, <MachineSecret>)

URL типов запросов различаются, т.е, например, запрос на проверку

`https://server:8443/bgbilling/mpsexecuter/10/1/Check`

проведение платежа - `https://server:8443/bgbilling/mpsexecuter/10/1/Pay`

статус - `https://server:8443/bgbilling/mpsexecuter/10/1/Status`

Здесь 10 - это код модуля, 1 - код системы платежей (аналогично с другими типами систем).

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=SFfour
mps.<mpsId>.protocol=sfoura
mps.<mpsId>.pid=
mps.<mpsId>.search.mode=
#
mps.<mpsId>.secret=
```


ОПТИМА plus

Аутентификация происходит по подписи закрытого ключа. Открытый ключ ПС необходимо указать в `mps.<x>.pem`. Им будет производиться проверка запросов. Сервер подписывает ответы своим закрытым ключом, поэтому необходимо передать открытый ключ (сертификат с открытым ключом) пс.

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=OptimaPlus
mps.<mpsId>.protocol=optimaplus
mps.<mpsId>.pid=
mps.<mpsId>.search.mode=
mps.<mpsId>.cert=1
#      header/footer (-----** PUBLIC KEY-----)
mps.<mpsId>.cert.pem=
```

Elecsnet

Аутентификация происходит по ЭЦП закрытого ключа. Открытый ключ ПС необходимо указать в **mps.<x>.pem**. Этим ключом будет производиться проверка запросов с сервера ПС. Сервер биллинга подписывает ответы своим закрытым ключом, поэтому необходимо передать свой открытый ключ ПС. В ответном запросе серверу ПС можно указать параметры поля протокола **ansid** для вывода их на терминале клиенту.

Конфигурация

```
#1-
#0-
mps.<mpsId>.mode=1
mps.<mpsId>.title=Elecsnet
mps.<mpsId>.protocol=elecsnet
#
mps.<mpsId>.pid=<pid>
mps.<mpsId>.cert=0
mps.<mpsId>.cert.pem=
# . .
mps.<mpsId>.search.mode=parameter
#
mps.<mpsId>.search.pid=<pid>
# ( ,
# +2 - 180)
mps.<mpsId>.timeoffset=180
# , timeoffset.
# ,
mps.<mpsId>.paytime=1440
# , .
# ansid elecsnet.
# .
# summ - elecsnet
mps.<mpsId>.response.attribute.summ=@sumin@3000
# . , . $title, $comment, $pid=32, 32 - id .
mps.<mpsId>.response.attribute.fio=$title
# regexp , , ...
# : ::: .
mps.<mpsId>.response.attribute.fio_regexp=[-]+:::\.
# ...
mps.<mpsId>.response.attribute.comment=
```

Юникасса

Аутентификация производится по логину/паролю. Используются SSL динамические ключи, не требующие подтверждения в центрах сертификации.

Запросы от сервера Ekassir приходят с IP-адресов: 195.96.80.186, 195.96.80.187, 195.96.80.188 и 195.96.80.182 и 2-х сетей: 195.239.136.160/27 и 217.148.219.32/28. Не забудьте добавить их в список исключений, если вы собираетесь фильтровать входящий трафик.

```
mps.<mpsId>.mode=1  
mps.<mpsId>.title=Unikassa  
mps.<mpsId>.protocol=unikassa  
mps.<mpsId>.pid=  
mps.<mpsId>.search.mode=  
mps.<mpsId>.cert=0  
mps.<mpsId>.login=  
mps.<mpsId>.passw=
```

Используется один тип поиска (с индексом 0), т.к. заявляется, что в сторонней системе менять соответствующий параметр (sid) нельзя, потому его значение в данный момент игнорируется.

Quickpay

Аутентификация производится сертификатом.

```
#1-
#0-
mps.<mpsId>.mode=1
mps.<mpsId>.title=Quickpay
mps.<mpsId>.protocol=quickpay
mps.<mpsId>.secret=<secret>
#
mps.<mpsId>.pid=<payment_id>
mps.<mpsId>.search.mode=login
mps.<mpsId>.search.mid=<module_id>
mps.<mpsId>.search.1.mode=contract
#mps.<mpsId>.search.1.mid=<module_id>
mps.<mpsId>.protocol.ext=noBaseAuth
```

Sberbank

Аутентификация производится по сертификату.

```
# (1) / (0)
mps.<mpsId>.mode=1
#
mps.<mpsId>.title=
mps.<mpsId>.protocol=sberbank
mps.<mpsId>.protocol.ext=noBaseAuth
#   Base-.
#   noBaseAuth
#mps.<mpsId>.login=sber
#mps.<mpsId>.passw=rgnko7fh
#
mps.<mpsId>.cert=0
mps.<mpsId>.cert.pem=sdfDdfs1....
#
mps.<mpsId>.pid=<payment_type_id>
#   -
mps.<mpsId>.search.mode=contract
#
mps.<mpsId>.min.summ=100
mps.<mpsId>.max.summ=15000
#   ,   . add,   .
# <address_param_id> -
#mps.<mpsId>.comment=$contract_param(<address_param_id>)
```

Адрес для приема реестров платежей: **[http\(s\)://<bgbilling_url>/mpsexecuter/<mid>/<mpsId>/register](http(s)://<bgbilling_url>/mpsexecuter/<mid>/<mpsId>/register)**

Сбербанк (sbrf)

Аутентификация производится по сертификату.

```
#!/ - 1/0 -
mps.<mpsId>.mode=1
mps.<mpsId>.title=
mps.<mpsId>.protocol=sbrf
mps.<mpsId>.protocol.ext=noBaseAuth
#mps.<mpsId>.login=sber // Base-
#mps.<mpsId>.passwd=
mps.<mpsId>.cert=
mps.<mpsId>.cert.pem=
mps.<mpsId>.pid=
mps.<mpsId>.search.mode=contract
mps.<mpsId>.min.summ=10
mps.<mpsId>.param.keys=sum, fio, address
mps.<mpsId>.sum=300.00
mps.<mpsId>.fio=<textParamId1>[,<textParamId2>[,<textParamId3>...]]
mps.<mpsId>.address=<addressParamId1>[,<addressParamId2>[,<addressParamId3>...]]
#mps.<mpsId>.request.encoding=utf-8
#mps.<mpsId>.response.encoding=utf-8
```

Bisys

Протокол для платежного сервиса от компании ООО "Биллинговые системы" (включает сервис "центральная касса"). Аутентификация в данном протоколе осуществляется по подписи запросов с помощью секрета. Настройки протокола:

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=
mps.<mpsId>.protocol=bisys
mps.<mpsId>.secret=<secret>
#
mps.<mpsId>.pid=<payment_id>
mps.<mpsId>.search.mode=login
mps.<mpsId>.search.mid=<dialup_module_id>
mps.<mpsId>.comment=[<contract_title> [<contract_comment> [<contract_param(4)>]
mps.<mpsId>.protocol.ext=noBaseAuth, needClientInfo
#
mps.<mpsId>.request.encoding=<encoding>
# ( utf-8) [7.0+]
mps.<mpsId>.response.encoding=<encoding>
```

Расширение **needClientInfo** позволяет опционально добавлять информацию о клиенте (ФИО) в ответ на запрос платежной системы о возможности совершения платежа. Содержимое информации о клиенте определяется параметром конфигурации **mps.<mpsId>.comment**.

Paynet

Протокол реализован совместно с компанией ООО «UZPAYNET» ([официальный сайт](#)). Протокол работает через SOAP, реализованный в биллинге в модуле MPS.

```
# Paynet
mps.<mpsId>.protocol=paynet
#
mps.<mpsId>.title=Paynet
# (1), (0)
mps.<mpsId>.mode=1
#
mps.<mpsId>.login=user
mps.<mpsId>.passwd=password
# ,      getInformation. , .
mps.<mpsId>.getInformation.class=
# . : mps.<mpsId>.pid.<serviceId>=<payment_type_id>
# mpsId -      (      , mpsId=1)
#serviceId -      .. paynet. serviceId      ,      serviceId      . serviceId = 1 (paynet), serviceId = 2 (click)
..
#payment_type_id -      ( - )
mps.<mpsId>.pid.1=
# . . true - . . , false - .
mps.<mpsId>.deny.jur=
```

Остановимся подробнее на опции **mps.<mpsId>.getInformation.class**. Класс, прописанный в данной строке конфигурации, должен реализовать метод **getResponseParameters** интерфейса **ru.bitel.bgbilling.modules.mps.server.bean.ws.paynet.ResponceParameters** и возвращать **java.util.Map<String, String>**. Список возвращаемых параметров определяется по договоренности с платежной системой. Ниже представлен пример реализации для случая, когда необходимо вернуть в платежную систему информацию о балансе договора.

Пример реализации интерфейса ResponceParameters

```
package ru.bitel.bgbilling.modules.mps.paynet;

import java.sql.Connection;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

import ru.bitel.bgbilling.common.BGException;
import ru.bitel.bgbilling.modules.mps.server.bean.ws.paynet.ResponceParameters;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.Utills;
import ru.bitel.billing.server.contract.bean.BalanceUtills;

public class GetResponceParameters
    implements ResponceParameters
{
    @Override
    public Map<String, String> getResponceParameters( Connection con, Setup setup, int contractId )
        throws BGException
    {
        Map<String, String> result = new HashMap<String, String>();
        try(BalanceUtills bu = new BalanceUtills( con ))
        {
            result.put( "balance", Utills.formatBigDecimalSumm( bu.getBalance( new Date(), contractId ) ));
        }
        return result;
    }
}
```


В paynet необходимо передать данные для аутентификации, которые прописываются в параметрах **mps.<mpsId>.login** и **mps.<mpsId>.passw**. Также необходимо передать url web-сервиса, к которому будут обращаться для уведомления о поступлении платежа. Url выглядит следующим образом:

```
http://<server_host>:<server_port>/bgbilling/mpsexecuter/mps-ws/ru.bitel.bgbilling.modules.mps.paynet/<mps_mid>/ProviderWebService/<mpsId>?wsdl
```

где:

- **<mps_mid>** - код модуля MPS
- **<mpsId>** - код протокола в конфигурации модуля

Также следует обратить внимание на то, что биллинг ожидает, что для идентификации абонента используется номер его договора, который передается в параметре **clientId**.

Обязательным условием со стороны paynet является наличие HTTPS.

В некоторых случаях бывает полезно изменить сумму платежа, которая заносится в БД. Например, в ситуации, когда баланс абонента ведется в некоторой валюте, а платежная система присылает сумму в местной валюте. Для такого варианта предусмотрено событие "Перед добавлением платежа", в обработчик которого передается сумма платежа в виде 123.45. Обработчик может установить свою сумму и вернуть ее в модуль. Событие называется **ru.bitel.bgbilling.modules.mps.server.events.MPSBeforePaymentEvent** и настраивается на вкладке **Сервис->Автоматизация->Функции глобальных событий**. Сумму платежа, пришедшую от платежной системы, можно получить с помощью метода **getPaymentSum()**, которая возвращает `java.util.BigDecimal`. Установить измененную сумму можно с помощью метода **setPaymentSum(BigDecimal newSum)**.

Пример обработчика события:

Пример обработчика события "Перед занесением платежа"

```
package ru.bitel.bgbilling.modules.mps.paynet;

import java.math.BigDecimal;
import ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase;
import ru.bitel.bgbilling.modules.mps.server.events.MPSBeforePaymentEvent;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.sql.ConnectionSet;

public class ModifyPaymentSumm
    extends EventScriptBase<MPSBeforePaymentEvent>
{
    @Override
    public void onEvent( MPSBeforePaymentEvent event1, Setup setup2, ConnectionSet connectionSet3 )
        throws Exception
    {
        BigDecimal sum = event1.getPaymentSum();
        sum = sum.multiply( new BigDecimal( 1.18 ) );
        event1.setPaymentSum( sum );
    }
}
```

Click

Протокол реализован для приема платежей через систему **ООО «CLICK»** (<http://click.uz>). Ниже приведен пример конфигурации для работы с платежной системой:

```
#
mps.1.protocol=click
mps.1.protocol.ext=noBaseAuth,denyForJur
#
mps.1.title=Click
# (1), (0)
mps.1.mode=1
#
mps.1.login=user
mps.1.passw=password
#
mps.1.pid=
#
mps.1.secret=
# " ",
# ( ,
# ),
# ,
# mps.
mps.1.store.original.sum=true
#
mps.1.search.mode=
```

В некоторых случаях бывает полезно изменить сумму платежа, которая заносится в БД. Например, в ситуации, когда баланс абонента ведется в некоторой валюте, а платежная система присылает сумму в местной валюте. Для такого варианта предусмотрено событие "Перед добавлением платежа", в обработчик которого передается сумма платежа в виде 123.45. Обработчик может установить свою сумму и вернуть ее в модуль. Событие называется **ru.bitel.bgbilling.modules.mps.server.events.MPSBeforePaymentEvent** и настраивается на вкладке **Сервис->Автоматизация->Функции глобальных событий**. Сумму платежа, пришедшую от платежной системы, можно получить с помощью метода **getPaymentSum()**, которая возвращает `java.util.BigDecimal`. Установить измененную сумму можно с помощью метода **setPaymentSum(BigDecimal newSum)**.

Пример обработчика события:

Пример обработчика события "Перед занесением платежа"

```
package ru.bitel.bgbilling.modules.mps;

import java.math.BigDecimal;
import ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase;
import ru.bitel.bgbilling.modules.mps.server.events.MPSBeforePaymentEvent;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.sql.ConnectionSet;

public class ModifyPaymentSumm
    extends EventScriptBase<MPSBeforePaymentEvent>
{
    @Override
    public void onEvent( MPSBeforePaymentEvent event1, Setup setup2, ConnectionSet connectionSet3 )
        throws Exception
    {
        BigDecimal sum = event1.getPaymentSum();
        sum = sum.multiply( new BigDecimal( 1.18 ) );
        event1.setPaymentSum( sum );
    }
}
```

MBank

Официальный сайт платежной системы. Проверка правильности запросов осуществляется с помощью алгоритма MD5, который применяется к входящим параметрам запроса. Авторизация не требуется. Минимальная конфигурация для работы с платежной системой.

Конфигурация. Протокол mbank

```
# . 0
mps.<mps_id>.mode=1
#
mps.<mps_id>.title=Mbank
# MPS, .
mps.<mps_id>.protocol=mbank
# , (noBaseAuth) , (action=1).
mps.<mps_id>.protocol.ext=noBaseAuth, needClientInfo
#
mps.<mps_id>.pid=37
# . , MPS (. " MPS")
mps.<mps_id>.search.mode=
# , ,
mps.<mps_id>.secret=123456
# (action=1). , . . , $contract_title $contract_param(<param_id>), <param_id> - .
mps.<mps_id>.comment=
```

URL, Mbank, : http(s)://<bgbilling_host>[:<port>]/<context>/mpsexecuter/<mid>/<mps_id>

Kaspi

Интеграция с платежной системой [Kaspi](#). Минимальная конфигурация для работы с платежной системой.

Конфигурация для Kaspi

```
# . 0
mps.<mps_id>.mode=1
#
mps.<mps_id>.title=
# MPS, .
mps.<mps_id>.protocol=kaspi
# , (noBaseAuth)
mps.<mps_id>.protocol.ext=noBaseAuth
#
mps.<mps_id>.pid=
# . , MPS (. " MPS")
mps.<mps_id>.search.mode=contract
# (action=1). , . . , $contract_title $contract_param(<param_id>), <param_id> - .
mps.<mps_id>.comment=
```

ВТБ

Пример конфигурации

```
mps.<mpsId>.mode=1
mps.<mpsId>.title=
mps.<mpsId>.protocol=vtb
mps.<mpsId>.protocol.ext=noBaseAuth
mps.<mpsId>.pid=10
mps.<mpsId>.search.mode=contract
mps.<mpsId>.search.allow.contract.groups=2
mps.<mpsId>.min.summ=1
mps.<mpsId>.max.summ=10000
#mps.<mpsId>.param.keys=accountPath[,...]
#mps.<mpsId>.accountPath=
```

По требованию протокола банка, ответы на сообщения должны подписываться с **MD5withRSA** и подпись передаваться в HTTP заголовке **x-sign**

Для подписи используется **Private Key** из **.keystore**

Пароль для **Private Key** задается в конфигурации сервера в параметре **keystore.password**, пароль по умолчанию **bgbilling**.

Сертификаты

ОСМП, Empray, Pegas, Rapida, Comeray, CyberPlat используют метод аутентификации по сертификатам. Для работы необходим [серверный сертификат](#) и [использование HTTPS](#).

1. Необходимо экспортировать серверный сертификат, чтобы удаленная сторона могла добавить его в доверенные:

```
keytool -keystore .keystore -alias bgbilling -exportcert -file bgbilling.cer
```

Сконвертировать сертификат в x509 base64 можно командой:

```
openssl x509 -inform der -in bgbilling.cer -out bgbilling.pem
```

2. Работа с клиентским сертификатом платежной системы может проходить в нескольких вариантах:

2.1. Платежная система присылает сертификат, который просто нужно добавить в доверенные:

```
keytool -keystore .keystore -alias pegas -importcert -file pegas.cer
```

Далее переходим к пункту 3.

2.2. Подпись запроса на сертификат или создание подписанного сертификата:

Сначала создаем главный сертификат для модуля, им будут подписаны клиентские сертификаты, а сам он будет добавлен в доверенные сертификаты в .keystore (таким образом).

Генерируем ключ, который будет зашифрован паролем:

```
[amir@ts01 keys]$ openssl genrsa -des3 -out mps.key 1024
Generating RSA private key, 1024 bit long modulus
e is 65537 (0x10001)
Enter pass phrase for mps.key:
Verifying - Enter pass phrase for mps.key:
```

Создаем на основе ключа сертификат, здесь понадобится ввести описание сертификата, точка ('.') означает пустое поле:

```
[amir@ts01 keys]$ openssl req -new -x509 -days 1001 -key mps.key -out mps.pem
Enter pass phrase for mps.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:RU
State or Province Name (full name) [Berkshire]:.
Locality Name (eg, city) [Newbury]:.
Organization Name (eg, company) [My Company Ltd]:Provider
Organizational Unit Name (eg, section) []:Provider BGBilling MPS
Common Name (eg, your name or your server's hostname) []:Provider BGBilling MPS
Email Address []:support@provider.ru
```

Добавляем сертификат в доверенные:

```
keytool -keystore .keystore -alias mps -importcert -file mps.pem
```

Далее, если нам прислали запрос на подпись сертификата (.csr) - переходим к пункту 2.2.2, иначе - к пункту 2.2.1.

2.2.1. Если нам нужно самим создать сертификат (нам не присылали запрос на подпись):

Генерируем ключ, osmp.key - в данном случае имя файла, здесь уже нужно использовать другой пароль, потому что его необходимо будет также переслать удаленной стороне:

```
[amir@ts01 keys]$ openssl genrsa -des3 -out osmp.key 1024
Generating RSA private key, 1024 bit long modulus
e is 65537 (0x10001)
Enter pass phrase for osmp.key:
Verifying - Enter pass phrase for osmp.key:
```

И создаем запрос на подпись:

```
[amir@ts01 keys]$ openssl req -new -key osmp.key -out osmp.csr
Enter pass phrase for osmp.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:RU
State or Province Name (full name) [Berkshire]:.
Locality Name (eg, city) [Newbury]:.
Organization Name (eg, company) [My Company Ltd]:OSMP
Organizational Unit Name (eg, section) []:OSMP-BGBilling
Common Name (eg, your name or your server's hostname) []:www.osmp.ru
Email Address []:support@osmp.ru

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Далее переходим к пункту 2.2.2.

2.2.2. Если нам прислали CSR-запрос на подпись или мы только что сами создали новый, создаем подписанный сертификат из запроса:

```
[amir@ts01 keys]$ openssl x509 -req -in osmp.csr -CA mps.pem -CAkey mps.key -out osmp.pem -days 1001 -
CAcreateserial -CAserial mps.seq
Signature ok
subject=/C=RU/O=OSMP/OU=OSMP-BGBilling/CN=www.osmp.ru/emailAddress=support@osmp.ru
Getting CA Private Key
Enter pass phrase for mps.key:
```

Если к нам приходил только CSR-запрос, то нужно послать только osmp.pem.

Если мы сами генерировали клиентский закрытый ключ, то необходимо передать как osmp.pem, так и osmp.key, т.е. сертификат и приватный ключ.

Связку закрытый ключ и сертификат можно сконвертировать в PKCS12 формат.

```
[amir@ts01 keys]$ openssl pkcs12 -export -in osmp.pem -inkey osmp.key -out osmp.p12
```

Сначала необходимо будет ввести пароль на закрытый ключ, который указали при генерации osmp.key, затем два раза - новый (можно тот же самый) уже для нового PKCS файла.

3. Открытый ключ добавленного в доверенные/созданного/подписанного клиентского сертификата необходимо указать в конфиге.

Для этого необходимо скопировать содержание открытого ключа одной строкой, без заголовка и окончания '-----BEGIN PUBLIC KEY-----'/-----END PUBLIC KEY-----'

Например:

```
mps.1.cert.pem=MIGfMA0GCSqGSIsb3DQE...EmO5Phqo2FG52KwIDAQAB
```

Извлечь открытый ключ из сертификата можно так:

```
[amir@ts01 keys]$ openssl x509 -inform der -in osmp.cer -pubkey  
#  
[amir@ts01 keys]$ openssl x509 -inform pem -in osmp.pem -pubkey
```


Менеджер платежей

Позволяет просматривать платежи за период.

Менеджер платежей | Сверка платежей | Конфигурация модуля

Система: ОСМП | Период: с 01.07.2006 по 31.08.2006 | Тип: Проведенные | Вывести

Платежи

#транзакции ПС	Договор	Внутренний #транзакции	Сумма	Время (часовой пояс ПС)	Комментарий платежа	Статус
222223	x0000	21	10.45	19.07.2006 13.13.33	OSMP	проведен
118969266	x0002	22	34.00	19.07.2006 12.01.36	OSMP	проведен
1223	x0000	20	10.45	19.07.2006 13.13.33	OSMP	проведен
4	x0000	23	10.45	12.07.2006 15.23.33	OSMP	проведен

Кол-во платежей: 4 на сумму 65.35

Внизу показываются количество платежей и сумма только с учётом проведённых платежей

Сверка платежей MPS

Позволяет сверить реестр системы платежей с данными сервера биллинга и при необходимости добавить, отменить или восстановить платежи. Не все протоколы поддерживают сверку. Если Вы загружаете файл реестра, но все платежи попадают в лишние - это может означать, что файл реестра парсится неверно. По умолчанию парсинг реестра включен в поддержку протокола модуля.

Для изменения параметров парсинга реестра можно задать параметры в конфигурации платежной системы:

```
#regexp
mps.1.register.pattern=^(.+)[;\t](\d+)[;\t](\d+:[\d+ -TZ+)[;\t](\d+\.*\d*)[;\t](\d+)$
# regexp
mps.1.register.patternOrder=account:1, time:3, sum:4, transId:5
#
mps.1.register.timeFormat=yyyy-MM-dd'T'HH:mm:ss'Z'
```

Менеджер платежей | Сверка платежей | Конфигурация модуля

Период: с 01.07.2006 по 01.07.2006 | Файл: С:\Текстовый документ.txt | ОСМП | Обработать

Результат

Недостающие платежи | Лишние платежи | Отмененные, но присутствующие в реестре | Отличающиеся платежи

#транзакции ПС	Договор	Внутренний #транзакции	Сумма	Время (часовой пояс ПС)
95752972	не найден	0	123.45	25.02.2005 12:13:14
95752982	не найден	0	0.01	25.02.2005 13:22:34
95752992	не найден	0	123.01	25.02.2005 14:55:11
95753002	не найден		1000.00	25.02.2005 14:55:12

Добавить

Web-интерфейс модуля MPS

Web-интерфейс пользователя позволяет просмотреть платежи за месяц и установить/сменить номер для поиска:

Договор N x0000

[Вернуться в меню](#)

Список платежей

Система: Месяц: Год:

внешний ID	внутренний ID	Дата/время оплаты	Сумма	Комментарий	Статус
------------	---------------	-------------------	-------	-------------	--------

Номер для поиска договора в платежных системах:

Модуль MTSBank (ПАО «МТС-Банк»)

Модуль MTSBank предназначен для оплаты картами через процессинг МТСБанка.

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.

Наименование	Значение	Тип
mtsbank.gate.url	https://payments.mtsbank.ru/ecommm/ClientHandler	Строка
mtsbank.description	Пополнение счета по договору №{CONTRACT_TITLE}	Строка
mtsbank.merchant.url	https://payments.mtsbank.ru/ecommm/MerchantHandler	Строка
mtsbank.cert.path.crt	cert/mtsbank/cert.pem	Строка
mtsbank.cert.path.key	cert/mtsbank/key.pem	Строка
mtsbank.cert.path.host	cert/mtsbank/host.pem	Строка
mtsbank.payment.type.id	1	Число
mtsbank.cert.key.password		Строка
web.menuitem1	Оплата картой через МТСБанк	Строка

Для банка нужно будет предоставить URL на который будут перенаправлять клиента после оплаты. URL формируется следующим образом:

для 7.0+

```
http[s]://host[:port]/bgbilling/webexecuter?action=DoTransaction&mid=<_>&module=mtsbank&check=1
```

для остальных

```
http[s]://host[:port]/bgbilling/webexecuter?action=DoTransaction&mid=<_>&module=mtsbank&operation=check
```

ВНИМАНИЕ!!! В связи с "особенностями" протокола шлюза, возможны ситуации с не получением биллингом информации о проведении платежа. В таких случаях требуется ручная проверка статуса платежа из интерфейса биллинга.

Модуль Netpay

После установки модуля создаем конфигурацию, заполняем параметры, сохраняем конфигурацию, делаем ее активной.

Список параметров модуля с описанием каждого параметра доступен в редакторе параметров конфигурации по кнопке **Шаблон**.

Тип	Наименование	Значение
Число	netpay.sum.min	100
Число	netpay.sum.max	5000
Строка	netpay.gate.url	https://demo.net2pay.ru/billingService/paypage/
Строка	netpay.api.key	js4cucpn4kkc6j1p95np054g2
Строка	netpay.auth.signature	1
Строка	netpay.expired.day	30
Строка	netpay.order.description	Пополнение счета по договору №{CONTRACT_TITLE}
Строка	netpay.order.success.url	http://192.168.184.33:8089/bgbilling/webexecuter?action=DoTransaction&mid...
Строка	netpay.order.fail.url	http://192.168.184.33:8089/bgbilling/webexecuter?action=DoTransaction&mid...
Строка	netpay.payment.comment	Оплата через платежный сервис NetPay [транзакция #{TRANSACTION_ID}]
Список	netpay.mnt.currency.code	RUB
Список	netpay.status.default	
Число	netpay.payment.type.id	54
Строка	web.menuitem1	Оплата через NetPay

Для оповещения о результатах оплаты, в Личном кабинете на сайте платежной системы нужно указать URL, по которому будет проходить оповещения.

Данный URL имеет следующий формат: **http(s)://host[:port]/bgbilling/netpay/<код_модуля_netpay>**

После настройки модуля, добавления модуля в договор, в Личном кабинете клиентов будет доступен для оплаты и просмотра транзакций следующая форма:

Оплата через NetPay

Платежный сервис NetPay

Сумма (от 100,00 до 5 000,00):

Текущий баланс: 99 878 861,1 руб.

Период: < Сентябрь, 2016 > Всего записей: 2 Страница: << < 1 из 1 [25] > >> [показать допол. фильтры]

№	Дата/Время оплаты	Статус	Сумма
7	01.09.2016 16:34:14	Завершена	107,00
6	01.09.2016 15:35:48	Новая	106,00

Модуль OnPay

Назначение модуля

Платежный модуль OnPay предназначен для осуществления безопасного приема платежей от абонентов. Более подробную информацию о сервисе, размере комиссии можно прочитать на сайте платежного сервиса OnPay (<http://onpay.ru>).

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.

Для возможности оплачивать через данный сервис необходимо добавить модуль на договор. После чего в личном кабинете будет доступен соответствующий пункт меню и страница оплаты.

Модуль PayKeeper

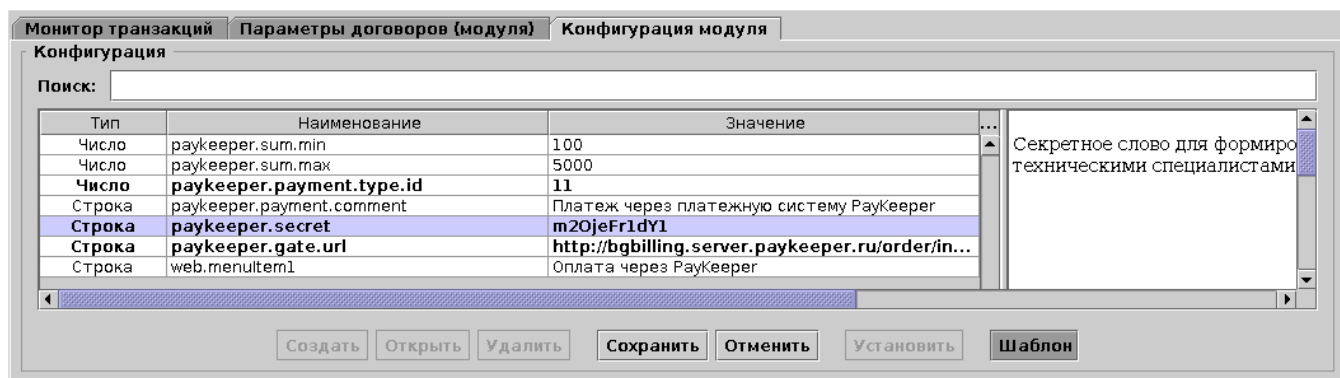
Назначение модуля

Модуль предназначен для интеграции биллинга с платежной системой PayKeeper.ru

Настройка модуля

После установки модуля создаем конфигурацию, заполняем параметры, сохраняем конфигурацию, делаем ее активной.

Список параметров модуля с описанием каждого параметра доступен в редакторе параметров конфигурации по кнопке **Шаблон**.

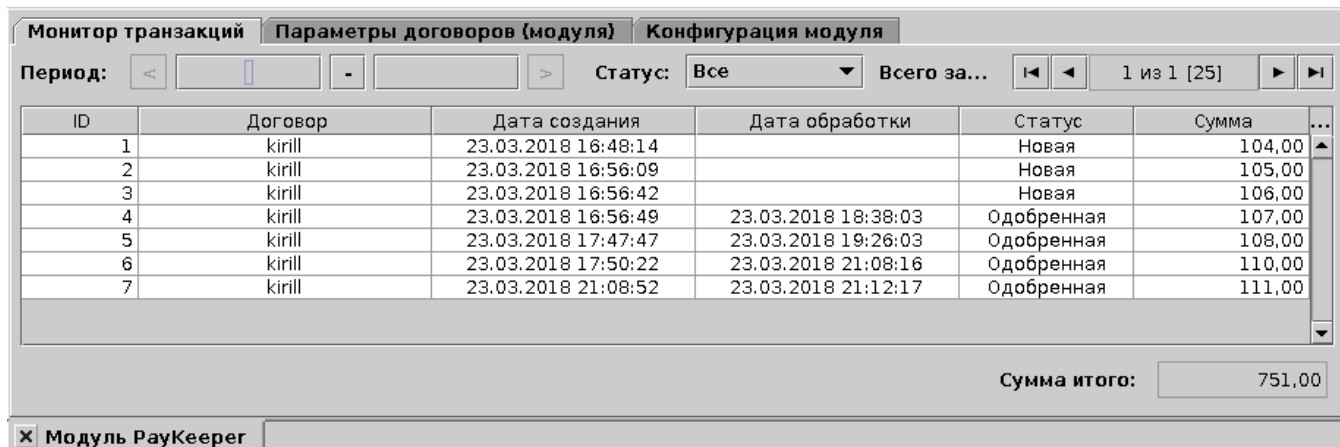


Для оповещения о результатах оплаты, в Личном кабинете на сайте платежной системы нужно указать URL, по которому будет проходить оповещения.

Данный URL имеет следующий формат: `http(s)://host[:port]/bgbilling/paykeeper/<код_модуля_paykeeper>`

Мониторинг транзакций

В клиенте биллинга доступен монитор всех транзакций



В карточке договора доступен список транзакций по данному договору

Транзакции

Период: 01.01.2018 - Статус: Все Всего записей: 7 1 из 1 [25]

ID	Дата создания	Дата обработки	Статус	Сумма
1	23.03.2018 16:48:14		Новая	104,00
2	23.03.2018 16:56:09		Новая	105,00
3	23.03.2018 16:56:42		Новая	106,00
4	23.03.2018 16:56:49	23.03.2018 18:38:03	Одобренная	107,00
5	23.03.2018 17:47:47	23.03.2018 19:26:03	Одобренная	108,00
6	23.03.2018 17:50:22	23.03.2018 21:08:16	Одобренная	110,00
7	23.03.2018 21:08:52	23.03.2018 21:12:17	Одобренная	111,00

Сумма итогов: 751,00

Интерфейс оплаты в Личном кабинете

После настройки модуля, в Личном кабинете клиентов будет доступна для оплаты и просмотра транзакций следующая форма:

Оплата через PayKeeper

Платежный сервис PayKeeper

Сумма (от 100,00 до 5 000,00):

Текущий баланс:
99 895 227,1 руб.

Период: < Март, 2018 > Всего записей: 7 Страница: << < 1 из 1 [25] > >>

Дата	Статус	Сумма
23.03.2018 16:48:14	Новая	104
23.03.2018 16:56:09	Новая	105
23.03.2018 16:56:42	Новая	106
23.03.2018 18:38:03	Одобренная	107
23.03.2018 19:26:03	Одобренная	108
23.03.2018 21:08:16	Одобренная	110
23.03.2018 21:12:17	Одобренная	111

Модуль Paylinks

Назначение модуля Paylinks

Модуль предназначен для предоставления пользователям личного кабинета способов оплаты через внешние платёжные системы с гибкой настройкой. Модуль похож на другие подобные модули, реализующие оплату через разные системы, но в отличие от них не предоставляет обработчика ответов от удалённой системы, помимо общих ответов, вроде "успешно", "неуспешно" и других. Остальное, например, некоторая отчётность по оплатам (общая и для договора) и пр. — присутствует.

Также модуль может предоставлять любое количество способов оплаты через любое количество платёжных систем. Обработка успешных оплат и, как следствие, занесение платежа в систему может проводиться либо существующим внешним обработчиком, либо вручную, либо через модуль `mps`, либо любым другим способом.

Модуль может также использоваться для любого договора-зависимого предоставления ссылок web-пользователю, например, переходов в какой-то веб-магазин и пр.

Настройка модуля Paylinks

Все методы конфигурируются в конфигурации модуля. Количество методов неограничено. Большинство из параметров метода параметризуются через макросы вида `{$var}`, количество которых, теоретически, неограничено.

В общем виде конфигурация модуля выглядит так (с подробными комментариями), используется реальный пример со шлюзом газпромбанка:

Конфигурация

```
# .
# ( title ) :
# 1) {$sum} - , .
# description .
# url confirm : , .
# result.* : url "sum", url .
# 2) {$contract.getTitle} - , .
# getTitle - , get- Contract (. API), "contract." - .
# :
# {$contract.*} - .
# : {$contract.getTitle} - , {$contract.getId} - (cid).
# 3) {$action}, {$mid}, {$module} - url, url.
# , ( , module=paylinks&mid=666&action=Paylinks).
#
# 1
# .
method.1.title= VISA, MasterCard
# url, .
# url, : http://webexecutor?mid={$mid}&module={$module}&action={$action}
&operation=result&result=success
# result , , &result=success .., .
# ( method.1.result.*), "sum" url.
# url methodId=1 ( ), .
# .., url , , .
# url - .
back_url_s=http://127.0.0.1:8080/bgbilling/webexecutor?mid={$mid}%26module={$module}%26action={$action}%
26operation=result%26result=success%26sum={$sum}%26methodId=1
back_url_f=http://127.0.0.1:8080/bgbilling/webexecutor?mid={$mid}%26module={$module}%26action={$action}%
26operation=result%26result=failed%26sum={$sum}%26methodId=1
method.1.url=https://www.pps.gazprombank.ru:443/payment/start.wsm?merch_id=xxx&o.account={$contract.getTitle}&o.
sum={$sum}&back_url_s=@back_url_s&back_url_f=@back_url_f
# , .
method.1.description= gazprombank, {$contract.getTitle}. : {$contract.getId}.
# , .
method.1.confirm= gazprombank {$contract.getTitle} {$sum}?
# , .
# url ( )
# (&result=anything).
method.1.result.success=, {$sum} {$contract.getTitle}
method.1.result.failed=, {$sum} {$contract.getTitle}
#
# -
method.2.title=Test
...
```

Обратите внимание, для удобства вынесены url успеха и неуспеха в отдельные параметры, а затем вставлены в основной параметр `method.1.url`.

Также обратите внимание, что в случае необходимости (например, при передаче через url других адресов возврата) экранировать символы надо заранее, и прописывать в конфигурацию уже экранированное, как, например, здесь `%26` вместо `&`.

В данный момент существует возможность перехода формы только с методом GET.

Использование модуля Paylinks

При добавлении модуля на договор в web-интерфейсе статистики появляется пункт меню "Пополнить счёт", внутри которого перечислены основные типы оплаты, настроенные в данном модуле.

The screenshot displays a web interface for a contract named "Договор № dimon". At the top, there is a blue header with the contract name. Below it, a dark blue bar indicates "Супердоговор: dimon". On the left side, there is a vertical menu titled "Меню" with various options, each preceded by a right-pointing arrow. The "Пополнение счёта" option is highlighted. On the right side, a blue header titled "Пополнение счёта" is visible, followed by a list of payment methods: "Карты VISA, MasterCard" and "Test Тест". A mouse cursor is positioned over the "Пополнение счёта" menu item.

Договор № dimon	
Супердоговор: dimon	
Меню	Пополнение счёта
»» Новости	• Карты VISA, MasterCard
»» Уведомления	• Test Тест
»» Просмотр баланса	
»» Смена пароля на доступ к статистике	
»» Смена тарифных планов	
»» Тарифные опции	
»» Карточки	
»» Управление лимитом	
»» Управление статусом	
»» Дополнительные действия	
»» Примечания	
»» Оплата через Assist.Ru	
»» Оплата через BVCom	
»» Управление подпиской SerberCrypt	
»» Виртуальный кинотеатр SerberCrypt	
»» Смена пароля на карте	
»» Магазин Epaza	
»» Пополнение счёта	

После выбора каждого из методов оплаты открывается страница с названием, описанием, а также приглашением ко вводу необходимой суммы. Все параметры гибко настраиваются в конфигурации модуля.

Договор № dimon

Супердоговор: dimon

Меню

- Новости
- Уведомления
- Просмотр баланса
- Смена пароля на доступ к статистике
- Смена тарифных планов
- Тарифные опции
- Карточки
- Управление лимитом
- Управление статусом

Пополнение счёта — Карты VISA, MasterCard.

Метод оплаты: Карты VISA, MasterCard
Описание: Оплата через gazprombank, на счёт dimon. Тест 352715.
Введите сумму:

[Далее](#) [Отмена](#)

После этого последует переход на внешнюю платёжную систему по сформированному по шаблону url. После возврата в личный кабинет выведется одно из настроенных сообщений об успехе или неуспехе.

Договор № dimon

Супердоговор: dimon

Меню

- Новости
- Уведомления
- Просмотр баланса
- Смена пароля на доступ к статистике
- Смена тарифных планов
- Тарифные опции
- Карточки
- Управление лимитом
- Управление статусом
- Дополнительные действия

Операция завершена

Плохо, платёж на сумму 1 не прошёл на счёт dimon

Все успешные и неуспешные переходы логируются в модуле, являясь по сути логом переходов по ссылкам, а не сколько-нибудь достоверной отчётностью по оплатам.

Лог переходов **Конфигурация модуля**

с по Применить 1 из 1 [25]

Код	Метод	Договор	Комментарий договора	Время	Сумма	Полный url
101	Карты VISA, Mas...	dim on	тестовый договор	03.09.2011 00:00:00	10.00	https://www.pps.gazprombank.ru:443/payme...
102	Карты VISA, Mas...	dim on	тестовый договор	05.09.2011 00:00:00	10.00	https://www.pps.gazprombank.ru:443/payme...
103	Карты VISA, Mas...	dim on	тестовый договор	07.09.2011 00:00:00	111.00	https://www.pps.gazprombank.ru:443/payme...
100	Карты VISA, Mas...	x0000	ФИО - Иванов Иван Иванович; сп...	07.09.2011 18:30:25	10.00	https://www.pps.gazprombank.ru:443/payme...
105	Карты VISA, Mas...	dim on	тестовый договор	08.09.2011 16:33:59	999.00	https://www.pps.gazprombank.ru:443/payme...
106	Карты VISA, Mas...	dim on	тестовый договор	08.09.2011 17:06:23	1.00	https://www.pps.gazprombank.ru:443/payme...
104	Карты VISA, Mas...	dim on	тестовый договор	09.09.2011 17:47:23	10.00	https://www.pps.gazprombank.ru:443/payme...
107	Карты VISA, Mas...	dim on	тестовый договор	12.09.2011 18:44:46	1.00	https://www.pps.gazprombank.ru:443/payme...
108	Карты VISA, Mas...	dim on	тестовый договор	14.12.2011 16:34:34	1.00	https://www.pps.gazprombank.ru:443/payme...

Модуль PayLinks

В договоре также можно посмотреть подобную статистику.

Параметры Отчет Карточки HelpDesk Рассылки CRM Документы

Договор

dimon [тестовый договор] 352715 Период с 04.02.2009 по

с по Применить 1 из 1 [25]

Код	Метод	Договор	Комментарий договора	Время	Сумма	Полн...
101	Карты VISA, M...	dim on	тестовый договор	03.09.2011 00:00:...	10.00	https://...
102	Карты VISA, M...	dim on	тестовый договор	05.09.2011 00:00:...	10.00	https://...
103	Карты VISA, M...	dim on	тестовый договор	07.09.2011 00:00:...	111.00	https://...
105	Карты VISA, M...	dim on	тестовый договор	08.09.2011 16:33:...	999.00	https://...
106	Карты VISA, M...	dim on	тестовый договор	08.09.2011 17:06:...	1.00	https://...
104	Карты VISA, M...	dim on	тестовый договор	09.09.2011 17:47:...	10.00	https://...
107	Карты VISA, M...	dim on	тестовый договор	12.09.2011 18:44:...	1.00	https://...
108	Карты VISA, M...	dim on	тестовый договор	14.12.2011 16:34:...	1.00	https://...

Assist.Ru
BVCom
CerberCrypt2
Enaza
Взлом
CerberCrypt2
Enaza
PayLinks
TrayInfo
Бухгалтерия
Группы
dimon
Web
Скрытый

Поиск договоров dimon

Модуль Paymaster

Назначение модуля

Модуль Paymaster предназначен для проведения платежей через платежную систему Paymaster с использованием пластиковых карт. Для проведения платежей Вашими клиентами у Вас должен быть заключен договор с системой.

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Оплата через PayMaster

Пополните счет с помощью сервиса PayMaster

Сумма:

Текущий баланс:
10 078,34 руб.

Период:

Дата	Сумма
Записи не найдены.	

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, введите требуемые параметры. После этого сохраните конфигурацию и сделайте её активной.

Монитор транзакций **Конфигурация**

Конфигурация

Поиск:

Тип	Наименование	Значение
Число	paymaster.sum.min	100
Число	paymaster.sum.max	5000
Строка	paymaster.secret	
Список	paymaster.digest.type	SHA-256
Строка	paymaster.merchant.id	
Строка	paymaster.currecy	RUB
Строка	paymaster.payment.confirmation.url	
Строка	paymaster.payment.notification.url	
Строка	paymaster.success.url	
Строка	paymaster.failure.url	
Строка	paymaster.payment.desc	Пополнение счета по договору №{CONTRACT_TITLE}
Строка	paymaster.payment.desc.base64	Пополнение счета по договору №{CONTRACT_TITLE}
Строка	paymaster.payment.comment	Платеж через платежную систему Paymaster ({PAYMENT_SYS...
Число	paymaster.payment.type.id	0
Число	paymaster.sim.mode	0
Строка	paymaster.payer.email	
Строка	paymaster.direct.secret	
Строка	paymaster.direct.scope	
Строка	paymaster.direct.redirect.uri	
Логический	paymaster.autopayment.enable	false
Число	paymaster.autopayment.mode.default	0
Логический	paymaster.autopayment.mode.week.enable	false
Число	paymaster.autopayment.mode.week.day.default	1
Число	paymaster.autopayment.mode.week.sum.min	100
Число	paymaster.autopayment.mode.week.sum.max	3000
Число	paymaster.autopayment.mode.week.sum.default	100
Логический	paymaster.autopayment.mode.month.enable	false
Число	paymaster.autopayment.mode.month.day.default	1
Число	paymaster.autopayment.mode.month.sum.min	100
Число	paymaster.autopayment.mode.month.sum.max	10000
Число	paymaster.autopayment.mode.month.sum.default	100
Логический	paymaster.autopayment.mode.auto.enable	false
Число	paymaster.autopayment.mode.auto.day.sum.max	3000
Число	paymaster.autopayment.mode.auto.day.sum.default	500
Число	paymaster.autopayment.mode.auto.week.sum.max	5000
Число	paymaster.autopayment.mode.auto.week.sum.default	1000
Число	paymaster.autopayment.mode.auto.month.sum.max	10000
Число	paymaster.autopayment.mode.auto.month.sum.defa...	3000
Строка	web.menuitem1	Оплата через PayMaster

Справка по данному параметру отсутствует

Создать Открыть Удалить Сохранить Отменить Установить Шаблон

Модуль PayMaster

Модуль имеет встроенную задачу планировщика для периодических автоплатежей (необходимо настроить если используются периодические автоплатежи).

Оплата через систему

Для предоставления возможности клиенту платить банковской картой через платежный шлюз **Paymaster** необходимо подключить данный модуль к договору клиента. В Web-интерфейсе клиента появится новый пункт в меню - **Оплата через Paymaster** (название по умолчанию).

Оплата через PayMaster

Пополните счет с помощью сервиса PayMaster

Сумма:

Текущий баланс: 10 078,34 руб.

Период: Всего записей: 0 Страница:

Дата	Сумма
Записи не найдены.	

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль **Paymaster** в дереве параметров договора. Для просмотра ВСЕХ платежей, проведенных через систему Paymaster, существует глобальный монитор в параметрах модуля. В открывшейся вкладке **Проведенные платежи** модуля **Paymaster** у Вас есть возможность просмотреть платежи с учетом фильтра по периоду, совершенные вашими абонентами.

Монитор транзакций		Конфигурация			
Период:	< 19.09.2009 - >	Статус:	Все		
Paymaster ID:	0	Всего записей:	19		
		1 из 1 [25]			
ID	Договор	ID транзакции	Дата обработки	Статус	Сумма
1		0		Не проведенная	1,26
2		0		Не проведенная	1,00
3		0		Не проведенная	2,00
4		0		Не проведенная	3,00
5		0		Не проведенная	12,00
6		0		Не проведенная	15,25
7		0		Не проведенная	12,00
8		0		Не проведенная	10,00
9		0		Не проведенная	1,00
10		0		Не проведенная	100,00
11		0		Не проведенная	1,00
12		0		Не проведенная	100,00
13		0		Не проведенная	100,00
14		0		Не проведенная	1,00
15		0		Не проведенная	10,00
16		0		Не проведенная	14,00
17		2939892	19.09.2012 17:50:38	Проведенная	0,15
18		2940816	19.09.2012 19:21:43	Проведенная	0,16
19		2941250	19.09.2012 20:18:00	Проведенная	1 000,11
Сумма итого:					1 383,93

По двойному клику левой кнопкой мыши на строке в таблице открывается соответствующий договор.

Модуль PayOnline

Назначение модуля

Модуль PayOnline предназначен для проведения платежей через платежную систему PayOnline с использованием пластиковых карт. Для проведения платежей Вашими клиентами у Вас должен быть заключен договор с системой.

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, введите требуемые параметры. После этого сохраните конфигурацию и сделайте её активной.

Конфигурация

```
# ,
contract.active.status.codes=

# ( PayOnline)
merchant.id=

# ( PayOnline)
private.security.key=

#URL, ( "/" )
pay.online.url=https://secure.payonlinesystem.com/ru/payment

# ,
payment.type.id=
# ,
refund.charge.typeid=
#
refund.charge.comment

# (4 ) rebillAnchor .
# : true - , false - .
payonline.store.data=false

#URL ,
redirect.url=

#
min.summa=100

#
max.summa=3000

#
#thread.count=20

# ( , , , )
check.org.header=
check.org.inn=
check.org.address=
check.org.phone=

# ( )
pdf.check.title.1.regex=^Pech+$
pdf.check.title.1.title=Good

#
# web-
rebill=true
#URL,
rebill.url=https://secure.payonlinesystem.com/payment/transaction/rebill
```

```

#
#
usual.comment=" "
#
rebill.comment=""

# . .
rebillUnique=
#=====
#. : 0 - ; 1 - ; 2 -
commission.type=0;
# 0-100 %
commission.percent=2
#
commission.comment= Payonline
#
commission.charge.type.id=
#
refund.payment.typeid=
#
refund.charge.comment= Payonline

```

Замечания:



1. Прежде, чем задавать [payment.type.id](#) необходимо создать соответствующий платеж в **Справочники->Другие->Типы платежей**
2. После заключения договора с системой PayOnline в их личном кабинете необходимо задать callback URL, который ждет результаты от платежной системы. URL будет выглядеть следующим образом: http://<адрес_машины_биллинга>/payonline/<mid>. Например, если у вас биллинг находится по адресу <http://billing.example.com/bgbilling/> и модуль PayOnline имеет mid=16, то результирующий URL, который нужно дать компании PayOnline, выглядит следующим образом: <http://billing.example.com/bgbilling/payonline/16>.
3. Чтобы использовать функционал автоплатежа необходимо заключить доп. соглашение с PayOnline. И в планировщик заданий добавить задачу **Автоплатеж(Rebill)**. В параметрах запуска задачи нужно указать код модуля PayOnline и время запуска 1 раз в сутки.

Оплата из Личного кабинета

Для предоставления возможности клиенту платить банковской картой через платежный шлюз **PayOnline** необходимо подключить данный модуль к договору клиента. В Web-интерфейсе клиента появится новый пункт в меню - **Оплата банковской картой** (название по умолчанию).

Оплата банковской картой

Пополните счет с помощью сервиса **PayOnline**

Сумма (от 100.00 до 3000.00): |

Текущий баланс:
284 945,00 руб.

Автоплатеж

На Вашем договоре с **16.08.2012** активирован автоплатеж на сумму **1.00 руб.**, который будет проводиться каждый месяц **4-го** числа.

Период:

Июль, 2014

Всего записей: **0** Страница:

1 из 1 [25]

Дата	Сумма	Номер
Записи не найдены.		

Простой платеж

Для совершения простого платежа необходимо ввести необходимую сумму в поле Сумма и нажать кнопку **Оплатить**. Далее клиент попадает на страницу подтверждения оплаты. На данном этапе уже сформирован запрос на платежный шлюз и от клиента требуется подтвердить оплату, либо **Отмена** для отмены платежа.

В случае, если клиент подтверждает оплату, то он перенаправляется на платежный шлюз, где может ввести информацию о своей пластиковой карте. Правильно заполнив предложенную форму и нажав кнопку **Оплатить**, клиент будет уведомлен о результате платежа.

В случае положительного результата биллинговая система начислит клиенту указанную им сумму на счет.

Автоплатеж

Данный функционал позволяет выполнять повторные транзакции без непосредственного участия плательщика ежемесячно. Как и остальные программные интерфейсы, Автоплатеж работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Для одного договора может быть активирован только 1 автоплатеж. Если необходимо изменить параметры автоплатежа после успешной активации, то нужно сначала удалить автоплатеж, а затем активировать новый.

Для того, чтобы активировать автоплатеж, необходимо ввести сумму автоплатежа в поле и указать число месяца автоплатежа. После заполнения всех полей необходимо нажать кнопку Активировать, после чего будет произведена стандартная процедура платежа.

Автоплатеж

Вы можете активировать функцию автоплатежа. После активации функции с вашей карты будет в определенный ниже день месяца будет списывать указанная ниже сумма. В дальнейшем в любой момент время Вы можете деактивировать данную функцию. **ВНИМАНИЕ!!!** Для активации автоплатежа необходимо будет совершить первый платеж с карты.

Сумма (от 100.00 до 3000.00):Число месяца:**Активировать**

Для активации функции автоплатежа, укажите сумму и число месяца в который Вы хотите, что бы данная сумма списывалась с вашей карты.

Автоплатеж осуществляется периодической задачей планировщика **Автоплатеж (Rebill)**. Для корректной работы задачи необходимо указать в конфигурации модуля параметр **contract.status.active.codes=<перечень кодов через запятую>**. Таким образом, не будут проводиться автоплатежи по закрытым договорам и договорам, имеющим неактивный статус.

Для того, чтобы удалить автоплатеж, необходимо нажать кнопку Отменить автоплатеж.

Монитор транзакций

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль **PayOnline** в дереве параметров договора. Здесь присутствует фильтр по типу платежа (все/только автоплатежи/только обычные) с указанием периода, когда производилась оплата. Также отображается состояние автоплатежа, если он активирован. Оператор биллинга может только удалить автоплатеж с договора, если клиент по каким-либо причинам не может это сделать самостоятельно, сняв соответствующее выделение **автоплатеж**.

Фильтр			
Показать за период с <input type="text" value="05.04.2011"/> по <input type="text"/>		тип платежа: <input type="text" value="Все"/>	<input type="button" value="Ok"/>
<input checked="" type="checkbox"/> автоплатеж	На договоре с 05.03.2013 активирован автоплатеж на сумму 100.00руб. , который будет проводиться каждый месц 15-го числа.		
Код транзакции	Дата		Сумма
3601608	28.11.2011 06:56:15		425.00
3601607	28.10.2011 06:56:15		425.00

Для просмотра ВСЕХ платежей, проведенных через систему PayOnline, существует глобальный монитор в параметрах модуля. В открывшейся вкладке **Проведенные платежи** модуля **PayOnline** у Вас есть возможность просмотреть платежи с учетом фильтра по типу платежа и указанному периоду, совершенные вашими абонентами. На вкладке модуля **Ошибки автоплатежей** выводятся ошибки, которые были получены от системы Payonline по автоплатежу для каждого договора.

Проведенные платежи				
Фильтр				
Показать за период с		05.10.2011	по	
			тип платежа:	Все
Ok				
Код транзакции	Договор	Комментарий договора	Дата	Сумма
3601607	kostya1		28.10.2011 06:56:15	425.00
3601608	kostya1		28.11.2011 06:56:15	425.00

По двойному клику левой кнопкой мыши на строке в таблице открывается соответствующий договор.

Сверка транзакций

Существует возможность произвести сверку платежей, используя выгружаемые из системы Payonline csv-файлы, сформированные за определенный период. Сверка осуществляется через меню Модули->Payonline->Сверка платежей.

После проведения сверки на вкладке Недостающие платежи появятся платежи, которые есть в csv-файле и отсутствуют в биллинге. На вкладке Лишние платежи - отсутствующие в csv-файле, но присутствующие в базе биллинга. Существует возможность добавить один и более недостающих платежей, используя контекстное меню на таблице недостающих платежей, либо удалить один и более лишних платежей на соответствующей вкладке.

BGBillingClient				
Договор Модули Плагины Справочники Сервис Утилиты Справка				
			БД: local	Логин: artur 04 Июль 18:08
Проведенные платежи				
Файл сверки				
Период		insactions20140616-20140622.csv		
с 16.06.2014 по 22.06.2014		Сверить		
Отсутствующие платежи				
ID в биллинге	ID в ПС	Договор	Сумма	Дата/Время платежа...
42	25901370	artur_test	200	17.06.2014 09:28:37
Всего платежей 1 на сумму 200.00				

Модуль PayOnline

Модуль Payture

Назначение модуля Payture

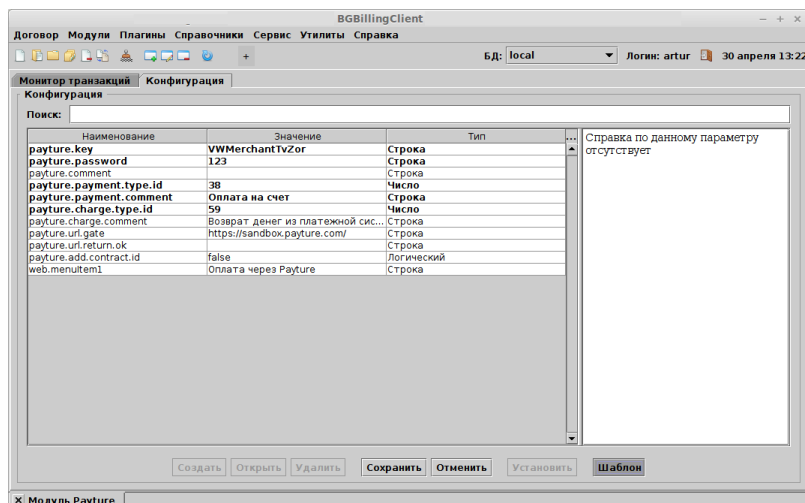
Платежный модуль Payture предназначен для осуществления безопасного приема платежей от абонентов банковскими картами Visa /MasterCard. Более подробную информацию о сервисе, размере комиссии можно прочитать на сайте платежного сервиса Payture (<http://payture.com/>).

Настройка модуля Payture

Скачайте модуль payture с нашего сервера, установите на сервере с помощью утилиты `bg_installer.sh (.bat)`.

В редакторе модулей и услуг создайте экземпляр модуля payture.

Для дальнейшей работы модуля необходимо создать конфигурацию модуля, нажать кнопку **Шаблон** и заполнить требуемые параметры. Описание параметров представлено справа.



Для получения статуса платежей необходимо передать в Payture адрес обработчика биллинга, который формируется следующим образом:

`http://host[:port]/bgbilling/payture/<mid>`

Проведение платежей Payture

Для совершения платежа необходимо перейти в личный кабинет, выбрать пункт меню, соответствующий платежной системе Payture. Перед абонентом появится окно следующего содержания:

Оплата через платежный сервис Payture

Пополните счет с помощью сервиса Payture

Сумма:

Текущий баланс:
284 945,00 руб.

Период:

Страница:

[\[показать допол. фильтры\]](#)

ID	Дата/время оплаты	Сумма	Статус
----	-------------------	-------	--------

В самом низу пользователь увидит свой текущий баланс, а также список всех своих платежей с фильтрацией по периоду и статусу (необходимо нажать на **показать допол. фильтры**). Для оплаты ему необходимо ввести сумму в поле **Сумма**, а затем нажать кнопку **Оплатить**. После он будет перенаправлен на сайт сервиса, где сможет ввести данные своей карты. После оплаты абонент будет перенаправлен на страницу, указанную в конфигурации модуля.

Мониторинг платежей Payture

Для просмотра общего списка платежей по всем договорам в системе существует Монитор транзакций, который доступен через меню **Модули->Экземпляр Payture ->Монитор транзакций**.

ID	Дата платежа	Договор	Статус	Сумма
21	14.08.2013 13:10:34	artur_test	Успешный платеж	10,00
23	14.08.2013 14:18:16	artur_test	Успешный платеж	12,00
24	14.08.2013 14:24:43	artur_test	Успешный платеж	12,00
26	19.08.2013 14:49:24	artur_test	Успешный платеж	10,00

Платежи можно отфильтровать по названию договора, по статусу и периоду.

Также можно отслеживать платежи по каждому договору в отдельности. Для этого необходимо открыть карточку договора, в списке модулей выбрать экземпляр модуль Payture. Справа появится окно со списком платежей договора.

ID	Дата платежа	Статус	Сумма
21	14.08.2013 13:10:34	Успешный платеж	10,00
23	14.08.2013 14:18:16	Успешный платеж	12,00
24	14.08.2013 14:24:43	Успешный платеж	12,00
26	19.08.2013 14:49:24	Успешный платеж	10,00

Возврат платежей Payture

В модуле существует возможность возврата платежей. Для этого необходимо выбрать в клиенте биллинга платеж (или в мониторе транзакций, или в карточке договора), нажать правую кнопку мыши и в появившемся контекстном меню выбрать пункт меню **Вернуть платеж**. После одобрения возврата на договор занесется расход с типом, указанным в конфигурации модуля, и суммой, равной сумме платежа. После этого можно отфильтровать возвращенные платежи с помощью фильтра статуса.

Модуль PSB (Промсвязьбанк)

Назначение модуля

Модуль PSB предназначен для оплаты картами через процессинг Промсвязьбанка.

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.

Проведенные платежи Конфигурация модуля

Конфигурация

Поиск:

Наименование	Значение	Тип	...
psb.back.url	http://billing.bitel.ru/webexecuter?action=PSB&module=psb&mid=216	Строка	▲
psb.description	Red Book	Строка	
psb.email	lakhtin@psbank.ru	Строка	
psb.gate.url	http://193.200.10.117:8080/cgi-bin/cgi_link	Строка	
psb.merchant	790367686219999	Число	
psb.merchant.name	TEST_MERCH	Строка	
psb.payment.type.id	43	Число	
psb.secret	C50E41160302E0F5D6D59F1AA3925C45	Строка	
psb.terminal	79036847	Число	
web.menuitem1	Оплата через ПСБ	Строка	

Создать Открыть Удалить Сохранить Отменить Установить Шаблон

Модуль PSB (ПромСвязьБанк)

Модуль PSCB (АО «Петербургский социальный коммерческий банк»)

После установки модуля создаем конфигурацию, заполняем параметры, сохраняем конфигурацию, делаем ее активной.

Список параметров модуля с описанием каждого параметра доступен в редакторе параметров конфигурации по кнопке **Шаблон**.

В Личном Кабинете банка должен быть настроен адрес по которому сервер банка будет оповещать биллинг о проведенных платежах, типа

```
http[s]://host[:port]/bgbilling/pscbexecuter/<__pscb>
```

Модуль Qiwi

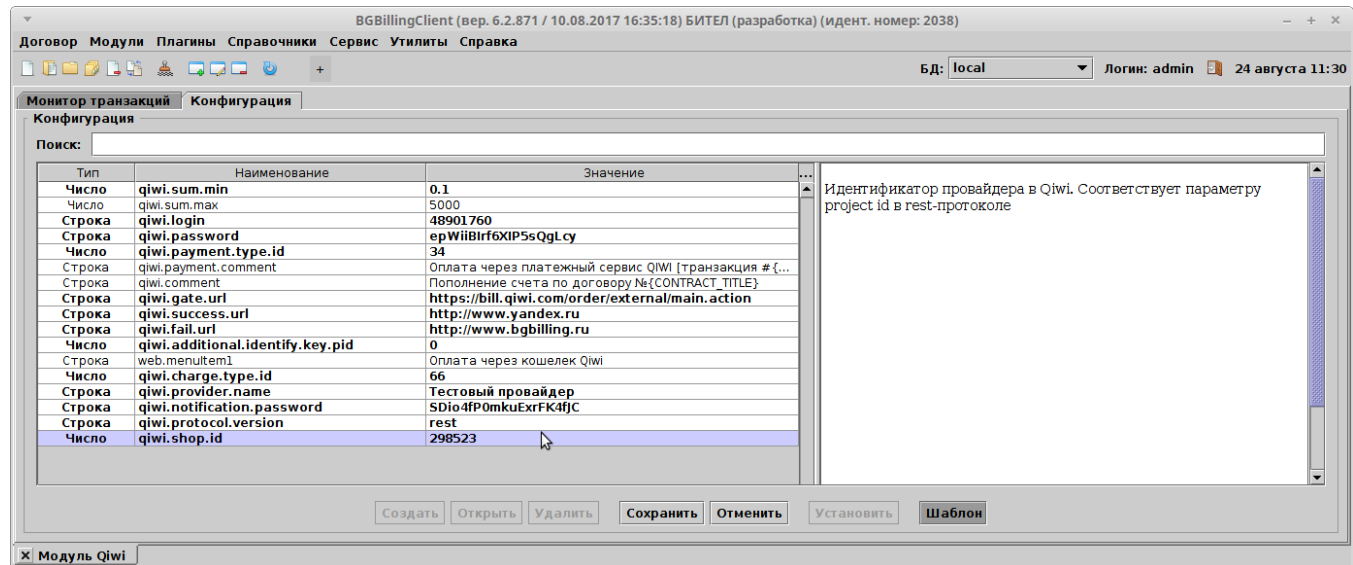
Назначение модуля

Модуль биллинга Qiwi предназначен для проведения платежей с использованием механизма выставления счетов и оплаты их с помощью кошелька в системе Qiwi. Для проведения платежей вашими абонентам у вас должен быть заключен договор с данной платежной системой.

Обратите внимание, что для приема платежей через терминалы Qiwi достаточно [модуля MPS](#).

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию.



После этого сохраните конфигурацию и сделайте её активной.

Модуль поддерживает 2 протокола взаимодействия с платежной системой: soap (устарел) и rest. Настройка используемого протокола осуществляется в ЛК Qiwi, а также в конфигурации биллинга с помощью параметра **qiwi.protocol.version**.

Замечания:

1. Прежде, чем задавать **qiwi.payment.type.id**, необходимо создать соответствующий тип платежа в Справочнике (**Справочники->Другие->Типы платежей**).
2. Прежде, чем задавать **qiwi.charge.type.id**, необходимо создать соответствующий тип расхода в Справочнике (**Справочники->Другие->Типы расходов**). Обращаем внимание, что возврат платежей поддерживается только в протоколе REST!
3. Номер транзакции создается следующим образом: берется ID транзакции из таблицы **qiwi_transaction_mid** и соединяется с шаблоном. Например: если шаблон "BG0000", а ID пусть будет 34, тогда номер транзакции, отсылаемый в Qiwi, будет иметь вид: **BG0034**.
4. **При использовании протокола SOAP.** После заключения договора с системой необходимо зайти в свой личный кабинет провайдера на стороне Qiwi и в настройках подключения в разделе SOAP ввести адрес веб-сервиса на стороне биллинга, на который будет приходить информация по статусу счета. Этот адрес формируется следующим образом: **http://<адрес_машины_биллинга>/bgbilling/qiwiexecuter/ru.bitel.bgbilling.modules.qiwi/<код_модуля_Qiwi>/IShopClientWS**. Например, если у вас биллинг находится по адресу **http://billing.example.com/bgbilling/** и модуль Qiwi имеет mid=16, то результирующий URL, который нужно ввести в личном кабинете, выглядит следующим образом: **http://billing.example.com/bgbilling/qiwiexecuter/ru.bitel.bgbilling.modules.qiwi/16/IShopClientWS**.
5. **При использовании протокола REST.** В ЛК Qiwi в разделе настройки REST-протокола необходимо ввести URL для уведомлений, который выглядит следующим образом: **http://<адрес_машины_биллинга>/bgbilling/qiwiexecuter/rest/<код_модуля_Qiwi>**
6. **При использовании протокола REST.** Параметр **qiwi.notification.password** используется для авторизации запросов-уведомлений о состоянии транзакций.
7. Проверка статуса транзакции, отмена и возврат поддерживаются **только** в протоколе REST!
8. При использовании протокола REST параметр **qiwi.shop.id** соответствует полю **project_id** в ЛК Qiwi.

Оплата через кошелек

Если у клиента подключен экземпляр модуля в дереве договора, то он может осуществлять оплату через свой Qiwi-кошелек, используя личный web-интерфейс.

В личном кабинете на странице отображается история платежей, совершенных клиентом:

Оплата через кошелек Qiwi

Пополнение счета с помощью электронного кошелька Qiwi

Сумма (от 100.00 до 5000.00):

Телефон:

Поле телефон заполняется без префикса. Пример: 9123459878.

Текущий баланс:
99 877 775,1 руб.

Период: < Октябрь, 2015 > Всего записей: 1 Страница: << < 1 из 1 [25] > >>

[показать допол. фильтры]

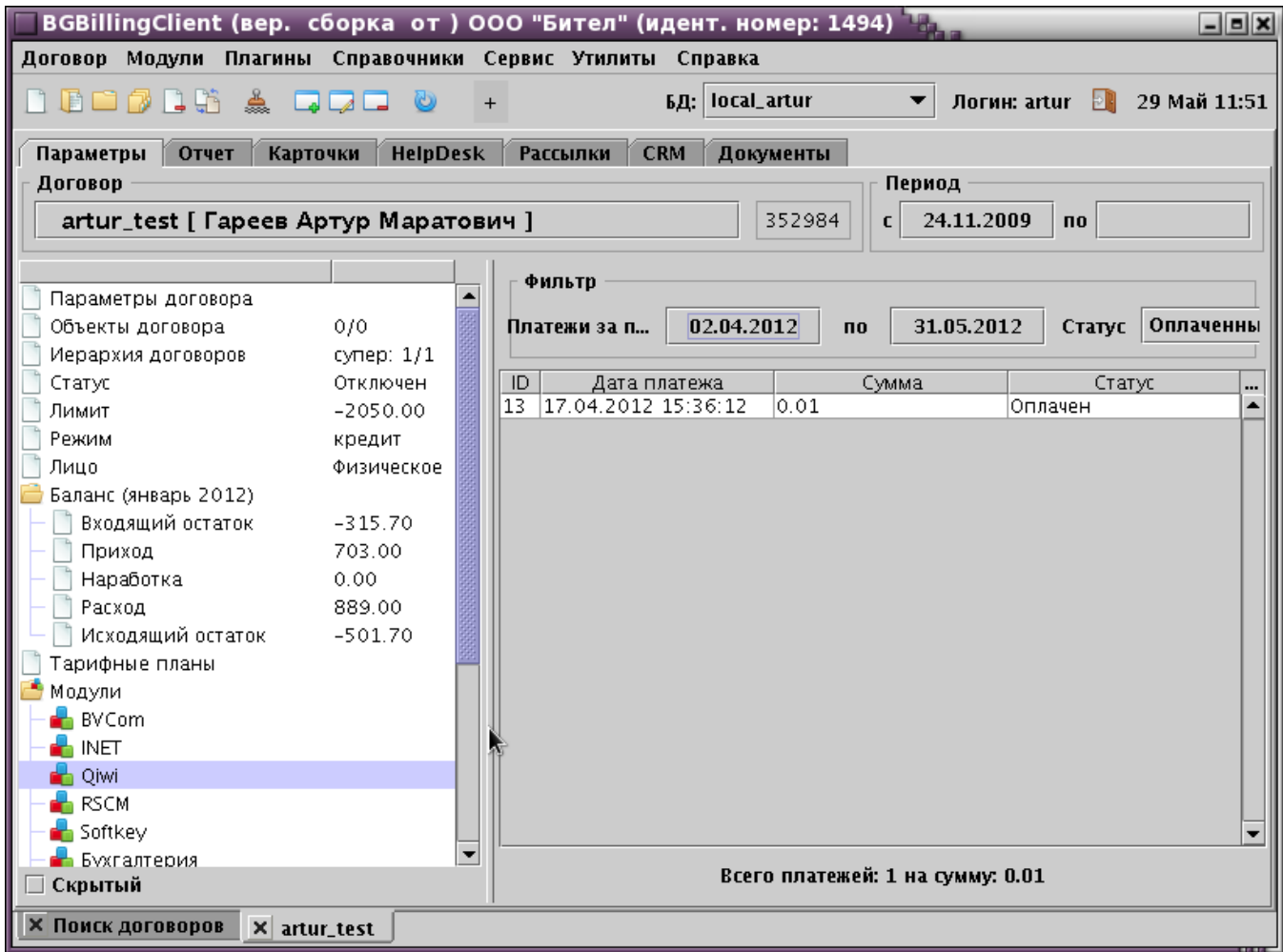
ID	Дата/Время оплаты	Статус	Сумма
31	09.10.2015 21:18:51	выставлен	111,00

Над таблицей с историей платежей расположена форма для совершения нового платежа. Чтобы осуществить платеж, необходимо заполнить обязательные поля **Сумма** и **Телефон**. Далее необходимо нажать кнопку **Оплатить** и подтвердить. После подтверждения клиент попадает на страницу системы Qiwi, где ему необходимо авторизоваться и подтвердить созданный счет с помощью своего мобильного телефона.

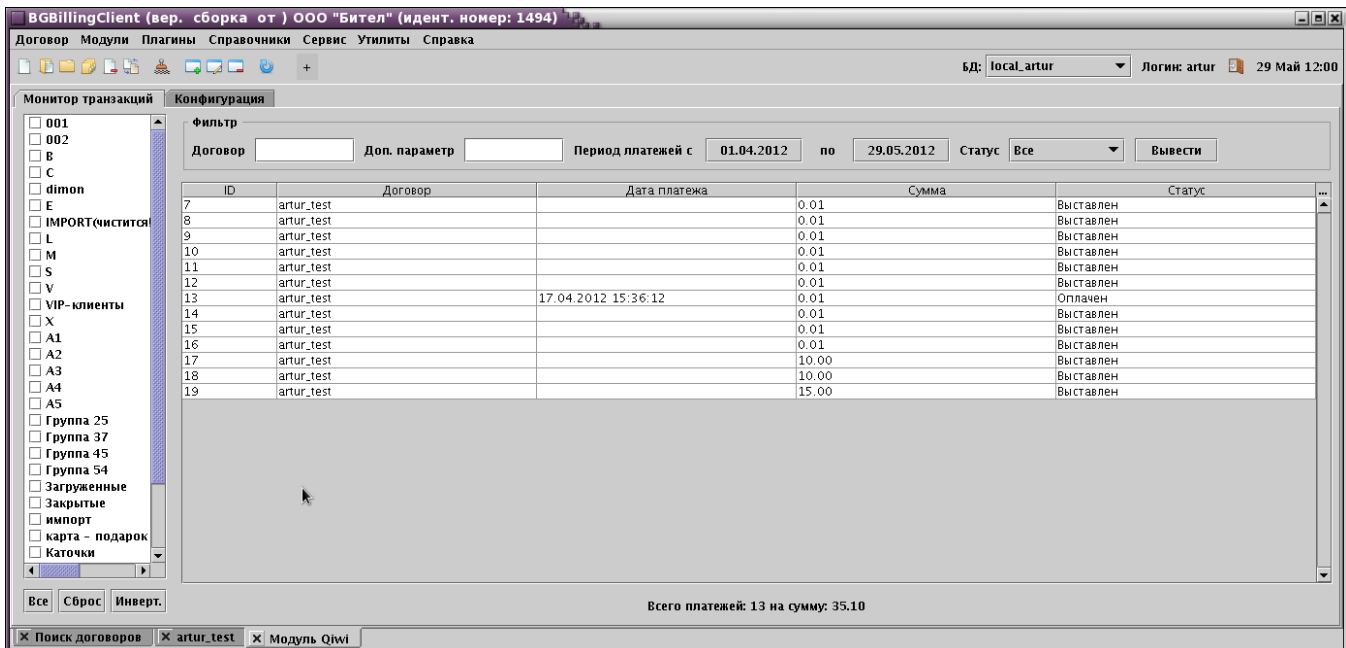
В случае успеха, клиент будет перенаправлен на страницу, указанную в конфигурации модуля в параметре **qiwi.success.url**.

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль Qiwi в дереве параметров договора. Здесь присутствует фильтр по статусу платежей (оплаченные, выставленные, проводимые, отмененные, все) с указанием периода, когда производилась оплата.



Для просмотра ВСЕХ платежей, проведенных с использованием модуля Qiwi, существует глобальный монитор транзакций в параметрах данного модуля биллинга (Модули -> Модуль Qiwi). На открывшейся вкладке у Вас есть возможность просмотреть все платежи, совершенные вашими абонентами за указанный временной период. Также можно установить фильтр платежей по группам договоров, по имени договора, по статусу, а также по произвольному текстовому параметру договора, по которому можно идентифицировать договор (например, в параметре договора хранится ИНН абонента, его расчетный счет и т.п.). В последнем случае код параметра договора задается в конфигурации модуля в опции `qiwi.additional.identify.key.pid`.



Модуль RBK.Money

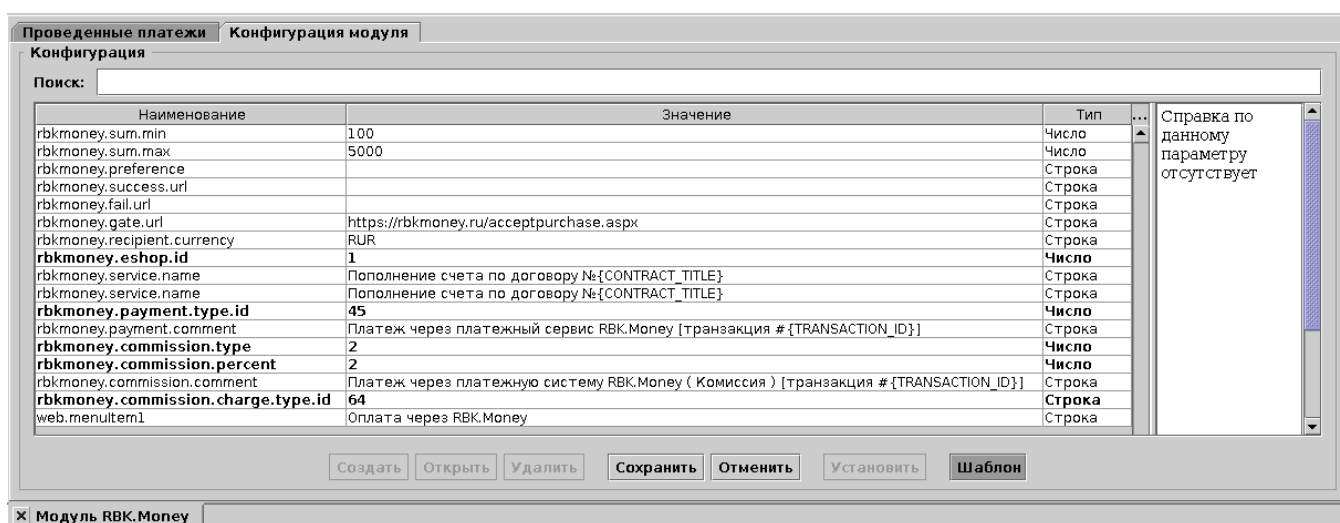
Назначение модуля

Модуль RBK.Money предназначен для проведения платежей через платежную систему RBK.Money с использованием пластиковых карт. Для проведения платежей Вашими клиентами у Вас должен быть заключен договор с системой.

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, скопируйте туда приведенный ниже текст и введите требуемые параметры. После этого сохраните конфигурацию и сделайте её активной.



The screenshot shows the 'Конфигурация модуля' (Module Configuration) window. It contains a table with the following data:

Наименование	Значение	Тип
rbkmoney.sum.min	100	Число
rbkmoney.sum.max	5000	Число
rbkmoney.preference		Строка
rbkmoney.success.url		Строка
rbkmoney.fail.url		Строка
rbkmoney.gate.url	https://rbkmoney.ru/acceptpurchase.aspx	Строка
rbkmoney.recipient.currency	RUR	Строка
rbkmoney.eshop.id	1	Число
rbkmoney.service.name	Пополнение счета по договору №{CONTRACT_TITLE}	Строка
rbkmoney.service.name	Пополнение счета по договору №{CONTRACT_TITLE}	Строка
rbkmoney.payment.type.id	45	Число
rbkmoney.payment.comment	Платеж через платежный сервис RBK.Money [транзакция #{TRANSACTION_ID}]	Строка
rbkmoney.commission.type	2	Число
rbkmoney.commission.percent	2	Число
rbkmoney.commission.comment	Платеж через платежную систему RBK.Money (Комиссия) [транзакция #{TRANSACTION_ID}]	Строка
rbkmoney.commission.charge.type.id	64	Строка
web.menuitem1	Оплата через RBK.Money	Строка

Below the table are buttons: Создать, Открыть, Удалить, Сохранить, Отменить, Установить, Шаблон. A search bar is located at the top left of the configuration area.

Замечания:

1. Прежде, чем задавать **rbkmoney.payment.type.id** необходимо создать соответствующий платеж в **Справочники->Другие->Типы платежей**
2. Параметры **rbkmoney.success.url** и **rbkmoney.fail.url** будет выглядеть следующим образом: **http://<адрес машины биллинга>/rbkmoney/<mid>**. Например, если у вас биллинг находится по адресу <http://billing.example.com/bgbilling/> и модуль RBK.Money имеет mid=16, то результирующий URL, который нужно дать компании RBK.Money, выглядит следующим образом: <http://billing.example.com/bgbilling/rbkmoney/16>.

Оплата через систему

Для предоставления возможности клиенту платить банковской картой через платежный шлюз **RBK.Money** необходимо подключить данный модуль к договору клиента. В Web-интерфейсе клиента появится новый пункт в меню - **Оплата через RBK.Money** (название по умолчанию).

Пополните счет с помощью шлюза RBK Money

Сумма: Текущий баланс:
99 877 778,1 руб.Период: < > Всего записей: 0 Страница: << < 1 из 1 [25] > >>[\[показать допол. фильтры\]](#)

№	Дата/Время оплаты	Статус	Карта	Сумма
Записи не найдены.				

Простой платеж в системе

Для совершения простого платежа необходимо ввести необходимую сумму в поле 2 и нажать кнопку **Оплатить**. Далее клиент попадает на страницу подтверждения оплаты. На данном этапе уже сформирован запрос на платежный шлюз и от клиента требуется нажать на кнопку **Да** для продолжения оплаты, либо **Отмена** для отмены платежа.

В случае, если клиент подтверждает оплату, то он перенаправляется на платежный шлюз, где может ввести информацию о своей пластиковой карте. Правильно заполнив предложенную форму и нажав кнопку **Оплатить**, клиент будет уведомлен о результате платежа.

В случае положительного результата биллинговая система начислит клиенту указанную им сумму на счет в виде нового платежа договора.

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль **RBK.Money** в дереве параметров договора. Здесь присутствует фильтр по статусу платежа с указанием периода, когда производилась оплата.

ID	Дата создания	Дата обработки	Карта	Статус	Сумма	...
1	03.05.2014 15:24:24			Новая	1,00	
2	03.05.2014 17:18:23			Новая	1,00	
3	03.05.2014 17:19:02			Новая	1,00	
4	03.05.2014 17:22:04			Новая	1,00	
5	03.05.2014 17:23:01			Новая	1,00	

Сумма итого:

Для просмотра ВСЕХ платежей, проведенных через систему RBK.Money, существует глобальный монитор в параметрах модуля. В открывшейся вкладке **Проведенные платежи** модуля **RBK.Money** у Вас есть возможность просмотреть платежи с учетом фильтра по статусу платежа и указанному периоду, совершенные вашими абонентами.

Проведенные платежи		Конфигурация модуля							
Период:	< 01.05.2011 - >	Статус:	Все	Договор:		Всего записей:	9	1 из 1 [25]	<< >>
ID	Договор	Дата создания	Дата обработки	Карта	Статус	Сумма	...		
1	kirill	03.05.2014 15:24:24			Новая	1,00	▲		
2	kirill	03.05.2014 17:18:23			Новая	1,00			
3	kirill	03.05.2014 17:19:02			Новая	1,00			
4	kirill	03.05.2014 17:22:04			Новая	1,00			
5	kirill	03.05.2014 17:23:01			Новая	1,00			
6	boris	24.11.2014 16:31:41			Новая	100,00			
7	boris	25.11.2014 16:33:15			Новая	55,00			
8	boris	28.11.2014 16:44:38			Новая	100,00			
9	boris	28.01.2015 13:20:44			Новая	113,22			
Сумма итого:						373,22			

По клику правой кнопкой мыши на строке в таблице открывается контекстное меню из которого можно открыть соответствующий договор или вернуть платеж.

Модуль RFI e-Commerce (шлюз ЗАО «РФИ БАНК»)

После установки модуля создаем конфигурацию, заполняем параметры, сохраняем конфигурацию, делаем ее активной.

Список параметров модуля с описанием каждого параметра доступен в редакторе параметров конфигурации по кнопке **Шаблон**.

Наименование	Значение	Тип
rfiec.sum.min	100	Число
rfiec.sum.max	5000	Число
rfiec.key	zO+TjycQu4pC4lvay10xf1eSfxq5HNTj0Ba1XuqxLZM=	Строка
rfiec.name	Пополнение баланса по договору №{CONTRACT_TITLE}	Строка
rfiec.secret	a35749aab5dbffafbd7cc7c877fd73a3	Строка
rfiec.url.gate	https://partner.rficb.ru/alba/input	Строка
rfiec.payment.type.id	47	Число
rfiec.payment.comment	Оплата через платежный шлюз РФИ Банка [транзакция #{TRANSACTION_ID}]	Строка
web.menuitem1	Оплата через платежный шлюз РФИ Банка	Строка

Для оповещения о результатах оплаты, в Личном кабинете на сайте платежной системы нужно указать URL, по которому будет проходить оповещения.

Данный URL имеет следующий формат: **http(s)://host[:port]/bgbilling/rfiec/<код_модуля_rfiec>**

После настройки модуля, в Личном кабинете клиентов будет доступен для оплаты и просмотра транзакций следующая форма:

Пополните счет с помощью шлюза РФИ Банка

Сумма (от 100.00 до 5000.00):

Оплатить

**Текущий баланс:
99 877 755,1 руб.**

Период:



Февраль, 2015



Всего записей: 13

Страница:



1 из 1 [25]

[\[показать допол. фильтры\]](#)

№	Дата/Время оплаты	Статус	Сумма
13	10.02.2015 19:12:59	Завершена	132,00
12	10.02.2015 19:07:30	Завершена	131,00
11	10.02.2015 19:03:08	Завершена	130,00
10	10.02.2015 18:55:34	Завершена	129,00
9	10.02.2015 18:48:29	Завершена	128,00
8	10.02.2015 18:38:41	Новая	127,00
7	10.02.2015 18:25:36	Новая	126,00
6	10.02.2015 18:22:19	Новая	125,00
5	10.02.2015 16:00:43	Новая	124,00
4	10.02.2015 15:56:37	Новая	112,00
3	10.02.2015 15:30:38	Новая	123,00
2	10.02.2015 1:35:19	Новая	111,00
1	10.02.2015 1:10:05	Новая	111,00

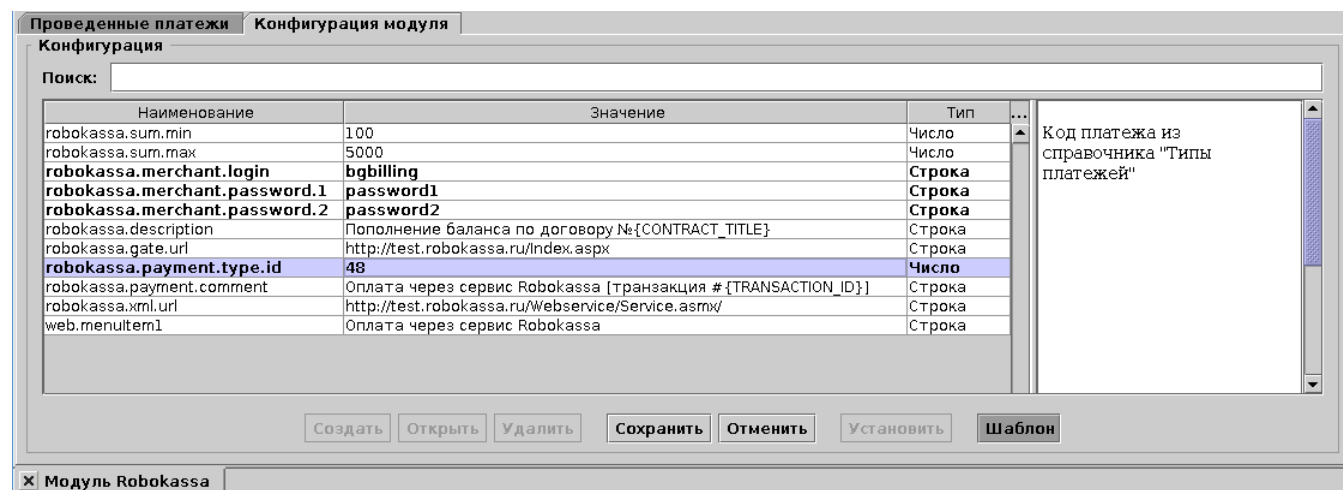
Модуль Robokassa

Назначение модуля

Платежный модуль Robokassa предназначен для осуществления безопасного приема платежей от абонентов. Более подробную информацию о сервисе, размере комиссии можно прочитать на сайте платежного сервиса Robokassa (<http://robokassa.ru>).

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.



Пояснения по параметрам:

- robokassa.sum.min - минимальная сумма
- robokassa.sum.max - максимальная сумма
- robokassa.merchant.login - логин
- robokassa.merchant.password.1 - пароль для формирования подписи при инициации платежа
- robokassa.merchant.password.2 - пароль для формирования подписи при поступлении уведомления о платеже
- robokassa.description - описание платежа, которое увидит абонент на сайте робокассы
- robokassa.gate.url - url сервиса, по умолчанию <https://merchant.roboxchange.com/Index.aspx>
- robokassa.payment.type.id - код типа платежа из справочника типов платежей
- robokassa.payment.comment - комментарий платежа, с которым занесется платеж на договор
- robokassa.xml.url - url для запроса статуса операции. по умолчанию <https://auth.robokassa.ru/Merchant/WebService/Service.asmx>
- robokassa.xml.method.state - по умолчанию OpState.
- robokassa.receipt - 0 - не формировать параметры чека в сторону сервиса, 1 - формировать
- robokassa.receipt.name - описание платежа (для чеков)
- robokassa.receipt.payment_method - Признак способа расчёта. По умолчанию full_payment. Все возможные значения можно посмотреть в доке робокассы: <https://docs.robokassa.ru/#6865> (для чеков)
- robokassa.receipt.payment_object - Признак предмета расчёта. Все возможные значения можно посмотреть в доке робокассы: <https://docs.robokassa.ru/#6865> (для чеков)
- robokassa.receipt.payment_tax - Это поле устанавливает номер налога в ККТ. Все возможные значения можно посмотреть в доке робокассы: <https://docs.robokassa.ru/#6865> (для чеков)

Для оповещения о результатах оплаты, в Личном кабинете на сайте платежной системы нужно указать URL, по которому будет проходить оповещения.

Данный URL имеет следующий формат: **http(s)://host[:port]/bgbilling/robokassa/<код_модуля_robokassa>**

URL успешной оплаты (SuccessURL) - **http[s]://hostname[:port]/bgbilling/webexecuter/robokassa/<код_модуля_robokassa>/DoTransaction/success**

URL ошибки (FailURL) - **http[s]://hostname[:port]/bgbilling/webexecuter/robokassa/<код_модуля_robokassa>/DoTransaction/fail**

Для возможности оплачивать через данный сервис необходимо добавить модуль на договор. После чего в личном кабинете будет доступен соответствующий пункт меню и страница оплаты.

Пополните счет с помощью сервиса **ROBOKASSA**

Сумма (от 100.00 до 5000.00):

Текущий баланс:
99 877 778,1 руб.

Период:

Всего записей: **0** Страница:

[\[показать допол. фильтры\]](#)

№	Дата/Время оплаты	Статус	Сумма
Записи не найдены.			

Модуль RuRuPay

Назначение модуля

Модуль предназначен для предоставления пользователям личного кабинета способа пополнения счёта через систему [RuRu](#). В личном кабинете работа ведётся через всплывающий виджет, позволяющий оплатить с помощью электронных валют, баланса сотового телефона, кредитных карт и других доступных способов оплаты системы. Также поддерживается оплата с витрины RuRu, с помощью СМС и т.п (см. возможности системы RuRu).

Настройка модуля

В общем виде конфигурация модуля выглядит так:

```
# ( RuRu )
partner_id=***
service_id=***
#
secretword=***
#
paymenttype=32
#
# contract.getTitle -
# contract.getComment -
# payment.getSum -
# payment.getDate -
# sum -
# date -
payment_comment= . {$contract.getTitle} ({$contract.getComment}) RuRu {$payment.getSum} {$date}
# / ( , : "https://wdemo.ruru.ru/" , : "https://widget.ruru.ru/" )
action_url=https://wdemo.ruru.ru/
```

Обмен данными со стороны RuRu нужно настроить так, чтобы приходил запрос init.

URL для оповещения настраивается со стороны RuRu и выглядит так: `http://<host>/bgbilling/rurupayexecuter/<mid>` , где mid - ID экземпляра [модуля](#).

Использование модуля

При добавлении модуля на договор в web-интерфейсе статистики появляется пункт меню "Пополнение счёта через RuRu". На странице оплаты — список операций со статусами и кнопка для вызова виджета оплаты.

Меню	
»	Новости
»	Уведомления
»	Просмотр баланса
»	Смена пароля на доступ к статистике
»	Смена тарифных планов
»	Тарифные опции
»	Карточки
»	Управление лимитом
»	Управление статусом
»	Дополнительные действия
»	Примечания
»	Параметры договора
»	Оплата через Assist.Ru
»	Управление подпиской SergeiCrypt
»	Смена пароля на карте
»	Пополнение счёта
»	Пополнение счёта через RuRu
»	Активация TrayInfo
»	Оплата через Yandex Деньги
»	Счета
»	Счета-фактуры
»	HelpDesk
»	Подписка на рассылки
»	Выход

Пополнение счёта через RuRu

Период: < Октябрь, 2013 >

Номер	Заказ	Дата/время оплаты	Сумма	Статус операции
5	352715_121	17.10.2013 20:02:21	12.00	В обработке
6	352715_123	17.10.2013 20:04:35	13.00	В обработке
18	352715_198	18.10.2013 13:44:09	30.00	Не проведён (Технологический сбой)
19	352715_206	18.10.2013 16:37:34	30.00	Проведён
20	352715_207	18.10.2013 16:38:37	31.00	Не проведён (Клиент отказался от резервирования)
21	352715_210	18.10.2013 16:39:41	40.00	Проведён
22	352715_215	18.10.2013 16:40:21	41.00	Не проведён (Клиент отказался от резервирования)
23	352715_218	18.10.2013 18:04:46	505.00	Проведён
24	352715_219	18.10.2013 18:05:02	600.00	Проведён
25	352715_220	18.10.2013 18:05:02	600.00	Проведён



Период: < Октябрь, 2013 >

Номер	Заказ	Дата/время оплаты
5	352715_121	17.10.2013 20:02:21
6	352715_123	17.10.2013 20:04:35
7	352715_125	17.10.2013 20:05:17
8	352715_128	17.10.2013 20:07:54
9	352715_130	17.10.2013 20:09:11
10	352715_131	17.10.2013 20:11:30
11	352715_134	17.10.2013 20:19:02
12	352715_136	17.10.2013 20:20:48
13	352715_169	17.10.2013 20:38:38
14	352715_170	17.10.2013 20:42:28
15	352715_172	17.10.2013 20:44:50
16	352715_178	17.10.2013 21:09:26
17	352715_179	17.10.2013 21:11:16
18	352715_198	18.10.2013 13:44:09
19	352715_206	18.10.2013 16:37:34
20	352715_207	18.10.2013 16:38:37
21	352715_210	18.10.2013 16:39:41
22	352715_215	18.10.2013 16:40:21
23	352715_218	18.10.2013 18:04:46
24	352715_219	18.10.2013 18:05:02
25	352715_220	18.10.2013 18:05:02



Номер заказа:

Выберите способ оплаты

- Со счёта мобильного телефона (Билайн, МегаФон, МТС)
- Банковская карта (Visa, Mastercard, Visa Electron)
- Электронные деньги (WebMoney)

Введите данные для оплаты

Номер мобильного телефона:

номер телефона, со счёта которого вы платите Вам придёт SMS для подтверждения платежа.

Сумма платежа: руб.

Все оплаты логируются в модуле.

VBillingClient (вер. 5.3.679 / 01.04.2013 14:03:39) ООО Бител (идент. номер: 2038)

Договор Модули Плагины Справочники Сервис Утилиты Справка

БД: Local | Логин: dimon | 30 Октябрь 16:54

Платежи | Конфигурация модуля

с [] по [] | Проведён | Не проведён | В обработке | 1 из 1 [25]

Код	Договор	Комментарий договора	Время	Сумма	Заказ	Статус	Причина отказа...
6	dimon	тестовый договор	17.10.2013 20:04:35	13.00	352715_123	В обработке	
7	dimon	тестовый договор	17.10.2013 20:05:17	14.00	352715_125	В обработке	
8	dimon	тестовый договор	17.10.2013 20:07:54	15.00	352715_128	В обработке	
9	dimon	тестовый договор	17.10.2013 20:09:11	16.00	352715_130	В обработке	
10	dimon	тестовый договор	17.10.2013 20:11:30	17.00	352715_131	В обработке	
11	dimon	тестовый договор	17.10.2013 20:19:02	18.00	352715_134	Проведён	
12	dimon	тестовый договор	17.10.2013 20:20:48	19.00	352715_136	Проведён	
13	dimon	тестовый договор	17.10.2013 20:38:38	20.00	352715_169	В обработке	
14	dimon	тестовый договор	17.10.2013 20:42:28	21.00	352715_170	В обработке	
15	dimon	тестовый договор	17.10.2013 20:44:50	22.00	352715_172	Не проведён	Клиент отказал...
16	dimon	тестовый договор	17.10.2013 21:09:26	23.00	352715_178	Проведён	
17	dimon	тестовый договор	17.10.2013 21:11:16	25.00	352715_179	В обработке	
18	dimon	тестовый договор	18.10.2013 13:44:09	30.00	352715_198	Не проведён	Технологически...
19	dimon	тестовый договор	18.10.2013 16:37:34	30.00	352715_206	Проведён	
20	dimon	тестовый договор	18.10.2013 16:38:37	31.00	352715_207	Не проведён	Клиент отказал...
21	dimon	тестовый договор	18.10.2013 16:39:41	40.00	352715_210	Проведён	
22	dimon	тестовый договор	18.10.2013 16:40:21	41.00	352715_215	Не проведён	Клиент отказал...

Модуль RuRuPay

В договоре также можно посмотреть подобную статистику.

БД: Local | Логин: dimon | 30 Октябрь 16:54

Параметры | Отчет | Карточки | Бонусы | HelpDesk | Рассылки | CRM | Документы

Договор

dimon [тестовый договор] | 352715 | Период с 04.02.2009 по []

с [] по [] | Проведён | Не проведён | В обработке | 1 из 1 [25]

Код	Договор	Комментарий договора	Время	Сумма	Заказ	Статус	Причина отказа...
6	dimon	тестовый договор	17.10.2013 20:04:35	13.00	352715_123	В обработке	
7	dimon	тестовый договор	17.10.2013 20:05:17	14.00	352715_125	В обработке	
8	dimon	тестовый договор	17.10.2013 20:07:54	15.00	352715_128	В обработке	
9	dimon	тестовый договор	17.10.2013 20:09:11	16.00	352715_130	В обработке	
10	dimon	тестовый договор	17.10.2013 20:11:30	17.00	352715_131	В обработке	
11	dimon	тестовый договор	17.10.2013 20:19:02	18.00	352715_134	Проведён	
12	dimon	тестовый договор	17.10.2013 20:20:48	19.00	352715_136	Проведён	
13	dimon	тестовый договор	17.10.2013 20:38:38	20.00	352715_169	В обработке	
14	dimon	тестовый договор	17.10.2013 20:42:28	21.00	352715_170	В обработке	
15	dimon	тестовый договор	17.10.2013 20:44:50	22.00	352715_172	Не проведён	Клиент отказал...
16	dimon	тестовый договор	17.10.2013 21:09:26	23.00	352715_178	Проведён	
17	dimon	тестовый договор	17.10.2013 21:11:16	25.00	352715_179	В обработке	
18	dimon	тестовый договор	18.10.2013 13:44:09	30.00	352715_198	Не проведён	Технологически...
19	dimon	тестовый договор	18.10.2013 16:37:34	30.00	352715_206	Проведён	
20	dimon	тестовый договор	18.10.2013 16:38:37	31.00	352715_207	Не проведён	Клиент отказал...
21	dimon	тестовый договор	18.10.2013 16:39:41	40.00	352715_210	Проведён	

Модуль RuRuPay | Поиск договоров | dimon

Модуль Sberbank (АО "Сбербанк")

Назначение модуля

Модуль биллинга **Sberbank** предназначен для оплаты картами через процессинг АО "Сбербанка".

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.

Тип	Наименование	Значение
Число	sberbank.sum.min	1
Число	sberbank.sum.max	5000
Число	sberbank.request.timeout	10000
Число	sberbank.request.read.timeout	3000
Число	sberbank.payment.type.id	46
Строка	sberbank.payment.comment	Платеж через платежную систему ...
Строка	sberbank.payment.features	
Строка	sberbank.rest.description	Пополнение счета по договору №{C...
Строка	sberbank.rest.register.url	https://securepayments.sberbank.ru/...
Строка	sberbank.rest.order.status.url	https://securepayments.sberbank.ru/...
Строка	sberbank.rest.bindings.url	https://securepayments.sberbank.ru/...
Строка	sberbank.rest.order.binding.url	https://securepayments.sberbank.ru/...
Строка	sberbank.rest.currency	643

В настройках Личного кабинета на сайте АО "Сбербанка" необходимо прописать Callback URL (merchant-url) для оповещения о статусах транзакций, в формате

`http[s]://host[:port]/bgbilling/sberbank/<код_модуля>`

Модуль имеет несколько встроенных задач для планировщика: для периодических автоплатежей (необходимо настроить если используются периодические автоплатежи) и проверка статуса транзакций (рекомендуется настроить для периодической проверки статуса платежей, необработанных по каким либо причинам другими способами)

Оплата через Личный кабинет

Если у клиента подключен экземпляр модуля в дереве договора, то он может осуществлять оплату через шлюз АО "Сбербанк", используя личный web-интерфейс.

В личном кабинете на странице отображается история платежей, совершенных клиентом:

Оплата через Сбербанк

Пополните счет с помощью шлюза Сбербанка

Сумма (от 1,00 до 5 000,00): Оплатить

Текущий баланс: 123,67 руб.

Период: < Октябрь, 2015 > Всего записей: 0 Страница: << < 1 из 1 [25] > >>

[показать допол. фильтры]

№	Дата/Время оплаты	Статус	Сумма
Записи не найдены.			

Над таблицей с историей платежей расположена форма для совершения нового платежа. Чтобы осуществить платеж, необходимо заполнить обязательные поля **Сумма**. Далее необходимо нажать кнопку **Оплатить** и подтвердить. После подтверждения клиент попадает на страницу с формой ввода реквизитов карты на сайте Сбербанка.

После оплаты (или отмены), клиент будет перенаправлен на страницу, указанную в конфигурации модуля в параметре [sberbank.rest.return.url](#).

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль Sberbank в дереве параметров договора. Здесь присутствует фильтр по статусу платежей с указанием периода, когда производилась оплата.

The screenshot shows a software interface for monitoring payments. On the left is a tree view of contract parameters, with 'Модуль Sberbank' selected. The main area displays a table of payments with the following columns: ID, Дата с..., Дата о..., Карта, Имя держателя карты, ApprovalCode, AuthCode, Ст..., and Сумма. The table is currently empty. At the top, there are filters for 'Период' (01.10.2006) and 'Статус' (Все). A 'Сумма итогов' field at the bottom right shows 0,00.

Для просмотра ВСЕХ платежей, проведенных с использованием модуля Sberbank, существует глобальный монитор транзакций в параметрах данного модуля биллинга (**Модули -> Модуль Sberbank**). На открывшейся вкладке у Вас есть возможность просмотреть все платежи, совершенные вашими абонентами за указанный временной период. Также можно установить фильтр платежей по имени договора, по статусу.

The screenshot shows the 'Проведенные платежи' (Completed Payments) table. The table has columns: ID, Договор, Дата создания, Дата обработки, Orderid, Карта, Имя держателя карты, Описание ошибки, Статус, and Сумма. The table contains 18 rows of data, including successful payments and failed transactions. A 'Сумма итогов' field at the bottom right shows 205,00.

ID	Договор	Дата создания	Дата обработки	Orderid	Карта	Имя держателя карты	Описание ошибки	Статус	Сумма
4		27.11.2014 20:50:...						Откло...	100,00
5		21.01.2015 19:33:...						Откло...	1,00
6		21.01.2015 19:43:...						Откло...	2,00
7		21.01.2015 19:53:...						Откло...	3,00
8		22.01.2015 01:13:...		7d1267f9-6be8-49bc-88e3-728d76526961			Платеж отклонен	Откло...	4,00
9		22.01.2015 01:24:...		44823963-19f4-48be-ac2e-b629ad3f1ae9			Платеж отклонен	Откло...	5,00
10		22.01.2015 01:28:...		52de9357-43f9-4423-9f9e-939930a5c396			Платеж отклонен	Откло...	6,00
11		22.01.2015 01:28:...		f9a03498-b423-47b3-9023-177464cd7ff			Платеж отклонен	Откло...	7,00
12		22.01.2015 01:33:...		e9eef27-e4e9-4577-8c2f-497dc0e02df4			Платеж отклонен	Откло...	8,00
13		22.01.2015 01:39:...		3b1ba0f2-d4dc-4c22-ba48-90c3a4a30b3			Платеж отклонен	Откло...	9,00
14		22.01.2015 01:42:...		a2c5882c-eb63-4ed7-bbfc-0a4d629477ad	411111**1111	TEST TEST	Платеж отклонен	Откло...	10,00
15		22.01.2015 02:35:...		44aeef89-37e0-467b-8ad1-6f9e9fdbcb67	639002**0003	TEST TEST	Платеж отклонен	Откло...	11,00
16		22.01.2015 18:55:...		198e0794-a6b8-4fd5-ba95-65df7fe23be0	555555**5599	TEST TEST	Платеж отклонен	Откло...	12,00
17		22.01.2015 19:21:...		3f476b63-c910-40f1-87bd-b91ce6da0e1d	555555**5599	TEST TEST	Платеж отклонен	Откло...	13,00
18		23.01.2015 20:30:...	23.01.2015 20:32:25	7ac7d28a-ae7f-4edf-ae1a-9ca80368d1c2	555555**5599	TEST TEST	Успешно	Одоб...	14,00

Настройка рекуррентных платежей (автоплатежи)

Модуль поддерживает автоплатежи. Для активации этой возможности, первое что нужно сделать, это связаться с сотрудниками Сбербанка для того что бы они настроили необходимые параметры со своей стороны и предоставили вам дополнительные данные для настройки в биллинге. После получения необходимых данных внесите их в конфигурацию модуля.

Модуль поддерживает три режима работы (каждый режим может включается/выключатся независимо от других):

- 1) Еженедельные платежи, клиенты через ЛК могут задать день недели и сумму, которая будет автоматически списываться с их карты.
- 2) Ежемесячные платежи, аналогично первому режиму, клиент задает день месяца и сумму.

Эти режимы не зависят от текущего состояния баланса клиентов, в указанный момент модуль будет пытаться пополнить баланс на заданную сумму. Есть возможность подменить сумму заданную клиентом, с помощью обработки соответствующего события. Для работы данных режимов должна быть настроена задача планировщика (часть запуска, раз в сутки)

- 3) Автоматический режим, при выборе этого режима, момент списания и сумма задаются скриптом по алгоритму провайдера (скрипт кидает событие с данными, какой договор и на какую сумму необходимо пополнить). В данном режиме также возможно выполнить автоплатеж из АРМа(админки биллинга) в ручном режиме.

Алгоритм работы автоплатежей:

Для активации автоплатежа клиент в ЛК задает параметры автоплатежа, после чего сервер биллинга посылает запрос регистрации (order.do) на сервер Сбербанка в запрос добавляется параметр clientId сигнализирующий о том что нужно создать связку для дальнейшего использования при автоплатежах. В ответ на этот запрос возвращается URL на который нужно перенаправить клиента для совершения платежа. После перехода клиента по указанному URL, ввода данных карты и совершения оплаты, клиент возвращается в ЛК. При возвращении в ЛК биллинг посылает запрос статуса (getOrderStatusExtended.do) платежа на сервер Сбербанка. В ответе на этот запрос, если платеж завершился удачно и в запросе регистрации присутствовал параметр clientId, должны присутствовать параметры clientId и bindingId. Параметр bindingId и есть связка, которую надо использовать при автоплатежах, это связка сохраняется в БД в таблице sberbank_autopayment_<mid> в поле access_token.

Оплата через СБП (система быстрых платежей)

Модуль поддерживает оплату с помощью QR кодов СБП. Для этого необходимо согласовать с банком данный режим и получить от банка: **сертификат, clientId, clientSecret, terminalId**

В клиенте биллинга из меню **Сервис - Настройка - Сертификаты** загрузить сертификат

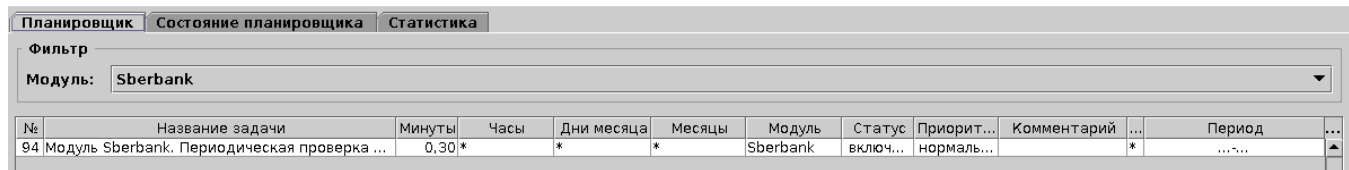
В конфигурации модуля включить поддержку оплаты через СБП и прописать параметры **clientId, clientSecret, terminalId**

После чего в Личном кабинете появится кнопка для оплаты через СБП.

Задачи планировщика

В модуле доступны две задачи:

- 1) Генерация заданий на списание автоплатежей (недельный и месячный режим). Задача должна запускаться раз в сутки.
- 2) Задача периодической проверки статуса транзакций (предназначена для обработки транзакций не обработанных по какой либо причине callback механизмом)



№	Название задачи	Минуты	Часы	Дни месяца	Месяцы	Модуль	Статус	Приорит...	Комментарий	...	Период	...
94	Модуль Sberbank. Периодическая проверка ...	0,30	*	*	*	Sberbank	включ...	нормаль...	*		...-...	

Модуль SimplePay

После установки модуля создаем конфигурацию, заполняем параметры, сохраняем конфигурацию, делаем ее активной.

Список параметров модуля с описанием каждого параметра доступен в редакторе параметров конфигурации по кнопке **Шаблон**.

Наименование	Значение	Тип
simplepay.sum.min	100	Число
simplepay.sum.max	5000	Число
simplepay.merchant.id	12	Строка
simplepay.outlet.id	12	Строка
simplepay.secret.key	secretKey	Строка
simplepay.description	Пополнение баланса по договору №{CONTRACT_TITLE}	Строка
simplepay.gate.url	https://api.simplepay.pro/sp/payment	Строка
simplepay.payment.type.id	49	Число
simplepay.payment.comment	Оплата через сервис SimplePay [транзакция #{TRANSACTION_ID}]	Строка
simplepay.lifetime	86400	Число
web.menuitem1	Оплата через сервис SimplePay	Строка

Для оповещения о результатах оплаты, в Личном кабинете на сайте платежной системы нужно указать URL, по которому будет проходить оповещения.

Данный URL имеет следующий формат: **http(s)://host[:port]/bgbilling/simplepay/<код_модуля_simplepay>**

Success URL:

[http\(s\)://host\[:port\]/bgbilling/webexecuter?module=simplepay&mid=<код_модуля_simplepay>&action=DoTransaction&operation=success](http(s)://host[:port]/bgbilling/webexecuter?module=simplepay&mid=<код_модуля_simplepay>&action=DoTransaction&operation=success)

Fail URL:

[http\(s\)://host\[:port\]/bgbilling/webexecuter?module=simplepay&mid=<код_модуля_simplepay>&action=DoTransaction&operation=fail](http(s)://host[:port]/bgbilling/webexecuter?module=simplepay&mid=<код_модуля_simplepay>&action=DoTransaction&operation=fail)

После настройки модуля, в Личном кабинете клиентов будет доступна для оплаты и просмотра транзакций следующая форма:

Оплата через сервис SimplePay

Пополните счет с помощью сервиса SimplePay

Сумма (от 100.00 до 5000.00): Оплатить

Текущий баланс: 99 877 775,1 руб.

Период: < Март, 2015 > Всего записей: 5 Страница: << < 1 из 1 [25] > >>

[показать допол. фильтры]

№	Дата/Время оплаты	Статус	Сумма
5	20.03.2015 17:38:53	Завершена	115,00
4	19.03.2015 17:35:25	Новая	114,00
3	19.03.2015 17:33:39	Новая	113,00
2	19.03.2015 17:25:47	Новая	112,00
1	19.03.2015 17:22:37	Новая	111,00

Модуль Tinkoff

Назначение модуля

Модуль биллинга **Tinkoff** предназначен для оплаты картами через процессинг банка "Тинькофф".

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.

Тип	Наименование	Значение
Число	tinkoff.sum.min	100
Число	tinkoff.sum.max	5000
Число	tinkoff.request.timeout	10000
Число	tinkoff.request.read.timeout	3000
Число	tinkoff.payment.type.id	61
Строка	tinkoff.payment.comment	Платеж через платежную систему Тинькофф
Строка	tinkoff.terminal.key	1563896358428DEMO
Строка	tinkoff.terminal.pswd	qeqli89o307imaf
Строка	tinkoff.init.url	https://securepay.tinkoff.ru/v2/init
Строка	tinkoff.init.notification.url	http://test33.bitel.ru/bgbilling/tinkoff/248
Строка	tinkoff.init.success.url	http://192.168.184.33:8089/bgbilling/webexecuter?action=D...
Строка	tinkoff.init.fail.url	
Строка	tinkoff.init.description	Пополнение счета по договору №{CONTRACT_TITLE}
Строка	tinkoff.rest.get.state.url	https://securepay.tinkoff.ru/v2/GetState
Строка	tinkoff.resend.url	https://securepay.tinkoff.ru/v2/Resend
Логический	tinkoff.fiscalization.enable	false
Список	tinkoff.fiscalization.receipt.taxation	osn
Строка	tinkoff.fiscalization.receipt.name	Пополнение счета по договору №{CONTRACT_TITLE}
Число	tinkoff.fiscalization.receipt.qty	1
Список	tinkoff.fiscalization.receipt.tax	none
Строка	tinkoff.fiscalization.receipt.email.company	
Список	tinkoff.fiscalization.receipt.payment.method	full_payment
Список	tinkoff.fiscalization.receipt.payment.object	payment

В настройках Личного кабинета на сайте Тинькофф можно прописать NotificationURL для оповещения о статусах транзакций, в формате `http[s]://host[:port]/bgbilling/tinkoff/<код_модуля>`

Модуль имеет несколько встроенных задач для планировщика: для периодических автоплатежей (необходимо настроить если используются периодические автоплатежи) и проверка статуса транзакций (рекомендуется настроить для периодической проверки статуса платежей, необработанных по каким либо причинам другими способами)

Модуль Uniteller

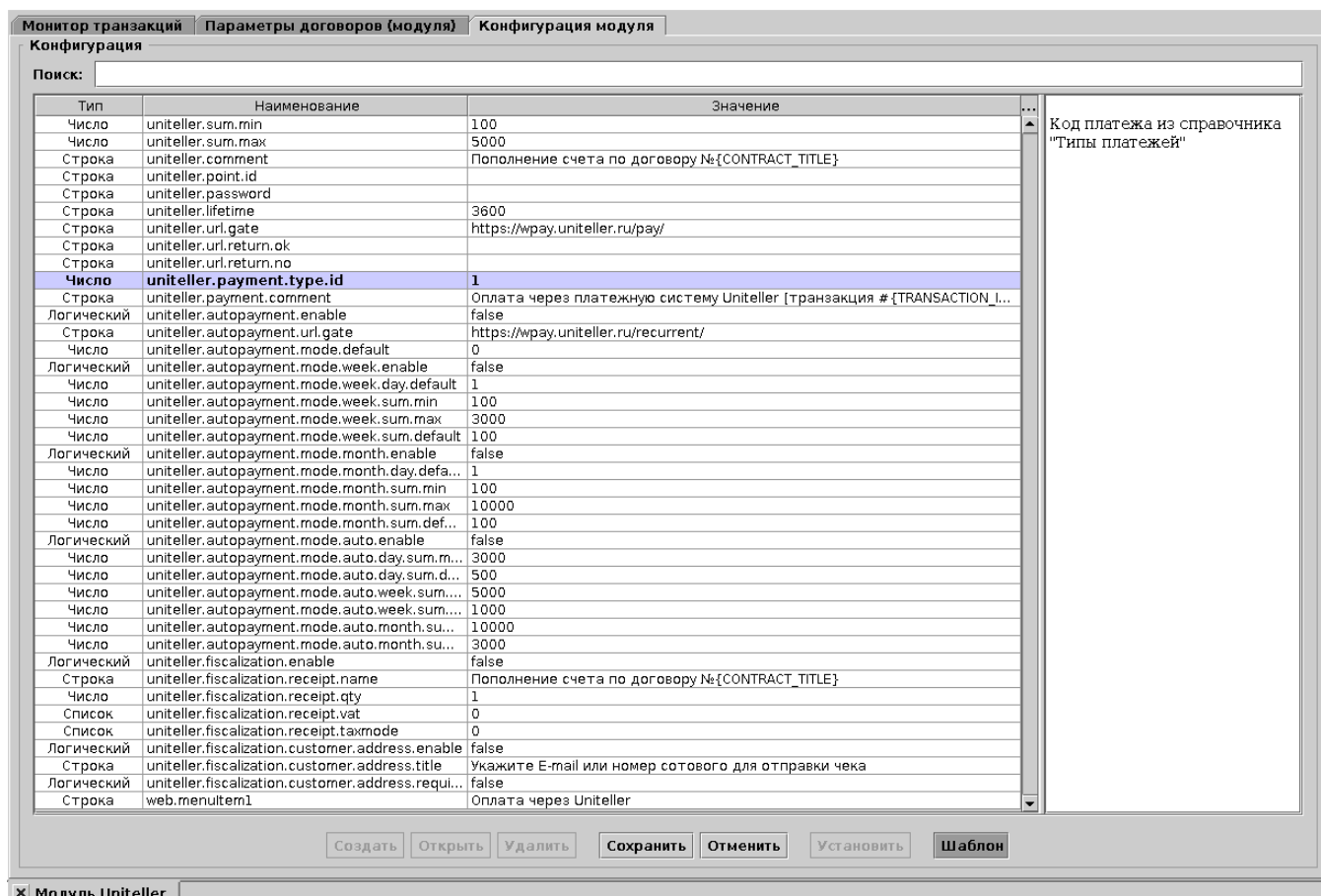
Назначение модуля

Модуль биллинга **Uniteller** предназначен для оплаты услуг с помощью сервиса <https://www.uniteller.ru>

Модуль работает по защищенному протоколу https с использованием шифрования SSL, что исключает возможность перехвата информации и нарушения целостности данных третьими лицами.

Настройка модуля

Установите модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, пропишите необходимые параметры. После этого сохраните конфигурацию и сделайте её активной.



The screenshot shows the 'Конфигурация' (Configuration) window of the Uniteller module. It features a search bar at the top and a table of parameters. The table has three columns: 'Тип' (Type), 'Наименование' (Name), and 'Значение' (Value). The 'uniteller.payment.type.id' parameter is highlighted in blue. To the right of the table is a text area containing the value 'Код платежа из справочника "Типы платежей"'. At the bottom of the window are several buttons: 'Создать', 'Открыть', 'Удалить', 'Сохранить', 'Отменить', 'Установить', and 'Шаблон'.

Тип	Наименование	Значение
Число	uniteller.sum.min	100
Число	uniteller.sum.max	5000
Строка	uniteller.comment	Пополнение счета по договору №{CONTRACT_TITLE}
Строка	uniteller.point.id	
Строка	uniteller.password	
Строка	uniteller.lifetime	3600
Строка	uniteller.url.gate	https://wpay.uniteller.ru/pay/
Строка	uniteller.url.return.ok	
Строка	uniteller.url.return.no	
Число	uniteller.payment.type.id	1
Строка	uniteller.payment.comment	Оплата через платежную систему Uniteller [транзакция # {TRANSACTION_I...}
Логический	uniteller.autopayment.enable	false
Строка	uniteller.autopayment.url.gate	https://wpay.uniteller.ru/recurrent/
Число	uniteller.autopayment.mode.default	0
Логический	uniteller.autopayment.mode.week.enable	false
Число	uniteller.autopayment.mode.week.day.default	1
Число	uniteller.autopayment.mode.week.sum.min	100
Число	uniteller.autopayment.mode.week.sum.max	3000
Число	uniteller.autopayment.mode.week.sum.default	100
Логический	uniteller.autopayment.mode.month.enable	false
Число	uniteller.autopayment.mode.month.day.defa...	1
Число	uniteller.autopayment.mode.month.sum.min	100
Число	uniteller.autopayment.mode.month.sum.max	10000
Число	uniteller.autopayment.mode.month.sum.def...	100
Логический	uniteller.autopayment.mode.auto.enable	false
Число	uniteller.autopayment.mode.auto.day.sum.m...	3000
Число	uniteller.autopayment.mode.auto.day.sum.d...	500
Число	uniteller.autopayment.mode.auto.week.sum...	5000
Число	uniteller.autopayment.mode.auto.week.sum...	1000
Число	uniteller.autopayment.mode.auto.month.su...	10000
Число	uniteller.autopayment.mode.auto.month.su...	3000
Логический	uniteller.fiscalization.enable	false
Строка	uniteller.fiscalization.receipt.name	Пополнение счета по договору №{CONTRACT_TITLE}
Число	uniteller.fiscalization.receipt.qty	1
Список	uniteller.fiscalization.receipt.vat	0
Список	uniteller.fiscalization.receipt.taxmode	0
Логический	uniteller.fiscalization.customer.address.enable	false
Строка	uniteller.fiscalization.customer.address.title	Укажите E-mail или номер сотового для отправки чека
Логический	uniteller.fiscalization.customer.address.requi...	false
Строка	web.menuitem1	Оплата через Uniteller

В ЛК Uniteller пропишите **URL-адрес уведомления магазина**:

```
http[s]://billing_host[:port]/[bgbilling/]uniteller/<_>
```

Модуль имеет несколько встроенных задач для планировщика: для периодических автоплатежей (необходимо настроить если используются периодические автоплатежи) и проверка статуса транзакций (рекомендуется настроить для периодической проверки статуса платежей, необработанных по каким либо причинам другими способами)

Оплата через Личный кабинет

Если у клиента подключен экземпляр модуля в дереве договора, то он может осуществлять оплату через платежную систему Uniteller, используя личный web-интерфейс.

В личном кабинете на странице отображается история платежей, совершенных клиентом:

Оплата через Uniteller

Пополнить баланс Настроить Автоплатеж

Текущий баланс: 99 895 227,1 руб.

Пополните счет с помощью сервиса Uniteller

Сумма (от 100,00 до 5 000,00):

Период: < Октябрь, 2018 > Всего записей: 0 Страница: << < 1 из 1 [25] > >>

[показать доп. фильтры]

№	Дата/Время оплаты	Статус	Сумма
Записи не найдены.			

Мониторинг платежей

В клиенте биллинга есть возможность отслеживать историю платежей по каждому абоненту. Для этого необходимо выбрать модуль Uniteller в дереве параметров договора. Здесь присутствует фильтр по статусу платежей с указанием периода, когда производилась оплата.

Транзакции Автоплатеж

Период: < > Статус: Все Всего записей: 16 1 из 1 [25]

ID	Дата создания	Дата обработки	Автоплатеж	Статус	Сумма
5	15.06.2017 16:37:08		<input type="checkbox"/>	Новая	123,00
6	15.06.2017 16:41:20		<input type="checkbox"/>	Новая	123,00
7	15.06.2017 16:49:18		<input type="checkbox"/>	Новая	222,00
8	15.06.2017 16:53:33		<input type="checkbox"/>	Новая	124,00
9	15.06.2017 17:07:58		<input type="checkbox"/>	Новая	125,00
10	15.06.2017 18:02:32		<input type="checkbox"/>	Новая	126,00
11	15.06.2017 18:03:19		<input type="checkbox"/>	Новая	127,00
12	15.06.2017 18:06:48		<input type="checkbox"/>	Новая	128,00
13	15.06.2017 18:09:08		<input type="checkbox"/>	Новая	129,00
14	15.06.2017 18:17:19		<input type="checkbox"/>	Новая	130,00
15	15.06.2017 18:21:20		<input type="checkbox"/>	Новая	131,00
16	15.06.2017 18:26:30		<input type="checkbox"/>	Новая	132,00
17	15.06.2017 23:46:55		<input type="checkbox"/>	Новая	133,00
18	16.06.2017 00:02:39	16.06.2017 00:03:51	<input type="checkbox"/>	Завершена	155,00
19	16.06.2017 00:44:26	16.06.2017 00:45:06	<input type="checkbox"/>	Завершена	101,00
20	16.06.2017 00:49:21	16.06.2017 00:50:28	<input type="checkbox"/>	Завершена	111,00

Сумма итогов: 2 120,00

Для просмотра ВСЕХ платежей, проведенных с использованием модуля Uniteller, существует глобальный монитор транзакций в параметрах данного модуля биллинга (**Модули -> Модуль Uniteller**). На открывшейся вкладке у Вас есть возможность просмотреть все платежи, совершенные вашими абонентами за указанный временной период. Также можно установить фильтр платежей по имени договора, по статусу.

Монитор транзакций Параметры договоров (модуля) Конфигурация модуля

Период: < > Статус: Все Всего записей: 18 1 из 1 [25]

ID	Договор	Дата создания	Дата обработки	Автоплатеж	RRN	Approval code	3DSECURE	Статус	Сумма
1				<input type="checkbox"/>				1	1,00
2				<input type="checkbox"/>				2	2,00
5	kirill	15.06.2017 16:37:08		<input type="checkbox"/>				Новая	123,00
6	kirill	15.06.2017 16:41:20		<input type="checkbox"/>				Новая	123,00
7	kirill	15.06.2017 16:49:18		<input type="checkbox"/>				Новая	222,00
8	kirill	15.06.2017 16:53:33		<input type="checkbox"/>				Новая	124,00
9	kirill	15.06.2017 17:07:58		<input type="checkbox"/>				Новая	125,00
10	kirill	15.06.2017 18:02:32		<input type="checkbox"/>				Новая	126,00
11	kirill	15.06.2017 18:03:19		<input type="checkbox"/>				Новая	127,00
12	kirill	15.06.2017 18:06:48		<input type="checkbox"/>				Новая	128,00
13	kirill	15.06.2017 18:09:08		<input type="checkbox"/>				Новая	129,00
14	kirill	15.06.2017 18:17:19		<input type="checkbox"/>				Новая	130,00
15	kirill	15.06.2017 18:21:20		<input type="checkbox"/>				Новая	131,00
16	kirill	15.06.2017 18:26:30		<input type="checkbox"/>				Новая	132,00
17	kirill	15.06.2017 23:46:55		<input type="checkbox"/>				Новая	133,00
18	kirill	16.06.2017 00:02:39	16.06.2017 00:03:51	<input type="checkbox"/>				Завер...	155,00
19	kirill	16.06.2017 00:44:26	16.06.2017 00:45:06	<input type="checkbox"/>				Завер...	101,00
20	kirill	16.06.2017 00:49:21	16.06.2017 00:50:28	<input type="checkbox"/>				Завер...	111,00

Сумма итогов: 2 123,00

Настройка рекуррентных платежей (автоплатежи)

Модуль поддерживает автоплатежи. Для активации этой возможности, первое что нужно сделать, это связаться с сотрудниками Uniteller для того, что бы они настроили необходимые параметры со своей стороны и предоставили вам дополнительные данные для настройки в биллинге. После получения необходимых данных внесите их в конфигурацию модуля.

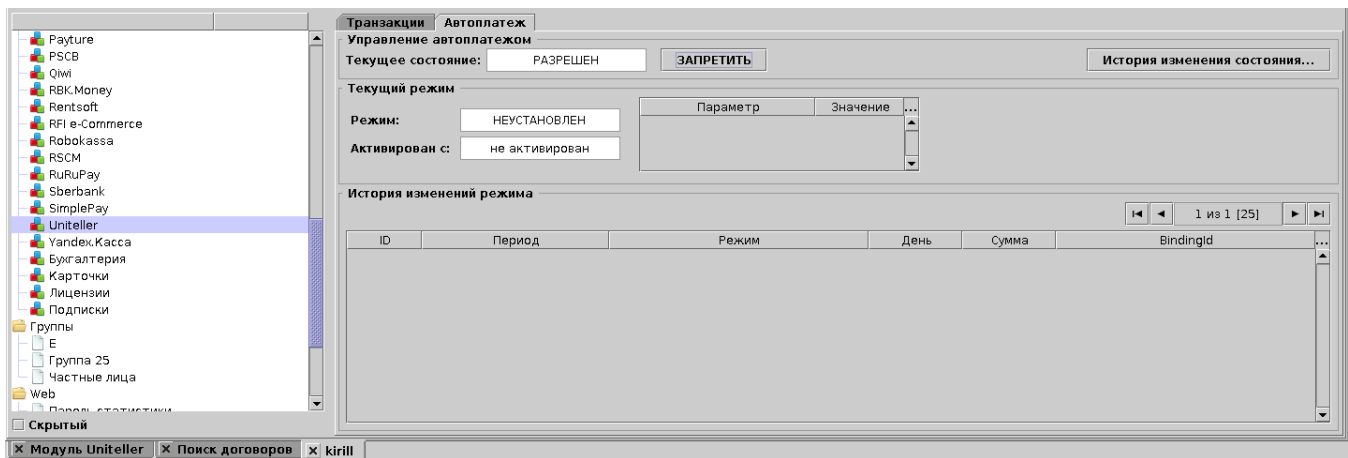
Модуль поддерживает три режима работы (каждый режим может включается/выключатся независимо от других):

- 1) Еженедельные платежи, клиенты через ЛК могут задать день недели и сумму, которая будет автоматически списываться с их карты.
- 2) Ежемесячные платежи, аналогично первому режиму, клиент задает день месяца и сумму.

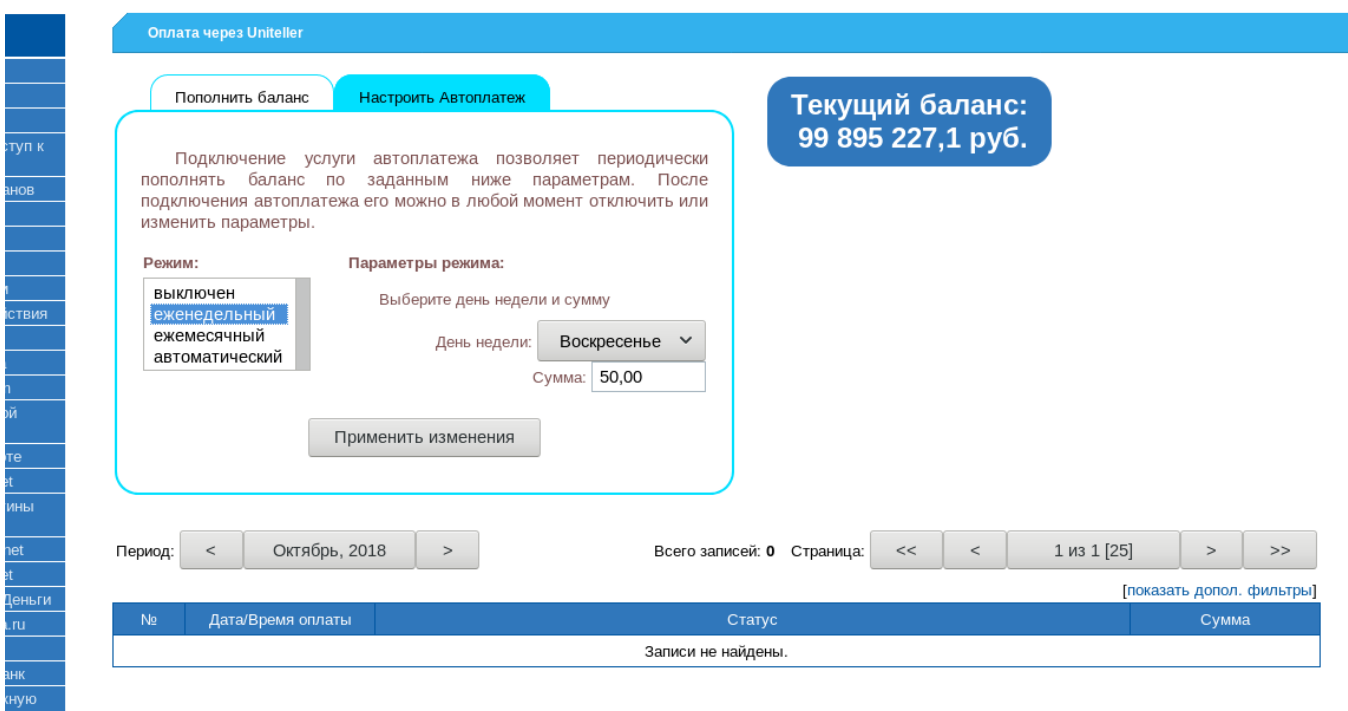
Эти режимы не зависят от текущего состояния баланса клиентов, в указанный момент модуль будет пытаться пополнить баланс на заданную сумму. Есть возможность подменить сумму заданную клиентом, с помощью обработки соответствующего события. Для работы данных режимов должна быть настроена задача планировщика (частота запуска, раз в сутки)

3) Автоматический режим, при выборе этого режима, момент списания и сумма задаются скриптом по алгоритму провайдера (скрипт кидает событие с данными, какой договор и на какую сумму необходимо пополнить). В данном режиме также возможно выполнить автоплатеж из АРМа(админки биллинга) в ручном режиме.

Данные о текущем режиме для конкретного клиента можно посмотреть в договоре



Настройка автоплатежа в Личном кабинете



Алгоритм работы автоплатежей:

Для активации автоплатежа клиент в ЛК задает параметры автоплатежа, после чего клиенту предлагается совершить платеж с использованием карты, в случае успешного проведения платежа, его идентификатор запоминается в параметрах текущего режима автоплатежа и в дальнейшем при проведении автоплатежей используется для получения необходимых данных о карте (в базе биллинга данные карты не сохраняются, только ссылка на успешный платеж).

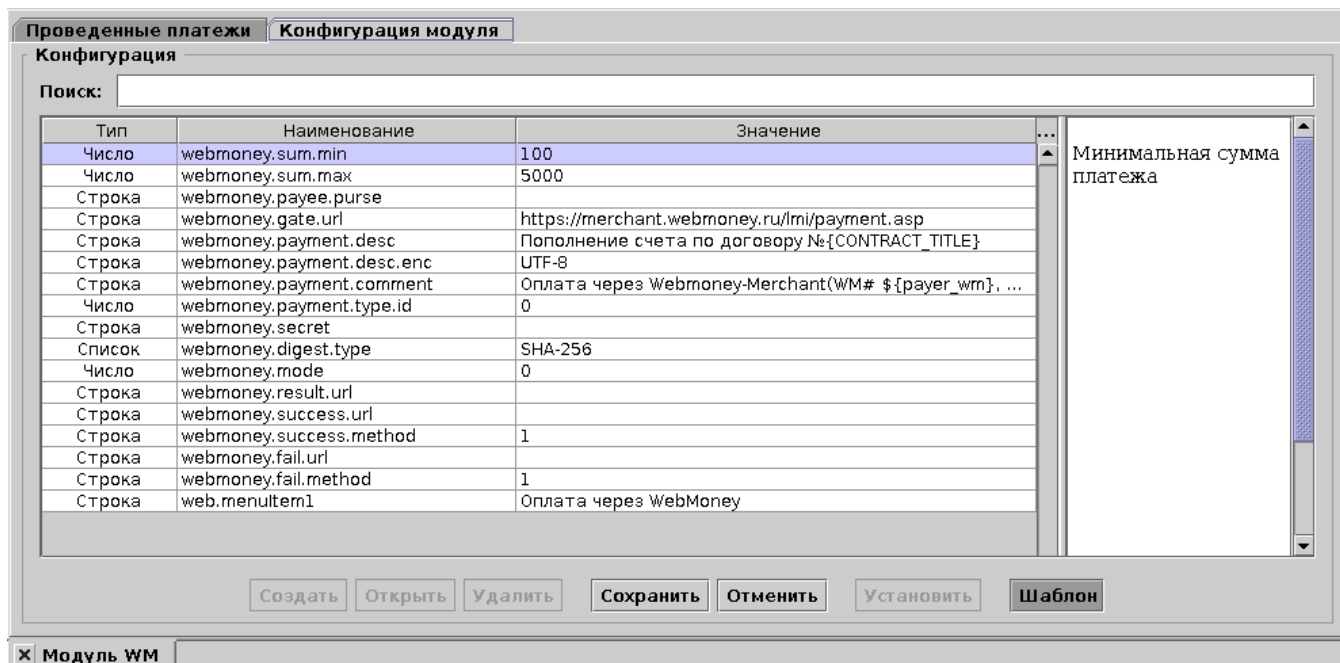
Модуль WebMoney

Назначение модуля

Модуль WebMoney предназначен для проведения платежей через систему Merchant WebMoney Transfer. Для работы в нетестовом режиме у вас должен быть аттестат продавца (разновидность персонального) системы [WebMoney](#)

Настройка модуля

Проинсталируйте модуль на сервер, обновите клиент. Затем создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, включите режим редактор **ШАБЛОН** и заполните необходимые параметры, сохраните конфигурацию и сделайте её активной.



Настройте сервис MerchantWM

На сайте Webmoney в Личном кабинете выполните настройки торгового кошелька

The screenshot shows the 'Настройки торгового кошелька' (Merchant Wallet Settings) form. The fields and their values are:

- Кошелек: R484945688783
- Торговое имя: BG Billing
- Secret Key: superpuperverylonghardandcomplicatedkey
- Result URL: (empty)
- Success URL: (empty) POST метод вызова Success URL
- Fail URL: (empty) POST метод вызова Fail URL
- Позволять использовать URL, передаваемые в форме:
- Высылать оповещение об ошибке платежа на кипер:
- Метод формирования контрольной подписи: 256
- Тестовый/Рабочий режимы: Тестовый (Тестовый режим! Перевод WM не происходит.)
- Активность: Вкл. (Прием WM на кошелек ВКЛЮЧЕН.)
- Прием чеков Paymer.com (ВМ-карт): Вкл. (Прием ВМ-карт или чеков Paymer на кошелек ВКЛЮЧЕН. Подробнее ...)
- Прием платежей с телефонов Telepat.ru: Вкл. (Прием платежей с телефонов Telepat.ru на кошелек ВКЛЮЧЕН. Подробнее ...)

A 'Сохранить' (Save) button is located at the bottom left.

Введите торговое имя, secretkey, оставьте пустыми поля URL, галочку на Высылать SecretKey. Поставьте галочку на "Передавать параметры в предварительном запросе", "Позволять использовать URL, передаваемые в форме" и ,если необходимо, "Высылать оповещение об ошибке платежа на кипер".

Установите метод вызова Succes URL и Fail URL на POST.

Переключите Активность на Вкл.

Переключение из тестового в рабочий режим возможен только при наличии аттестата продавца.

Проведение платежа из Личного кабинета

Если у абонента подключён экземпляр модуля, то он может произвести оплату услуг через Web-интерфейс (пункт меню **Оплата через WebMoney**). Здесь же клиент может просмотреть все проведённые им платежи через WM:

Список выписанных счетов

Месяц: январь Год: 2006

Номер	Дата/время оплаты	Кошелек	WM-ID	Сумма
1	2006.01.26 14:53:09	R484945688783	299497894311	1.00
18	2006.01.26 15:41:13	R484945688783	299497894311	1.00
42	2006.01.26 17:43:03	R484945688783	299497894311	1.00
43	2006.01.26 17:45:40	R484945688783	299497894311	10.00

При нажатии кнопки **Пополнить счёт** клиент попадает на новую страницу, с формой для ввода суммы.

Пополнение счета

Новый платеж	
Текущий баланс	10042.02
Введите сумму для оплаты	<input type="text" value="100"/>
<input type="button" value="Принять"/> <input type="button" value="Отмена"/>	

При нажатии кнопки **Принять** клиент сначала переходит на страницу подтверждения биллинга, а затем к системе [Merchant WebMoney Transfer](#), где может выбрать удобный для себя способ оплаты.

Безопасность

При нажатии клиентом кнопки **Подтверждаю платёж** Merchant WM последовательно производит два запроса в биллинг: предварительный и оповещающий об оплате. При первом запросе модуль проверяет правильность данных и разрешает или же запрещает (например, если модуль в режиме выключен) проведение оплаты на кошелёк. Если биллинг разрешил платёж WebMoney Transfer сразу же (!) производит транзакцию и отправляет второй, оповещающий запрос, в котором, кроме служебных данных, посылается контрольная подпись, созданная из суммы этих служебных данных (почти все из них уникальны) и secretkey (который в запросе не посылается). Модуль проверяет контрольную сумму и, если она совпала, пополняет баланс клиента.

Таким образом secretkey должен быть длинным и не подбираемым (в системе Merchant WM secretkey может быть длиной до 50 символов).

Внимание! В предварительном запросе Merchant WM не посылает контрольную подпись, поэтому если в момент платежа в конфиге биллинга и настройках Merchant WM secretkey будут различными, то транзакция будет проведена, но баланс в биллинге не пополнится. Поэтому при изменении secretkey и вообще при изменении каких-либо параметров модуля настоятельно рекомендуется перевести режим модуля в состояние выключен опцией wm.mode=0, подождать завершения транзакций, которые могут проходить в данный момент (при стабильной работе оба запроса происходят практически в одно время) и только потом изменять настройки.

Все платежи, не прошедшие проверку контрольной подписи (но прошедшие все другие проверки), попадают в базу как не проведённые.

Монитор транзакций

В параметрах договора можно посмотреть все платежи, как проведённые, так и не проведённые (см. Безопасность)

Параметры Журнал Отчет Полная карта Другие карты

Договор: x0000 (Иванов Петр Сидорович) Период: с 03.08.2003 по

Свойства Услуги

Дата транзакции с 01.01.2006 по 01.02.2006 проведённые Поиск

ID	WM #	Кошелёк	Время	Сумма	INVS #	TRANS #
1	299497894311	R484945688783	2006.01.26 14:53:09	1.00	612	547
18	299497894311	R484945688783	2006.01.26 15:41:13	1.00	78	528
19	299497894311	R484945688783	2006.01.26 15:42:15	1.00	968	488
20	299497894311	R484945688783	2006.01.26 15:43:07	1.00	628	738
21	299497894311	R484945688783	2006.01.26 15:45:12	1.00	876	173
29	299497894311	R484945688783	2006.01.26 16:47:43	1.00	748	287
30	299497894311	R484945688783	2006.01.26 16:48:01	2.00	885	599
31	299497894311	R484945688783	2006.01.26 16:51:10	1.00	26	538
32	299497894311	R484945688783	2006.01.26 16:53:10	1.00	160	209
33	299497894311	R484945688783	2006.01.26 16:54:39	1.00	387	371
34	299497894311	R484945688783	2006.01.26 16:55:17	1.00	566	403
37	299497894311	R484945688783	2006.01.26 16:58:45	1.00	761	866
38	299497894311	R484945688783	2006.01.26 17:36:20	1.00	806	45
39	299497894311	R484945688783	2006.01.26 17:38:36	1.00	249	481
40	299497894311	R484945688783	2006.01.26 17:39:06	2.97	608	799
41	299497894311	R484945688783	2006.01.26 17:39:32	1.07	633	203
42	299497894311	R484945688783	2006.01.26 17:43:03	1.00	660	175
43	299497894311	R484945688783	2006.01.26 17:45:40	10.00	878	666

В параметрах модуля есть глобальный монитор

Проведенные платежи		Конфигурация модуля	
Группы:			
<input type="checkbox"/>	B		
<input type="checkbox"/>	C		
<input type="checkbox"/>	E		
<input type="checkbox"/>	L		
<input type="checkbox"/>	M		
<input type="checkbox"/>	S		
<input type="checkbox"/>	V		
<input type="checkbox"/>	VIP-клиенты		
<input type="checkbox"/>	X		
<input type="checkbox"/>	Группа 25		
<input type="checkbox"/>	Закрытые		
<input type="checkbox"/>	карта - подарок		
<input type="checkbox"/>	Каталоги		
<input type="checkbox"/>	Организации		
<input type="checkbox"/>	Сотрудники		
<input type="checkbox"/>	Физические		
<input type="checkbox"/>	Хостинг		
<input type="checkbox"/>	Частные лица		
<div style="text-align: right;"> <input type="button" value="Все"/> <input type="button" value="Сброс"/> <input type="button" value="Инверт."/> </div>			

Фильтр									
Номер договора	WM #	Номер платежа							
				проведенные	с	01.01.2006	по	01.02.2006	<input type="button" value="Поиск"/>
Договор	ID	WM #	Кошелёк	Время	Сумма	INVS #	TRANS #	Поступило на	
x0000	1	299497894311	R484945688783	2006.01.26 14:53:09	1.00	612	547	R484945688783	
x0000	18	299497894311	R484945688783	2006.01.26 15:41:13	1.00	78	528	R484945688783	
x0000	19	299497894311	R484945688783	2006.01.26 15:42:15	1.00	968	488	R484945688783	
x0000	20	299497894311	R484945688783	2006.01.26 15:43:07	1.00	628	738	R484945688783	
x0000	21	299497894311	R484945688783	2006.01.26 15:45:12	1.00	876	173	R484945688783	
x0000	29	299497894311	R484945688783	2006.01.26 16:47:43	1.00	748	287	R484945688783	
x0000	30	299497894311	R484945688783	2006.01.26 16:48:01	2.00	885	599	R484945688783	
x0000	31	299497894311	R484945688783	2006.01.26 16:51:10	1.00	26	538	R484945688783	
x0000	32	299497894311	R484945688783	2006.01.26 16:53:10	1.00	160	209	R484945688783	
x0000	33	299497894311	R484945688783	2006.01.26 16:54:39	1.00	387	371	R484945688783	
x0000	34	299497894311	R484945688783	2006.01.26 16:55:17	1.00	566	403	R484945688783	
x0000	37	299497894311	R484945688783	2006.01.26 16:58:45	1.00	761	866	R484945688783	
x0000	38	299497894311	R484945688783	2006.01.26 17:36:20	1.00	836	45	R484945688783	
x0000	39	299497894311	R484945688783	2006.01.26 17:38:36	1.00	249	481	R484945688783	
x0000	40	299497894311	R484945688783	2006.01.26 17:39:06	2.97	608	799	R484945688783	
x0000	41	299497894311	R484945688783	2006.01.26 17:39:32	1.07	633	203	R484945688783	
x0000	42	299497894311	R484945688783	2006.01.26 17:43:03	1.00	660	175	R484945688783	
x0000	43	299497894311	R484945688783	2006.01.26 17:45:40	10.00	878	666	R484945688783	

Коды ошибок

Ошибки проведения платежа представлены в битовой маске, т.е ошибка 384(11000000) это две ошибки из списка:

- 0x01 - 1 - неверный запрос (не все и/или неправильные параметры, подразумевается запрос не от merchant)
- 0x02 - 10 - 0x04 - 100 - неверная подпись в запросе (либо неверно указан secretkey в конфиге, либо запрос не от merchant)
- 0x08 - 1000 - в запросе указан кошелёк, отличный от указанного в конфигурации
- 0x10 - 10000 - тестовый режим merchant, но не тестовый режим модуля
- 0x20 - 100000 - в запросе неверно указан номер (id) платежа
- 0x40 - 1000000 - в запросе неверно указан sid договора
- 0x80 - 10000000 - в запросе неверно указана сумма платежа
- 0x0100 - 100000000 - платёж уже помечен как оплаченный
- 0x0200 - 1000000000 - платёж с таким инвойсом уже есть в базе
- 0x0400 - 10000000000 - платёж с таким номером транзакции уже есть в базе
- 0x0800 - 100000000000 - модуль в режиме выключен
- 0x1000 - 1000000000000 - в конфиге указан id не существующего типа платежа

Модуль ЮKassa

Назначение модуля

Чтобы настраивать прием платежей по этой инструкции, нужно подать [заявку на подключение ЮKassa](#) и заключить договор с компанией «ЮKassa».

Когда у вас появится личный кабинет на сайте ЮKassa, можно приступать к настройке.

В ЛК ЮKassa при настройке Магазина выберите способ подключения **протокол API**

Настройки магазина

shopId	637669
Способ подключения к ЮKassa	протокол API

Если вы хотите изменить данные, напишите нам: merchants@yoomoney.ru

Настройка модуля

Установите модуль на сервер, обновите клиент, создайте экземпляр модуля. Создайте в редакторе конфигурации модуля новую конфигурацию, сделайте её активной. Можно воспользоваться шаблоном конфигурации. Конфигурация в общем виде такая (минимально необходимо указать shopId, api.key и payment.type.id):

Конфигурация

Поиск:

Тип	Наименование	Значение
Число	yamoney.sum.min	100
Число	yamoney.sum.max	5000
Строка	yamoney.url	https://demomoney.yandex.ru/eshop.xml
Строка	yamoney.payment.types	PC:Оплата с кошелька Яндекс.Деньги;AC:Оплата с банковской кар...
Строка	yamoney.scid	
Строка	yamoney.payment.api.url	https://payment.yandex.net/api/v3/payments
Строка	yamoney.shopId	
Строка	yamoney.api.key	
Число	yamoney.payment.type.id	0
Строка	yamoney.payment.sum.amount	
Строка	yamoney.payment.sum.percent	
Строка	yamoney.authMethod	
Строка	yamoney.shopPassword	
Строка	yamoney.certificateFile	
Строка	yamoney.payment.comment	Платёж через Яндекс Касса
Логический	yamoney.autopayment.enable	true
Число	yamoney.autopayment.mode.default	0
Логический	yamoney.autopayment.mode.week.enable	false
Число	yamoney.autopayment.mode.week.day.default	1
Число	yamoney.autopayment.mode.week.sum.min	100
Число	yamoney.autopayment.mode.week.sum.max	3000
Число	yamoney.autopayment.mode.week.sum.default	100
Логический	yamoney.autopayment.mode.month.enable	false
Число	yamoney.autopayment.mode.month.day.default	1
Число	yamoney.autopayment.mode.month.sum.min	100
Число	yamoney.autopayment.mode.month.sum.max	10000
Число	yamoney.autopayment.mode.month.sum.default	100
Логический	yamoney.autopayment.mode.auto.enable	true
Число	yamoney.autopayment.mode.auto.day.sum.max	3000
Число	yamoney.autopayment.mode.auto.day.sum.default	500
Число	yamoney.autopayment.mode.auto.week.sum.max	5000
Число	yamoney.autopayment.mode.auto.week.sum.default	1000
Число	yamoney.autopayment.mode.auto.month.sum.max	10000
Число	yamoney.autopayment.mode.auto.month.sum.default	3000
Логический	yamoney.fiscalization.enable	true
Строка	yamoney.fiscalization.receipt.name	Пополнение счета по договору №{CONTRACT_TITLE}
Число	yamoney.fiscalization.receipt.qty	1
Список	yamoney.fiscalization.receipt.tax	2
Список	yamoney.fiscalization.receipt.tax.system	1
Список	yamoney.fiscalization.receipt.payment.method.type	full_prepayment
Список	yamoney.fiscalization.receipt.payment.subject.type	service
Логический	yamoney.fiscalization.customer.address.enable	true
Строка	yamoney.fiscalization.customer.address.title	Укажите E-mail или номер сотового для отправки чека
Логический	yamoney.fiscalization.customer.address.required	false
Строка	web.menuitem1	Оплата через Яндекс.Кассу

В ЛК ЮKassa пропишите URL уведомлений об операциях:

```
https://host[:port]/[bgbilling/]yakassa/<_> ( 8.2104)
https://host[:port]/[bgbilling/]yookassa/<_> ( 8.2104)
```

HTTP-уведомления

Уведомления позволяют автоматически отслеживать статусы платежей, возвратов и других объектов API. Подробности в [документации](#)

URL для уведомлений

О каких событиях уведомлять

- payment.succeeded
- payment.waiting_for_capture
- payment.canceled
- refund.succeeded

Отменить

Сохранить

Оплата из Личного кабинета

Если у абонента подключён экземпляр модуля, то он может произвести оплату услуг через Web-интерфейс (пункт меню **Оплата через ЮKassa**). Здесь же клиент может просмотреть все проведённые им платежи:

Оплата через Яндекс.Деньги

Платежный сервис Яндекс.Деньги

Оплата со счета Яндекс.Денег

Сумма (от 100.00 до 5,000.00):

Внимание! На счёт будет зачислена не вся сумма, а введённая сумма за вычетом комиссии Яндекс. Комиссия в данный момент составляет 5%. Сумма к зачислению: 0 рублей.

Текущий баланс:
99,878,861.1 руб.

Период: Август, 2016

Всего записей: 5 Страница: 1 из 1 [25]

№	Транзакция	Дата/время оплаты	Кошелёк	Сумма общая	Сумма на счёт
1	1099520089059	Aug 10, 2016 2:54:37 PM	4100312840388	10.00 демо-рубли	9.55 демо-рубли
2	1099520089066	Aug 10, 2016 2:54:37 PM	4100312840388	23.00 демо-рубли	21.96 демо-рубли
3	1099520089087	Aug 10, 2016 2:54:37 PM	4100312840388	58.00 демо-рубли	55.39 демо-рубли
4	1099520089147	Aug 10, 2016 2:54:37 PM	4100312840388	33.00 демо-рубли	31.51 демо-рубли
5	1099520089145	Aug 10, 2016 2:54:37 PM	4100312840388	22.00 демо-рубли	21.01 демо-рубли

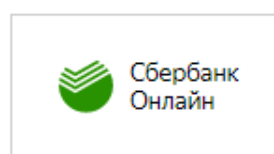
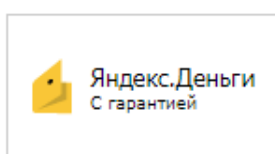
При вводе суммы и нажатии кнопки **Оплатить** клиент попадает на страницу системы ЮKassa, где может выбрать удобный для себя способ оплаты.

Яндекс Касса

Тестовый магазин

Идентификатор
плательщика 932489203

Сумма 584 Р



Все способы оплаты

Form for card payment. Includes logos for MasterCard, Maestro, VISA, and MIR. Fields for card number, expiration date (VALID THRU MM / GG), and CVC (Three digits from the back of the card).

Получить чек по электронной почте

Нажимая на кнопку, вы соглашаетесь с условиями использования сервиса

Заплатить

[← Вернуться в магазин](#)



Монитор платежей

В параметрах договора можно посмотреть все платежи и их параметры.

Транзакции		Автоплатеж					
Период:		<	>	Проверить...	Всего записей: 5		1 из 1 [25]
ID	YandexID	Время оплаты	Способ оплаты	Причина отказа	Статус	Сумма	...
1	1099520089059	10.08.2016 14:54:37			New	10,00	▲
2	1099520089066	10.08.2016 14:54:37			New	23,00	
3	1099520089087	10.08.2016 14:54:37			New	58,00	
4	1099520089147	10.08.2016 14:54:37			New	33,00	
5	1099520089145	10.08.2016 14:54:37			New	22,00	

В параметрах модуля есть глобальный монитор платежей.

Проведенные платежи		Параметры договоров (модуля)		Конфигурация модуля					
Период:		<	>	Договор:	Проверить...	Всего записей: 5		1 из 1 [50]	
ID	Договор	Комментарий догов...	YandexID	Время оплаты	Способ оплаты	Причина отказа	Статус	Сумма заказа	...
1	kinill	Сергеев Кирилл	1099520089059	10.08.2016 14:54:37			New	10,00	▲
2	kinill	Сергеев Кирилл	1099520089066	10.08.2016 14:54:37			New	23,00	
3	kinill	Сергеев Кирилл	1099520089087	10.08.2016 14:54:37			New	58,00	
4	kinill	Сергеев Кирилл	1099520089147	10.08.2016 14:54:37			New	33,00	
5	kinill	Сергеев Кирилл	1099520089145	10.08.2016 14:54:37			New	22,00	

Модуль Card

Назначение модуля

Модуль Карточки предназначен для совместного использования с модулями Inet, DialUP, и VoiceIP для автоматического создания договоров и логинов, а также для пополнения баланса договора. В модуле происходит загрузка, учёт и активация prepaid-интернет-карт. Кроме того поддерживается учёт дилеров-распространителей. С версии модуля 3.0 для дилеров поддерживается возможность удалённых платежей.

Настройка модуля

В редакторе модулей и услуг создайте модуль типа **Карточки** и добавьте туда одну услугу, например **Интернет-карты**. Подключите этот модуль всем тем договорам, которым хотите предоставить возможность пополнения баланса prepaid-картами. Услугу **Интернет-карты** добавьте в список разрешённых услуг договоров.

Перезагрузите клиент биллинга. Зайдите на вкладку **Конфигурация модуля**, с помощью кнопки **Создать** создайте конфигурацию и добавьте в неё следующие параметры:

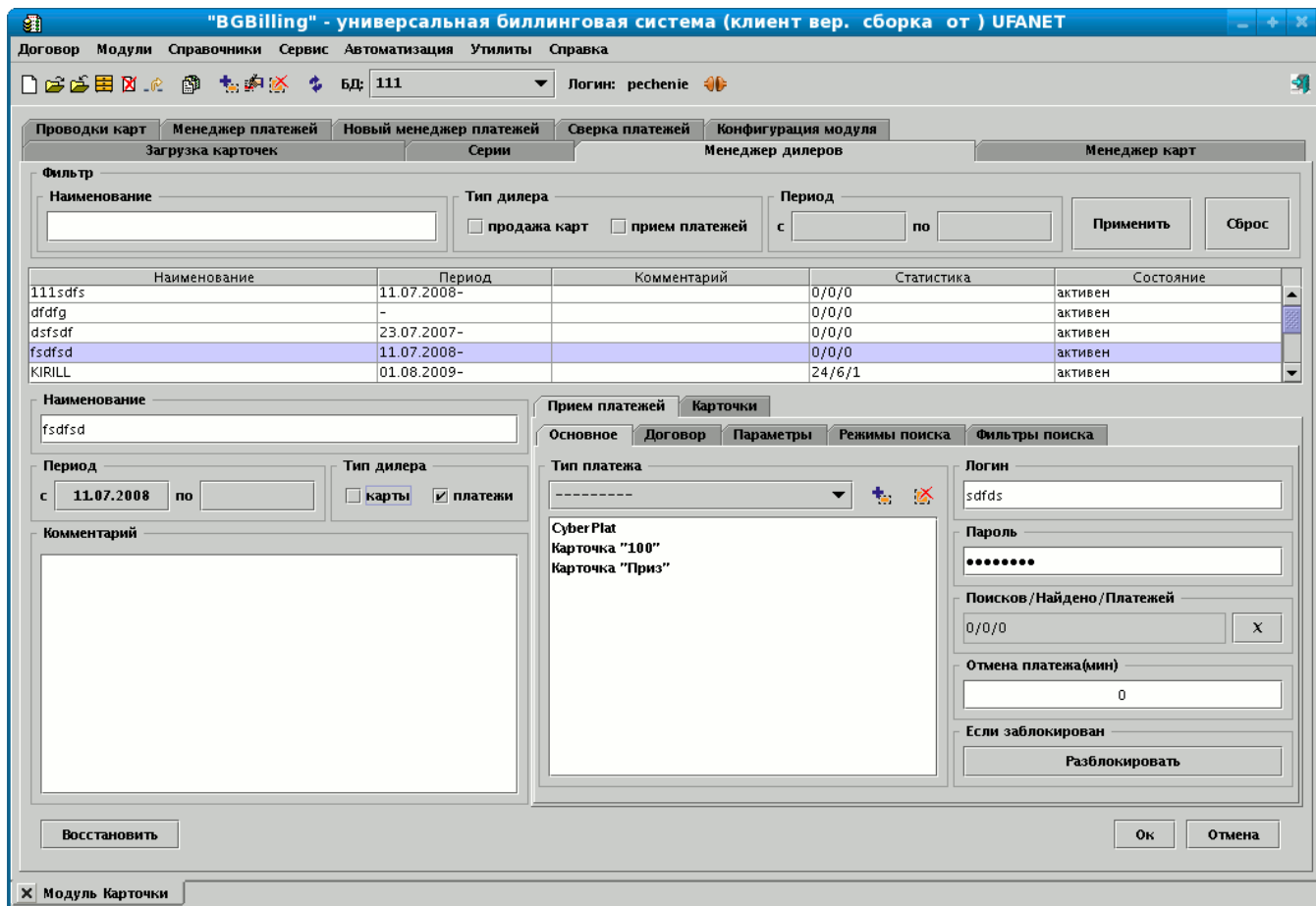
```
#
contract.pattern=T####-##
#
status.title.lock=
status.title.pay=
status.title.contract=
status.title.balance=
#
web.menuItem1=    -
#
# , :
# ${dealer} - , ${trans} - , ${contract} - , ${comment} - , ${summ} -
payment.comment= . ${contract} ${dealer} (${comment})
#
statistics.url=http://localhost:8080/bgbilling/webexecuter
```

Обратите внимание на параметр **contract.pattern**, он задаёт шаблон имени договоров, которым дилеры могут заносить платежи. Если шаблонов несколько, их нужно ввести через ";". В данном случае разрешено пополнять баланс договоров вида T5454-04, Twr34-03 и т.д. Символ # обозначает любой символ.

Дилеры

Дилеры модуля карточек могут в зависимости от установленных опций заниматься реализацией карт предоплаты и/или производить платежи в пользу провайдера через Web-интерфейс дилера.

Для добавления дилера перейдите на вкладку **Менеджер дилеров** и нажмите кнопку **Добавить** на панели инструментов.



В основных свойствах дилера присутствуют название и галочки разрешения приёма им платежей и карт. Кроме того указывается скидка в % на получение карт. Скидка на приём платежей в данной версии никак не используется.

Остальные параметры дилера, относящиеся к приёму платежей и работы с картами, разделены по вкладкам и описаны в соответствующих разделах.

Работа с карточками

Каждая карточка характеризуется 3 параметрами: **серийный код**, **логин** и **пароль**. Логин и пароль карты должны быть введены пользователем для её активации как DialUP/VoIP аккаунт, либо указаны при пополнении картой баланса через Web интерфейс, либо IVR-систему.

Серийный код и логин - числа до 2000000000, пароль - строка длиной до 32 символов латинского алфавита, либо цифр. При создании карт, предназначенных для активации посредством IVR, либо иным способом, не позволяющим вводить буквы, рекомендуется делать цифровые пароли.

Серийный номер карты используется для всех операций по управлению картой; для удобства его рекомендуется делать равным логину при первом выпуске карт. В дальнейшем по мере повторного выпуска карт с теми же логинами образовывать добавлением к логину слева цифр 1, 2, 3. Карты следует выпускать сериями с возрастающими на единицу логинами и паролями.

Файл для загрузки карт должен выглядеть подобным образом:

```
serial1<>login1<>pswd1  
serial2<>login2<>pswd2
```

Например:

```
10001 1 545454  
10002 2 545dd4
```

Вы можете использовать стандартный генератор карт, доступный на сайте, либо реализовать свой. Стандартный генератор позволяет сохранять карты в указанном выше формате.

Жизненный цикл карты включает в себя несколько состояний:

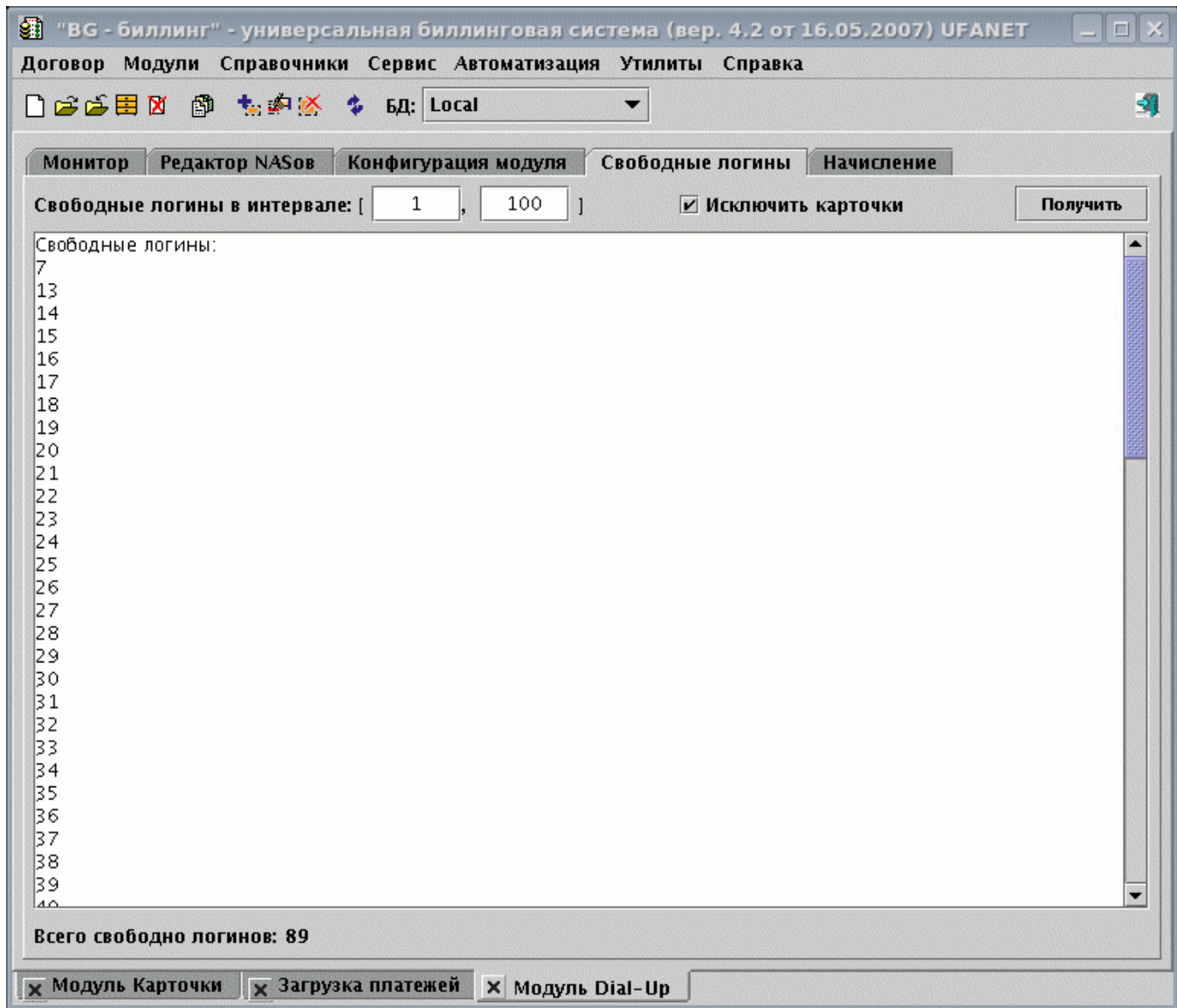
- **Заблокирована** - только что загруженная в базу карта, не назначенная дилеру и не поступившая в продажу, она не может быть активирована;
- **Активна** - карта передана дилеру и может быть активирована либо для создания договора и логина DialUP/VoiceIP, либо для оплаты услуг;
- **Договор** - карта была использована для автоматического создания договора и DialUP/VoiceIP логина в нем;
- **Баланс** - карта была использована для пополнения баланса.

Серийный номер карты должен быть уникален и никогда не повторяться, менеджер карт не даст вам загрузить карточку, если такой код уже есть в базе.

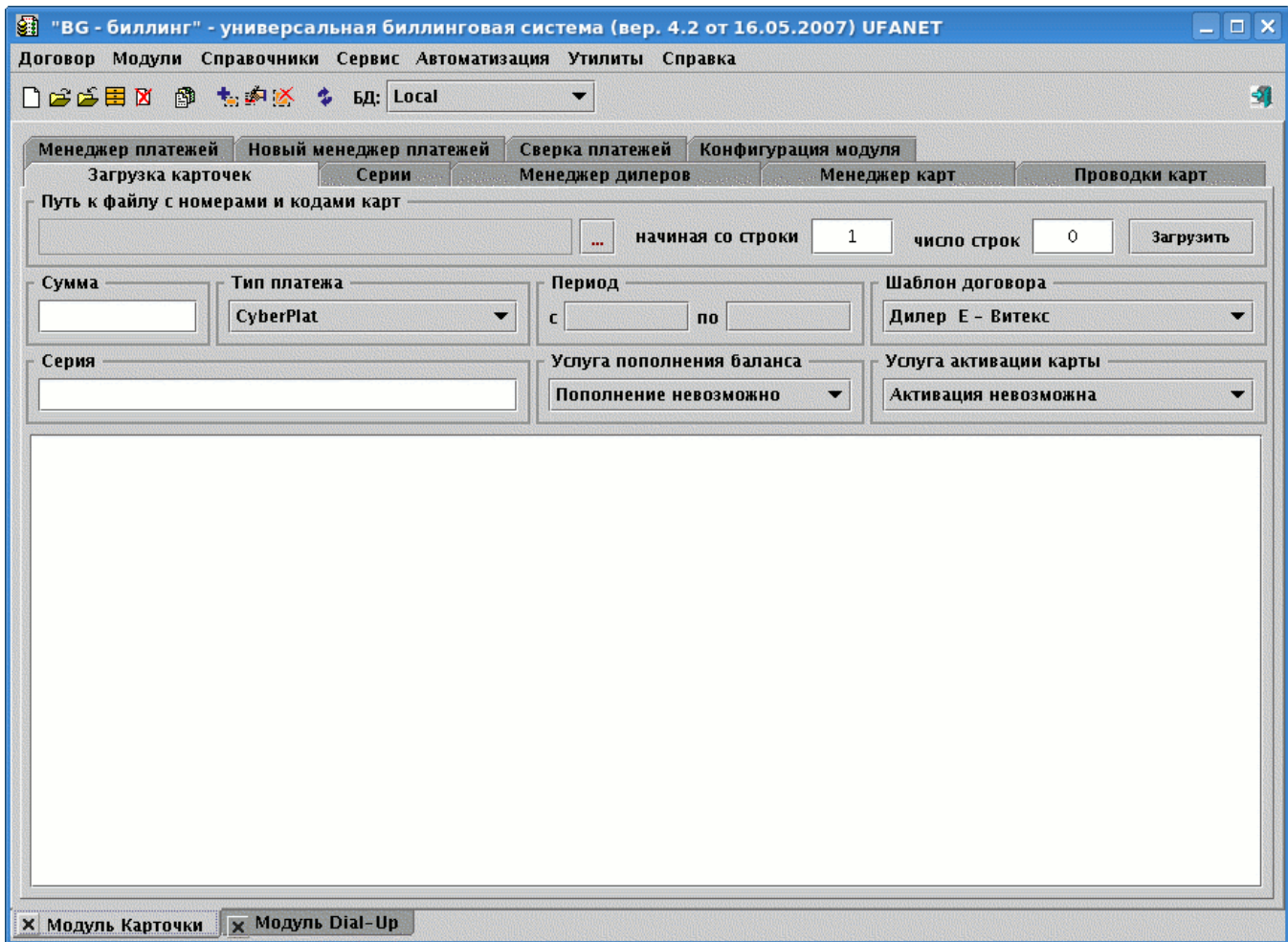
При активации карты для создания договора и логина логин и пароль карты преобразуется в логин и пароль модуля DialUP, либо модуля VoiceIP. Использовать данный логин для новой карты повторно можно только после того, как логины в модуле DialUP/VoiceIP будут освобождены и договоры удалены. Для проверки свободных логинов во всех модулях интегрирующихся с модулем карт есть вкладка **Свободные логины**.

Перед выпуском очередной серии карт следует проверить во всех модулях, где возможна активация карт данного типа в качестве логина, незанятость логинов карт. В противном случае при авторизации RADIUS-сервер будет находить старый логин и писать ошибку пароля карты. Галочка **Исключить карточки** исключает помимо логинов модуля логины карт из подключённого карточного модуля.

Карта, активированная для пополнения баланса, привязывается к договору, на который она была активирована и может оставаться в системе неограниченно долго.



Для загрузки карточек используется первая вкладка модуля **Загрузка карточек**.

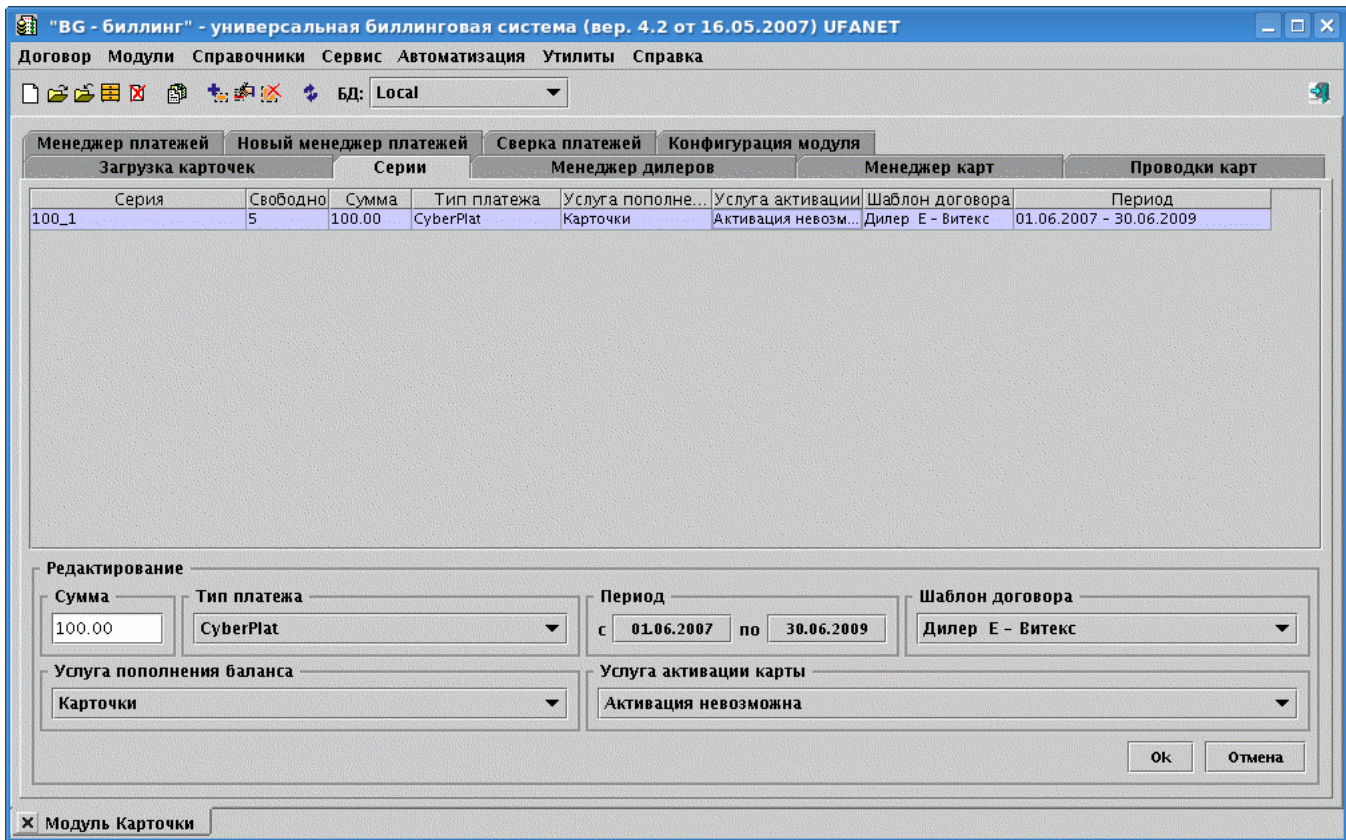


При загрузке карт указываются следующие параметры:

- **путь к файлу карт** - выбирается файл с данными карт, разделёнными табуляторами, каждая карта на новой строке (см. формат выше), при необходимости можно загрузить только часть строк из файла;
- **шаблон договора** - по этому шаблону будет создан договор при активации карты для создания договора и логина;
- **сумма** - номинал карты;
- **тип платежа** - такого типа платёж будет занесён в созданный договор; тип платежа должен быть создан в справочнике типов платежа с галочкой **нераз редактируемый**. Сумма платежа равна номиналу карточки;
- **услуга пополнения баланса** - баланс договора может быть пополнен этой карточкой только, если у договора есть в разрешённых данная услуга. Нужно это затем, чтобы возможно было создать карточки, которыми можно будет только создать договор, но не пополнять баланс;
- **услуга активации** - фильтр по услугам активации можно указывать в конфигурации NASов модуля VoiceIP, либо DialUP. Так, например, можно создать карты, активируемые только на VoiceIP NASax, и универсальные карты, активируемые везде. Реализуется заведением 2х услуг и прописыванием их двоих на VoiceIP NASax и только услуги универсальной карты на всех других;
- **период** - в течении данного периода карта должна быть активирована;
- **серия** - название серии карт; серия используется для более удобного управления загруженными картами в дальнейшем.

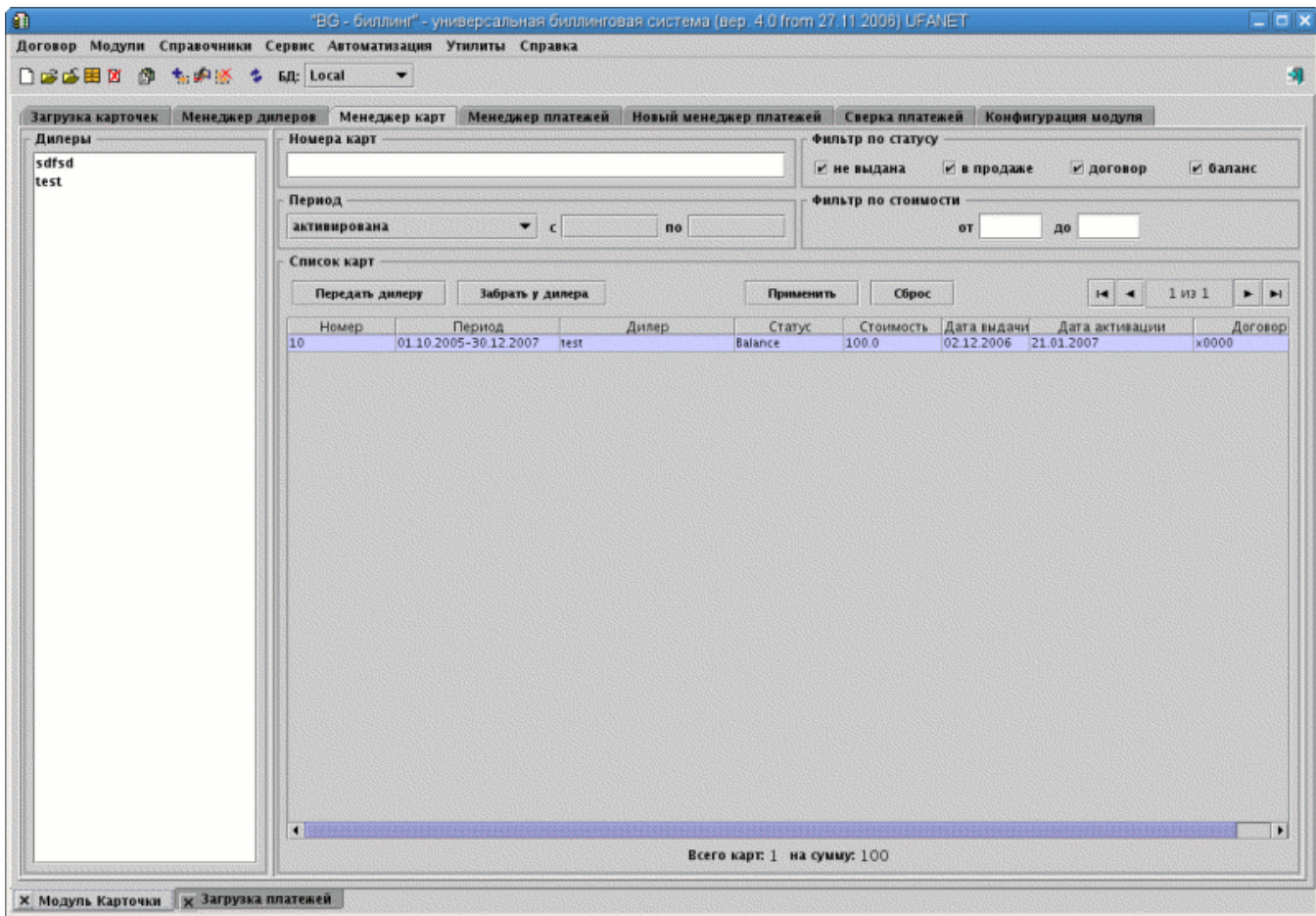
Серии следует именовать подобным образом - "Универсальные 200 ки от 24.06.07 1000 штук" - для простой идентификации группы карт в дальнейшем. Если при загрузке какие-то данные были указаны неверно, вы можете редактировать карты, используя вкладку **Серии**. Также возможно удаление ошибочно загруженной серии.

Обязательно указывайте серию для загружаемых карт.



При изменении серии изменяются свойства всех карт, входящих в неё. Обратите внимание на столбец **Свободно** - это количество свободных карт, оставшихся в серии.

Для распределения карточек по дилерам и просмотра текущего состояния базы карточек воспользуйтесь вкладкой **Менеджер карт**.



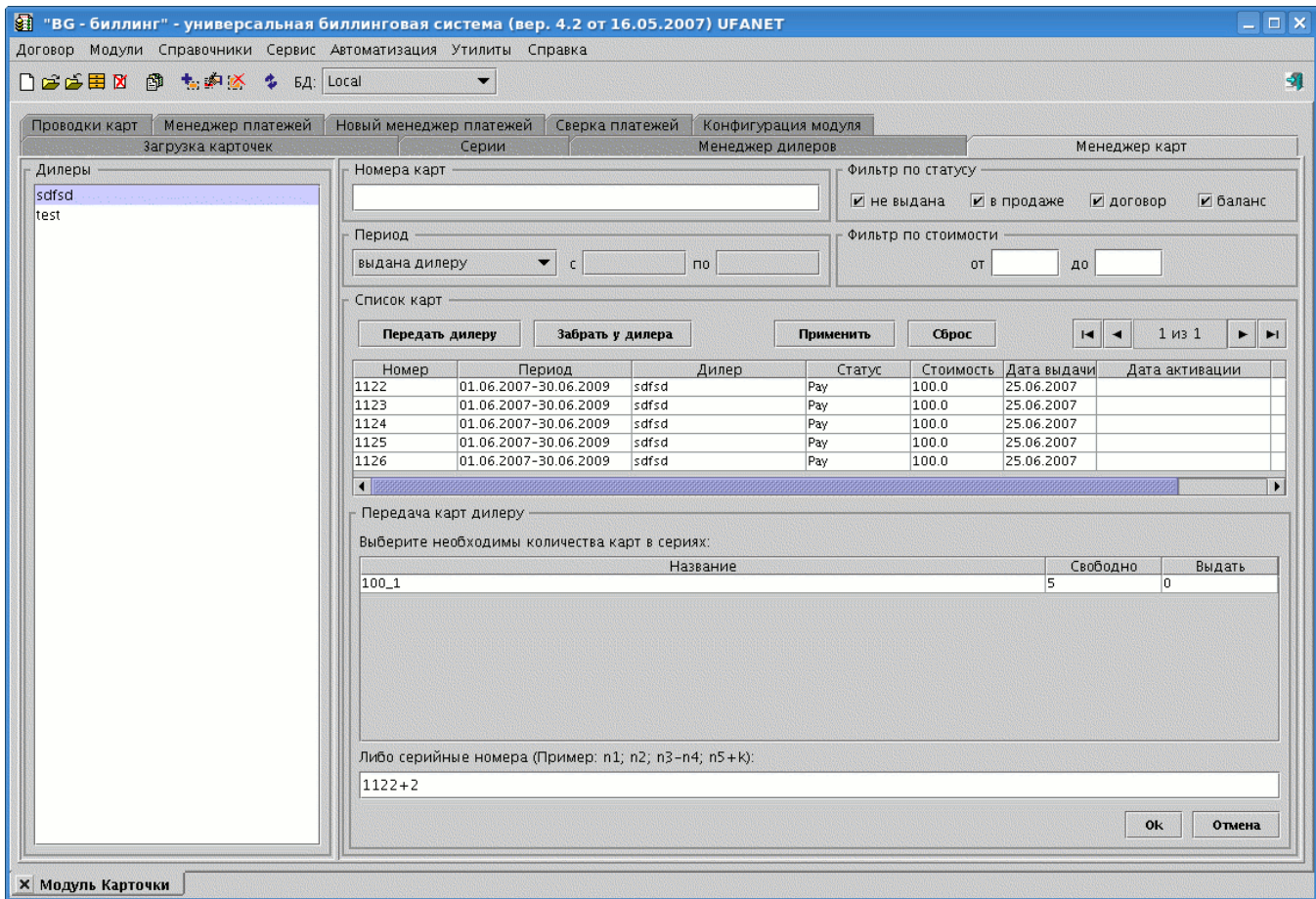
Текстовая область в верхней области фильтра - для поиска карты. Введите туда номер карты, будет отображён дилер, получивший эту карту, и её статус.

Кроме ввода одиночных номеров возможен ввод конструкций вида "N+K" (вывести карту с номером N и ещё K карт после неё). Таким образом можно выбирать диапазоны карт по серийным номерам. При пустых фильтрах система выводит все карты в постраничном режиме.

С 4.0 версии в таблице карт отображается номер договора, для создания или пополнения которого была использована карта. Двойной клик по строке открывает договор.

Набор флажков сверху позволяет выбирать нужные типы карточек. Для выбора карт, отданных какому-либо дилеру, выберите этого дилера в списке слева. Кнопка **Сброс** возвращает все фильтры в исходное состояние и убирает выделение со списка дилеров.

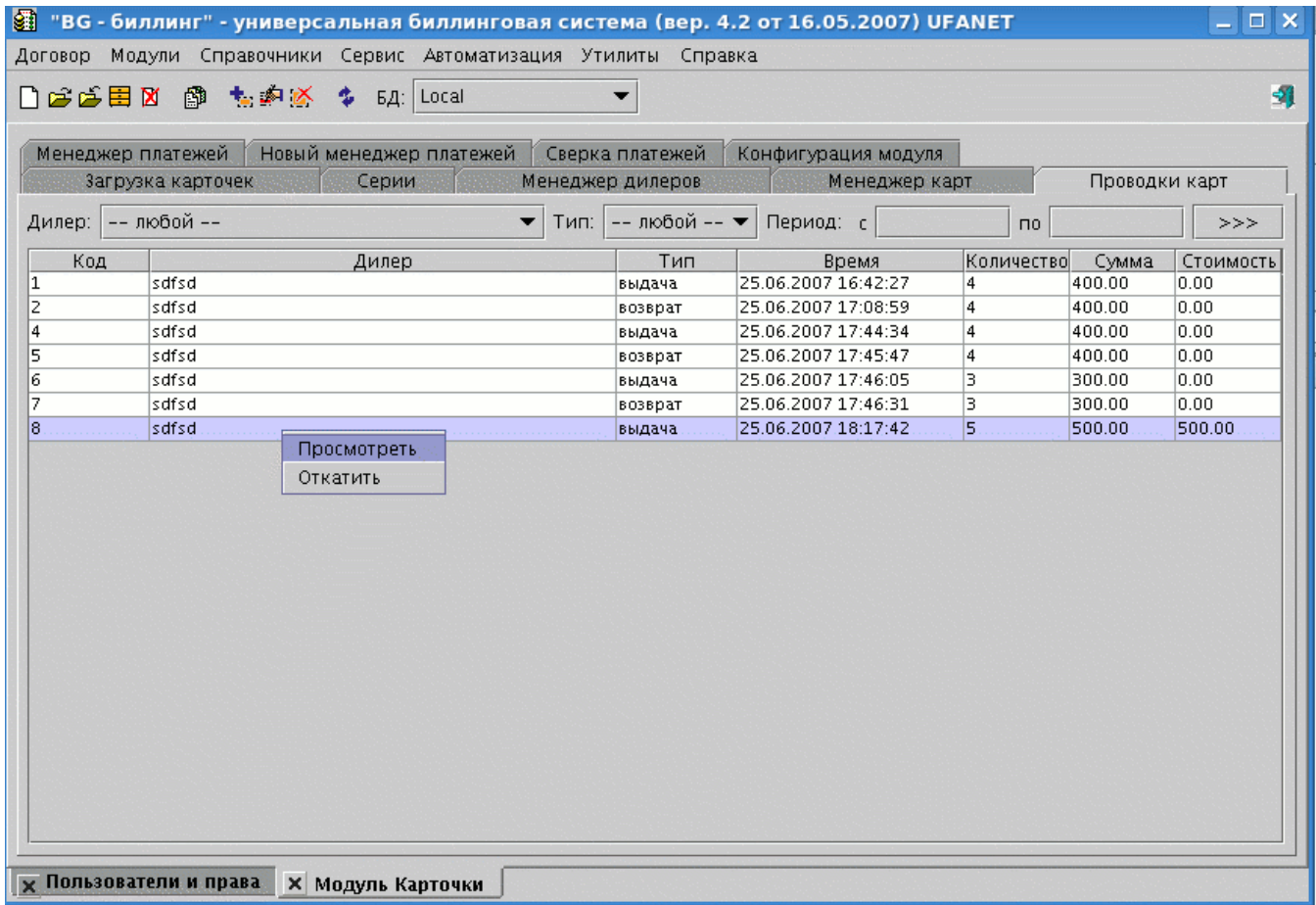
Чтобы отдать карточки дилеру на реализацию, выберите дилера в списке слева и нажмите **Передать дилеру**.



При передаче карт можно указать как количество карт выдаваемое из каждой серии (отображаются только не пустые серии), так и перечислить серийные номера карточек. При перечислении можно использовать следующие конструкции, разделённые точкой с запятой:

- **n1** - одиночный номер карты;
- **n3-n4** - серийные номера от n3 до n4 включительно;
- **n5+k** - серийный номер n5 и k карт после него.

При осуществлении передачи карт дилеру или возврата карт формируется **Проводка**, список проводок доступен на отдельной вкладке.



В таблице возможна установка фильтров по типу проводки (выдача/возврат) и по дилеру. Колонка **Стоимость** формируется с вычетом скидки, указанной в свойствах дилера. При просмотре проводки формируются Акт передачи/возврата и Заявка на выдачу/возврат карт.

Внешний вид документов задаётся шаблоном **card_action_pdf.xml**. Исходный XML документ выводится в лог **server.log** при установке его в режим **DEBUG**.

В XML документ передаётся название и дата договора с дилером, которые должны быть указаны в свойствах дилера следующим образом:

0/0/0
0/0/0

активен
активен

Прием платежей | Карточки

Договор: 4444 от 33.33.3333 г.

Ю

Ок

Отмена

"ВГ - биллинг" - универсальная биллинговая система (вер. 4.2 от 16.05.2007) UFANET

Договор Модули Справочники Сервис Автоматизация Утилиты Справка

БД: Local

Менеджер платежей | Новый менеджер платежей | Сверка платежей | Конфигурация модуля

Загрузка карточек | Серии | Менеджер дилеров | Менеджер карт | Проводки карт

1 из 2 | Печать | Размер: 100

АКТ ПРИЕМА-ПЕРЕДАЧИ № 00008 от 25.06.2007
карт предварительной оплаты

г. Уфа 25.06.2007

Мы, нижеподписавшиеся, представитель ДИЛЕРА "sdfsd" с одной стороны и представитель ПРОВАЙДЕРА ООО "Билл", с другой стороны, составили настоящий Акт о том, что во исполнение обязательств по Договору ddfdf ПРОВАЙДЕР передает, а дилер принимает на реализацию карты для продажи третьим лицам.

№	Номинал	Кол-во, шт.	Пропускно	Сумма, руб.
1	100.00	5		500.00

Серийные номера: 1122-1 126

Итого:	500.00
Скидка:	0.00 %
Итого со скидкой:	500.00

Все передаваемые карты видимых повреждений поверхности и защитного слоя не имеют.
Претензий по качеству передаваемых карт СТОРОНЫ друг к другу не имеют.
Настоящий акт составлен в двух экземплярах, имеющих одинаковую юридическую силу, по одному для каждой из сторон.

ЗАВЕРЯЮ
ООО "Иреглов"

ЗАВЕРЯЮ
sdfsd

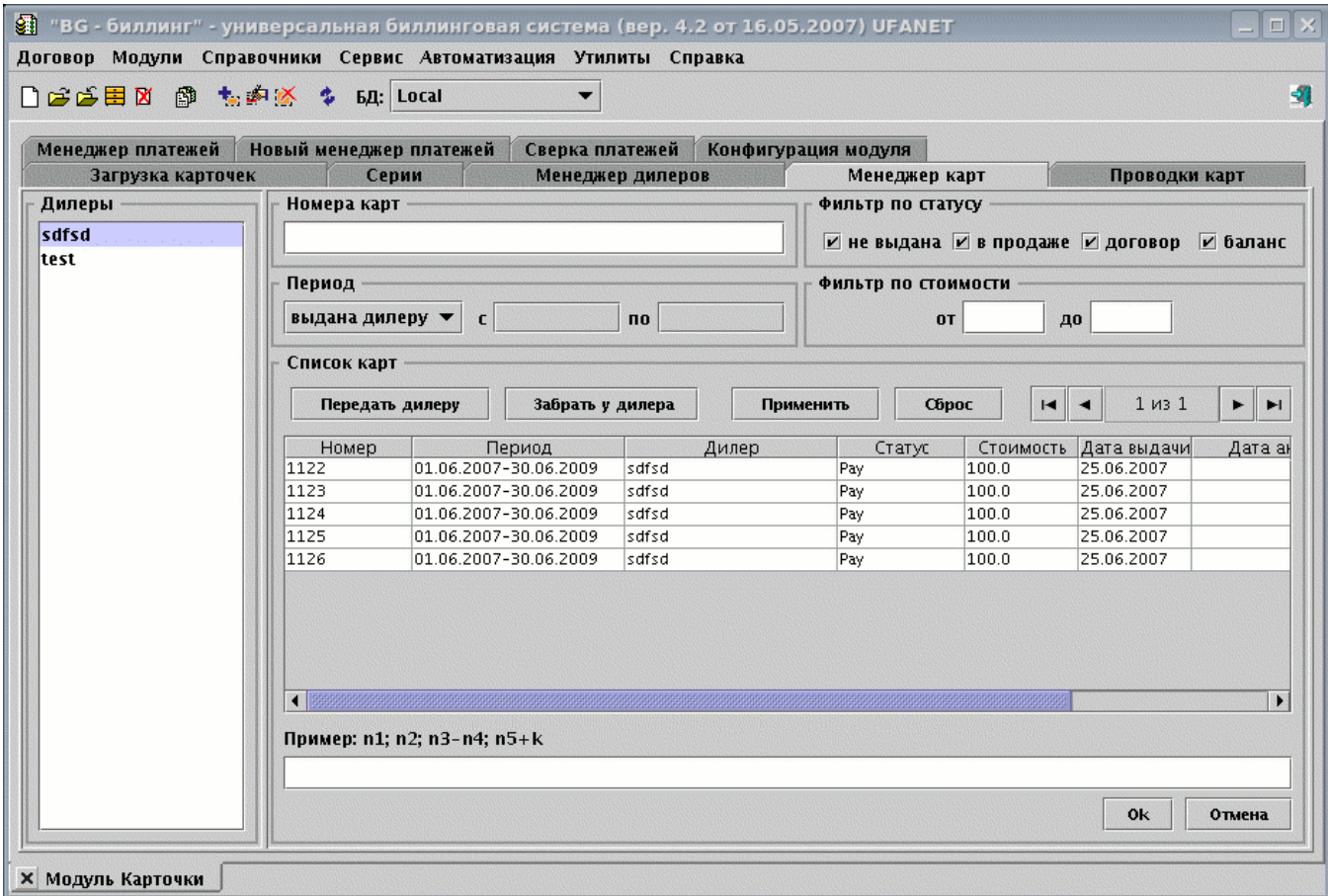
_____/_____
М.П. М.П.

Закреть

Пользователи и права | Модуль Карточки

Каждую проводку можно откатить до того момента пока какая-либо карта, участвующая в проводке, не изменила свой статус (не была активирована для выдачи, либо не была передана другому дилеру для проводок возврата).

Для возврата карт от дилера в **Менеджере карт** выберите дилера и нажмите **Забрать у дилера**. При этом можно указывать только серийные номера по тем же правилам, что и при передаче карт.



Генератор логинов и паролей для модуля "Карточки"

Установка генератора

Генератор логинов и паролей написан на Java. Следовательно, для его работы на машине должен быть установлен JDK или JRE версии 1.5.0 или выше.

Распакуйте архив с генератором в любое место вашего диска и запускайте "run.bat"

Использование генератора

После запуска генератора отображается окно.

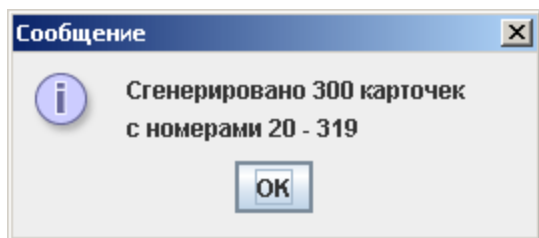
The screenshot shows the 'BGBilling - Генератор карточек' window. It features several input fields and a grid. The 'Диапазон номеров' section has 'Номера с:' and 'Количество:' fields. The 'Логины' section has radio buttons for 'Численные логины, начиная с:' and 'Файл с логинами:'. The 'Параметры пароля' section has a text field for 'Набор символов для генерации паролей:' and a dropdown for 'Длина пароля:'. The 'Экспорт' section has radio buttons for 'В файл XML' and 'В текстовый файл'. A 'Сгенерировать' button is at the bottom. The grid on the right is titled 'Заштрихуйте всю область' and is currently empty.

Допустим, вам нужно сгенерировать карточки с серийными номерами 20 - 300, логины начиная с 3.

Вводим параметры в форму, заполняем мышкой правую область (ваши движения послужат основанием для генерации случайной последовательности):

The screenshot shows the 'BGBilling - Генератор карточек' window with the following parameters filled in: 'Номера с:' is 20, 'Количество:' is 300, 'Численные логины, начиная с:' is 3, and 'Длина пароля:' is 5. The 'В текстовый файл' option is selected under 'Экспорт'. The grid on the right is now shaded yellow, indicating it has been marked for generation.

Если все прошло благополучно, будет выдано сообщение:



Далее выводится окно выбора файла, в который нужно сохранить пароли. Файл уже пригоден для загрузки в модуль Карточки.

Файл XML-формата, в котором возможно сохранение сгенерированных карт, в данный момент не загружается модулем карточек.

Web-интерфейс

Добавив какому-либо пользователю услугу из модуля Карточки, вы предоставляете ему возможность пополнять баланс карточками, которые были загружены с этим типом услуги.

Web-активация

В конфигурации модуля карточек можно включить активацию карт на dialup/voiceip модули.

В конфигурации заводится тип активации activate.login.service.x :

activate.login.service.x.title - Название активации

activate.login.service.x.sids - Услуга карты, с которой возможна активация, 0 - любая

activate.login.service.x.mid - Код модуля dialup/voiceip/inet

activate.login.service.x.tariffs - Коды тарифов через запятую, из которых клиент может выбрать нужный

activate.login.service.x.tariffs.hidden - Коды тарифов через запятую, которые будут включены в договор, дополнительно к выбранному.

activate.login.service.x.group - Группа договоров, добавляемая в договор при создании

А также параметры договора, заполняющиеся при активации activate.login.param.x:

activate.login.param.x.title - Название параметра

activate.login.param.x.pid - Код параметра

activate.login.param.x.type - Тип параметра, text - текстовый, list - список, email, flag - флаг, phone - телефон

activate.login.param.x.require - 1 - обязательно для заполнения

activate.login.param.x.require.error - Выводимая ошибка, если параметр обязателен для заполнения, но не заполнен.

activate.login.param.x.pattern - regexr параметра, если есть и не совпадает со значением, то выводится ошибка:

activate.login.param.x.pattern.error - Выводимая ошибка при несоответствии с шаблоном

Активация будет доступна по такому URL:

[http://provider:port/bgbilling/pubexecuter?action=CreateContract&module=card&mid=\\${mid}&activateType=\\${activateType}](http://provider:port/bgbilling/pubexecuter?action=CreateContract&module=card&mid=${mid}&activateType=${activateType})

где \${mid} - код модуля карточек, \${activateType} - код типа активации

При изменении xml-шаблона можно создать другой файл и указать его в конфигурации activate.login.xml=... По умолчанию используется card_create_contract.xml

Пример конфигурации:

Конфигурация

```
# 1 (activateType=1)
#
activate.login.service.1.title=Wifi
# , , 0 -
activate.login.service.1.sids=0
#
activate.login.service.1.mid=21
# ,
activate.login.service.1.tariffs=52
#
activate.login.service.1.group=25
#
#
#
activate.login.param.1.title= :
#
activate.login.param.1.pid=3
#
activate.login.param.1.require=1
# ,
activate.login.param.1.require.error=
#
activate.login.param.2.title= :
activate.login.param.2.pid=28
```

```
# -
activate.login.param.2.type=list
activate.login.param.2.require=1
activate.login.param.2.require.error=
#
activate.login.param.3.title= :
activate.login.param.3.pid=29
activate.login.param.3.type=list
activate.login.param.3.require=0
#
activate.login.param.4.title= email:
activate.login.param.4.pid=20
# - email
activate.login.param.4.type=email
# (regexp)
activate.login.param.4.pattern=^[a-zA-Z0-9_\. \-]+\@(([a-zA-Z0-9\-]+\.)+([a-zA-Z0-9]{2,4}))+$
# ,
activate.login.param.4.pattern.error= email
activate.login.param.4.require=0
#
activate.login.param.5.title=
activate.login.param.5.pid=26
# -
activate.login.param.5.type=flag
activate.login.param.5.require=1
activate.login.param.5.require.error=

#
activate.login.param.6.title=
activate.login.param.6.pid=40
activate.login.param.6.type=phone
activate.login.param.6.require=0
activate.login.param.6.require.error=
```

Создание договора по карте

Введите ФИО:

Выберите обслуживающее лицо:

Выберите статус:

Введите email:

Нужность

Выберите тарифный план:

Я принимаю договор оферты.

Введите символы, показанные на картинке:



Буквы вводятся без учета регистра.

Введите логин и пароль карты:

Номер карты:

PIN-код карты:

Создать договор

Суперкарты

Суперкарты предназначены для использования единой базы карт на нескольких серверах биллинга. При этом один из карточных модулей на одном из серверов выступает первичным хранилищем - в нем заводятся карты, идёт учёт дилеров. Остальные модули на других серверах импортируют карты из супермодуля, если карта не найдена в самом модуле.

Супермодуль должен выступать только в таком качестве. Если на этой же машине необходимы его карты следует создать ещё один модуль и установить супермодуль для него.

Подключение супермодуля производится в конфигурации клиентского карточного модуля:

```
#
#
super.db.driver=com.mysql.jdbc.Driver
super.db.url=jdbc:mysql://<IP >/bgbilling?useUnicode=true&characterEncoding=UTF-
8&allowUrlInLocalInfile=true&zeroDateTimeBehavior=convertToNull
super.db.user=<mysql >
super.db.pswd=<mysql >
super.db.maxActive=5
super.db.maxIdle=2
#
super.mid=X
# ,
super.cid=< >
#
#
super.dealer.id=X
#
super.act.sid=X
#
super.pay.sid=X
#
super.pattern.id=X
#
# <1:1>;<2:2>.. , - , -
super.payment.type=X
```

При импорте карты из супермодуля в супербазе происходит пометка карты как активированной и привязка к указанному в параметре **super.cid** договору. При импорте все параметры карты выставляются так, как указано в конфигурации, значения параметров в супермодуле значения не имеет. Вместо X должны быть установлены числовые параметры.

Для активации работы с супермодулем у клиентских модулей карточек **ОБЯЗАТЕЛЬНО** необходимо правильно заполнить **ВСЕ** параметры! Если клиентский модуль при попытке доступа к супермодулю обнаруживает какое-либо несоответствие (отсутствующий параметр, неверные реквизиты базы и т.п.), то он более не пытается осуществить подключение к супербазе до следующей перезагрузки сервера. Это объясняется тем, что, возможно, от модуля и не требуется обращения к супермодулю (вообще не указаны никакие параметры), поэтому каждый раз заново перечитывать конфигурацию нет смысла.

IVR-Система

Модуль карточек поддерживает следующие функции IVR: озвучивание информации о балансе пользователя и пополнение счета картой. При проверке баланса у пользователя запрашивается пароль доступа к Web-статистике, который должен быть цифровым. Система построена на VXML.

Набор тестовых IVR-скриптов вы можете загрузить с сервера. Архив ivr.zip содержит:

- bgivr.vxml - основной скрипт
- menu.vxml - пункты меню
- prompts - фразы

Архив необходимо распаковать и разместить на HTTP сервере, доступным с CISCO.

Общий алгоритм работы скрипта следующий:

1. с помощью голосового меню пользователь выбирает тип требуемого договора (при этом происходит определение адреса сервера биллинга и кода типа договора);
2. запрашивается номер договора без тире, при этом клиент вводит слитно и номер и две последних цифры года, происходит запрос на сервер для определения существования договора;
3. пользователь выбирает вид операции: активация карты, либо произнесение остатка на счёте;
4. если выбрано произнесение остатка, то запрашивается пароль и происходит обращение на сервер биллинга, который либо возвращает ошибку, если пароль не верен, либо генерирует VXML скрипт, произносящий баланс;
5. если выбрана активация карты, то запрашиваются слитно логин и пароль карты, которые отправляются в запросе на сервер. Карта либо успешно активируется, либо сервер возвращает код ошибки.

Для настройки скриптов в файле bgivr.vxml необходимо настроить адреса серверов биллинга и коды карточных модулей. Также необходимо установить параметр prompt_url - путь к папке с фразами на вашем HTTP сервере.

```
<var name="module" expr="'card'"/>
<var name="contentType" expr="'stream'"/>
<var name="prompt_url" expr="'http://bgbilling.bitel.ru/patch/vxml/prompts'"/>

<var name="billing_ds" expr="'http://stat1.ufanet.ru/bgbilling/pubexecuter'"/>
<var name="mid_ds" expr="'8'"/>

<var name="billing_bis" expr="'http://stat2.ufanet.ru/bgbilling/pubexecuter'"/>
<var name="mid_bis" expr="'1'"/>

<var name="billing_tks" expr="'http://stat3.ufanet.ru/bgbilling/pubexecuter'"/>
<var name="mid_tks" expr="'1'"/>
```

Также необходимо установить адреса серверов и коды карточных модулей. В приведённом примере скрипт обслуживает 3 сервера биллинга: Кабельное ТВ, Скоростной интернет и DialUP+VOIP. Далее клиент выбирает нужный тип договора.

На сервере биллинга в модуле карт должны быть настроены:

- 1) правила разделения логинов+паролей карт различной длины на, собственно, логин и пароль, например, данная строка означает, что если длина логина с паролем составляет 11 символов, то 6 из них - логин

```
#
ivr.contract.size.11.login.length=6
```

- 2) правила преобразования номера договора, введённого слитно (в данном примере - числовая часть и две цифры года на конце договора) в зависимости от кода типа договора (приходит в запросе)

```
#phone - , 1 ,
#2 phone, - 2 phone
# 444406 -> 4444-06
#
ivr.contract.type.1.pref=
ivr.contract.type.1.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
#
ivr.contract.type.2.pref=NK
ivr.contract.type.2.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
ivr.contract.type.3.pref=N
```

```
ivr.contract.type.3.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
ivr.contract.type.4.pref=NA
ivr.contract.type.4.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
ivr.contract.type.5.pref=R
ivr.contract.type.5.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
ivr.contract.type.6.pref=RK
ivr.contract.type.6.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
ivr.contract.type.7.pref=
ivr.contract.type.7.prepare=phone.substring( 0, phone.length() - 2 ) + "-" + phone.substring( phone.length() - 2, phone.length() );
```

В общем случае преобразование может быть и более сложным, язык макроса - BeanShell. Запросы на IVR-сервер и ответы сервера доступны в логe server.log, для этого необходимо установить его уровень логирования в DEBUG в файле data/log4j_server.properties.

Удалённые платежи

Система дилерских платежей. Стандартный клиент интерфейса дилера реализован на AJAX и, запускаясь в браузере дилера, позволяет проводить платежи в пользу провайдера. Обмен AJAX приложения с сервером биллинга происходит по протоколу, описанному в [Wiki](#) (раздел **Протоколы**). Данный протокол может быть использован также сторонними клиентами для интеграции систем приёма платежей с провайдером.

Настройка модуля Card

Добавьте в конфигурацию модуля карточек следующие параметры:

```
findmode.x.mode=  
findmode.x.mid=id  
findmode.x.pid=id  
findmode.x.title=
```

Возможные сочетания.

Поиск по номеру договора:

```
##( - )  
findmode.x.mode=contract  
findmode.x.title=
```

Поиск по адресу:

```
##( - )  
findmode.x.mode=address  
findmode.x.title=  
#(id , . )  
findmode.x.pid=x
```

Поиск по текстовому параметру

```
##( - . )  
findmode.x.mode=parameter  
findmode.x.title=  
#(id , . . )  
findmode.x.pid=x
```

Поиск по комментарию договора

```
##( - )  
findmode.x.mode=comment  
findmode.x.title=
```

*id параметра в справочнике можно узнать выбрав нужный элемент и нажав Ctrl+i

Поиск по логину модуля Inet

```
##( - )  
findmode.x.mode=login_inet  
#(id )  
findmode.x.mid=x  
findmode.x.title= Inet
```

Поиск по логину модуля DialUp (VPN и т.д.)

```
##( - )  
findmode.x.mode=login_dialup  
#(id )
```

```
findmode.x.mid=x
findmode.x.title= Dialup
```

Поиск по логину модуля IP телефонии (VoiceIP)

```
#{ - )
findmode.x.mode=login_voip
#{id )
findmode.x.mid=x
findmode.x.title= VoIP
```

Поиск по номеру модуля телефонии (Phone)

```
#{ - )
findmode.x.mode=phone
#{id )
findmode.x.mid=x
findmode.x.title=
```

Поиск по карточке модуля CerberCrypt

```
#{ - cerbercrypt)
findmode.x.mode=cerbercrypt
#{id )
findmode.x.mid=x
findmode.x.title=
```

Далее можно указать группы договоров или шаблон, названия договора которые разрешено находить (выбрать необходимые нужно будет в настройках параметров дилера):

```
dealer.allow.contract.x.title=
dealer.allow.contract.x.group=
dealer.allow.contract.x.regex=regex
```

При установке regex-фильтра используется [MySQL REGEXP](#). Например, фильтр по договорам частных (префиксы AA, AD, AL и т.п.) может выглядеть так:

```
dealer.allow.contract.1.title=
dealer.allow.contract.1.regex=(AA*)|(AD*)|(AL*)
```

Еще один параметр указывает разрешать ли выдавать часть результата поиска, если было найдено несколько договоров

```
dealer.findcontract.fewresults= 1 | 0
```

В web-интерфейсе дилера существует возможность вывода тарифных планов договора при проведении платежа. Для этого следует добавить в конфигурацию параметр

```
idealer.tariff.ids=X,Y,...,Z
```

Здесь **X**, **Y** и **Z** - это коды тарифных планов. Порядок их расположения определяет порядок вывода тарифных планов в интерфейсе дилера. Например, сперва можно указать все тарифные планы телефонии (например, 10,11,12), а далее все тарифные планы абонлат (например, 23, 24, 26). Тогда при различных комбинациях этих тарифов у клиентов в любом случае на первом месте будет стоять один из тарифов телефонии, а затем уже какой-либо тариф абонлат. Не указанные в перечислении тарифные планы не отображаются. Для отображения персональных тарифных планов (всех сразу) следует указать код 0. При этом порядок также имеет значение.

Далее пример конфигурации:

Конфигурация

```
#
findmode.1.mode=contract
findmode.1.title=

#
findmode.2.mode=address
findmode.2.title=
#id -
findmode.2.pid=19

# () dialup
findmode.3.mode=login_dialup
findmode.3.mid=21
findmode.3.title= Dialup

# () VPN
findmode.4.mode=login_voip
findmode.4.mid=22
findmode.4.title= VPN

# Voip
findmode.5.mode=login
findmode.5.mid=6
findmode.5.title= VoIP

# Phone
findmode.6.mode=phone
findmode.6.mid=20
findmode.6.title=

# Cerbercrypt
findmode.7.mode=cerbercrypt
findmode.7.mid=43
findmode.7.title=

#

#
#
dealer.allow.contract.1.title= VIP
dealer.allow.contract.1.group=1,10,17
# , regexp
#( : .- , \d-, \w-;/ * - - ,
#+ - , ? - )
dealer.allow.contract.2.title=
dealer.allow.contract.2.regexp=K.*
# regexp group ,
#
dealer.findcontract.fewresults=1
#
# idealer'. ,
# 10, 13, 24 ( )
idealer.tariff.ids=0,10,13,24
```

Настройка дилера

В свойствах дилера по приёму платежей должны быть указаны следующие опции:

Прием платежей | Карточки

Основное | Договор | Параметры | Режимы поиска | Фильтры поиска

Тип платежа

Cyber Plat
Карточка "100"
Карточка "Приз"

Логин

sdfds

Пароль

.....

Поисков/Найдено/Платежей

0/0/0

X

Отмена платежа(мин)

0

Если заблокирован

Разблокировать

- **Логин** - логин дилера в системе удалённых платежей;
- **Пароль** - пароль дилера в системе удалённых платежей;
- **Тип платежа** - каким типом могут быть обозначены платежи в приходе договора, варианты типов платежа должны быть помечены как **нередактируемые** в справочнике типов платежей. При проведении платежа в интерфейсе дилера имеется возможность выбрать один из указанных типов платежей;
- **Отмена платежа** - в течении скольких минут после проведения дилер может отменить платёж;
- **Поисков/Найдено/Платежей** - сколько было попыток поиска/найдено договоров и проведено платежей - для отслеживания попыток дилера просканировать абонентскую базу;
- **Разблокировать** - в случае нескольких неудачных попыток ввода пароля логин дилера блокируется и может быть разблокирован этой кнопкой.

Прием платежей | Карточки

Основное | Параметры | Режимы поиска | Фильтры поиска

Тестовый режим

Все | Сброс | Инверт.

Ok | Отмена

- **Тестовый режим** - платежи дилера не будут реально попадать на баланс договора, необходим для отладки взаимодействия;

Прием платежей | Карточки

Основное | Параметры | Режимы поиска | Фильтры поиска

- Номер договора
- Адрес
- Логин Dialup
- Логин VoIP
- Телефон
- Карта цифрового телевидения
- Комментарий договора
- Параметр
- Цена тарифа

Все Сброс Инверт.

Ok Отмена

На вкладке **Режимы поиска** указываются признаки, по которым дилер может искать абонента. Сами признаки описываются в конфигурации модуля (см. выше).

Прием платежей | Карточки

Основное | Параметры | Режимы поиска | Фильтры поиска

- Частники

Все Сброс Инверт.

Ok Отмена

Если не указаны **Фильтры поиска**, дилеру будет позволено находить любые договоры. Фильтры позволяют выделить только те договоры, на которые дилер может принимать платежи.

Начиная с 4.3 версии, каждый дилер может быть привязан к договору, на балансе которого ведётся учёт платежей от дилера с учётом комиссии.

Прием платежей Карточки

Основное Договор Параметры Режимы поиска Фильтры поиска

Тип комиссии
сложная комиссия ▼

Размер комиссии

Договор
 >>> X

Ок Отмена

Привязываемый договор должен быть открыт. Система поддерживает два вида комиссии: **простую** и **сложную**. Кроме того, возможна работа без комиссии. Размер комиссии указывается в процентах в свойствах дилера.

При простой комиссии на баланс договора будет положена наработка, равная сумме, оплаченной клиентом, минус агентское вознаграждение (размер комиссии).

При сложной комиссии на баланс договора начисляется наработка, равная сумме, внесённой клиентом, и начисляется отрицательная наработка, равная агентскому вознаграждению. В результате на балансе дилера образуется долг, равный сумме проведённых платежей, минус его агентское вознаграждение. Например, баланс дилера может выглядеть так:

с 01.10.2006 по 23.10.2007		М П 01 02 03 04 05 06 07 08 09 10 11 12											
0.0 дебет		Приход		Расход		Нарработка		Баланс					
Дата	Услуга												
2007 июль	Платежи									291,04			
2007 июль	Агентское вознаграждение									-58,21			

Коды услуг наработки агента и агентского вознаграждения задаются в конфигурации модуля.

```
# ""
dealer.pay.sid=105
# ""
dealer.bonus.sid=106
```

При привязке дилера к договору дилер сможет проводить платежи только, если у него в договоре сумма остатка больше лимита, либо если в договоре стоит режим КРЕДИТ.

Web-клиент

Web-клиент по умолчанию находится по адресу **http(s)://server:port/bgbilling/id/**

web-клиент является AJAX (active javascript and xml) приложением, работающим под IE, Firefox, Opera, реализующим интерфейс удалённых платежей модуля.

Дополнительно необходимо произвести настройку файла **BGBillingServer/webroot/id/conf.txt**, указав код модуля карт.

```
//  
mid = < >  
//  
page_size = 15  
  
// ,  
// ,  
// cookies  
//prefix = ""  
  
// ,  
//( )  
// ,  
// cookies  
//postfix = ""
```

При открытии страницы запрашивается логин и пароль дилера. Далее дилер может осуществлять поиск договора, пополнение счета, отмену платежа (при ошибочном платеже, см. Отмена платежа выше), просмотр выполненных/отмененных платежей.

Поиск договора

Номер договора	<input type="text"/>
Адрес	<input type="text"/> / <input type="text"/> кв. <input type="text"/>
Логин Dialup	<input type="text"/>
Логин VoIP	<input type="text"/>
Телефон	<input type="text"/>
Карта цифрового телевидения	<input type="text"/>

Найти

Сверка платежей

Формат реестра платежей:

<номер платежа>\t<время проведения>\t<договор>\t<сумма>

\t - знак табуляции;

время проведения в формате dd.MM.yyyy HH:mm:ss или yyyy-MM-dd HH:mm:ss

Например: 2147483647 2006-03-03 13:56:39 x0000 10

Модуль RentSoft

Назначение модуля

Данный модуль позволяет абонентам работать с услугами платформы [Rentsoft](#) от компании ООО "Рентсофт". Компания Рентсофт является дистрибьютором программных продуктов и цифрового контента с ежемесячной оплатой по подписке. Рентсофт заключила лицензионные соглашения и осуществила интеграции с большинством производителей антивирусного программного обеспечения. Перечень программного обеспечения и цифрового контента, распространяемых по подписке, постоянно расширяется, делая предложение все более и более востребованным партнерами и конечными пользователями. Интеграция BGBilling с данной платформой производится посредством вставки IFRAME на страницу пользователя в личном кабинете.

Базовые понятия и алгоритм работы модуля

Модуль состоит из трех базовых частей:

- страничка в личном кабинете, в которую посредством вставки IFRAME интегрируется площадка по продаже ПО по подписке;
- сервлет на сервере BGBilling, посредством которого удаленные сервера Rentsoft "уведомляют" биллинговую систему о покупке, изменении статуса подписок и т.п. (должен быть доступен "извне");
- клиентская часть (для операторов и администраторов биллинга), через которую возможен просмотр наличия и статуса подписок и платежей конкретного клиента, а также список продуктов (пополняется в реальном времени в процессе обращения пользователей к тем или иным продуктам).

Таким образом все операции по покупке, продлению подписок, отказа от них и т.п. осуществляются на серверах Rentsoft посредством встроенного в личный кабинет IFRAME. На сервер биллинг через открытое API (сервлет) отправляются уведомления об этих операциях, в соответствие с чем договорам устанавливается определенная наработка и списание средств.

API модуля (сервлет) биллинга доступно по URL:

```
http[s]://host[:port]/bgbilling//rentsoft_api/<__Rentsoft>
```

Установка и настройка модуля

Модуль устанавливается с помощью утилиты **bg_installer**, после чего необходимо создать его экземпляр. После создания экземпляра модуля необходимо создать услугу (одну) для данного модуля.

На вкладке **Конфигурация** создайте и установите конфигурацию модуля.

```
# ,
contract.status.active.codes=0

# c ( "   IFRAME"
#   "   API   " rentsoft.ru)
rentsoft.secret=some-secret-string
#   ( "   AG_NAME"   rentsoft.ru)
rentsoft.ag_name=your-operator-name
# IFRAME   .
rentsoft.iframe_width=880px
```

Инструкции по активации услуг компании RentSoft

После установки и настройки модуля необходимо заключить сублицензионный договор с компанией RentSoft, передающий интернет-провайдеру права на предоставление услуги подписки конечным пользователям.

Контакты RentSoft:

- e-mail: partner@rentsoft.ru
- тел. +7 (499) 504-98-75

Клиентская часть модуля RentSoft

В меню модуля Rentsoft на вкладке **Используемые продукты** отображаются продукты, подписки на которые уже были куплены клиентами биллинга. В общем случае здесь не хранятся все возможные для покупки продукты, эта таблица носит информационный характер и заполняется по мере поступления запросов на покупку подписок от клиентов.

Используемые продукты		Конфигурация модуля	
Код	Ключ продукта	Название продукта	...
4	first_product	Первый продукт	▲

После добавления модуля к договору можно просматривать его подписки, их статусы, а также все транзакции, производимые по конкретным подпискам. Также в договоре возможно вызвать переобсчет начислений для нужного месяца.

Свойства		
Подписки		
Выполнить переначисление за		
◀	Август ▼ 2010 ▼ ▶	
OK		
Продукт	Статус	...
Первый продукт	активна	▲

Код транзакции	Дата	Стоимость	Комментарий	Статус	...
tx_0_1281515228721	11.08.2010 14:27:19	10	Первое списание	отменен	▲
tx_0_1281513653801	11.08.2010 14:26:43	10	Списание, которое будет от...	отменен	
tx_3_1281515198414	11.08.2010 14:26:39	10	Первое списание	оплачен	
tx_3_1281515189936	11.08.2010 14:26:35	10	Первое списание	оплачен	
tx_0_1281515187812	11.08.2010 14:26:25	10	Первое списание	оплачен	
tx_0_1281515185567	11.08.2010 14:26:24	10	Первое списание	оплачен	
tx_0_1281515181874	11.08.2010 14:26:22	10	Списание, которое будет от...	оплачен	
tx_0_1281513404821	11.08.2010 13:56:45	10	Списание, которое будет от...	отменен	
tx_0_1281513340953	11.08.2010 13:55:44	10	Списание, которое будет от...	отменен	
tx_0_1281513336416	11.08.2010 13:55:37	10	Списание, которое будет от...	отменен	
tx_0_1281513330873	11.08.2010 13:55:33	10	Первое списание	оплачен	
tx_0_1281513334514	11.08.2010 13:55:33	10	Первое списание	оплачен	


Работа с юридическими лицами

Предоставление услуги "Подписка на ПО" для юридических лиц несколько сложнее, чем для физических лиц. Главное отличие заключается в том, что согласно правилам ведения бухгалтерии в РФ списания за услугу необходимо привязывать к отдельному договору (или субдоговору) данного юридического лица с отдельным балансом, созданному специально для услуги "Подписка". В момент заказа юрицом услуги (набора юрицом корзины в кабинете статистики) субдоговор для Рентсофт может еще у него не существовать. Поэтому в качестве базового идентификатора, к которому осуществляется привязка услуг системой Рентсофт, выступает идентификатор основного договора (того, под которым юрицо заходит в личный кабинет). Это позволяет юрицу начать пользоваться услугами с триальным периодом (которых абсолютное большинство) сразу же, не ожидая заключения нового договора, а просто приняв оферту.

Схема, когда провайдер самостоятельно подключает продукты юрицу

Если продажа продукта осуществляется самим провайдером при посещении им клиента, реализуется следующий сценарий.

1. Провайдер создает юрицу договор (или субдоговор), предназначенный для работы с услугами Рентсофт, и выдает клиенту логин и пароль от этого договора (или субдоговора). Внимание: в названии или комментарии договора обязательно должна присутствовать подстрока "rentsoft". Это разрешает модулю Рентсофт осуществлять списания по данному договору, а заодно и служит целям наглядности при будущих поисках договоров по реквизитам юриц. Для удобства можно добавить строку "rentsoft" в шаблон названия договора – например, "rentsoft- $\{N6\}$ ".
2. Клиент заходит в кабинет статистики, выбирает необходимые услуги.
3. В некоторый момент юрицо попадает на страницу "Пополнить счет" в IFRAME (это происходит, например, когда подключается платная услуга):

 Данные сохранены.

Рекомендуемая сумма пополнения счёта — **450 руб.** После поступления денег на счёт нужно будет зайти в личный кабинет и активировать выбранные подписки.

Суммарная стоимость подписок до конца следующего месяца составит **450 руб.**

Счёт будет выставлен на сумму руб.


Выставить счёт

4. Юрицо нажимает кнопку "Выставить счет", и в бухгалтерию интернет-провайдера высылается письмо-уведомление следующего содержания: "Юрицо с реквизитами XXX просит заключить договор (если он еще не заключен ранее) и выставить ему счет на сумму YYY руб за услуги подписки".
5. Бухгалтерия получает письмо, находит в административном интерфейсе биллинга договор юрица. Провайдер выставляет юрицу счет на указанную сумму. Как только деньги поступают, провайдер зачисляет их на счет договора.
6. Юрицу автоматически приходит уведомление о том, что деньги поступили, и он может зайти в свой личный кабинет и включить /разблокировать услуги.
7. В дальнейшем всякий раз, когда юрицу будет требоваться пополнение счета, ответственному лицу в провайдере будет отправляться уведомление об этом, и работа будет продолжаться с п. 6. Для новых пополнений создавать субдоговор, конечно, уже не потребуется – можно будет использовать уже существующий.

Схема, когда юрицо самостоятельно выбирает и подключает услуги

Отличается от случая выше тем, что в момент, когда юрицо принимает решение воспользоваться услугами Рентсофт на витрине с продуктами, у него еще не существует отдельного договора/субдоговора для подписок. Поэтому в качестве идентификатора договора, по которому Рентсофт идентифицирует абонента, всегда выступает идентификатор основного договора. Сценарий следующий:

1. Юридическое лицо заходит в свой кабинет статистики и в IFRAME Рентсофт выбирает продукты, которыми хочет пользоваться.
2. Для большинства продуктов имеется триальный период, так что клиент может подключить их и начать работать немедленно: подписки активизируются и сразу же становятся доступными в личном кабинете.
3. В процессе подключения запрашиваются реквизиты юрица.
4. В некоторый момент юрицо попадает на страницу "Пополнить счет" в IFRAME (это происходит, например, когда подключается платная услуга):

 Данные сохранены.

Рекомендуемая сумма пополнения счёта — **450 руб.** После поступления денег на счёт нужно будет зайти в личный кабинет и активировать выбранные подписки.

Суммарная стоимость подписок до конца следующего месяца составит **450 руб.**


Счёт будет выставлен на сумму руб.

Выставить счёт

- Юрлицо нажимает кнопку "Выставить счет", и в бухгалтерию интернет-провайдера высылается письмо-уведомление следующего содержания: "Юрлицо с реквизитами XXX просит заключить договор (если он еще не заключен ранее) и выставить ему счет на сумму YYY руб за услуги подписки".
- Бухгалтерия получает письмо, находит в административном интерфейсе биллинга основной договор юрлица и создает к нему субдоговор для услуг Рентсофт (если он еще не был ранее создан). Внимание: в имени субдоговора или в комментарии к нему обязательно должна присутствовать подстрока "rentsoft". Именно по наличию данной подстроки модуль определяет, что списания за подписки (которые, напомним, уже могут быть созданы на шаге 2) надо привязывать к данному субдоговору.
- Провайдер выставляет юрлицу счет на указанную сумму. Как только деньги поступают, провайдер зачисляет их на счет субдоговора.
- Юрлицу автоматически приходит уведомление о том, что деньги поступили, и он может зайти в свой личный кабинет и включить /разблокировать услуги:


Все продукты

Мои подписки


-  У Вас есть подписки, которые Вы должны включить вручную. Для этого:
- 1) нажмите на ссылку названия продукта, рядом с которой расположена красная точка;
 - 2) отметьте требуемые подписки;
 - 3) нажмите кнопку "Включить".
- Если на Вашем счете достаточно средств, подписки станут активными.

Артикул

Наименование

KK5450  [Kaspersky Small Office Security \(лицензия на 5 ПК\) \(1\)](#) 450 руб/мес x 1 шт = 450 руб/мес



 **Укажите, кто пользуется**

Ожидает явного включения Вами - включите, чтобы оплатить и начать использование

[Ссылка на скачивание, пароли](#)

9. В дальнейшем всякий раз, когда юрлицу будет требоваться пополнение счета, бухгалтерии провайдера будет отправляться уведомление об этом, и работа будет продолжаться с п. 6. Для новых пополнений создавать субдоговор, конечно, уже не потребуется – можно будет использовать уже существующий.

Автоматизация: создание шаблона для договора по услугам Рентсофт

Т.к. создание договора для Рентсофт – частая операция, можно создать специальный шаблон договора со следующими параметрами:

- Название шаблона: "Договор Рентсофт".
- Шаблон имени договора: "rentsoft-{\$N6}" (к примеру). Если вы не хотите указывать строку "rentsoft" в имени договора, можно добавить ее в комментарий, воспользовавшись функцией "Шаблоны комментариев".
- Список модулей: добавлен модуль "Рентсофт".
- Лицо: "Ю" (юридическое).
- Режим: "К" (для предоплатной системы предоставления услуги) или "Д" (для предоставления в постоплатном режиме; в этом случае нужно также задать лимит).

Шаблоны

Договор RentSoft

Название шаблона

Шаблон имени

Лимит

Статус

Лицо

Режим

Время жизни (дни, 0 - неограниченно)

Модули | Группы | Шаблоны комментариев | Параметры | Группы тарифов и тарифы | Скрипт поведения | Объекты

РентСофт

Свойства недоступны

Восстановить

Ок Отмена

В дальнейшем при поступлении от юрлица заявки на пополнение счета, если субдоговор еще не существует, его нужно создать с параметрами:

- Шаблон – "Договор Рентсофт".
- Создать субдоговор как договор для ... - выбрать основной договор юрлица.

Web интерфейс модуля RentSoft

При добавлении модуля в договор, в web интерфейсе пользователя появляется пункт меню **Ваши подписки**. Здесь располагается iframe с онлайн витриной подписок от компании RentSoft.

Ваши подписки

К сожалению, у Вас нет ни одной подписки.

Если Вы точно знаете, какой продукт Вам нужен, воспользуйтесь блоком "Быстрая подписка" на этой странице.

Быстрая подписка

Outpost Antivirus Service — 49 руб.

Не хотите спешить? [Изучите продукты повнимательнее.](#)

Рекомендуем попробовать один из этих продуктов



После выбора необходимого продукта появится окно с информацией о покупке, полем для ввода E-mail, на который придет информация о ней, а также лицензионное соглашение. При согласии с условиями необходимо нажать кнопку Оплатить и подключить, после чего на указанный E-mail придет письмо с дальнейшими инструкциями.



i Для чего нужен e-mail?

На указанный e-mail придет письмо с данными для скачивания и активации ПО

1. Проверьте параметры подключения

Вы собираетесь подписаться на «Outpost Antivirus Service» за **49** руб. в месяц.

С учётом перерасчёта за неполный расчётный период стоимость подписки составит **31** руб. до **31.08.2010** включительно.

После оплаты Вы получите доступ к установочному файлу, и подписка станет активной.

2. Введите E-mail

на него придет письмо с параметрами для подключения ПО

3. Подтвердите согласие с сублицензионным договором

Уважаемый пользователь! Перед загрузкой, установкой, копированием или использованием программного обеспечения (как определено ниже) прочитайте изложенные ниже положения о применении этого программного обеспечения. Отметив пункт "Я принимаю условия договора" внизу, равно как загружая, устанавливая, копируя или иным образом используя программное обеспечение, вы (от своего имени как физическое лицо или от имени организации как ее агент и/или работник) соглашаетесь со всеми условиями

Я принимаю условия договора

Оплатить и подключить

[Вернуться к списку подписок](#)

Модуль TrayInfo

Назначение модуля

Модуль предназначен для предоставления пользователям использования утилиты TrayInfo — это маленькая программа, висящая в System Tray Windows и отображающая баланс пользователя. Кроме того она позволяет осуществлять мгновенный доступ к странице статистики, скрывая от пользователя процесс авторизации. Имеется версия TrayInfo3 под linux.

Настройка модуля

В конфигурации модуля можно указать название пункта меню Web-статистик пользователя.

В **Редакторе модулей и услуг** создайте экземпляр модуля TrayInfo. Зайдите в **Модули=><Название созданного экземпляра>**, откройте вкладку **Конфигурацию модуля**, создайте конфигурацию и укажите в ней.

```
web.menuItem1= TrayInfo
```

Подключите модуль всем клиентам, которым нужно предоставить возможность использования TrayInfo.

Создание типов логинов

После открытия созданного модуля TrayInfo на вкладке **Логины TrayInfo** — перечень видов логинов. Для каждого вида указывается его цена, тип расхода который будет добавлен в договор, период действия вида логина и срок действия логина. Для установки бесконечного срока действия, выберите соответствующий пункт из выпадающего списка или укажите срок ноль дней. Тип расхода должен быть помечен как **нередатируемый** в справочнике расходов.

Активация TrayInfo клиентом

Для того чтобы клиент смог использовать утилиту TrayInfo необходимо подключить к договору модуль TrayInfo. После этого клиент должен зайти на страницу статистики выбрать меню **Активация логина TrayInfo** и выбрать PIN2. После этого клиент получит PIN1 + PIN2 и с его счета будет взыскана сумма соответствующая выбранному типу логина. Чтобы посмотреть PIN2 в таблице его нужно выделить мышью.

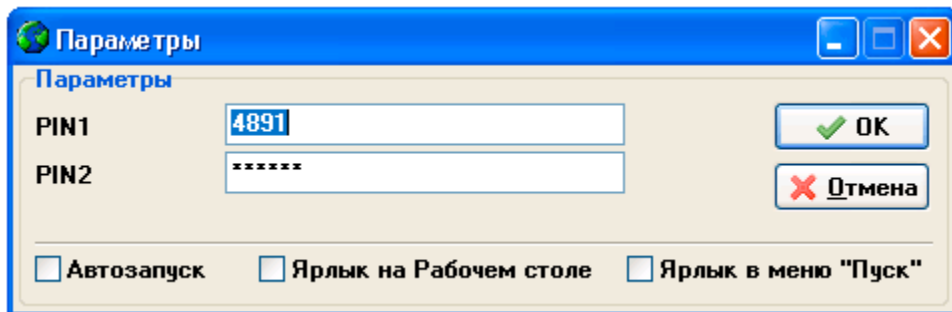
Настройка утилиты TrayInfo

Необходимо изменить ресурсы exe-файла приложения, чтобы указать URL баланса и Web-статистики.

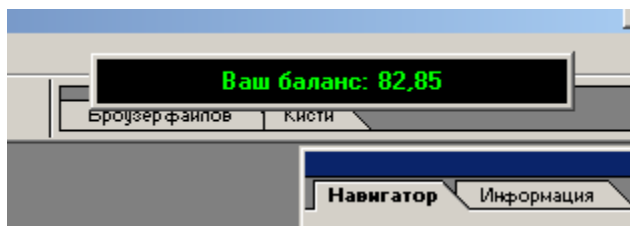
Для этого можно воспользоваться утилитой Resource Hacker™ или любым аналогичным редактором ресурсов: <http://www.google.com/search?q=exe+resource+editor>

Измените строковые ресурсы 1, 2, 3 (для старой версии утилиты 65257, 65258, 25259), соответственно http://provider/bgbilling/balance_sender, <http://provider/bgbilling/webexecuter>, mid на адреса вашего сервера биллинга и код модуля TrayInfo (вы можете его посмотреть в редакторе модулей и услуг).

Запускаем утилиту, жмём правой кнопкой мыши и выбираем **Параметры**:



Далее в PIN1 и PIN2 клиент вводит свой PIN1 и PIN2 TrayInfo, нажимает Ок и может смотреть баланс в висящем окошке.



Для того, чтобы клиент смог переходить на свою страницу статистики необходимо в настройках сервера (**Сервис=>Настройка=>Конфигурация**) добавить в конце значения переменной **web.auth.modes** точку с запятой и разрешение авторизоваться через модуль TrayInfo:

```
; < TrayInfo>:1
```

Строка примет, например, такой вид (код экземпляра модуля TrayInfo равен 10, помимо этого разрешена авторизация по договору):

```
web.auth.modes=0:1;10:1
```

Затем перезагрузите BGBillingServer. Теперь клик правой мышкой по окошку TrayInfo, затем выбор в меню **Статистика** и вы переходите на страницу статистики клиента.

Настройка утилиты TrayInfo3

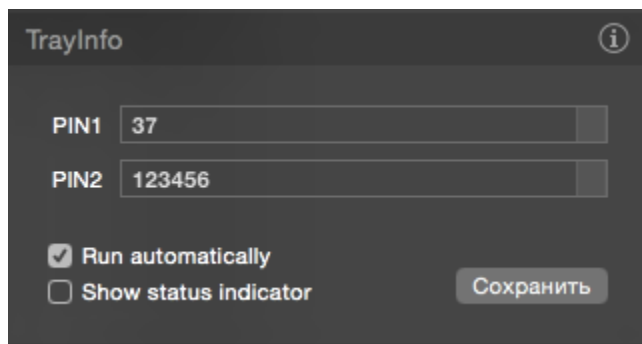
Утилита написана с использованием Qt, никаких ресурсов не содержит. Патчер для настройки утилиты находится внутри модуля в виде GUI-фрейма. Для его запуска необходимо запустить класс, например, в папке с корнем в клиенте или сервере:

```
java -cp ./*:./*lib/* ru.bitel.bgbilling.modules.trayinfo.common.BinaryPatch
```

Далее вводятся все нужные параметры. Дефолтные PIN1 и PIN2 можно не вводить (сделано на будущее для автоматической генерации настроек пользователя на лету).

Настройка утилиты TrayInfo для Mac Os

Для Mac Os можно запустить виджет для Notification Center, минимальная версия Mac Os 10.10(Yosemite)



TrayInfo ⓘ

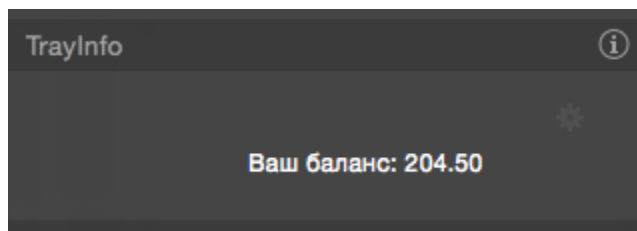
PIN1 37

PIN2 123456

Run automatically

Show status indicator

Сохранить



Перед распространением программы необходимо обратиться к нам, чтобы мы скомпилировали виджет с вашими данными(URL баланса, код модуля).

По желанию можно заменить изображение иконки, для этого надо пройти в Contents->Resources и заменить файл AppIcon.icns нужным, для конвертированной изображения можете воспользоваться утилитой Img2icns(MacOs)

Отображение в утилите произвольной информации вместо баланса

Имеется возможность генерировать любую информацию для отображения её в утилите. Возможность запроса баланса действует по умолчанию и оставлена для обратной совместимости. Для использования нового функционала необходимо генерировать нужную строку с помощью динамического кода.

```
#
# : ru.bitel.bgbilling.modules.trayinfo.server.bean.TrayInfoReplyBuilder
replybuilder=ru.bitel.bgbilling.trayinfo.SimpleReply
```

Пример класса идёт в комплекте.

Для отправки суммы:

```
#      summa ( ). .
#      ,      .
#      (1), .
use.summa.reply=0
```

Итак, если не настроить дополнительно, то будет генерироваться баланс без участия скрипта и отображаться в утилите. Можно использовать и скрипт и баланс, тогда старые утилиты будут отображать только баланс, а новые - сгенерированную строку.

Модуль Bill

Назначение модуля

Модуль предназначен для автоматизации выставления/обработки счетов и счетов-фактур, совмещённых с актами или УПД выполненных работ в среде биллинга, включает в себя Web-интерфейс пользователя для создания и просмотра счетов и просмотра счетов-фактур с актами.

Краткое описание алгоритма работы модуля

Модуль поддерживает два вида документов: **Счета** и **Счета-фактуры-Акт/УПД**. Счёт - уведомление клиенту о сумме, необходимой к оплате. Счёт-фактура-Акт/УПД - документ, определяющий размер оказанных услуг абоненту. Реально он представляет из себя одновременно счет-фактуру и акт выполненных работ или УПД. В стандартном шаблоне они разделены на разных листах: на первом листе отображается счет-фактура, а на втором - акт. А также есть шаблон УПД.

Оба вида документов выставляются для конкретного договора и делятся на **Типы**. Пример типов: Счёт телефония физ. лица, Счёт интернет юр. лица. Тип определяет набор **Позиций** для документа. Позиция - строка документа, состоящая из:

- **названия**, например "Абонплата за февраль";
- **суммы**, например 10.40;
- **количества**, например 10;
- **число знаков после запятой для количества**, например 2;
- **единицы измерения**, например Мб.

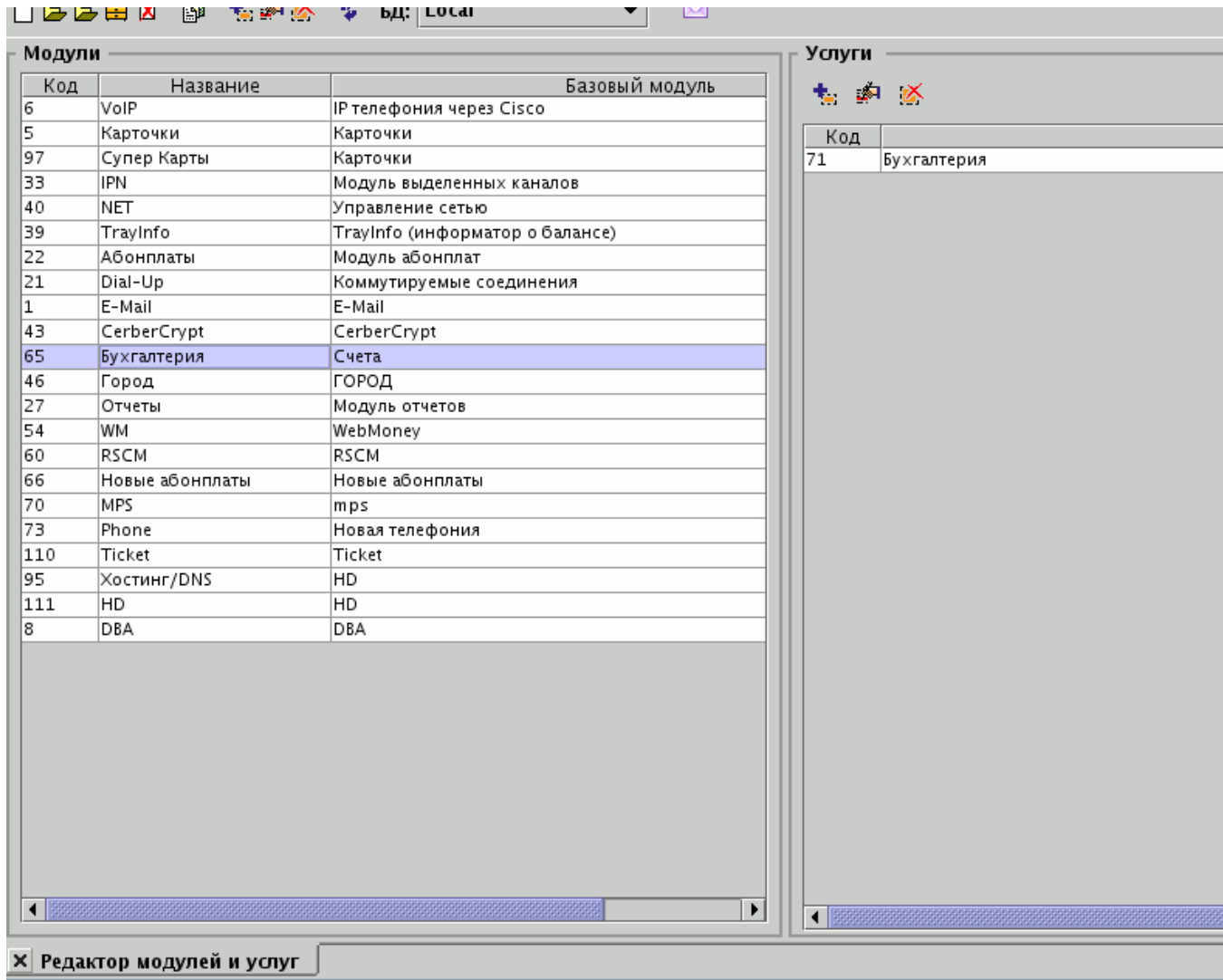
Каждый документ состоит из набора позиций и некоторых параметров для всего документа. Общие для обоих типов документов параметры:

- **Дата создания;**
- **Номер;**
- **Тип;**
- **Номерной пул;**
- **Сумма;**
- **Месяц**, за который выставлен документ.

Позиции могут быть гибко настроены, что позволяет выводить в счетах и фактурах различную информацию. Сформированный документ представляет из себя **XML** документ, который отображается в печатную форму посредством **XSLT**-шаблона. Данная прослойка позволяет, в общем случае, делать произвольное оформление счета. В виде квитанции физ. лицу, простого счета юр. лицу и т.п.

Настройка модуля Bill

Установите модуль на сервер с помощью утилиты **bg_installer**, обновите клиент. Добавление модуля в договор будет означать подключение функционала модуля к договору (возможность выставления счетов и счетов-фактур договору в АРМ администратора и пользователю самостоятельно через сайт).



The screenshot shows a software interface with two main panels: 'Модули' (Modules) and 'Услуги' (Services). The 'Модули' panel contains a table with columns 'Код', 'Название', and 'Базовый модуль'. The 'Услуги' panel contains a table with columns 'Код' and 'Название'.

Код	Название	Базовый модуль
6	VoIP	IP телефония через Cisco
5	Карточки	Карточки
97	Супер Карты	Карточки
33	IPN	Модуль выделенных каналов
40	NET	Управление сетью
39	TrafficInfo	TrafficInfo (информатор о балансе)
22	Абонплаты	Модуль абонплат
21	Dial-Up	Коммутируемые соединения
1	E-Mail	E-Mail
43	SerberCrypt	SerberCrypt
65	Бухгалтерия	Счета
46	Город	ГОРОД
27	Отчеты	Модуль отчетов
54	WM	WebMoney
60	RSCM	RSCM
66	Новые абонплаты	Новые абонплаты
70	MPS	mrs
73	Phone	Новая телефония
110	Ticket	Ticket
95	Хостинг/DNS	HD
111	HD	HD
8	DBA	DBA

Код	Название
71	Бухгалтерия

Х Редактор модулей и услуг

Перезапустите клиент, откройте в меню **Модули** созданный вами экземпляр модуля. Создайте в редакторе конфигурации новую конфигурацию и произведите настройку модуля. Значения параметров указаны после символа комментария (#).

Далее будет приведена конфигурация модуля, включающая в себя все имеющиеся параметры.

Конфигурация

```
#-----  
# Web-  
#-----  
#  
web.menuItem1=  
web.menuItem2=-  
# , .  
allowed.web.bill.types=36  
# , web ( 0)  
web.bill.generate.position.quantity=1  
# - - / web.  
# 1 - , 0 - .
```

```

invoice.generate.web.visible=0
#-----
#
#-----
#
#
bill.nds=18
#
#-----
# INLINE-
#-----
# Inline-
# : , , . Inline
#
# ->->
inline.params=1;43|name;dover;dolz;prikaz
inline.param.1.name= ..
inline.param.1.dover= 2
inline.param.1.prikaz= 32 12.33.06
inline.param.1.dolz= 2
inline.param.43.name= ..
inline.param.43.dover= 1
inline.param.43.prikaz= 33 12.33.06
inline.param.43.dolz= 1
#
#-----
#
#-----
#
# 0 1.
# 0 // .
# . / ,
#
#
# 1 // .
doc.num.mode=0
#
#-----
#
#-----
bill.pos.< >.title=< , >
bill.pos.< >.name=< >
bill.pos.< >.summ=< >
bill.pos.< >.quantity=< >
bill.pos.< >.qtynums=< >
bill.pos.< >.unit=< >
bill.pos.< >.unitCode=< >
#
# , 0 -
bill.pos.< >.insum=1
#
# 0
bill.pos.< >.awlz=1
#
#-----
#
#-----
bill.pos.<id>.title=< >
bill.pos.<id>.name=< >
bill.pos.<id>.extractor=<>
bill.pos.<id>.unit=< >
#
#-----
#
#-----
#
# xsl-
#
preempt.bill.reestr.over.xsl=preempt_bill_reestr.xsl
#
# xsl-
#
preempt.invoice.reestr.over.xsl=preempt_invoice_reestr.xsl
#
#
preempt.bill.reestr.to.server.path=/home/kostya/preempt.csv
#

```

```

preempt.invoice.reestr.to.server.path=/home/kostya/preemptInvoice.csv
#
#-----
# e-mail
#-----
# e-mail ,
mail.contract.param=20
# e-mail
mail.send.sub.as.parent=1
# ${fileName} , mail.filename
mail.subject= BGBilling ${fileName}
# ( mail.subject)
mail.text=
# , PLAIN, HTML,
# mail.text.type=html
# .
#
# ${type} - ,
# ${N} - ,
# ${contract.title}${contract.id} - id , , mail.send.sub.as.parent=1
# ${contract.title.sub}${contract.id.sub} - id
# ${contract.comment.sub} ${contract.comment} - , mail.send.sub.as.parent=1
# ${param_N} - , N , , mail.send.sub.as.parent=1
# ${sub.param_N} - , N
# ${formatN} - ,
mail.filename=${type}_${N}_${contract.title}(${contract.id})_for_${param_4}_document
#
#-----
#
#-----
# checkbox " "
bill.generate.new.checked=1
#
# checkbox " " -
invoice.generate.new.checked=1
#
# : 0 - ; 1 - ; 2 - ( )
pay=0
#
# XML-
add.contract=1
#
# , " , .
# package.param.id=< >
#
# , (//)
# address.sort.param.id=< >
#
# / . .
name.sort.param.id=<id . >,<id .>
#
# csv . " ,"
csv.separator=,
#
#use.master.connection=1

#
#
#closed.date.disabled.ActionBillUpdatePositions=1
#closed.date.disabled.ActionInvoiceUpdatePositions=1
#
#closed.date.disabled.ActionDeleteBill=1
#closed.date.disabled.ActionDeleteInvoice=1
#
#closed.date.disabled.ActionGenerateBills=1
#closed.date.disabled.ActionGenerateInvoices=1
# +
#closed.date.disabled.ActionGetTemplateBillList=1
# xml
#closed.date.disabled.ActionBillUpdateXml=1
# ,

```

```
#closed.date.disabled.BillService.billDocumentCreate=1  
#  
#closed.date.disabled.BillService.billDocumentDelete=1
```

Шаблоны оформления печатных форм описаны на языке [XSLT](#) и генерируют на выходе [FO](#)-документы, основываясь на [XML](#)-дереве с данными, предоставляемым модуле. Более подробно о настройке шаблонов описано [далее](#).

Настройка позиций

Позиция - это составная часть документа, обладающая суммой, количеством и единицей измерения, которую биллинг вычисляет для конкретного договора и месяца и подставляет в документ. Позиции счетов и счет-фактур задаются в **конфигурации модуля**, каждая позиция идентифицируется числовым кодом, коды позиций счетов и счет-фактур могут пересекаться.

Позиции счетов/счетов-фактур задаются аналогично, с той лишь разницей, что префикс **bill.pos** заменяется на **invoice.pos**. Разберём более подробно различные параметры позиции.

В **name** может быть указана произвольная строка с подстановками **\$month**, **\$nextmonth**, **\$prevmonth**, **\$prev2month**, **\$prev3month**. Подстановка означает строковое название месяца за который выставляется счёт, следующего месяца, предыдущего месяца, и ещё двух предыдущих месяцев. Также возможно вывести значение месяца в произвольном формате, вместо **\$month** (**\$nextmonth** и т.п.) указать в виде **{\$month,date,формат}**, например **{\$month,date,MMMM yyyy г.}** - результатом будет апрель 2009 г.

В **summ** могут быть указаны следующие макросы, связанные знаками "+" и "-". Вместо **<month>** могут быть подставлены значения **\$month**, **\$nextmonth**, **\$prevmonth**, **\$prev2month**, **\$prev3month**, означающие в данном контексте месяц, за который производится выборка.

- **DEBT**(**<month>**) - долг на конец месяца, равен минус исходящему остатку;
- **IN_REST**(**<month>**) - входящий остаток на месяц;
- **SERVICE_ACCOUNT**(**<month>**, **<коды услуг через запятую>**) - наработка по определённым услугам;
- **FULL_ACCOUNT**(**<month>**) - суммарная наработка по всем услугам;
- **CHARGE**(**<month>**, **<коды расходов через запятую, не обязательный параметр>**) - сумма расходов за месяц;
- **PAYMENT**(**<month>**, **<коды типов платежей, не обязательный параметр>**) - сумма платежей за месяц;
- **NPAY_MIN_ACCOUNT**(**<код экземпляра модуля абонплат>**,**<month>**,**<код услуги модуля абонплат>**) - для "доводящей абонплаты" сумма, до которой она доводится;
- **BILL_BILL_SUM**(**<код экземпляра модуля бухгалтерия>**,**<month>**,**<коды типов документов>**) - сумма выставленных счетов за определённый месяц для данного договора, перечень кодов типов документов через запятую - необязательный параметр;
- **BILL_INVOICE_SUM**(**<код экземпляра модуля бухгалтерия>**,**<month>**,**<коды типов документов>**) - сумма выставленных счетов-фактур за определённый месяц для данного договора, перечень кодов типов документов через запятую - необязательный параметр;
- **CONST**(**<число>**) - числовая константа.

В **quantity** могут быть указаны следующие макросы, связанные знаками "+" и "-". Вместо **<month>** могут быть подставлены значения **\$month**, **\$nextmonth**, **\$prevmonth**, **\$prev2month**, **\$prev3month**, означающие в данном контексте месяц, за который производится выборка. Вместо **<mid>** подставляется код экземпляра соответствующего модуля. Вместо **<sids>** подставляются коды услуг через запятую, это уточняющий и не обязательный параметр, если он не указан, будут взяты все услуги.

- **KERNEL_SERVICE_COUNT**(**<month>**, **<коды услуг через запятую>**) - количество разрешенных услуг, определённых перечнем, заведённых в договоре на начало указанного месяца;
- **NPAY_SERVICE_COUNT**(**<mid>**, **<month>**, **<sids>**) - количество услуг абонплаты, определённых перечнем, заведённых в договоре на начало указанного месяца, учитывая параметр кол-во в свойствах услуги абонплат;
- **NPAY_SERVICE_COUNT_MONTH**(**<mid>**, **<month>**, **<sids>**) - аналогично предыдущему, но считается количество услуг абонплат не на первое число месяца, а всего попавших в данный месяц отрезков данных услуг;
- **DIALUP_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг DialUP модуля, делитель определяет число, на которое будет разделен объем услуги (байты для трафика, секунды для времени); Например, делитель 1048576 даст объем в МБ для услуг типа трафик;
- **DIALUP_MAX_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг DialUP модуля типа "максимальный трафик", делитель определяет число, на которое будет разделен объем услуги (байты для трафика). Например, делитель 1048576 даст объем в МБ для услуг типа трафик;
- **DIALUP_LOGIN_COUNT**(**<mid>**, **<month>**) - количество DialUP логинов на начало месяца;
- **VOICEIP_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг VoiceIP модуля в секундах округлённого времени, делитель определяет число, на которое будет разделен объем услуги. Например, делитель 60 даст объем в минутах;
- **VOICEIP_LOGIN_COUNT**(**<mid>**, **<month>**) - количество VoiceIP-логинов на начало месяца;
- **IPN_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг IPN модуля в байтах, делитель определяет число, на которое будет разделен объем услуги. Например, делитель 1048576 даст объем в МБ;
- **IPN_MAX_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг IPN модуля типа "максимальный трафик" в байтах, делитель определяет число, на которое будет разделен объем услуги. Например, делитель 1048576 даст объем в МБ;
- **PHONE_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг Phone модуля в секундах округлённого времени, делитель определяет число, на которое будет разделен объем услуги. Например, делитель 60 даст объем в минутах;
- **PHONE_ZERO_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - то же, что предыдущее, но считаются только сессии с нулевой стоимостью;
- **PHONE_NOZERO_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - то же, но считаются только сессии с ненулевой стоимостью;
- **PHONE_NOZERO_SERVICE_COUNT**(**<mid>**, **<month>**, **<sids>**) - считается кол-во сессий с ненулевой стоимостью;
- **PHONE_SERVICE_COUNT**(**<mid>**, **<month>**, **<sids>**) - общее количество сессий;
- **PHONE_POINT_COUNT**(**<mid>**, **<month>**) - количество Phone-поинтов на начало месяца;
- **RSCM_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг RSCM-модуля в единицах, делитель определяет число, на которое будет разделен объем услуги;
- **INET_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг Inet-модуля, делитель определяет число, на которое будет разделен объем услуги (байты для трафика, секунды для времени). Например, делитель 1048576 даст объем в МБ для услуг типа трафик;
- **INET_SERV_COUNT**(**<mid>**, **<month>**) - количество Inet-сервисов на начало месяца.
- **VOICE_SERVICE_AMOUNT**(**<mid>**, **<month>**, **<делитель>**, **<sids>**) - объем услуг Voice модуля в секундах округлённого времени, делитель определяет число, на которое будет разделен объем услуги. Например, делитель 60 даст объем в минутах;

- **VOICE_ZERO_SERVICE_AMOUNT**(<mid>, <month>, <делитель>, <sids>) - то же, что предыдущее, но считаются только сессии с нулевой стоимостью;
- **VOICE_NOZERO_SERVICE_AMOUNT**(<mid>, <month>, <делитель>, <sids>) - то же, но считаются только сессии с ненулевой стоимостью;
- **VOICE_NOZERO_SERVICE_COUNT**(<mid>, <month>, <sids>) - считается кол-во сессий с ненулевой стоимостью для модуля Voice;
- **VOICE_SERVICE_COUNT**(<mid>, <month>, <sids>) - общее количество сессий Voice;
- **VOICE_ACCOUNT_COUNT**(<mid>, <month>) - количество Voice-аккаунтов на начало месяца;

Если **quantity** не указано, то позиция принимается за единицу.

Начиная с версии 4.6, добавлен необязательный параметр позиции - число знаков после запятой для количества (**qtnums**). По умолчанию он принимается за 0 (т.е. количество округляется соответствующей логике предыдущих версий).

В **unit** указывается просто строка вида "Мб", "Кб", "мин.". Если параметр не указан, то единицы принимаются за "шт.". Параметр **unitCode** определяет цифровой код единицы измерения. Если параметр не указан, то подставляется код 796, который, согласно Общероссийскому классификатору единиц измерений, соответствует единице измерения "шт."

Если параметр **insum** не указан, позиция включается в сумму документа, для исключения необходимо установить его в 0. Данный режим может быть полезен для позиций, несущих вспомогательную информацию.

Если параметр **awlz** указан, позиция включается в сумму документа, даже если сумма по позиции 0. Данный режим может быть полезен в случаях, когда нужно показать отсутствие расходов по позициям.

Рассмотрим несколько примеров построения позиций.

Пример 1

Клиенту необходимо выставить суммарный долг за услуги интернета.

```
bill.pos.1.title=
bill.pos.1.name=      $month
bill.pos.1.summ=DEBT($month)
```

Пример 2

А корпоративным клиентам необходимо выставить отдельными строчками абонплату, наработку по интернет и телефонии. За каждый месяц необходимо выставить счета с наработкой за данный месяц и предоплатой абонплаты за следующий. Предположим, что у нас в системе установлен модуль абонплат, IPN и VoiceIP. Предположим также, что существуют следующие коды услуг:

- Абонплата - 1;
- Трафик -2;
- Телефония - 3.

Так будет выглядеть настройка позиций счетов и счетов-фактур:

```
bill.pos.1.title=
bill.pos.1.name=      $month
bill.pos.1.summ=SERVICE_ACCOUNT($month, 2, 3 )
bill.pos.2.title=
bill.pos.2.name=      $nextmonth
bill.pos.2.summ=SERVICE_ACCOUNT($nextmonth,1 )
#
invoice.pos.1.title=
invoice.pos.1.name=    $month
invoice.pos.1.summ=SERVICE_ACCOUNT($month, 2, 3)
invoice.pos.2.title=
invoice.pos.2.name=    $month
invoice.pos.2.summ=SERVICE_ACCOUNT($month, 1)
```

Возможно суммирование и вычитание макросов суммы и количества. Например, следующим образом можно добавить в наработку определённые виды расходов:

```
invoice.pos.3.summ=SERVICE_AMOUNT($month,101,102)+CHARGE($month,14,8,1,54,32)
```

Экстракторы

Зачастую в системе присутствуют большое количество однотипных списаний и абонентских плат. В этом случае конфигурирование под каждую отдельную позицию достаточно обременительно и могут использоваться экстракторы. Позиция с экстрактором заводится в [конфигурации модуля](#).

- Параметры **title** и **unit** имеют значение, аналогичные обычным позициям. В параметр **extractor** может быть подставлен один из перечисленных ниже макросов. Вместо `<month>` могут быть подставлены значения **\$month**, **\$nextmonth**, **\$prevmonth**, **\$prev2month**, **\$prev3month**, означающие в данном контексте месяц, за который производится выборка. Вместо `<mid>` подставляется код экземпляра соответствующего модуля.
- **KERNEL_CHARGES**(`<month>`, `<коды типов расх>`) - расходы договора в каком-то месяце. Тип расхода становится названием позиции, количество расходов данного типа - количеством. При не указании кодов типов расходов выбираются все типы расходов;
- **KERNEL_CHARGES_EXCEPT**(`<month>`, `<коды типов расходов, которые исключаются>`) - аналогичен предыдущему, но указываются коды типов расходов, которые не выбираются;
- **NPAY_SERVICES**(`<mid>`, `<month>`, `<коды услуг>`) - начисления абонплат в каком-то месяце. Название услуги абонплаты становится названием позиции, количество абонплат данного типа, установленных в договоре на данный месяц - количеством. При не указании кодов услуг выбираются все начисленные на договор абонплаты;
- **NPAY_SERVICES_EXCEPT**(`<mid>`, `<month>`, `<коды услуг, которые исключаются>`) - аналогичен предыдущему, но указываются коды услуг, которые не выбираются;
- **NPAY_SERVICES_DETAIL**(`<mid>`, `<month>`, `<коды услуг>`) - детализация начислений абонплат в каком-то месяце. Аналогичен **NPAY_SERVICES**, но данные берутся из таблицы `npay_detail_<mid>_уууumm`, то есть данные будут разнесены по кодам сущностей других модулей(например поинты модуля телефонии);
- **RSCM_SERVICES**(`<mid>`, `<month>`, `<коды услуг>`) - начисления по услугам RSCM в каком-то месяце. Название услуги RSCM становится названием позиции, количество услуги - количеством оказанной услуги . Единица измерения услуги - единица измерения счета При не указании кодов услуг выбираются все услуги RSCM оказанные для данного договора;
- **RSCM_SERVICES_EXCEPT**(`<mid>`, `<month>`, `<коды услуг, которые исключаются>`) - аналогичен предыдущему, но указываются коды услуг, которые не выбираются;
- **EXT**-экстракторы (**KERNEL_CHARGES_EXT**, **KERNEL_CHARGES_EXT_EXCEPT**, **NPAY_SERVICES_EXT**, **NPAY_SERVICES_EXT_EXCEPT** и т.д.) - расширенные (EXTended) экстракторы, которые в отдельную позицию выделяют каждый расход (для ядра) и каждую услугу (для модулей), даже если они имеют одинаковый тип (код). Аргументы для экстрактора те же, что и для аналогичных нерасширенных.

Позиция экстрактора "распадается" в момент генерации документа на множество позиций в зависимости от реального количества начислений на договоре.

В **unit** указывается единица измерения - простая строка.

Получение всех расходов за месяц:

```
bill.pos.3.title=  
bill.pos.3.name=  
bill.pos.3.extractor=KERNEL_CHARGES($month)
```

NPay экстракторы

Отдельно стоит отметить, что экстракторы модуля NPay поддерживают следующие макросы в имени позиции(`name`):

- **\$title** - название услуги
- **\$period** - период услуги
- **\$month**, **\$nextmonth**, **{\$month,date,MMMM yyyy г.}** итд - стандартные макросы даты счета

И только расширенные(EXT) еще следующие:

- **\$comment** - комментарий услуги
- **\$oid.name** - название объекта, если таковой имеется у услуги

Если в конфигурации отсутствует параметр `bill.pos.<№ позиции>.name`, то для обычных экстракторов будет использовано название услуги, а для расширенных название услуги и комментарий.

Детализация по тарифу

С версии 4.5 возможна детализация по тарифу, т.е. получение и вывод информации, например, по количеству и наработке по предоплаченному трафику и превышению трафика.

Для этого сначала в тарифицируемом модуле (dialup или ipn) необходимо указать коды начислений:

```
#
tariff_detail.cost_type.1=
# , range_take_all=1 ,
# , ,
# -
tariff_detail.cost_type.1.range_take_all=1
# , range_take_all
# (.. )
tariff_detail.cost_type.1.range_take_all.sid=
tariff_detail.cost_type.2=
```

Услуга: Dial-Up(входящий)

- 60000 байт за месяц безусловно
 - 0.0 за 1 MB
 - Включенный трафик
 - Зона: one
 - Канал 256
- 0 байт за месяц безусловно
 - Зона: two
 - 1.0 за 1 MB (Превышение трафика)
 - Канал 128

В приведённом выше примере два кода начисления, Включённый трафик (60000 байт) и Превышение трафика. Т.к. опция range_take_all=1 включена, то кол-во по Включённому трафику будет всегда 60000 байт, даже если договор наработал меньше.

Для детализации максимальных трафиков необходимо для кода начисления указать calc_type=2:

```
#
tariff_detail.cost_type.1= ( )
tariff_detail.cost_type.1.calc_type=2
```

Для вычисления детализации по тарифу необходимо запускать переобсчет за весь месяц в тарифицируемом модуле. При этом можно переобсчитывать как группу договоров, так и отдельный договор. Иначе в выставленных после этого счетах/счетах-фактурах вы можете получить неполные данные.

Рекомендуемая процедура следующая: по окончании рассчитываемого месяца (в начальных числах следующего) запускаете ручную переобсчет, а после него формируете счета, счета-фактуры. Для модуля IPN существует еще одно [решение](#).

Теперь для поля summ в конфигурации модуля бухгалтерии доступен следующий параметр:

- **TARIFF_DETAIL_COST**(<mid>, <month>, коды начисления) - сумма начисления в детализации по тарифу для модуля.

а для поля quantity:

- **DIALUP_TARIFF_DETAIL_AMOUNT**(<mid>, <month>, <делитель>, коды начисления) - объем начисления в детализации по тарифу для модуля dialup;
- **IPN_TARIFF_DETAIL_AMOUNT**(<mid>, <month>, <делитель>, коды начисления) - объем начисления в детализации по тарифу для модуля ipn.

Номерной пул

Номерные пулы - это сущности, в рамках которых выдаются номера очередному созданному документу. Номерной пул может быть один общий для нескольких документов. Обязательными и неудаляемыми являются номерные пулы **По умолчанию (счета)** и **По умолчанию (счета-фактуры)**.

Для каждого документа запоминается его номерной пул, номер порядковый абсолютный, номер порядковый в пределах года и номер в пределах месяца. Отображаемый номер документа формируется из данных номеров, его вид определяется шаблоном.

Код	Название	Шаблон
1	По умолчанию (счета)	123123
2	По умолчанию (счета-фактуры)	12312321321
3	dfasdfsdfasdf	

Для настройки номерных пулов необходимо указать шаблон, по которому будет формироваться номер документа. Номер документа может состоять из произвольных символов и следующих макроподстановок:

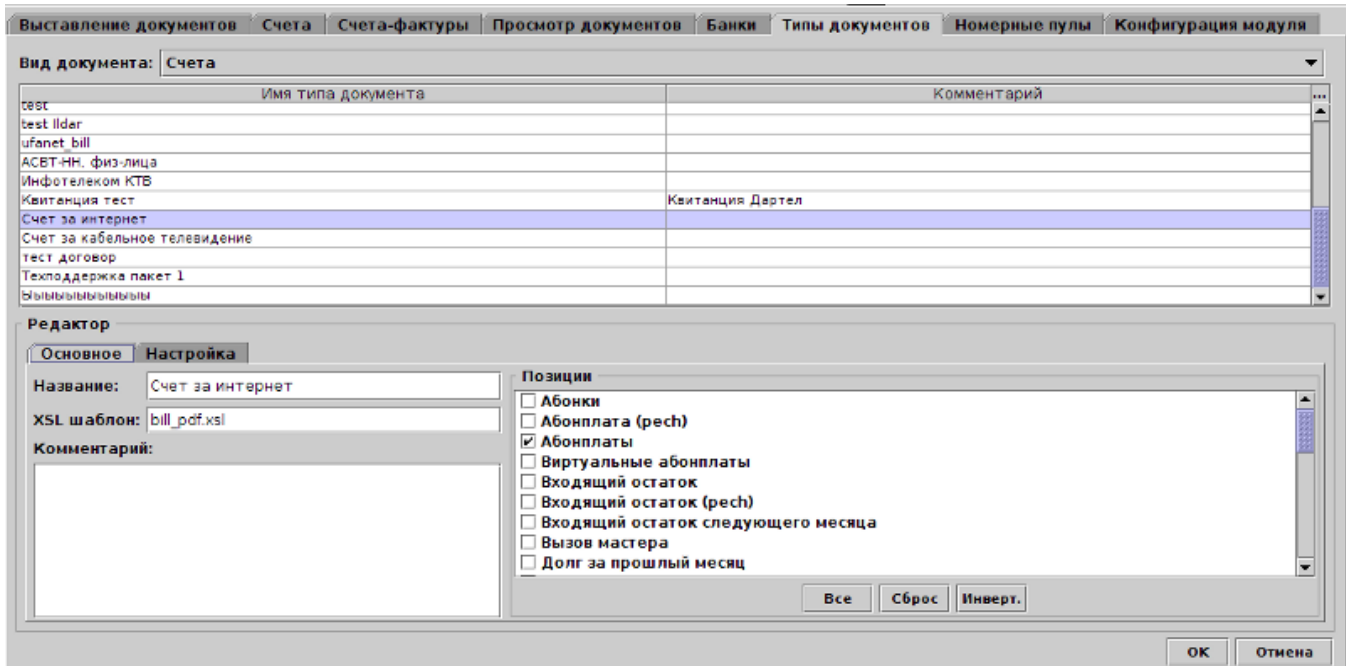
- **\$number{X}** - абсолютный порядковый номер;
- **\$number{START,X}** - абсолютный порядковый номер, начиная со START;
- **\$number_in_year{X}** - номер в пределах года;
- **\$number_in_month{X}** - номер в пределах месяца;
- **\$month** - номер месяца (двухзначный - от 01 до 12);
- **\$year** - номер года (четырёхзначный);
- **\$contract_title** - номер договора

Здесь X - это количество цифр, выделяемых под соответствующий номер, т.е. подставляемые номера дополняются нулями слева до длины, определенной этим числом.

Редактор	
Имя пула: <input type="text" value="Новый номерной пул"/>	Шаблон номера: <input style="border: 1px solid gray;" type="text" value="АБВ-\$number{4}/\$number_in_month{2}"/>
<input type="button" value="OK"/> <input type="button" value="Отмена"/>	

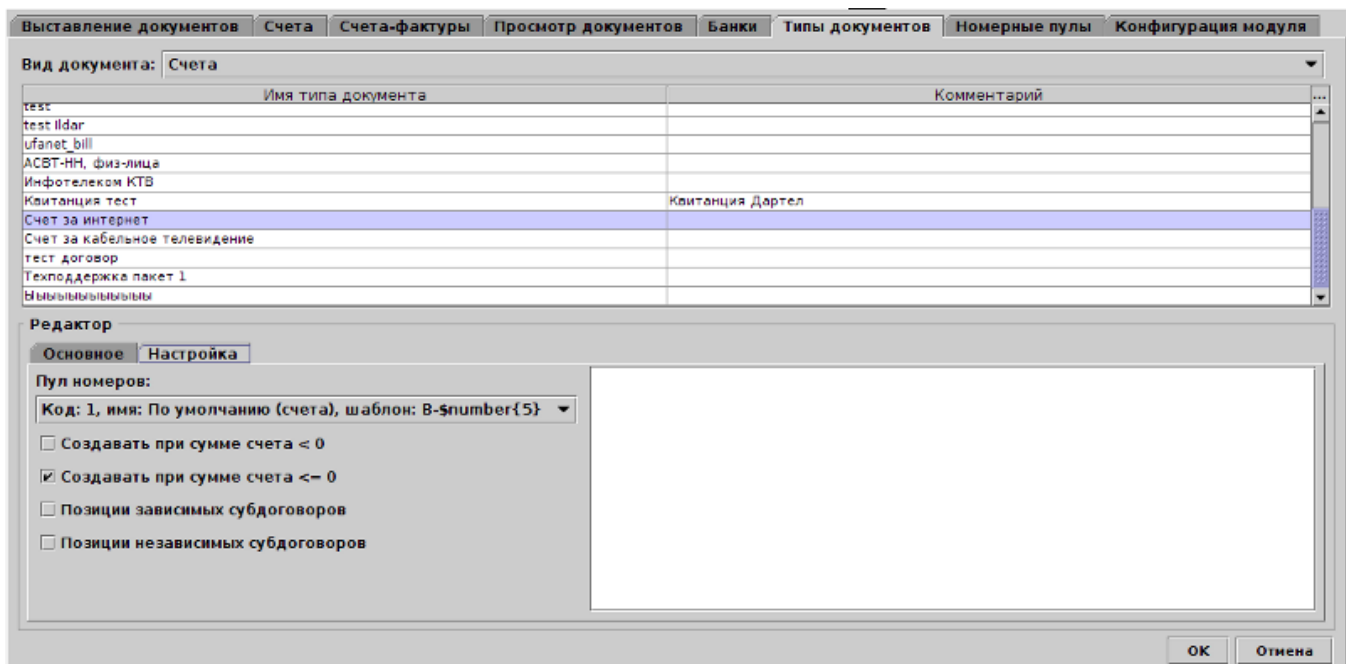
Типы документов

Тип документа определяет, какие позиции включает счет или счет-фактура. Типы определяются отдельно для счетов и счетов-фактур на вкладке **Типы документов** конфигурации модуля.



Для каждого типа документа указываются активные позиции, название. Поле **XSLT-шаблон** задаёт имя шаблона, используемого для создания печатной формы. Шаблон располагается в каталоге `BGBillingServer/webroot/xsl`. Если шаблон не указан, то для счетов используется `bill_pdf.xml`, а для счет-фактур `invoice_pdf.xml`. О генерации печатных форм будет подробно описано [далее](#).

Далее переходим во вкладку **Настройка**. Здесь можно указать номерной пул, по которому будет формироваться очередной номер для документа. Опция **Создавать при сумме счета ≤ 0** означает требование к созданию документов данного типа даже с неположительной суммой. Опция **Создавать при сумме счета < 0** означает требование к созданию документов данного типа во всех случаях, кроме случая равенства суммы нулю. Опция **Позиции субдоговоров** вызывает генерацию в исходное XML-дерево документа вычисленных значений позиций с разбивкой по субдоговорам. В шаблоне счета и счет-фактуры по умолчанию эти данные не используются, использование опции возможно только с модификацией шаблонов документов.



Для более тонкой настройки типа документа возможно указывать ключи в текстовом поле конфигурации. Возможные ключи для типов документов:

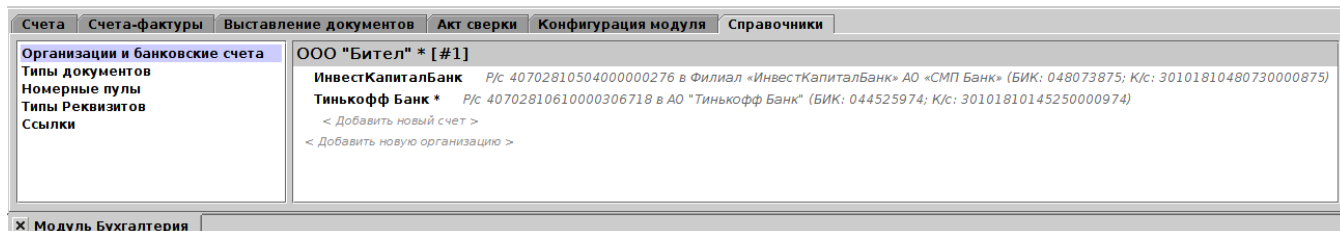
- **create.only.when.pos.positive=x,y,...,z** - создание счета\счета-фактуры только в том случае, если указанные позиции с кодами **x,y,...,z** положительны.

Т.к. файлы **bill_pdf.xml** и **invoice_pdf.xml** перетираются при каждой установке обновления модуля, мы рекомендуем вам скопировать их в файлы с иными именами (**bill_pdf_my.xml**, **invoice_pdf_my.xml**) и ссылать ваши типы документов уже на ваши файлы.

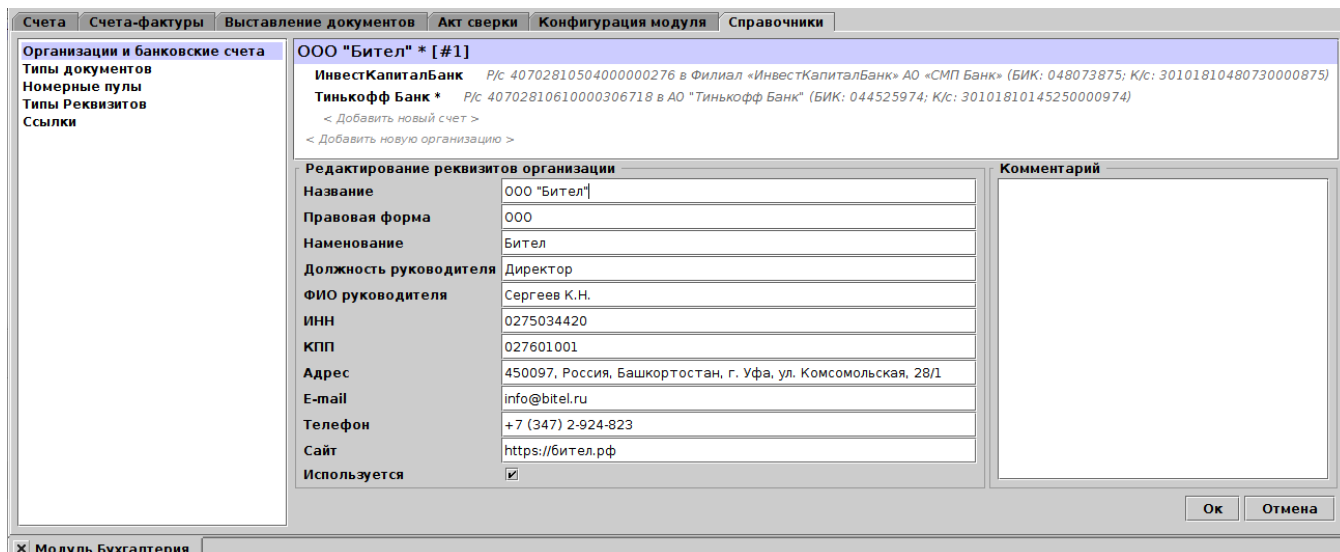
Настройка реквизитов организации(ий) и банковских счетов

для нормальной работы модуля в справочнике должны быть заведены как минимум одна организация и у нее один счет и они выбраны как **организация по умолчанию** и **счет по умолчанию**

При формировании документов (счета, акты, счет-фактуры, УПД и т.д.) требуется указание реквизитов организации от имени которой эти документы созданы. Данные реквизиты настраиваются в справочнике **Организации и банковские счета**

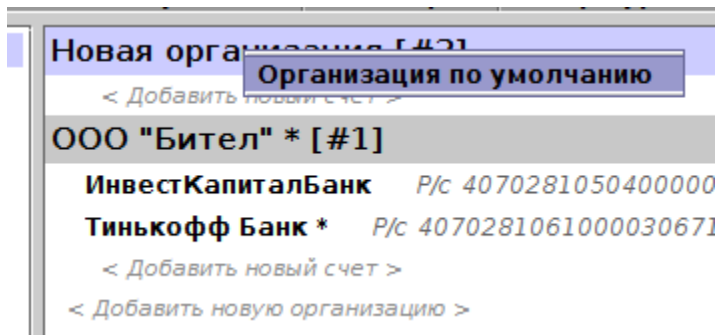


Для того, что бы добавить **Новую организацию** кликните двойным кликом по пункту **< Добавить новую организацию >** и заполните поля в открывшемся редакторе.



Сохраните реквизиты кликнув по клавише **Ок**

Выделите организацию в списке левой клавишей, затем правой клавишей мыши вызовите контекстное меню и выберите из него пункт **"Организация по умолчанию"**. Справа от названия организации должна появиться звездочка **"*"**



Для добавления банковского счета организации двойным кликом кликните по пункту **< Добавить новый счет >** после той организации счет которой вы хотите добавить. в Открывшемся редакторе заполните реквизиты счета.

ООО "Бител" * [#1]

ИнвестКапиталБанк Р/с 40702810504000000276 в Филиал «ИнвестКапиталБанк» АО «СМП Банк» (БИК: 048073875; К/с: 30101810480730000875)

Тинькофф Банк * Р/с 40702810610000306718 в АО "Тинькофф Банк" (БИК: 044525974; К/с: 30101810145250000974)

< Добавить новый счет >

< Добавить новую организацию >

Редактирование банк. реквизитов		Комментарий
Название	Тинькофф Банк	
Название Банка	АО "Тинькофф Банк"	
Счет	40702810610000306718	
БИК	044525974	
Кор. счет	30101810145250000974	
Тип платежа	Банк ("Тинькофф Банк")	
Используется	<input checked="" type="checkbox"/>	

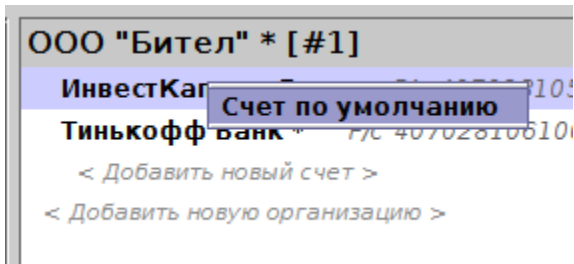
Ок Отмена

Параметр **Название** задаёт обозначение счета для идентификации его внутри модуля. **Название банка** - точное название банка, используемое в образце платёжного поручения.

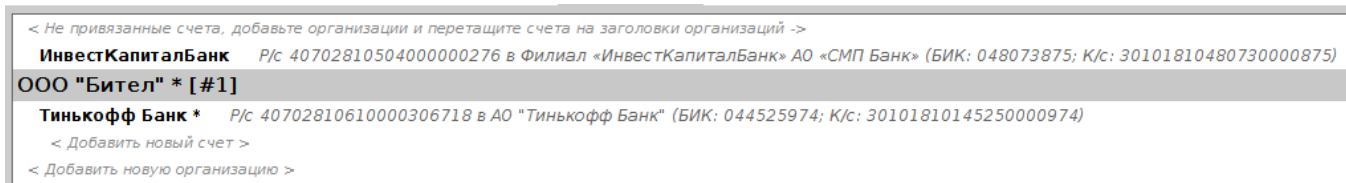
Тип платежа - не редактируемый тип платежа, который будет заносится в договор при пометке счета оплаченным и установленной опции **pay=1** в конфигурации модуля.

Сохраните счет кликнув по кнопке **Ок**.

Выделите счет в списке левой клавишей, затем правой клавишей мыши вызовите контекстное меню и выберите из него пункт **"Счет по умолчанию"**. Справа от названия счета должна появиться звездочка "*"



Если вы обновляетесь с более старых версий у вас может быть в справочнике быть указан счет без привязки к организации см. скриншот ниже



в такой ситуации вы можете перетащить счет мышкой на заголовок той организации к которой этот счет относится.

Настройка параметров договора

Реквизиты позволяют указывать в свойствах договора специфичные для него параметры. В отличие от параметров договора атрибуты поименованы (поле название в справочнике). Следует учитывать, что в шаблоне генерации печатной формы можно использовать как атрибуты, так и параметры договора.

Выбор (между параметрами и реквизитами) места хранения данных по ИНН, полному названию организации производится администратором биллинга. Возможна и комбинация этих способов. По-умолчанию шаблоны счета и фактуры ссылаются на реквизиты, указываемые в конфигурации. Реквизитов может быть сколько угодно много, добавляются в справочнике модуля бухгалтерии.

Для выставления документов необходимо добавить модуль в договор. Далее необходимо добавить реквизиты и выбрать какие документы будут выставляться договору. Также вместо реквизитов при генерации документов могут использоваться параметры договора, в таком случае реквизиты добавлять не нужно.

Для добавления реквизитов выберите в дереве модуль бухгалтерии и перейдите на вкладку **Реквизиты**.

Код	Реквизит	Значение	Период
46	инн	23	... - ...
47	ФИО	Параметр договора: Текст. поле: Ф.И.О...	... - ...

Редактирование реквизита

Название:

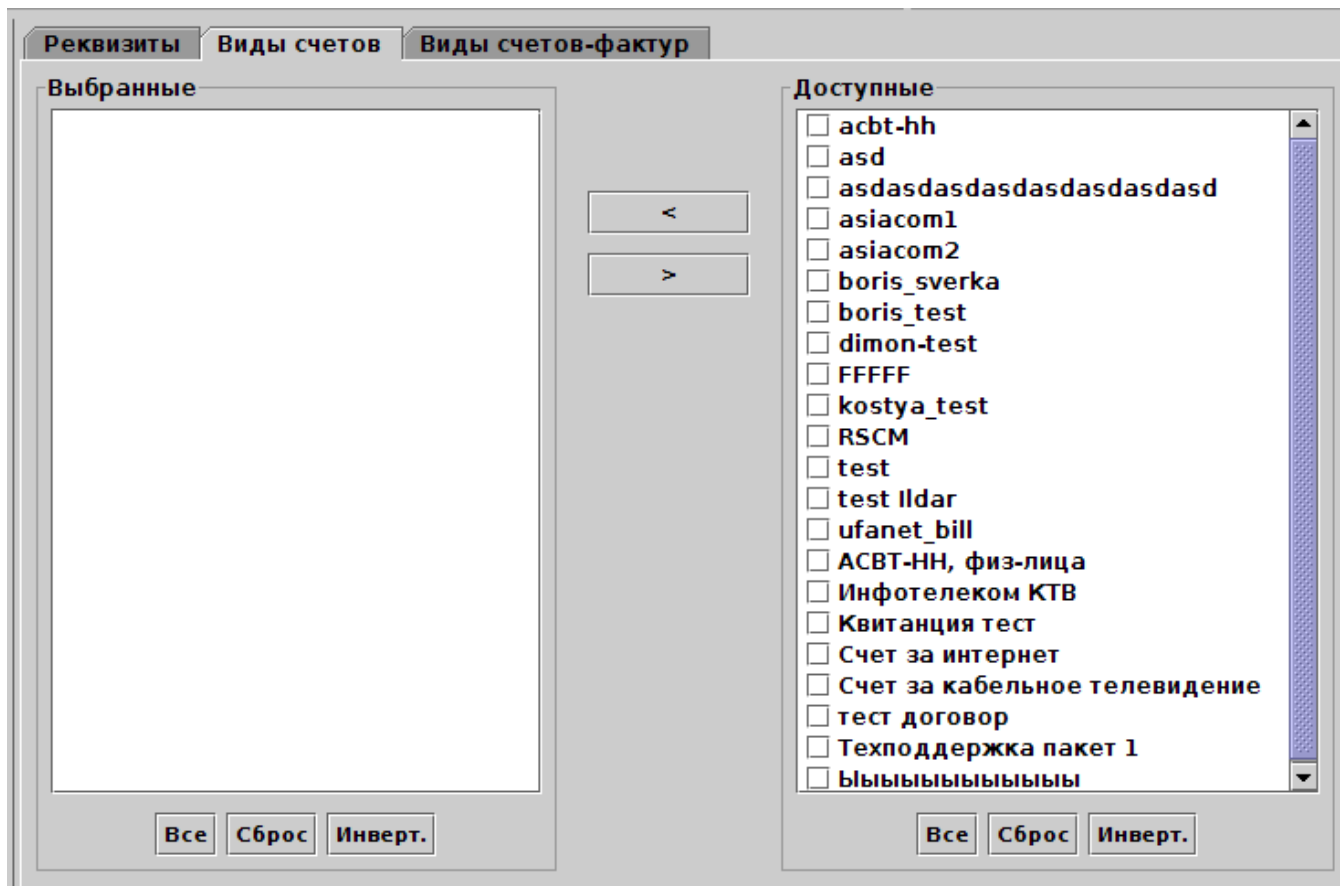
Тип значения:

Значение:

Период: с по

Ок Отмена

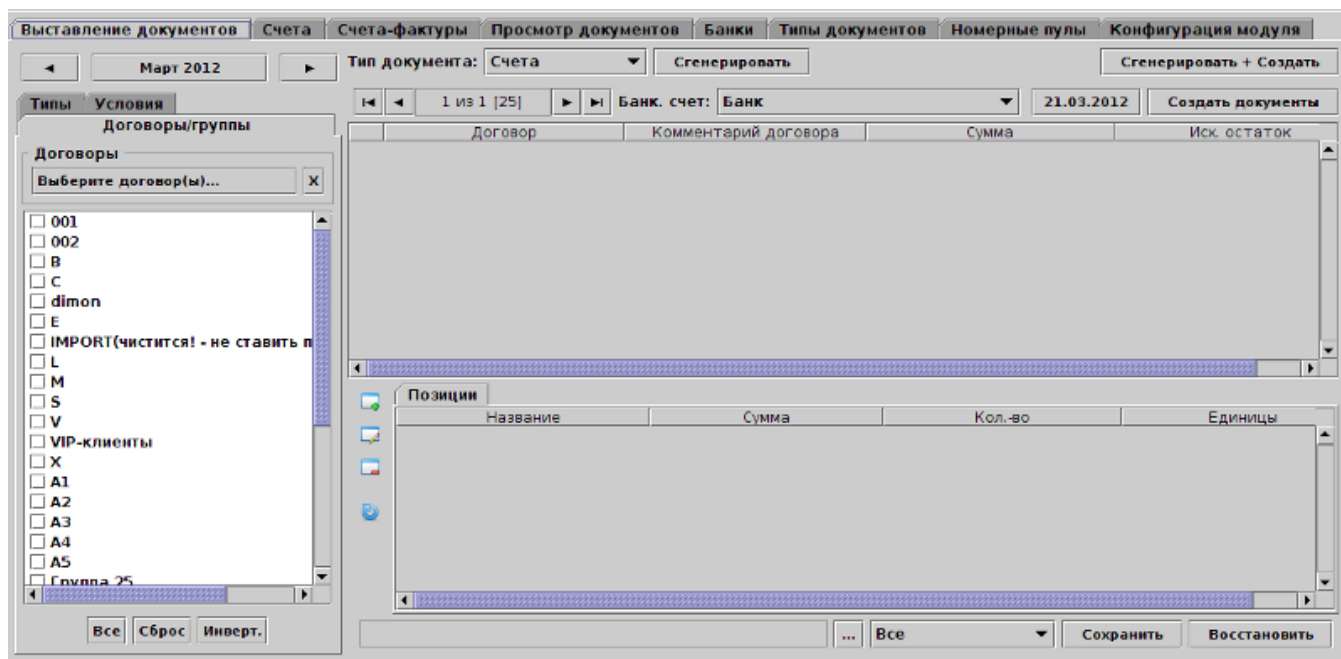
Для добавления документов договору перейдите на вкладки **Виды счетов** и **Виды счетов-фактур**. Затем выделите и переместите в левую часть необходимые документы.



Для массовой установки типов документов и услуги модуля договорам можно использовать [Групповые операции](#). С 4.4 версии требуемые типы документов можно указывать в [шаблонах договоров](#).

Выставление счетов и счетов-фактур администратором

Для выставления счетов и счет-фактур используется вкладка **Выставление документов**



В левом верхнем углу выбирается месяц, за который выставляются документы. В выпадающем списке **Тип документа** выбирается вид документа для генерации: либо **Счета**, либо **Счета-фактуры**. На вкладке **Договоры/группы** возможно задание фильтров для генерации документов конкретным договорам и группам договоров. На вкладке **Типы** устанавливаются типы документов для генерации. На вкладке **Условия** устанавливаются дополнительные условия фильтрации договоров. На данный момент существует возможность выставления счетов только тем договорам, у которых исходящий остаток на месяц выставления документа находится в заданных пределах. Крайние точки указанного предела включаются в поиск. Если же одно из двух значений не указано, то это означает "до бесконечности".



На скриншоте показан пример фильтра, по которому будут сгенерированы документы для договоров с исходящим остатком менее, либо равным минус одной копейке.

Все вышеперечисленные фильтры можно использовать одновременно. Если фильтры не установлены, генерируются все возможные типы документов для всех договоров в базе, к которым привязан модуль. Для договора генерируются только те типы документа, которые прописаны в свойствах модуля договора на вкладках **Виды счетов** и **Виды счетов-фактур**. Период договора должен пересекаться с месяцем, за который генерируются документы. После установки фильтров нажмите кнопку **Сгенерировать**.

В таблице в постраничном виде появятся подготовленные для создания документы. Каждый документ можно отредактировать в поле редактирования позиций, добавив, изменив или удалив позицию.

08_12-001-0001

Название	Сумма
Платежи за месяц	0.00
Долг на начало март	0.00
Платежи за март 2012 гц.	0.00
Платежи пред. месяца	0.00

Позиция	Сумма	Кол.-во	Единицы
Долг на начало месяца	0.00	1	шт.

Название
Долг на начало март

Ок Отмена

Вы можете сохранять промежуточные результаты правки в файл, используя область внизу окна (поле выбора файла, кнопки **Сохранить** и **Восстановить**).

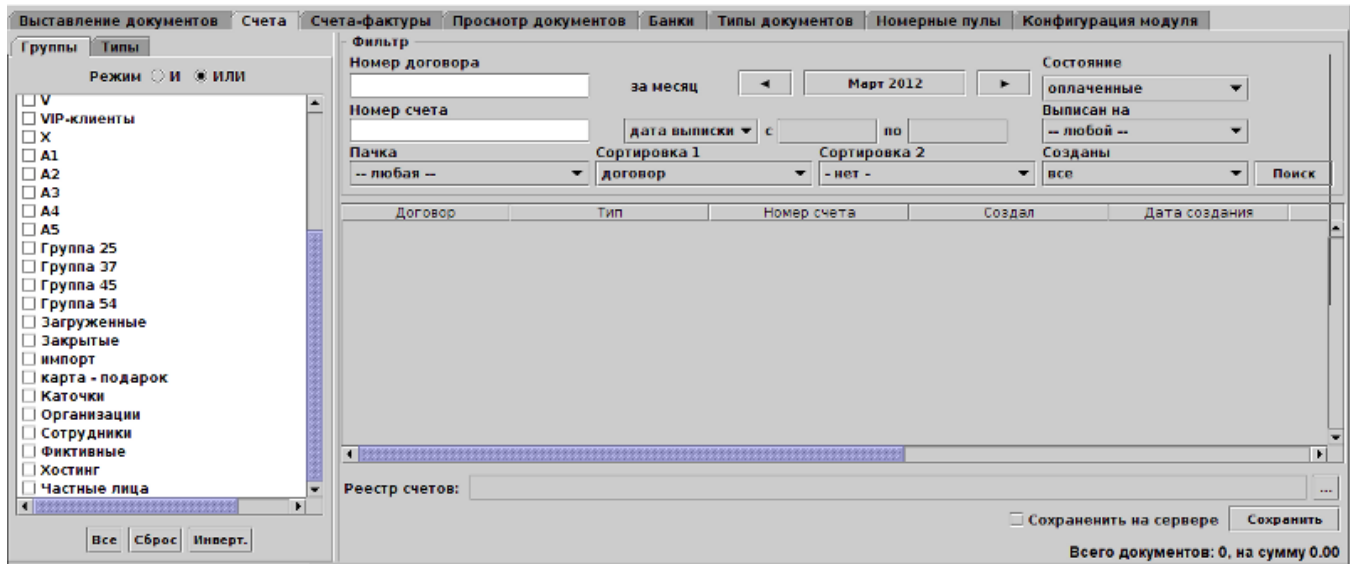
Отредактируйте позиции, если необходимо, уберите галочки в таблице с документов, которые создавать не нужно и нажмите кнопку **Создать счета**, либо **Создать счета-фактуры**. Создадутся только те документы, сумма которых больше нуля! Чтобы документ создавался с отрицательной позицией в типе документа должна стоять опция **Создавать при сумме счета <= 0**. Начальное состояние галочек в таблице задаётся опциями **конфигурации модуля**. При создании счетов необходимо выбрать счёт в банке, это делается в выпадающем списке **Банк. счет**, к которому привязывается документ, данные используются для генерации образца платёжного поручения в печатной форме счета.

Выставить счета можно также без предварительного просмотра сгенерированных счетов и редактирования их позиций. Для этого сначала задаются все необходимые фильтры и потом нажимается кнопка **Сгенерировать+Создать**.

Созданным документам присваиваются номера и они сохраняются в базе, дальнейшая работа с ними происходит на вкладках **Счета** и **Счета-фактуры** модуля, либо в отчёте модуля бухгалтерии в карточке договора.

Работа со счетами

Производится на вкладке **Счета**.



Здесь доступны множество различных фильтров для счетов, с помощью которых легко найти необходимые счета. При нажатии кнопки **Поиск** происходит поиск счетов и вывод их в таблицу. Возможно отсортировать найденные документы при помощи выпадающих списков **Сортировка 1** и **Сортировка 2** (сперва сортируется по сортировке 1, затем по сортировке 2). Поддерживаются следующие режимы сортировки документов:

- **договор/номер** - стандартный режим, сортировка по номеру договора;
- **договор (учитывая суб.)** - стандартный режим, сортировка по номеру договора с учетом номеров суб договоров;
- **номер** - сортировка по номеру документа по возрастанию;
- **номер обр.** - сортировка по номеру документа по убыванию;
- **ID** - сортировка по коду документа в БД (по возрастанию);
- **ID обр.** - сортировка по коду документа в БД (по убыванию);
- **улица/дом** - сортировка по улице/номеру дома/квартире;
- **индекс** - сортировка по индексу/улице/номеру дома/квартире;
- **город/квартал** - сортировка по городу/кварталу;
- **ФИО/наименование** - сортировка по параметру ФИО/наименование договора.

При использовании 4-х последних режимов сортировки в [конфигурации модуля](#) должны быть установлены соответствующие параметры - код параметра договора типа **Адрес** и код параметра договора типа **Текст**.

Для выбора нескольких строк таблиц воспользуйтесь кнопками **Ctrl, Shift**, либо комбинацией **Ctrl+A** для выбора всех строк. Нажатием правой кнопки мыши может быть вызвано всплывающее меню, применяющее одно из следующих действий ко всем выбранным документам:

- При пометке счета оплаченным автоматически проводится платёж с типом, указанным в банковском счёте, на который создан счёт и текущей датой. Автоматическое проведение платежей можно отключить, установив опцию в [конфигурации модуля](#).
- **Оплачено сегодня(dd.MM)** - все выбранные счета помечаются как оплаченные с сегодняшней датой;
- **Оплачено вчера(dd.MM)** - все выбранные счета помечаются как оплаченные вчерашней датой;
- **Оплачено на дату** - все выбранные счета помечаются как оплаченные с выбранной датой;
- **Не оплачено** - снять метку оплаченности с выбранных документов;
- **Просмотр** - поместить выбранные документы в панель просмотра (вкладка **Просмотр документов**). В панели просмотра вы можете распечатать выбранные документы, сохранить их, либо отправить на почту (см. далее про панель просмотра документов);
- **Вставить в просмотр** - документы вставляются в окне просмотра вслед за документами, принадлежащими тому же договору. Таким образом для распечатки 2х копий счет-фактур и одной копии счета необходимо выбрать счета-фактуры, выбрать **Просмотр**, затем это же выделение **Вставить в просмотр**, после чего выбрать счета для этих же договоров, а также **Вставить в просмотр**. В окне просмотра появится последовательность листов "счет-фактура - счет-фактура - счет";
- **Отослать на почту** - создается задание для планировщика, которое отправляет все выбранные документы на почту. Код параметра договора для почты устанавливается в конфигурации модуля.

Двойным щелчком мыши по любой из строк вы можете вызвать редактор документа:

Номер:
 Номер (год):
 Номер (месяц):
 Дата выписки:

Позиция	Сумма	Кол.-во	Единицы
Наработка по Телефонии	0.00	1	шт.

Название:

Вы можете скорректировать все три порядковых номера, на основании которых формируется номер документа (абсолютный порядковый, в году, в месяце), дату выписки и позиции документа. Редактирование позиций идентично тому, что производится при генерации документов. При сохранении документа с уже занятыми номерами абсолютным в году, либо в месяце выводится предупредительное сообщение, однако сохранение номера производится.

Перечень счетов и счет-фактур можно выгрузить в реестр CSV-формата. В реестре отображается номер счета, сумма, дата выставления, тип и адрес. Параметр адреса задаётся переменной [конфигурации модуля](#). Для выгрузки используется панель под таблицей счетов. Также можно настроить формат выгрузки счетов и счет-фактур при помощи шаблонов xsl. Для использования этого способа выгрузки необходимо в [конфигурации модуля](#) указать шаблоны для счетов и счетов-фактур. Шаблоны по умолчанию для выгрузки счетов и счет-фактур лежат в директории **BGBillingServer/webroot/xsl** и называются соответственно **preempt_bill_reestr.xsl** и **preempt_invoice_reestr.xsl**. Собственные шаблоны выгрузки необходимо разместить в той же директории, где находятся базовые шаблоны, а имена шаблонов прописать в [конфигурации модуля](#). Если параметры для выгрузки через шаблоны не будут указаны, то выгрузка будет происходить в старом формате, т.е. будет выгружаться таблица со списком найденных счетов или счетов-фактур.

Также возможно выгрузить реестр на стороне сервера. Для этого необходимо перед выгрузкой в клиенте указать данную опцию, выбрав галочку, и в конфигурации модуля указать следующие параметры: **preempt.bill.reestr.to.server.path=<Path>** - для выгрузки счетов и **preempt.invoice.reestr.to.server.path=<Path>** - для выгрузки счетов-фактур, где **<Path>** - путь с именем файла, куда будет происходить выгрузка. Необходимо проветить права на запись у указанной директории, чтобы сервер смог туда записать файлы. При выгрузке реестра, если файл не существует, то он будет создан, а если существует, то все данные, содержащиеся в файле, будут заменены новыми.

artur_test	RSCM	B-00474	kostya	01.12.2011
artur_test	test lldar	B-00553	kostya	05.12.2011
artur_test	asiacom2	B-00626	kostya	05.12.2011
artur_test	test lldar	B-00513	kostya	05.12.2011
artur_test	test lldar	B-00593	kostya	05.12.2011
artur_test	test lldar	B-00473	kostya	01.12.2011
artur_test	АСВТ-НН, физ-лица	B-00552	kostya	05.12.2011
artur_test	boris_sverka	B-00625	kostya	05.12.2011
artur_test	АСВТ-НН, физ-лица	B-00512	kostya	05.12.2011
artur_test	АСВТ-НН, физ-лица	B-00592	kostya	05.12.2011

Реестр счетов:

Сохранить на сервере

Первичная подготовка для курьерской службы

Модуль позволяет проводить первичную подготовку пакетов документов для курьерской службы: осуществляет сортировку по улице/дому и группировку по пачкам (маршрутам). Для активизации данной функции в конфигурации модуля должны быть указаны параметры **package.param.id** и **address.sort.param.id**. Параметр **package.param.id** определяет код параметра договора типа **Список**, который задаёт маршрут (пачку) для договора. Перечень допустимых значений данного параметра указывается в **Справочники=>Другие=>Значения списков**. Параметр определяется для каждого договора и задаёт маршрут, к которому относится договор.

Если код спискового параметра указан верно в фильтре **Пачка**, в окне управления счетами будут отображаться допустимые значения спискового параметра договора. Например, это может быть: Запад, Центр, Восток. Или иное деление. В общем случае, пачка может нести любой смысл, позволяя разделить документы.

Для субдоговоров при отсутствии явно заданного параметра с пачкой используется значение из супердоговора.

Работа со счетами-фактурами, Актами и УПД

Производится на вкладке **Счета-фактуры**.

ID	Номер	Дата создания	Договор (Комментарий)	Тип	Создал	Разрешен	Сумма
1	345	11.02.2019	kirill [Сергеев Кирилл]	УПД	<Пользователь неизвестен>	<input type="checkbox"/>	111,00

По сравнению со счетами набор фильтров уменьшен.

В всплывающем меню доступны опции добавления и вставки в панель просмотра, а также дополнительные пункты **Разрешено к показу** и **Запрещено к показу**. Разрешенный к показу документ доступен на странице статистики пользователя. Возможность реализована для проведения предварительной сверки и корректировки фактур.

Разрешенность к показу сгенерированного счета-фактуры в Web задается опцией в [конфигурации модуля](#).

Двойным щелчком мыши по любой из строк вы можете вызвать редактор документа:

Название	Сумма	Количество	Единицы
Абонентская плата	111,00	1	юниты

Поле **Исправление** предназначено для подстановки в счет-фактуру при выгрузке печатной формы порядкового номера исправлений, внесенных в счет-фактуру. Поле **Платежно-расчетный документ** позволяет сделать ссылку на документ, к которому относится данная счет-фактура (например, счет).

В случае, если поля **Платежно-расчетного документа** пустые, а поле **Исправление** равно 0, то в счет-фактуре будет отображаться прочерк на соответствующем месте.

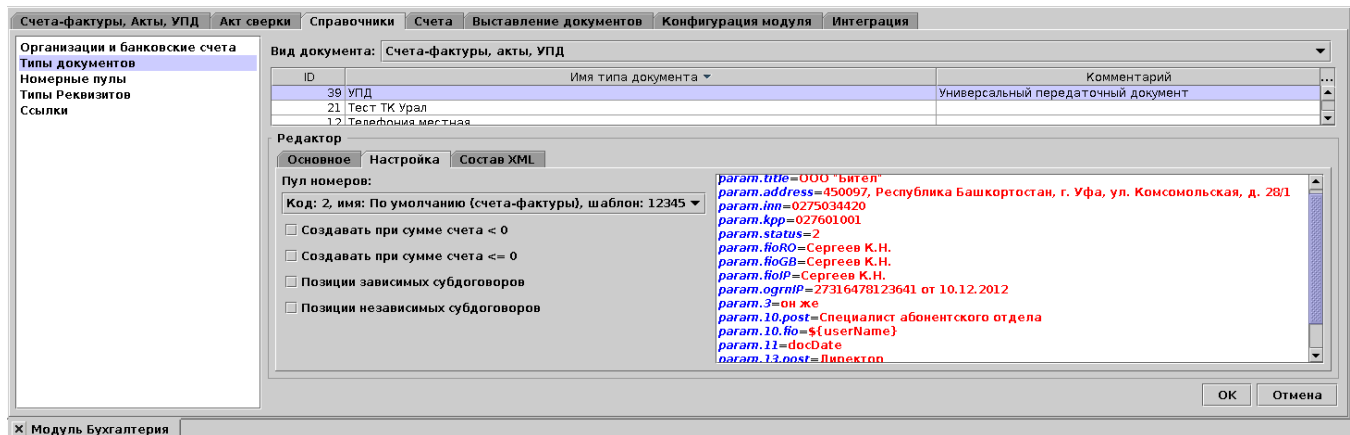
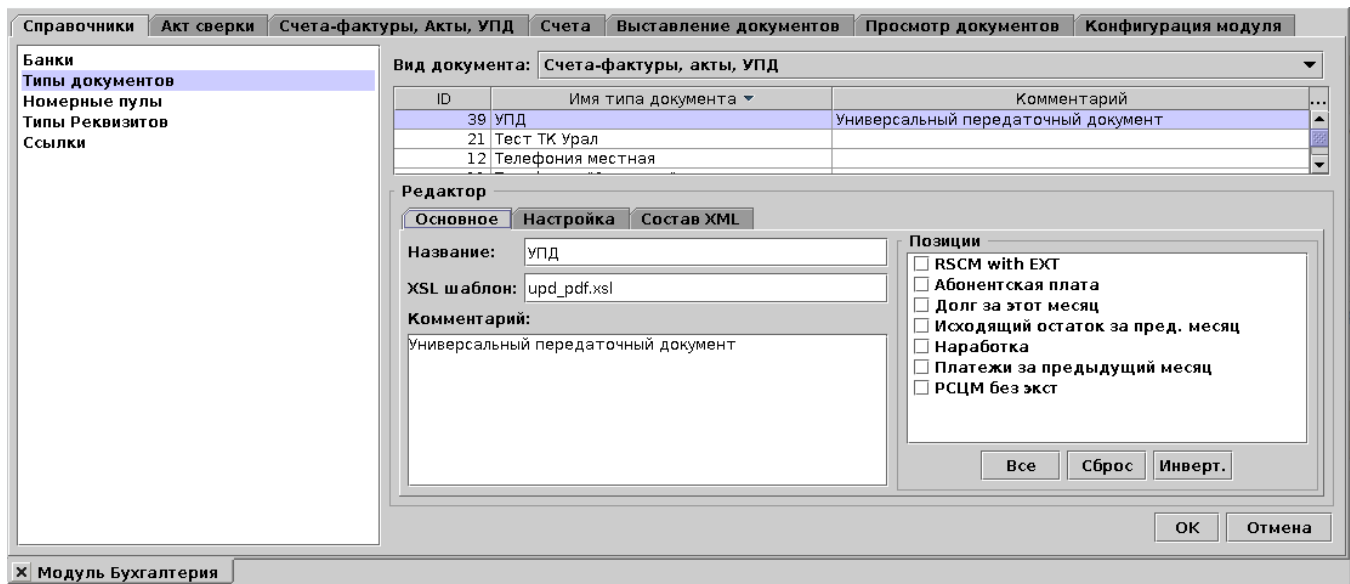
Настройка УПД

В биллинге есть шаблон УПД, если вы планируете его использование без модификаций, то необходимо сделать дополнительные настройки

1) создать тип документа УПД

Из меню Модули - Модуль Bill - Закладка Справочники - Справочник Типы документов, выбрать Вид документа - Счета-фактуры, Акты, УПД.

Добавить новый документ, и настроить его, см. скриншоты ниже



где

Универсальный передаточный документ

Счет-фактура № 12345 от 11.02.2019 (1)
Исправление № - (1a)

Продавец: ООО "Бител" param.title
Адрес: 450097, Республика Башкортостан, г. Уфа, ул. Комсомольская, д. 28/1 param.address
ИНН / КПП продавца: 0275034420 / 027601001 param.inn / param.kpp
Грузополучатель и его адрес: param.3
К платежно-расчетному документу: -
Покупатель: ООО "Рога и копыта"
Адрес: 45000, Республика Башкортостан, г. Уфа, ул. Ленина, д.12, оф. 56
ИНН / КПП покупателя: 0275032044 / 027601001
Валюта: наименование, код: Российский рубль, 643

Приложение №1 к постановлению Правительства Российской Федерации от 26 декабря 2011 г. № 1137

param.status
Статус: 2

1 - счет-фактура и передаточный документ (акт)
2 - передаточный документ (акт)

№	Код товара / работ, услуг	Наименование товара (описание выполненных работ, оказанных услуг), имущественного права	Единица измерения		Цена (тариф) за единицу измерения	Стоимость товаров (работ, услуг), имущественных прав без налога - всего	В том числе сумма акциза	Налоговая ставка	Сумма налога, подлежащая уплате покупателем	Стоимость товаров (работ, услуг), имущественных прав с налогом - всего	Страна происхождения товара		Номер таможенной декларации
			код	условное обозначение (национальное)							Коды по наименованию	Коды по наименованию	
A	B	T	2	2a	4	5	6	7	8	9	10	10a	11
1		Абонентская плата	499	юниты	92,50	92,50	-	20	18,50	111,00	-	-	-
Всего к оплате						92,50	X	X	18,50	111,00			

Документ составлен на 1 листе

Руководитель организации или иное уполномоченное лицо: Сергеев К.Н. param.fioRO
Индивидуальный предприниматель: Сергеев К.Н. param.fioIP
Главный бухгалтер или иное уполномоченное лицо: Сергеев К.Н. param.fioGB
Индивидуальный предприниматель: 27316478123641 от 10.12.2012 param.ogrnIP

Основание передачи(сдачи)/получения(приема): Договор № Kp11 от 29.01.2008 [8]

Данные о транспортировке и грузе: [9]

Товар(груз) передан(услуг, результаты работ, права сданы) param.10.post param.10.fio [10]

Дата отгрузки, передачи(сдачи) " " 20 г. [11]

Иные сведения об отгрузке, передаче [12]

Ответственный за правильность оформления факта хозяйственной жизни: param.13.post param.13.fio [13]

Наименование экономического субъекта - составителя документа (в т.ч. комиссионера /агента) [14]

в значение параметров можно указать макрос `{userName}`, он будет заменен на имя пользователя генерирующего документ. При указании параметра `param.11=docDate`, в поле [11] будет подставлена дата документа

2) настроить типа Реквизитов

Из меню Модули - Модуль Вил - Закладка Справочники - Справочник Типы Реквизитов. Добавить типы как на скриншоте ниже.

Выставление документов | Просмотр документов | Конфигурация модуля

Справочники | Акт сверки | Счета-фактуры, Акты, УПД | Счета

Банки
Типы документов
Номерные пулы
Типы Реквизитов
Ссылки

Код	Название	Описание	Проверка	...
1	inn	ИНН		
2	org_name	Покупатель		
3	address	Адрес		
4	kpp	КПП		
5	pol_org_name	Грузополучатель и его адрес		

Модуль Бухгалтерия

Затем в договоре клиента Добавить модуль Вил на договор, выбрать его, перейти на вкладку **Реквизиты, банковский счет**. Добавить реквизиты как на скриншоте ниже.

Параметры Отчет Карточки Рассылки CRM HelpDesk Бонусы Документы

Договор kirill [Сергеев Кирилл] 32603 Период с 29.01.2008 по

Реквизиты, Банковский счет Виды счетов Виды счетов-фактур, Актов, УПД

Банковский счет По умолчанию

Код	Реквизит	Значение	Период
46	ИНН	0275032044	...
47	Покупатель	000 "Рога и копыта"	...
48	Адрес	45000, Республика Башкортостан, г. Уфа, ул. Ленина, д.12, оф. 56	...
49	КПП	027601001	...
50	Грузополучатель и его адрес	000 "Рога и копыта", 45000, Республика Башкортостан, г. Уфа, ул. Ленина, д.12, оф. 56	...

МобСаш MPS Rentsoft RSCM Бухгалтерия Карточки Лицензии Новые абонлаты Подписки Группы Флиал УФА Скрытый kirill

Универсальный передаточный документ № 12345 от 11.02.2019 (1) (1a) Приложение №1 к постановлению Правительства Российской Федерации от 26 декабря 2011г. № 1137

Статус: 2

1 - с/фактура и передаточный документ (акт)
2 - передаточный документ (акт)

Продавец: ООО "Бител" param.title
Адрес: 450097, Республика Башкортостан, г. Уфа, ул. Комсомольская, д. 28/1 param.address
ИНН / КПП продавца: 0275034420 / 027601001 param.inn / param.kpp
Грузополучатель и его адрес: pol_org_name
К платежно-расчетному документу: *
Покупатель: ООО "Рога и копыта" org.name
Адрес: 45000, Республика Башкортостан, г. Уфа, ул. Ленина, д.12, оф. 56 address
ИНН / КПП покупателя: 0275032044 / 027601001 inn / kpp
Валюта: наименование, код: Российский рубль, 643

№ п/п	Код товара / работ, услуг	Наименование товара (описание выполненных работ, оказанных услуг), имущественного права	Единица измерения		Количество (объем)	Цена (тариф) за единицу измерения	Стоимость товаров (работ, услуг), имущественных прав без налога - всего	В том числе сумма акциза	Налоговая ставка	Сумма налога, предъявляемая покупателю	Стоимость товаров (работ, услуг), имущественных прав с налогом - всего	Страна происхождения товара		Номер таможенной декларации
			код	условное обозначение (национальное)								цифровой код	Краткое наименование	
1		Абонентская плата	499	кюпиты	1	92.50	92.50	-	20	18.50	111.00	-	-	-
Всего к оплате							92.50	X	X	18.50	111.00			

Документ составлен на 1 листе

Руководитель организации или иное уполномоченное лицо: param.fioRO Сергеев К.Н.
Индивидуальный предприниматель: param.fioIP Сергеев К.Н.
Главный бухгалтер или иное уполномоченное лицо: param.fioGB Сергеев К.Н.
param:ogrnIP 27316478123641 от 10.12.2012

Основание передачи(сдачи)/получения(приема) Договор № kirill от 29.01.2008 [8]

Данные о транспортировке и грузе [9]

Товар(груз) передан(услуга, результаты работ, права сданы) [10] Товар(груз) получен(услуга, результаты работ, права приняты) [15]

Дата отгрузки, передачи(сдачи) " " 20 г. [11] Дата получения(приема) " " 20 г. [16]

Иные сведения об отгрузке, передаче [12] Иные сведения о получении, приеме [17]

Ответственный за правильность оформления факта хозяйственной жизни [13] Ответственный за правильность оформления факта хозяйственной жизни [18]

Наименование экономического субъекта - составителя документа (в т.ч. комиссионера/агента) [14] Наименование экономического субъекта - составителя документа [19]

М.П. М.П.

Работа с актами-сверки

Панель просмотра документов

На панели просмотра отображаются счета и счета-фактуры добавленные, либо вставленные в просмотр.

Выставление документов | Счета | Счета-фактуры | **Просмотр документов** | Банки | Типы документов | Номерные пулы | Конфигурация модуля

Печать блоками | Печать по номерам (включительно)

Размер блока: 100 с блока: 1 Печать С документа: 1 * по: 100 * Печать Масштаб: 150

Документ: 1 из 16 [25] Страница: 1 из 1 [25]

1) В-00699
2) В-00739
3) В-00705
4) В-00704
5) В-00703
6) В-00702
7) В-00700
8) 123-0094
9) В-00756
10) В-00701
11) В-00716
12) В-00715
13) В-00714
14) В-00713
15) 123-0089
16) В-00712

450001, Российская Федерация, Республика Башкортостан, г.XXX, проспект Октября, д.Х, адм. здание
ИНН 0000000000, код по ОКВЭД 82000, код по ОКПО 45219144, КПП 0000000000

Получатель ИНН 0000000000 КПП 0000000000 ОАО "XXXXXXX"	Сч. № БИК Сч. №	0000000000 1111111111111111 33333333333333
Банк получателя БАНК		
Назначение платежа	Оплата по дог. 001482 и счету В-00699 от 02.03.2012. НДС в том числе -364.12 руб.	

счет действителен в течении 30 дней

Счет № В-00699 от 02.03.2012

Плательщик:
Телефон:
Договор: 001482 валюта: рубль

№	Товар	Ед.	Кол-во	Цена *	Сумма	НДС	Всего с НДС
1	Долг на начало март	шт.	1	-2022.88	-2022.88	-364.12	-2387.00
2	Платежи за март 2012 гц	шт.	0	0.00	0.00	0.00	0.00
Итого:					-2022.88	-364.12	-2387.00

* Цена не включает НДС

Итого к оплате: минус две тысячи триста восемьдесят семь рублей 00 копеек
Счет выдал
Для своевременного поступления суммы оплаты на Ваш Лицевой счет в платежном поручении просим указывать
НОМЕР КОНТРАКТА!

Модуль Бухгалтерия

Элемент управления в верхней части окна позволяет перелистывать документы. Список документов также находится в левой части окна просмотра и позволяет быстро перейти к нужному документу. Следующий элемент - перелистывание страниц в пределах одного документа. Кнопка **XML** позволяет просмотреть исходный XML-документ, на основании которого генерируется печатная форма. Этот режим полезен при модификации и отладке XSLT-шаблонов для генерации печатных форм (см. далее). Возврат в режим просмотра печатной формы осуществляется повторным нажатием кнопки **XML**. По нажатию на кнопку **Разослать на почту** происходит добавление задачи в планировщик «Задача рассылки документов с компоновкой». В этом режиме для отправки документа e-mail берется из конфигурации модуля. Отправляются все документы, добавленные в просмотрщик, по всем договорам, у которых они добавлены.

Выставление документов | Счета | Счета-фактуры | **Просмотр документов** | Банки | Типы документов | Номерные пулы | Конфигурация модуля

Разослать на почту | Размер блока печати: 100 с блока: 1 Печать | Размер: 150

Документ: 1 из 1 [25] Страница: 1 из 1 [25]

```
<?xml version="1.0" encoding="windows-1251"?><data secret="F8089A44071F149D6BF8AC9C274DA8229" status="ok">
<inline_params doc="Разработчик" docver="Двадцатьность 1" name="оператор 1" prikaz="Плрказаа 1"/>
<bill account="0000000000" amounts="0.00" balance_mode="0" bank title="БАНК" bik="11111111111111111111" bill_number="В-00722" charges="0.00" cid="352984" contract="artur_test" contract_date
<pos awlz="0" insum="1" name="Долг на начало март" position id="1" qtynums="0" quantity="1" sumUnscaled="1003.400000" summ="501.70" unit="шт." unitCode="796"/>
<pos awlz="1" insum="1" name="Платежи за март 2012 гц." position id="2" qtynums="0" quantity="0" sumUnscaled="0.000000" summ="0.00" unit="шт." unitCode="796"/>
<tariff_bill/>
<contract_data cid="352984">
<contract balance_rest="501.70" cid="352984" comment="Гареев Артур Маратович" date1="24.11.2009" date2="" fc="0" gr="11529215046006845976" limit="-2050.00" mode="0" param_gr
<module id="143" name="bvcom" title="BVCom"/>
<module id="179" name="inet" title="NET"/>
<module id="80" name="rscm" title="RSCM"/>
<module id="187" name="softkey" title="Softkey"/>
<module id="65" name="bill" title="Бухгалтерия">
<bill contract_doc type_65 cid="352984" doc_type="2" id="292"/>
<bill contract_doc type_65 cid="352984" doc_type="22" id="293"/>
<bill contract_doc type_65 cid="352984" doc_type="29" id="294"/>
<bill contract_doc type_65 cid="352984" doc_type="13" id="295"/>
<bill contract_doc type_65 cid="352984" doc_type="27" id="296"/>
<bill contract_doc type_65 cid="352984" doc_type="11" id="297"/>
<bill contract_doc type_65 cid="352984" doc_type="12" id="298"/>
<bill contract_doc type_65 cid="352984" doc_type="21" id="299"/>
<bill contract_doc type_65 cid="352984" doc_type="20" id="300"/>
<bill contract_doc type_65 cid="352984" doc_type="8" id="431"/>
<bill contract_doc type_65 cid="352984" doc_type="4" id="432"/>
<bill contract_doc type_65 cid="352984" doc_type="26" id="433"/>
<bill contract_doc type_65 cid="352984" doc_type="16" id="434"/>
<bill contract_doc type_65 cid="352984" doc_type="17" id="435"/>
<bill contract_doc type_65 cid="352984" doc_type="26" id="436"/>
<bill contract_doc type_65 cid="352984" doc_type="14" id="437"/>
<bill contract_doc type_65 cid="352984" doc_type="33" id="438"/>
<bill contract_doc type_65 cid="352984" doc_type="15" id="439"/>
<bill contract_doc type_65 cid="352984" doc_type="32" id="440"/>
<bill contract_doc type_65 cid="352984" doc_type="25" id="441"/>
<bill contract_doc type_65 cid="352984" doc_type="34" id="442"/>
<bill contract_doc type_65 cid="352984" doc_type="23" id="443"/>
<bill contract_doc type_65 cid="352984" doc_type="10" id="444"/>
<bill contract_doc type_65 cid="352984" doc_type="31" id="445"/>
<bill contract_doc type_65 cid="352984" doc_type="11" id="446"/>
```

Кнопки сохранения и отправки на почту позволяют сохранить/отправить текущий просматриваемый документ. Печать документов производится в поблочном режиме, размер блока печати задаётся в окне. После отправки на печать каждого блока выводится новый диалог печати. Данная функция предназначена для потоковой печати большого количества документов.

Печать по номерам документов позволяет печатать диапазон добавленных в просмотр документов. Клик на звёздочку - быстрый переход к просмотру документа с введённым номером .

Генерация печатных форм

Генерация печатной формы производится XSLT-шаблоном, указанным в типе счета или счета-фактуры. Если шаблоны не указаны, используются шаблоны по умолчанию **bill_pdf.xml**, который используется при генерации визуального представления счета и **invoice_pdf.xml** для счет-фактуры. Все XSLT-шаблоны располагаются в каталоге **BGBillingServer/webroot/xsl**.

Минимальная настройка стандартных шаблонов включает в себя изменение реквизитов организации:

```
<!-- -->

<xsl:variable name="postal_address" select="'450001, ., ., .4, . '" />

<xsl:variable name="INN" select="' 0275023387'" />
<xsl:variable name="KPP" select="' 027801001'" />
<xsl:variable name="title" select="' &quot;&quot;'" />

<xsl:variable name="OKVED" select="' 82000'" />
<xsl:variable name="OKPO" select="' 45219144'" />

<!-- -->
```

Они используются для построения верхней части PDF-документа (см. ниже). В каталоге **BGBillingServer/webroot** находится изображение печати - файл **stamp.gif**. Замените его на отсканированную копию печати вашей организации.

При каждом обновлении модуля bill файл перетирается, так же как и шаблоны по умолчанию. Вы можете переименовать файл и ссылки на него в шаблонах счетов и счетов-фактур.

Шаблоны по умолчанию используют для оформления печатных форм реквизиты договора из свойств модуля (перечень приведен в конфигурации модуля по умолчанию). Вы можете использовать для этих же целей параметры договоров, но это потребует модификации XSLT-шаблонов.

Банковские реквизиты берутся из справочника банковских реквизитов (текущий указывается при создании счета). В графе **Плательщик** указывается комментарий договора. Вычисление суммы НДС производится в момент генерации печатной формы.

В базе данных информация о счёте, либо счёте-фактуре сохраняется в виде XML-документа и содержит следующие поля:

1. сумму документа;
2. набор позиций с суммами;
3. параметр НДС из конфигурации модуля;
4. параметры договора;
5. реквизиты модуля Бухгалтерия из договора;
6. INLINE-параметры - специфичные для текущего пользователя (номера доверенностей, Ф.И.О.).

Модуль осуществляет подстановку INLINE-параметров в XML-документ в зависимости от пользователя, просматривающего документы. Перечень INLINE-параметров определяется в **конфигурации модуля**. Для просмотра исходного XML-документа, из которого генерируется печатная форма, необходимо воспользоваться панелью просмотра документа.

При модификации и отладке XSLT-шаблона следует отключить кэширование XSLT-шаблонов (опция **xslt.cache** в конфигурации сервера биллинга).

Отчёты договора модуля

В сводном отчёте отображаются счета и счет-фактуры договора с обратной сортировкой по месяцу, за который они выставлены.

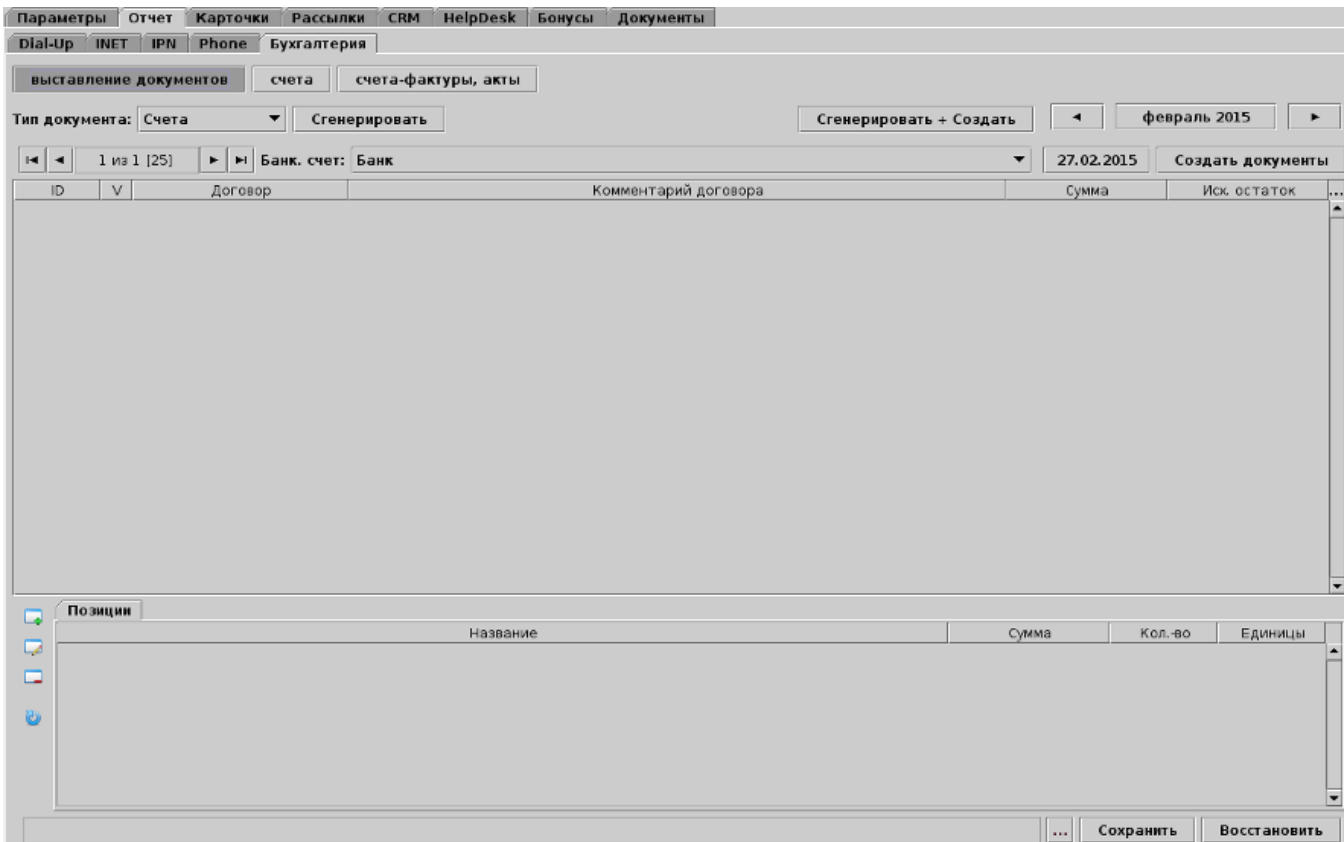
Месяц	Тип	Номер счета	Дата создания	Создал	Дата оплаты	Сумма	Отметил оплату	Статус
2012.03	Счет за интернет	В-00772	15.03.2012	kostya		10.00		не оплачен
2012.03	FFFF	В-00771	15.03.2012	kostya		0.00		не оплачен
2012.03	boris_sverka	В-00770	15.03.2012	kostya		-10.00		не оплачен
2012.03	Квитанция тест	В-00757	15.03.2012	kostya		0.00		не оплачен
2012.03	Счет за интернет	В-00755	15.03.2012	kostya		10.00		не оплачен
2012.03	FFFF	В-00754	15.03.2012	kostya		0.00		не оплачен
2012.03	boris_sverka	В-00753	15.03.2012	kostya		-10.00		не оплачен
2012.03	Квитанция тест	В-00740	15.03.2012	kostya		0.00		не оплачен
2012.03	Счет за интернет	В-00737	02.03.2012	Гареев Артур		10.00		не оплачен
2012.03	FFFF	В-00736	02.03.2012	Гареев Артур		0.00		не оплачен
2012.03	boris_sverka	В-00735	02.03.2012	Гареев Артур		-10.00		не оплачен
2012.03	Квитанция тест	В-00734	02.03.2012	Гареев Артур		10.00		не оплачен
2012.01	Квитанция тест	В-00690	11.01.2012	Гареев Артур		-1778735.08		не оплачен
2012.01	Счет за интернет	В-00691	11.01.2012	Гареев Артур		-1778735.08		не оплачен
2012.01	FFFF	В-00690	11.01.2012	Гареев Артур		0.00		не оплачен
2012.01	boris_sverka	В-00689	11.01.2012	Гареев Артур		1778735.08		не оплачен
2011.12	boris_sverka	В-00653	05.12.2011	kostya		-1775593.81		не оплачен
2011.12	Квитанция тест	В-00652	05.12.2011	kostya		1775593.81		не оплачен
2011.12	Счет за интернет	В-00646	05.12.2011	kostya		-1775593.81		не оплачен
2011.12	FFFF	В-00645	05.12.2011	kostya		0.00		не оплачен
2011.12	Квитанция тест	В-00613	05.12.2011	kostya		-1775593.81		не оплачен
2011.12	Счет за интернет	В-00600	05.12.2011	kostya		-1775593.81		не оплачен
2011.12	FFFF	В-00607	05.12.2011	kostya		0.00		не оплачен
2011.12	boris_sverka	В-00606	05.12.2011	kostya		1775593.81		не оплачен
2011.12	Квитанция тест	В-00573	05.12.2011	kostya		-1775593.81		не оплачен

Для редактирования счетов или счет фактур вызовите редактор двойным кликом по строке.

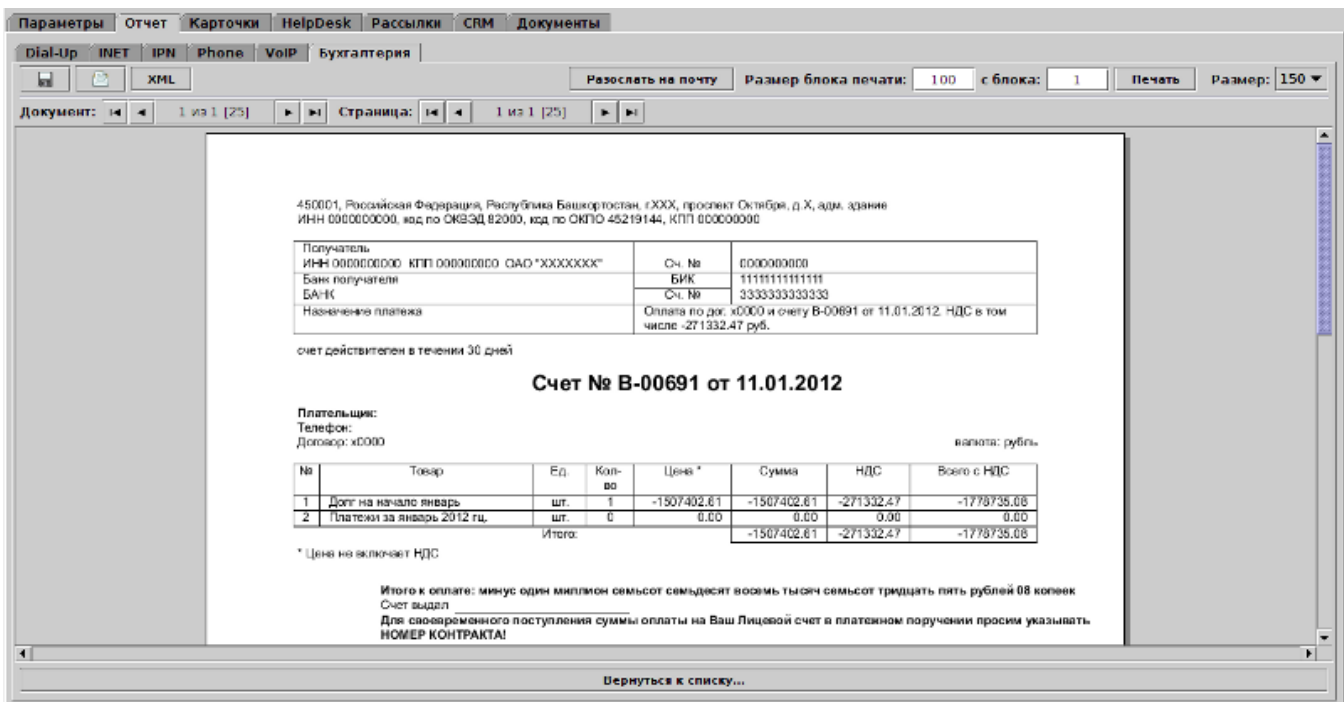
Название	Сумма	Кол-во	Единицы
Долг на начало декабря	510.00	1	шт.
Платежи за декабрь 2014 г.	0.00	0	шт.
Абонплата 60 в месяц	60.00	1	шт.

Название	Сумма	Кол-во	Единицы
Наработка за май	137.80	1.000	шт.

Также имеется возможность выставление счетов и счет-фактур в отчетах договора.



Для выбора документов в таблице воспользуйтесь мышью с зажатыми клавишами **Shift** и **Ctrl**. При нажатии правой кнопкой мыши появляется контекстное меню, позволяющее просмотреть выбранные документы (пункт **Просмотреть**). Для счетов доступны пункты **Оплачено сегодня**, **Оплачено вчера**, **Оплачено на дату** и **Не оплачено**. Панель просмотра отображается на месте таблицы, для возвращения к таблице документов воспользуйтесь кнопкой **Вернуться к списку**.



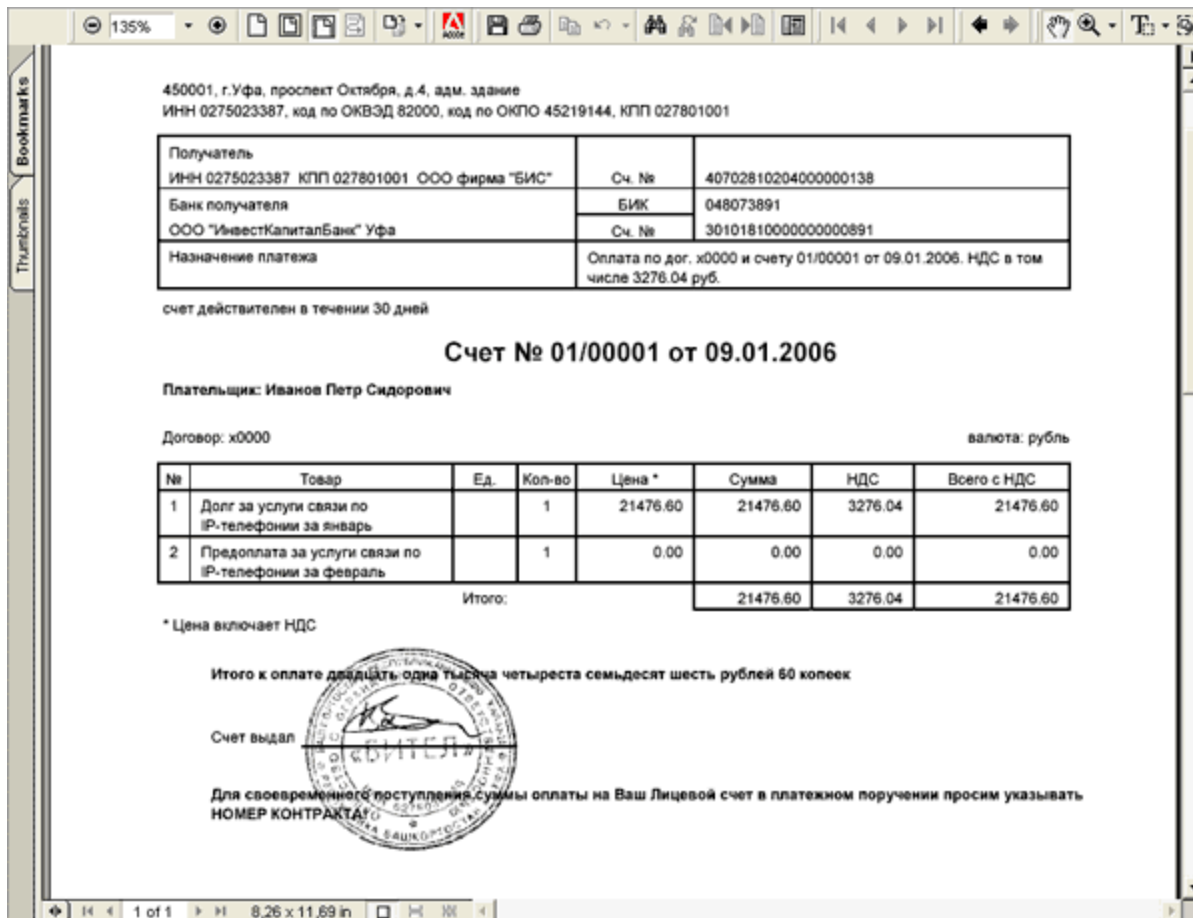
Web-интерфейс абонента

В личном кабинете пользователь может просматривать счета и счет-фактуры в пунктах меню **Счета** и **Счета-фактуры** (названия пунктов меню могут быть изменены в конфигурации модуля). Отображаются все документы договора в режиме обратной сортировки по месяцу, за который они были выписаны.

Счета-фактуры не отображаются в личном кабинете до тех пор, пока не будут разрешены к показу администратором (пункт **Разрешено к показу** во всплывающем меню таблицы счетов-фактур). Данная возможность реализована для возможности предкорректировки документов.

Меню	Список выписанных счетов								
Новости	Страница 1 из 2								
Просмотр баланса	Месяц	Номер	Дата выписки	Дата оплаты	Сумма	Состояние	Тип	PDF	Удаление
Смена пароля на доступ к статистике	2008.09	BC-0900023	22.09.2008	-	7422.00	не оплачен	создан администратором	PDF	
Подписка на рассылки	2008.09	BC-0900022	16.09.2008	-	7400.00	не оплачен	создан администратором	PDF	
Смена тарифных планов	2008.09	BC-0900021	11.09.2008	-	12500.00	не оплачен	создан администратором	PDF	
Дополнительные действия	2008.09	BC-0900020	11.09.2008	-	0.00	не оплачен	создан администратором	PDF	
Примечания	2008.09	BC-0900019	08.09.2008	-	-1340	не оплачен	создан администратором	PDF	
Управление ящиками	2008.09	BC-0900018	03.09.2008	-	77852.48	не оплачен	создан администратором	PDF	
Пересылки	2008.09	BC-0900017	03.09.2008	-	36926.24	не оплачен	создан администратором	PDF	
Смена пароля на ящики	2008.09	BC-0900016	03.09.2008	-	36926.24	не оплачен	создан администратором	PDF	
Платежные системы	2008.09	BC-0900015	03.09.2008	-	36926.24	не оплачен	создан администратором	PDF	
IP-статистика (IPN)	2008.09	BC-0900014	03.09.2008	-	36926.24	не оплачен	создан администратором	PDF	
Управление шлюзом (IPN)	2008.09	BC-0900013	03.09.2008	-	36926.24	не оплачен	создан администратором	PDF	
Пополнение счета с помощью Интернет-карт	2008.09	BC-0900011	29.08.2008	-	52296.64	не оплачен	создан администратором	PDF	
Активация T4uInfo	2008.08	BC-0800010	16.08.2008	-	3855.94	не оплачен	создан Вами	PDF	Удалить
Счета	2008.03	BC-0300009	17.03.2008	02.09.2008	0.00	оплачен	создан администратором	PDF	
Счета-фактуры	2008.03	BC-0300008	17.03.2008	-	0.00	не оплачен	создан администратором	PDF	
Просмотр сессий Voicetr	2008.03	BC-0300007	17.03.2008	-	0.00	не оплачен	создан администратором	PDF	
Наработка по линиям Voicetr									
Смена пароля на логины Voicetr									
Наработка логина Voicetr по направленным									
Пополнение счета с помощью Интернет-карт (new)									

Нажав на ссылку **PDF**, пользователь получит готовый для распечатки счёт в виде PDF-документа:

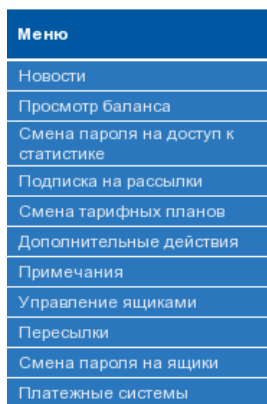


При нажатии на кнопку **Новый счёт** под таблицей просмотра счетов пользователь попадает на страницу создания счета. Если в справочнике банковских реквизитов несколько записей, то рядом с кнопкой **Принять** отображается выпадающий список. Далее он может использовать либо подготовленный счёт, либо ввести свою сумму и создать собственный счёт.

Перечень типов счетов, разрешённых к созданию пользователем через Web, задаётся опцией **конфигурации модуля**. Также тип счета должен быть добавлен в договоре.

Для того, чтобы узнать код типа счета выберите в таблице типов документов нужный тип и нажмите Ctrl+i.

Пользователь может скорректировать сумму счета. Разумеется, самостоятельное создание счетов разрешается только для договоров, работающих по предоплате.



Сформированный счет

Долг на начало октября	1р14.33
------------------------	---------

Пользователь может удалять неоплаченные счета, созданные самостоятельно.

Количество, которое ставится в позиции при генерации счетов из web можно настроить в [конфигурации модуля](#) (хотя это число не имеющее особого значения).

Тонкости интеграции с внешними (1С) системами

В запросе выдачи счетов и счет-фактур (action=Bill, action=Invoice) возможна передача параметра **get_xml=1**. При этом в таблице документов будут выдаваться полные XML со счетами и счет-фактурами.

В запросе пометки счета оплаченным (action=SetPayed) возможна передача параметра **pay=0**. При этом счёт просто будет помечен оплаченным, а приход на баланс проведён не будет.

Групповые операции модуля Bill

Операция позволяет добавить или удалить счета и счет-фактуры в договорах.

Бухгалтерия: Добавление(Удаление) типов документов

Изменяется таблица
bill_contract_doc_type_65

Добавить **Удалить**

Добавлять только отсутствующие типы документов

Счета

- acbt-hh
- asd
- asiacom1
- asiacom2
- boris_test
- FFFFF
- ufanet_bill

Счета- фактуры

- acbt-hh
- asiacom1
- asiacom2
- FDGFDGDFG
- pechtest
- test2page
- Телефония "Совинтел"
- Телефония местная

Модуль DBA

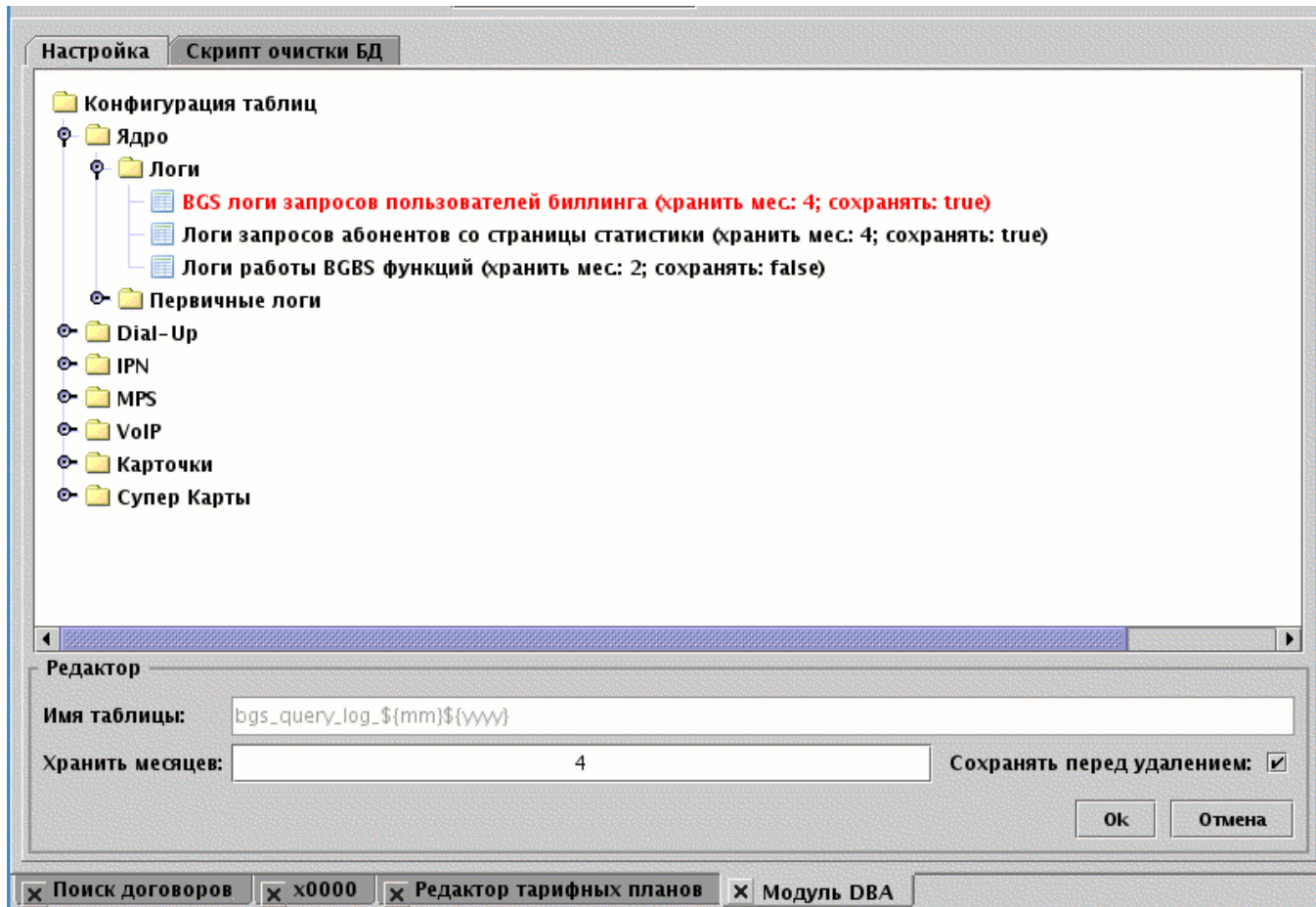
Назначение модуля

Модуль предназначен для выявления старых таблиц в базе данных по заданной конфигурации и генерации скриптов их резервного копирования и удаления из БД.

На данный момент модуль не предназначен для создания актуальных бакапов системы, только для очистки старых данных.

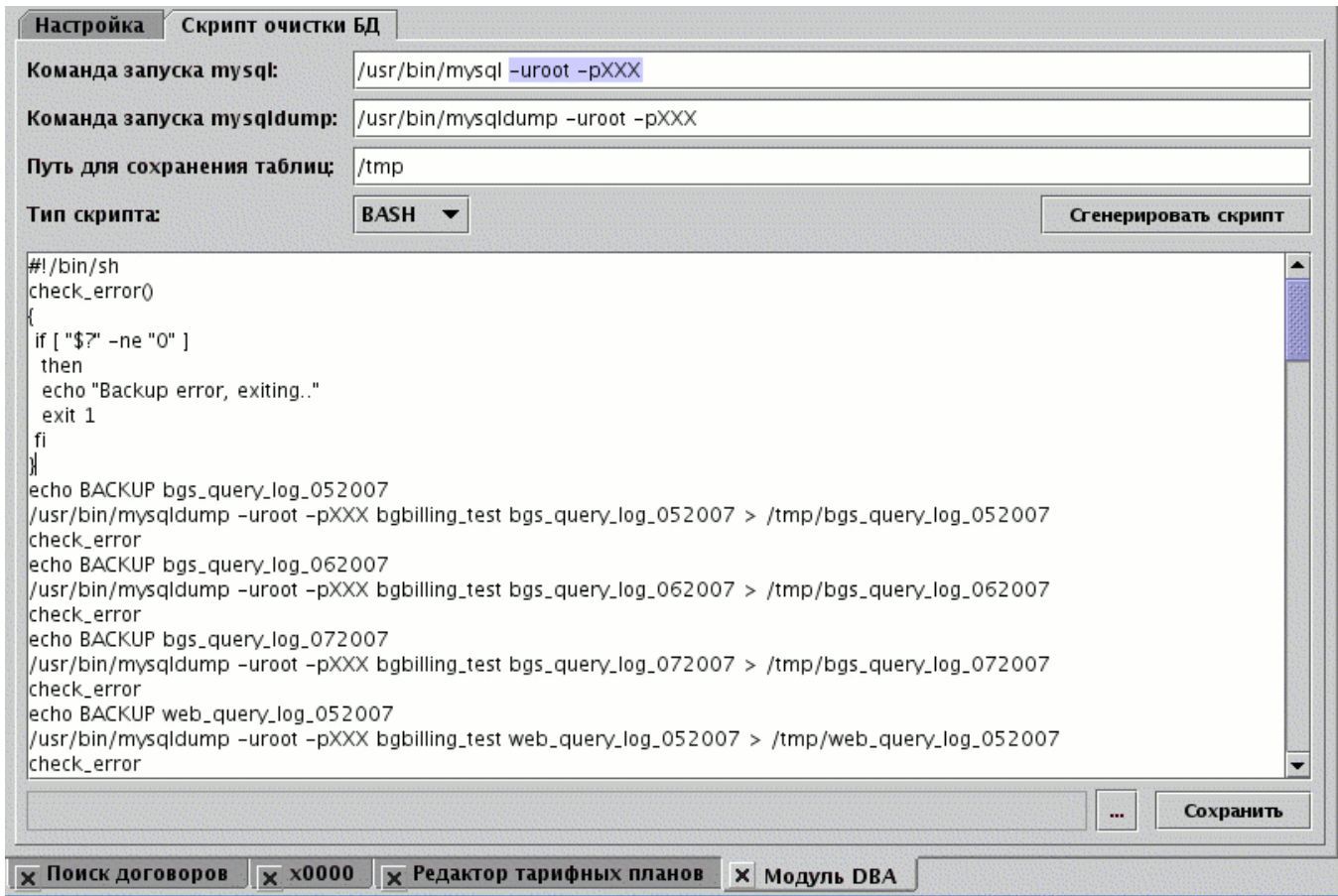
Установка и настройка модуля

Модуль устанавливается с помощью утилиты **bg_installer**, после чего создаётся его экземпляр. На вкладке **Настройка** модуля необходимо выбрать какие таблицы в течении какого срока хранить и нужно ли резервное копирование при их удалении.



Использование модуля

После произведённой настройки по времени жизни таблиц на вкладке **Скрипт очистки БД** можно сгенерировать BASH или Batch файл для резервного копирования и очистки БД.



Имя БД сервер биллинга подставляет из **data.properties**.

В первой части скрипт сохраняет все требуемые таблицы с помощью утилиты mysqldump. Если какая-то из таблиц не будет сохранена успешно, скрипт прерывает свою работу. Далее при успешном резервном копировании удаляются все требуемые таблицы из БД.

Полученный скрипт можно перенести в файл копированием через буфер обмена, либо использованием кнопки **Сохранить**, выбрав предварительно файл в поле ввода файла внизу окна.

Мы рекомендуем вам визуально контролировать содержимое файла для большей надежности. Также на первых этапах возможно следует удалять все таблицы с предварительным резервным копированием.

Модуль DrWeb

Назначение модуля

Модуль автоматизирует работу с Dr.Web AV-Desk. Позволяет пользователям подписываться на услугу, менять тариф, блокировать на время и прекращать подписку. Производит списание наработки с баланса пользователя в соответствии старифами. В настоящий момент поддерживается версия Dr.Web AV-Desk 5 с API2.0.

Базовые понятия и алгоритм работы модуля

При запросе агента через web интерфейс пользователя, проверяется наличие достаточного количества денег, после чего посылается запрос на сервер Dr.Web AV-Desk, полученная ссылка показывается пользователю для загрузки агента.

Задача планировщика "Обработка задач Dr.Web (режим 1)", выполняет несколько задач.

- За определенное количество дней до конца месяца проверяет наличие необходимых средств на балансе и продлевает подписки на следующий месяц. В случае нехватки средств, выполняется запрос на сервер, ограничивающий срок подписки последним днем текущего месяца;
- При поступлении денег на счет пользователя, у которого прекращена подписка из-за нехватки денег, продлевает подписку;
- В первые дни месяца посылает запрос на изменение тарифа для пользователей, которые инициировали смену тарифа.

Задача планировщика "Начисление Dr.Web", выполняет начисление наработки, по активным агентам.

Установка и настройка модуля

Модуль устанавливается с помощью утилиты **bg_installer**, после чего создаётся его экземпляр и прописываются необходимые услуги.

На вкладке **Конфигурация** создайте и установите конфигурацию модуля.

```
# api2 AVDESK
drweb.api.url=http://localhost:9080/api/2.0/
#
drweb.api.user=
#
drweb.api.pswd=
#
drweb.api.encoding=UTF-8
#id Drweb
drweb.charge.id=54
#
drweb.charge.comment= Dr.Web AV-Desk test;
#
drweb.error.balance.message= test
#
graceperiod=yes
#
graceperiod.count=1
# web-
web.menuItem1=Dr.Web AV-Desk -
# -
#drweb.agent.max.count=5
#
#drweb.error.max.count.message=
```

На вкладке модуля **Тарифы** создайте тарифы, соответствующие тарифам в Dr.Web AV-Desk.

Для тарифа можно выбрать следующие параметры:

- **Название** - определяет как тариф будет показываться пользователю;
- **Услуга** - определяет услугу, по которой будет начисляться наработка;
- **Услуга активации** - определяет услугу, по которой будет начисляться наработка в месяц активации агента (месяц первого платежа);
- **Группа** - соответствует группе на сервере Av-Desk;
- **Действует** - задает период действия тарифа;
- **Показывать** - задает период, когда пользователь видит тариф и может на него переключаться (подключаться);

- **Следующий тариф** - определяет тариф, на который нужно переключить агента по истечении срока действия данного тарифа (применяется для акционных тарифов);
- **Показывать для групп** - определяет группы договоров, пользователи которых видят тариф и могут на него переключаться (подключаться);
- **Только для новых** - тариф виден только для первичного подключения, но не для перехода на него в последующем.

Тарифы Начисление Конфигурация модуля

id	Название	Услуга	Услуга активации	Название группы	Показывать	Период действия
3	классик	Dr. Web Классик	Dr. Web Классик	AV	-	-
5	премиум	Dr. Web Премиум	Dr. Web Классик	AV+AS	-	-
6	премиум сервер	Dr. Web Премиум Сервер	Dr. Web Классик	AV+AS+PC	-	-
7	стандарт	Dr. Web Стандарт	Dr. Web Классик	0	-	-

Редактор

Название: классик Услуга: Dr. Web Классик Услуга активации: Dr. Web Классик Группа: AV

Действует: с [] по [] Показывать: с [] по [] Следующий тариф: классик

Показывать для групп: [] Только для новых

Ок Отмена

Модуль DrWeb

В планировщике заданий необходимо добавить задачу **Обработка задач Dr.Web (режим 1)**. Данная задача отвечает за продление подписки и изменение тарифов на сервере Dr.Web AV-Desk. Период запуска устанавливаем в начале каждых суток. В конфигурации задачи должно быть указано

```
#mid drweb
mid=
# . AV-Desk
change.tariff.days=1
# ( )
prolong.days=2
```

В планировщике заданий необходимо добавить задачу **Начисление Dr.Web**. Данная задача производит начисление за подписку. В конфигурации задачи должно быть указано

```
#mid drweb
mid=
```

Добавить модуль в договоры, в которых планируется применение данной услуги.

Управление подписками

В договоре при выборе модуля drweb имеется возможность управления агентами пользователя.

Свойства

Агенты

Описание	UUID	Тариф	Состояние	Статус	...
boris_1	3b6f50da-b5bc-4916-a004-5...	Dr. Web Стандарт	Активна	-	▲
boris_4	8f27820f-a340-43fc-9b70-23...	Dr. Web Классик	Активна	-	
boris_5	bfeafaa8-e26f-4ee5-ae58-b2...	Dr. Web Классик	Активна	-	
boris_3	d7e15459-cba7-4df6-bdbb-a...	Dr. Web Стандарт	Активна	-	
boris_2	c6024146-52de-4445-a09a-...	Dr. Web Стандарт	Активна	-	
boris_0	1b41206b-e59f-4e3a-af61-4...	Dr. Web Стандарт		-	

Редактор

Название: boris_2 UUID: c6024146-52de-4445-a09a-9035b9fafd6d

Состояние: Активна Статус: -

Приостановить подписку на 1 месяц ▼ Сменить тариф на Dr.Web Классик ▼

Активировать подписку Прекратить подписку

Ok Отмена

Приостановка и прекращение подписки, производится с 1 числа месяца, следующего за последним месяцем подписки.

Web-интерфейс модуля DrWeb

При добавлении модуля в договор, в Web-интерфейсе пользователя появляется пункт меню **Dr.Web AV-Desk - подписка**. Здесь имеется возможность получить новый агент и управлять подписками имеющихся агентов.

Dr.Web AV-Desk - подписка

Агент	Ссылка на загрузку	Тариф	Состояние подписки	Статус
boris_1 (3b6f50da-b5bc-4916-a004-5f51de0ee79e)	Скачать	Dr.Web Стандарт	Активна	-
boris_4 (8f27820f-a340-43fc-9b70-2353928ec276)	Скачать	Dr.Web Классик	Активна	-
boris_5 (bfeafaa8-e26f-4ee5-ae58-b20fcedfd44f)	Скачать	Dr.Web Классик	Активна	-
boris_3 (d7e15459-cba7-4df6-bdbb-a29270426f76)	Скачать	Dr.Web Стандарт	Активна	-
boris_2 (c6024146-52de-4445-a09a-9035b9fafd6d)	Скачать	Dr.Web Стандарт	Активна	-
boris_0 (1b41206b-e59f-4e3a-af61-4ac17ae1efb7)	Скачать	Dr.Web Стандарт		-

Dr.Web Классик

Прочтите лицензионное соглашение и нажмите '**Я принимаю условия лицензионного соглашения**' только в том случае, если Вы принимаете данное лицензионное соглашение.

Я принимаю условия лицензионного соглашения

Получить Dr.Web

Для получения нового агента пользователь должен выбрать галочку **Я принимаю условия лицензионного соглашения** и нажать кнопку **Получить Dr.Web**. Для управления агентом, необходимо выбрать агент по ссылке, откроется страница управления агентом.

Dr.Web AV-Desk - Управление агентом

Агент	Ссылка на загрузку	Тариф	Состояние подписки	Статус
(3b6f50da-b5bc-4916-a004-5f51de0ee79e)	Скачать	Dr.Web Стандарт	Активна	-

<input type="radio"/> Сменить тариф на Dr.Web Классик	<input type="radio"/> Приостановить подписку на 1 месяц
<input type="radio"/> Активировать подписку	<input type="radio"/> Прекратить подписку

Применить

Настройка тарифных планов модуля DrWeb

В один момент времени на договоре может действовать только один тарифный план, включающий в себя поддереву экземпляра модуля DrWeb. Если у вас не было тарифного плана, создайте его, создайте для него поддереву, либо расширьте от другого тарифа. Как это сделать можете прочитать [здесь](#). Там же описана логика работы тарифных деревьев и поведения стандартных узлов тарифных деревьев, общих для всех модулей. После того как вы откроете дерево в нем должен отобразиться узел со значком паука и названием экземпляра модуля DrWeb.

В тарифном запросе модуля DrWeb передаются следующие параметры:

- 1) код потребляемой услуги;
- 2) время момента потребления;
- 3) код тарифного плана.

Поиск Управление поддеревьями

Тарифный план 'drweb'

- DrWeb
 - Услуга: Dr.Web Классик
 - 59.0 за месяц
 - Услуга: Dr.Web Премиум
 - 79.0 за месяц
 - Услуга: Dr.Web Премиум Сервер
 - 88.0 за месяц

Модуль E-Mail

Назначение модуля

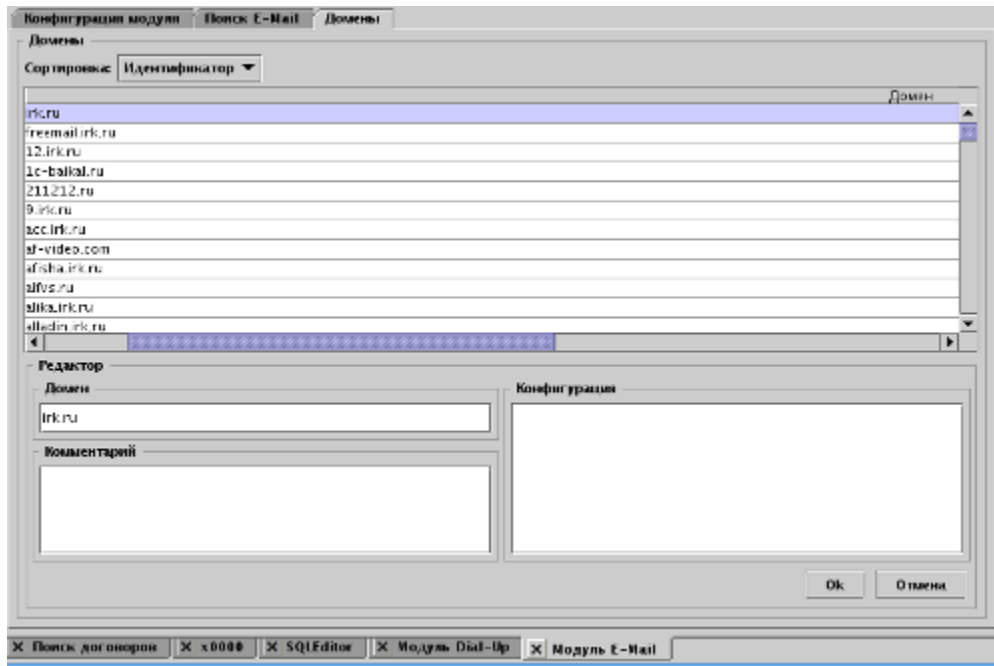
Модуль предназначен для упрощения процесса создания и управления почтовыми ящиками из интерфейса биллинга и из Web-интерфейса пользователя. Работа модуля построена на взаимодействии с LDAP или SQL базой данных, используемой для хранения почтовых аккаунтов и информации о них.

Установка и настройка модуля E-Mail

После того как вы проинсталировали модуль с помощью утилиты **bg_installer** и создали его экземпляр в редакторе модулей и услуг создайте услугу одну или несколько в данном модуле. Количество услуг может варьироваться в зависимости от того сколько типов почтовых пользователей вы хотите завести. Наличие разрешённой услуги в договоре определяют права по управлению почтовыми аккаунтами данного договора через Web интерфейс.

Домены

На вкладке **Домены** определите перечень доменов, аккаунты в которых можно создавать в модуле.



Обязательным параметром является лишь имя домена. Идентификаторы доменов указаны в столбце **ID** таблицы. В конфигурации домена могут быть указаны опции LDAP или SQL-хранилища, если для данного домена требуются иные от общей конфигурации модуля настройки хранилища (см. далее).

Настройка конфигурации

Откройте модуль E-Mail и внесите конфигурацию. Здесь определяются права услуг - разрешения для пользователей с определёнными услугами по манипуляции ящиками. Общая часть конфигурации:

Конфигурация

```
#
contract.status.active.codes=0

# Web-
web.menuItem1= E-Mail
web.menuItem2= E-Mail
web.menuItem3= E-Mail
#
#
#change.by.status=1
#
password.length.min=5
password.length.max=10
#
password.length.auto=6
#
password.chars=1234567890
#
quota.list=1:1:1024:1MB;10240:10MB
# , REGEXP
email.account.regex=^[^-\\w]{1,40}$
# , REGEXP
email.forward.regex=^[\\w]{4,20}@[\\.\\w]{5,20}$
#-----
#
#
#closed.date.disabled.ActionUpdateAccount=1
#-----
```

Строки с разрешениями для различных услуг:

```
# E-Mail
service.{ 1}=view;create:{, 0 - }:{ };delete;forward:{ };password
service.{ 2}=view;create:{, 0 - }:{ };delete;forward:{ };password
```

Строка-разрешение выглядит следующим образом:

```
service.{ }=view;create:{}:{ };delete;forward:{ }
```

Рассмотрим составляющие привилегии:

- **{код услуги}** - код услуги модуля E-Mail, можете посмотреть его в редакторе модулей и услуг; **view** - разрешение просмотра списка ящиков через Web-интерфейс; **create** - разрешение создания ящиков;
- **{сколько}** - максимальное число ящиков, которые можно создать договору;
- **{где}** - перечень кодов разрешённых доменов через запятую (в данном примере один домен с кодом 1);
- **{квота}** - квота создаваемых через Web ящиков; **delete** - привилегия удалять ящики; **forward** - разрешение вешать пересылки на ящик;
- **{форвардов на аккаунт}** - сколько пересылок можно вешать на один аккаунт;
- **password** - разрешение менять пароли на ящики;
- **view** - разрешение просматривать свои ящики;

Данные ограничения работают только на Web-интерфейс пользователя.

```
quota.list=1:1:1024:1MB;10240:10MB;0:
# , 1 1 1, ,
service.138=view;create:1:1:1;delete;password
# , 2 1 2 , 2
service.138=view;create:2:1,2:0;delete;forward:2
```

Если вы предоставите договору несколько услуг модуля E-Mail, их разрешения будут складываться.

Параметр **quota.list** задаёт список разрешённых квот и фактические значения, которые будут передаваться на LDAP сервер. В приведённом примере, например, указаны значения для Exim, т.к. он принимает значения квот в килобайтах.

Хранилище почтовых аккаунтов

Параметры хранилища почтовых аккаунтов можно указать как в конфигурации модуля, так и отдельно для каждого домена. При этом параметры сначала будут браться из конфигурации домена, а при отсутствии какого-либо - из конфигурации модуля. В случае персональной настройки доменов в конфигурации домена можно указать параметр **inherit=<id>**, означающий, что параметры хранилища данного домена такие же как и у домена идентификатором равным **<id>**.

Таким образом, аккаунты различных отдельных доменов могут храниться в разных LDAP/SQL-серверах, а также в разных типах хранилищ.

LDAP база

В конфигурации модуля/домена необходимо указать параметры подключения к LDAP-базе (пример):

```
# LDAP
sa=ru.bitel.bgbilling.modules.email.server.bean.LdapEmailServiceActivator
#,
sa.ldap.class=bitel.billing.server.email.bean.EximLDAPDriver
# LDAP-
sa.ldap.host=ldap.bill.ru
# LDAP-
sa.ldap.port=389
#
sa.ldap.root=dc=bill,dc=ru
# LDAP-
sa.ldap.user=cn=admin,dc=bill,dc=ru
sa.ldap.password=admin
# LDAP-
sa.ldap.base=bgbilling
```

Примеры настройки почтовой системы и LDAP-хранилища описаны в [Wiki](#) в разделе **Решения для модулей и плагинов => Модуль E-Mail**.

Начиная с версии 4.0 возможно добавление собственных LDAP-атрибутов. Для этого в конфигурации прописываются записи. name - имя атрибута, которое будет передано в LDAP. Атрибуты должны быть добавлены в схеме! title - отображение атрибута в биллинге.

```
ldap.attribute.1.title=1
ldap.attribute.1.name=attr1
ldap.attribute.2.title=2
ldap.attribute.2.name=attr2
```

В дальнейшем будут поддержаны LDAP-схемы для других почтовых клиентов.

SQL база

В конфигурации модуля/домена необходимо указать параметры подключения к SQL-базе, параметры домена (пример):

```
# SQL-
sa=ru.bitel.bgbilling.modules.email.server.bean.JDBCEmailServiceActivator
#JDBC- SQL-
#sa.jdbc.driver=com.mysql.jdbc.Driver
#URL
sa.jdbc.url=jdbc:mysql://127.0.0.1:3306/email
# SQL-
sa.jdbc.user=
sa.jdbc.password=
# ('LOCAL','RELAY','VIRTUAL')
sa.jdbc.domain.type=
#
sa.jdbc.domain.uid=
sa.jdbc.domain.gid=
```

Структура базы хранилища аккаунтов представлена ниже:

Код

```
CREATE TABLE domains (
  domain varchar(128) NOT NULL,
  type enum('LOCAL','RELAY','VIRTUAL') default 'LOCAL',
  uid int(10) unsigned default '1003',
  gid int(10) unsigned default '6',
  PRIMARY KEY (domain)
);

CREATE TABLE users (
  login varchar(64) NOT NULL,
  name varchar(128) NOT NULL,
  password varchar(64) NOT NULL,
  domain varchar(128) NOT NULL,
  quota tinyint(4) default '0',
  status enum('0','1') default '1',
  PRIMARY KEY (login,domain)
);

CREATE TABLE userforward (
  local_part varchar(64) NOT NULL,
  domain varchar(128) NOT NULL,
  recipients text,
  PRIMARY KEY (local_part,domain)
);

CREATE TABLE aliases (
  local_part varchar(64) NOT NULL,
  domain varchar(128) NOT NULL,
  recipients text,
  PRIMARY KEY (local_part,domain)
);
```

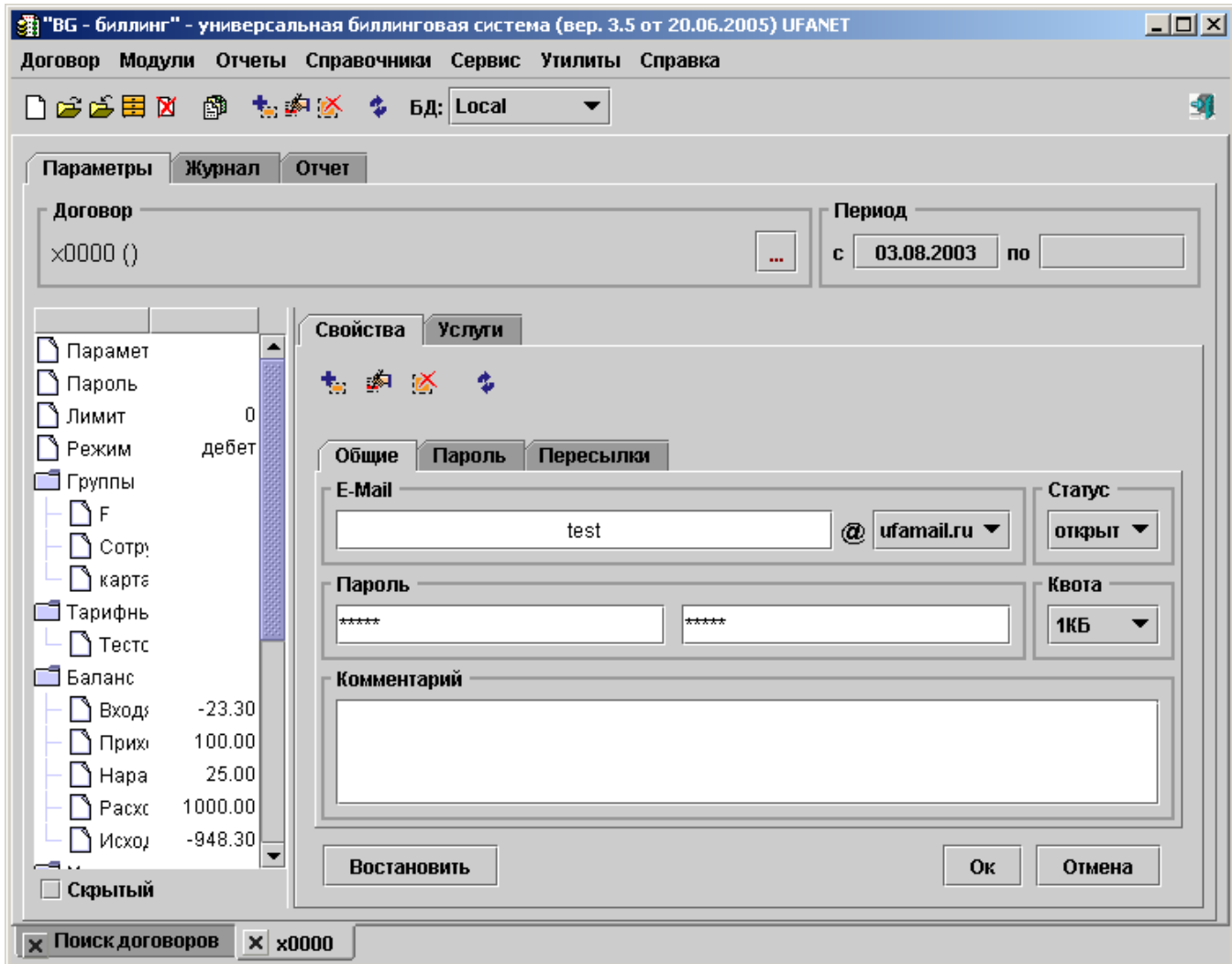
Настройка планировщика для модуля E-Mail

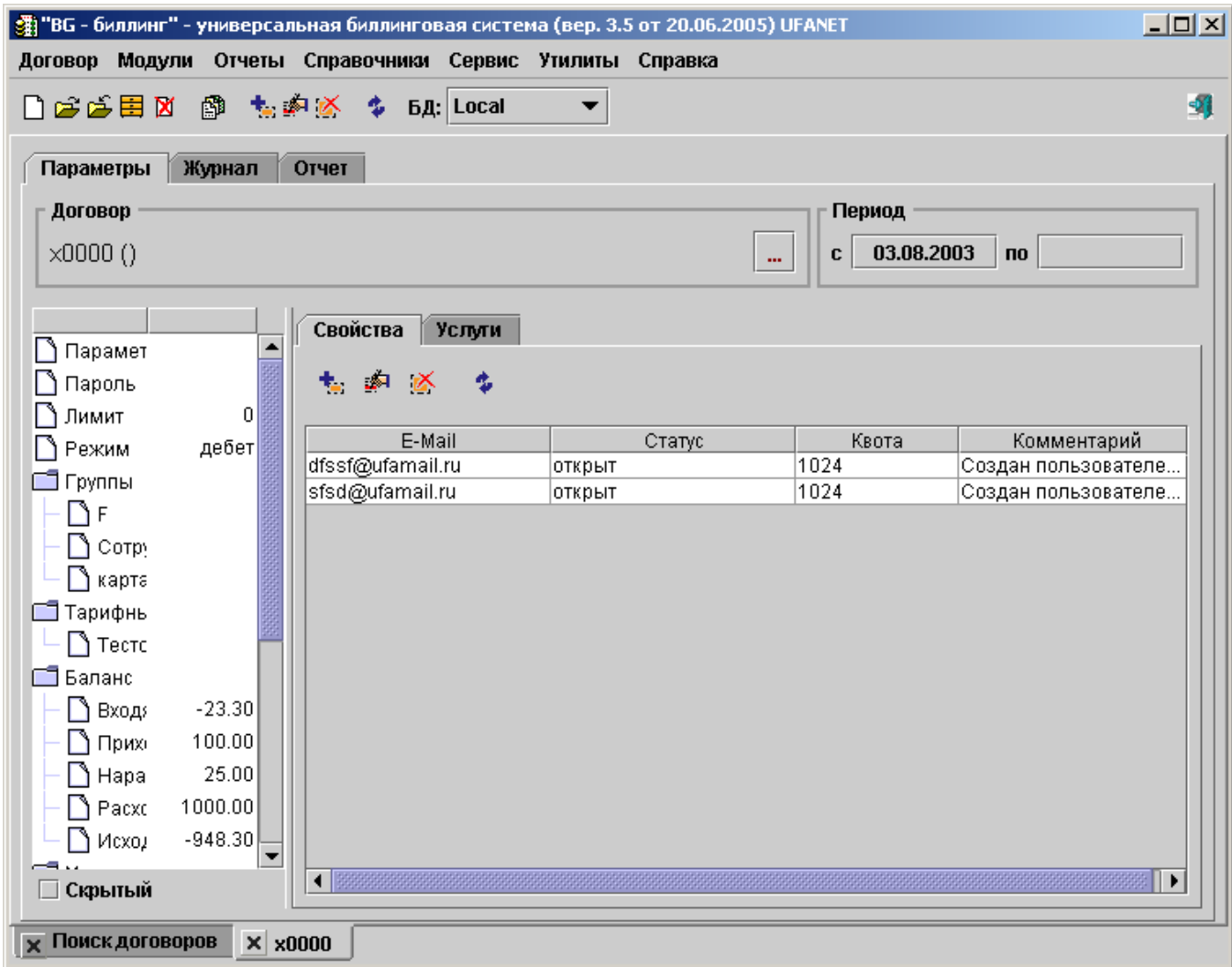
В планировщике заданий необходимо настроить запуск задачи **Синхронизация E-Mail аккаунтов** в 0 часов 3 минуты каждых суток. В параметрах запуска укажите **mid=<mid>**, где <mid> - код экземпляра модуля E-Mail. В хранилище почтовых аккаунтов хранится только актуальное на текущий момент состояние аккаунтов, данная задача необходима для обновления актуального состояния аккаунтов при переходе дат. При добавлении аккаунта в договор будущим числом, он будет реально добавлен в хранилище только при наступлении дня начала периода действия данной задачи. Аналогично, задача удалит аккаунт в хранилище при истечении периода его действия в биллинге.

Использование модуля E-Mail

Подключение экземпляра модуля к договору осуществляется путем выбора узла **Модули** дерева договора и нажатия кнопки **Новый элемент** стандартной панели инструментов. Разрешённые услуги устанавливают ограничения на управление ящиками через Web-интерфейс.

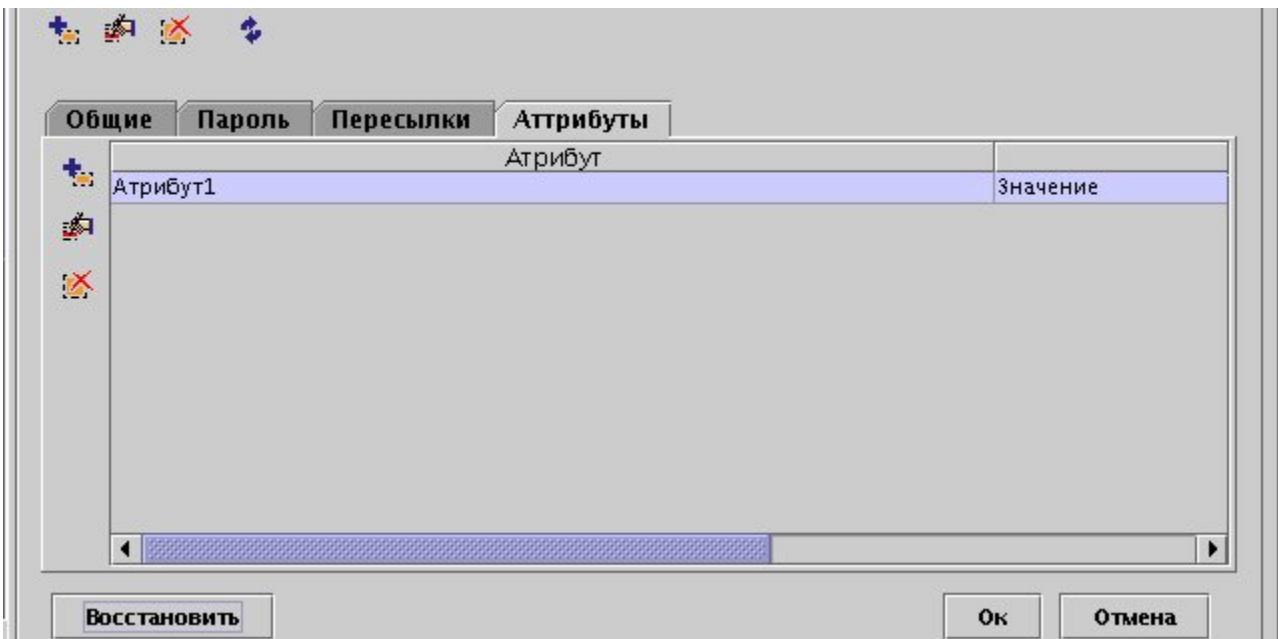
Для добавления/удаления/редактирования ящиков выберите модуль в дереве договора и используйте кнопки панели инструментов открывшегося редактора. При управлении ящиками через клиент биллинга не действуют ограничения, как при создании через Web.





Для переноса ящика на другой активный договор следует открыть целевой договор, выбрать ящик в таблице, нажатием правой кнопкой мыши вызвать всплывающее меню, выбрать пункт **Перенести на другой договор**. Целевой договор должен быть открыт на вкладке.

Вкладка **Атрибуты** - позволяет редактировать пользовательские LDAP-атрибуты аккаунта.



Модуль способен автоматически блокировать аккаунты договоров с остатком на балансе менее лимита. Блокировку осуществляет задача планировщика **Блокировка E-Mail аккаунтов**. В параметрах запуска задачи должно быть установлено:

```
mid=< E-Mail>
```

При этом аккаунт переходит в статус **заблокирован** и происходит обновление записи в хранилище.

При приходе платежа в случае, если баланс превысит лимит, **статус договора активен** и переменная конфигурации модуля **change.by.status** установлена в 1, сервер разблокирует аккаунты договора со статусом **заблокирован**.

Если установлена опция **change.by.status=1**, статусы ящиков также меняются вслед за статусами договоров. При всех неактивных статусах договора ящики переводятся в статус **заблокирован**, при подключении договора все **заблокированные** ящики становятся **активными**.

Модуль Reports


Назначение модуля

Модуль представляет из себя набор различных отчётов как по ядру системы, так и по модулям. Отказ от классической схемы генератора отчётов позволяет строить максимально оптимизированные и быстрые запросы в базу для каждого из отчётов, снижая тем самым нагрузку на сервер. При необходимости модуль позволяет разрабатывать недостающие отчёты самостоятельно силами клиента. Отчет можно экспортировать во внешние форматы - csv, xls, odt, ods, pdf, html.

Установка и настройка модуля

Достаточно проинсталлировать модуль с помощью утилиты **update.sh (.bat)**, а затем создать его экземпляр в редакторе модулей и услуг.

Для работы некоторых отчетов необходимо, что бы на сервере был установлен системый пакет **fontconfig**

 `dnf install fontconfig`

`apt-get install fontconfig`

При отсутствии надписей в клиенте, запущенном на Windows, необходимо установить на сервер биллинга пакета `ttf-mscorefonts-installer`.

Отчёты основного модуля

1. Отчет по договорам

Позволяет просмотреть существующие в системе договоры, фильтруя их по группам, виду лица (юридическое или физическое), статусу договора и другим критериям. Возможно отслеживание созданных, удалённых договоров.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																											
<p>Параметры</p> <p>Период</p> <p>с <input type="text"/> по <input type="text" value="16.02.2009"/></p> <p>Вывести</p> <p>Открытые договора</p> <p>Фильтр</p> <p>Группа Договор Списки Даты</p> <p>Режим: <input type="radio"/> И <input checked="" type="radio"/> ИЛИ</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> 001<input checked="" type="checkbox"/> 002<input checked="" type="checkbox"/> C<input checked="" type="checkbox"/> dimon<input checked="" type="checkbox"/> E<input checked="" type="checkbox"/> IMPORT(чистится! – не ставить просто так!!)<input checked="" type="checkbox"/> L<input checked="" type="checkbox"/> M<input checked="" type="checkbox"/> S<input checked="" type="checkbox"/> V<input checked="" type="checkbox"/> VIP-клиенты<input checked="" type="checkbox"/> X<input checked="" type="checkbox"/> A1<input checked="" type="checkbox"/> A2<input checked="" type="checkbox"/> A3<input checked="" type="checkbox"/> A4<input checked="" type="checkbox"/> A5<input checked="" type="checkbox"/> Группа 25<input checked="" type="checkbox"/> Дольваоылдва<input checked="" type="checkbox"/> Загруженные<input checked="" type="checkbox"/> Закрытые<input checked="" type="checkbox"/> импорт<input checked="" type="checkbox"/> карта – подарок<input checked="" type="checkbox"/> Каточки<input checked="" type="checkbox"/> Организации<input checked="" type="checkbox"/> Сотрудники<input checked="" type="checkbox"/> Фиктивные<input checked="" type="checkbox"/> Хостинг <p>Все Сброс Инверт.</p>	<p>Отчет по договорам</p> <p>Открытые за период: 12.07.2005-13.07.2005</p> <table border="1"><thead><tr><th>Договор</th><th>Дата</th><th>Договор</th></tr></thead><tbody><tr><td>R1817-05 (ООО "Дельта Трейд")</td><td>12.07.2005</td><td>NK2881-05 (Кораллина Оле</td></tr><tr><td>NK2699-05 (Копейкин Игорь)</td><td>12.07.2005</td><td>NK2815-05 (Герасимова Лар</td></tr><tr><td>RK0957-05 (ИП Новохова Н.А.)</td><td>12.07.2005</td><td>Local - 10.7.26.0/27 ДРС (Сег</td></tr><tr><td>NK2775-05 (Шадров Дмитрий)</td><td>12.07.2005</td><td>NK2728-05 (Сергина Елена</td></tr><tr><td>NK2727-05 (Яналин Тимур)</td><td>12.07.2005</td><td>NK2726-05 (Буляков Рамс</td></tr><tr><td>NK2725-05 (Уразаев Равиль)</td><td>12.07.2005</td><td>NK2656-05 (Рахимкулова</td></tr><tr><td>NK2531-05 (Зинин Александр)</td><td>12.07.2005</td><td>NK2655-05 (Низафоров Алек</td></tr><tr><td>NK2620-05 (Волков Михаил)</td><td>12.07.2005</td><td>NK2718-05 (Сидорова</td></tr></tbody></table>	Договор	Дата	Договор	R1817-05 (ООО "Дельта Трейд")	12.07.2005	NK2881-05 (Кораллина Оле	NK2699-05 (Копейкин Игорь)	12.07.2005	NK2815-05 (Герасимова Лар	RK0957-05 (ИП Новохова Н.А.)	12.07.2005	Local - 10.7.26.0/27 ДРС (Сег	NK2775-05 (Шадров Дмитрий)	12.07.2005	NK2728-05 (Сергина Елена	NK2727-05 (Яналин Тимур)	12.07.2005	NK2726-05 (Буляков Рамс	NK2725-05 (Уразаев Равиль)	12.07.2005	NK2656-05 (Рахимкулова	NK2531-05 (Зинин Александр)	12.07.2005	NK2655-05 (Низафоров Алек	NK2620-05 (Волков Михаил)	12.07.2005	NK2718-05 (Сидорова
Договор	Дата	Договор																										
R1817-05 (ООО "Дельта Трейд")	12.07.2005	NK2881-05 (Кораллина Оле																										
NK2699-05 (Копейкин Игорь)	12.07.2005	NK2815-05 (Герасимова Лар																										
RK0957-05 (ИП Новохова Н.А.)	12.07.2005	Local - 10.7.26.0/27 ДРС (Сег																										
NK2775-05 (Шадров Дмитрий)	12.07.2005	NK2728-05 (Сергина Елена																										
NK2727-05 (Яналин Тимур)	12.07.2005	NK2726-05 (Буляков Рамс																										
NK2725-05 (Уразаев Равиль)	12.07.2005	NK2656-05 (Рахимкулова																										
NK2531-05 (Зинин Александр)	12.07.2005	NK2655-05 (Низафоров Алек																										
NK2620-05 (Волков Михаил)	12.07.2005	NK2718-05 (Сидорова																										

2. Отчет по платежам

Выводит принятые системой платежи с фильтрацией по типу, группе договора. Возможно использовать для анализа суммы поступлений, с фильтром по группам и типам. Также пригоден для сверки с кассой.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																																																																																																																												
<div style="border: 1px solid gray; padding: 5px;"> <p>Параметры</p> <p>Период</p> <p>с <input type="text" value="30.06.2010"/> по <input type="text" value="30.06.2010"/></p> <p>Фильтр</p> <p>Маска договора: <input type="text"/></p> <p>Сортировать по</p> <p><input type="button" value="сумма"/> <input type="button" value="договор"/></p> <p>Типы платежа Пользователи Группы</p> <div style="border: 1px solid gray; padding: 5px;"> <p><input type="checkbox"/> Все типы</p> <ul style="list-style-type: none"> <input type="checkbox"/> CyberPlat <input type="checkbox"/> EPort <input type="checkbox"/> Банк <input type="checkbox"/> Банк (Альфа банк) <input type="checkbox"/> Банк (Башкредит банк) <input type="checkbox"/> Банк (Социнвест-агро банк) <input type="checkbox"/> Взаимозачеты <input type="checkbox"/> Виртуальный <input type="checkbox"/> владивосток - корректирующий <input type="checkbox"/> владивосток-виртуальный <input checked="" type="checkbox"/> Группа <ul style="list-style-type: none"> <input type="checkbox"/> BVCom <input type="checkbox"/> WebMoney </div> <p style="text-align: center;"><input type="button" value="Сгенерировать"/></p> </div>	<div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: right;">Отчет по плат</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #92d050;"> <th>Договор</th> <th>Пользователь</th> <th>Сумма</th> <th>Дата</th> </tr> </thead> <tbody> <tr><td>159-01</td><td>Селиванов Михаил Александрович</td><td>0</td><td>12.07.2</td></tr> <tr><td>163-01</td><td>Савельев Дмитрий Александрович</td><td>0</td><td>12.07.2</td></tr> <tr><td>242-01</td><td>Рубцов В.Н.</td><td>0</td><td>12.07.2</td></tr> <tr><td>274-02</td><td>Шалгунов Марат Михайлович</td><td>0</td><td>12.07.2</td></tr> <tr><td>287-02</td><td>Яковлев Сергей Михайлович</td><td>0</td><td>12.07.2</td></tr> <tr><td>296-02</td><td>Садриетдинов Рамель Рамисович</td><td>0</td><td>12.07.2</td></tr> <tr><td>359-02</td><td>Шашкин Хазрет Фатхович</td><td>0</td><td>12.07.2</td></tr> <tr><td>364-02</td><td>Чернышев Владимир Юрьевич</td><td>0</td><td>12.07.2</td></tr> <tr><td>407-03</td><td>Ураманов Азур Ринатович</td><td>3,0</td><td>12.07.2</td></tr> <tr><td>422-03</td><td>Кареев Э.Ф.</td><td>0</td><td>12.07.2</td></tr> <tr><td>424-03</td><td>Наумова Олеся Львовна</td><td>0</td><td>12.07.2</td></tr> <tr><td>476-03</td><td>Халиков Владимир Рахматович</td><td>0</td><td>12.07.2</td></tr> <tr><td>567-03</td><td>Климуко Татьяна Викторовна</td><td>0</td><td>12.07.2</td></tr> <tr><td>580-04</td><td>Алимов Абдураман Абдураманович</td><td>0</td><td>12.07.2</td></tr> <tr><td>584-05</td><td>Зуляев Константин Антонович</td><td>0</td><td>12.07.2</td></tr> <tr><td>0014-03</td><td>Вологов Владимир Михайлович</td><td>0</td><td>12.07.2</td></tr> <tr><td>0006-02</td><td>Чернышев Андрей Николаевич</td><td>0</td><td>12.07.2</td></tr> <tr><td>0030-03</td><td>Кликин В.И.</td><td>0</td><td>12.07.2</td></tr> <tr><td>0030-03</td><td>Терехов Г.Р.</td><td>0</td><td>12.07.2</td></tr> <tr><td>0074-06</td><td>Полынов Александр Васильевич</td><td>0</td><td>12.07.2</td></tr> <tr><td>075-03 вкл.</td><td>ГОУ СПО "ИИП"</td><td>0</td><td>12.07.2</td></tr> <tr><td>851-03</td><td>Ковалев Александр Иванович</td><td>3,0</td><td>12.07.2</td></tr> <tr><td>850-03</td><td>Беляев Александр Радикович</td><td>3,0</td><td>12.07.2</td></tr> <tr><td>865-04</td><td>Бийчиев Хан</td><td>3,0</td><td>12.07.2</td></tr> <tr><td>871-04</td><td>Рудкович И.А.</td><td>3,0</td><td>12.07.2</td></tr> <tr><td>882-05</td><td>Голованов Владимир Евгеньевич</td><td>3,0</td><td>12.07.2</td></tr> <tr><td>897-05</td><td>Курапов Иван Михайлович</td><td>0</td><td>12.07.2</td></tr> <tr><td>0009-03</td><td>Мильников Илья Викторович</td><td>0</td><td>12.07.2</td></tr> <tr><td>0013-03</td><td>Харисов Марат Хабирович</td><td>0</td><td>12.07.2</td></tr> <tr style="background-color: #92d050;"> <td>Итого:</td> <td></td> <td>18,0</td> <td></td> </tr> </tbody> </table> </div>	Договор	Пользователь	Сумма	Дата	159-01	Селиванов Михаил Александрович	0	12.07.2	163-01	Савельев Дмитрий Александрович	0	12.07.2	242-01	Рубцов В.Н.	0	12.07.2	274-02	Шалгунов Марат Михайлович	0	12.07.2	287-02	Яковлев Сергей Михайлович	0	12.07.2	296-02	Садриетдинов Рамель Рамисович	0	12.07.2	359-02	Шашкин Хазрет Фатхович	0	12.07.2	364-02	Чернышев Владимир Юрьевич	0	12.07.2	407-03	Ураманов Азур Ринатович	3,0	12.07.2	422-03	Кареев Э.Ф.	0	12.07.2	424-03	Наумова Олеся Львовна	0	12.07.2	476-03	Халиков Владимир Рахматович	0	12.07.2	567-03	Климуко Татьяна Викторовна	0	12.07.2	580-04	Алимов Абдураман Абдураманович	0	12.07.2	584-05	Зуляев Константин Антонович	0	12.07.2	0014-03	Вологов Владимир Михайлович	0	12.07.2	0006-02	Чернышев Андрей Николаевич	0	12.07.2	0030-03	Кликин В.И.	0	12.07.2	0030-03	Терехов Г.Р.	0	12.07.2	0074-06	Полынов Александр Васильевич	0	12.07.2	075-03 вкл.	ГОУ СПО "ИИП"	0	12.07.2	851-03	Ковалев Александр Иванович	3,0	12.07.2	850-03	Беляев Александр Радикович	3,0	12.07.2	865-04	Бийчиев Хан	3,0	12.07.2	871-04	Рудкович И.А.	3,0	12.07.2	882-05	Голованов Владимир Евгеньевич	3,0	12.07.2	897-05	Курапов Иван Михайлович	0	12.07.2	0009-03	Мильников Илья Викторович	0	12.07.2	0013-03	Харисов Марат Хабирович	0	12.07.2	Итого:		18,0	
Договор	Пользователь	Сумма	Дата																																																																																																																										
159-01	Селиванов Михаил Александрович	0	12.07.2																																																																																																																										
163-01	Савельев Дмитрий Александрович	0	12.07.2																																																																																																																										
242-01	Рубцов В.Н.	0	12.07.2																																																																																																																										
274-02	Шалгунов Марат Михайлович	0	12.07.2																																																																																																																										
287-02	Яковлев Сергей Михайлович	0	12.07.2																																																																																																																										
296-02	Садриетдинов Рамель Рамисович	0	12.07.2																																																																																																																										
359-02	Шашкин Хазрет Фатхович	0	12.07.2																																																																																																																										
364-02	Чернышев Владимир Юрьевич	0	12.07.2																																																																																																																										
407-03	Ураманов Азур Ринатович	3,0	12.07.2																																																																																																																										
422-03	Кареев Э.Ф.	0	12.07.2																																																																																																																										
424-03	Наумова Олеся Львовна	0	12.07.2																																																																																																																										
476-03	Халиков Владимир Рахматович	0	12.07.2																																																																																																																										
567-03	Климуко Татьяна Викторовна	0	12.07.2																																																																																																																										
580-04	Алимов Абдураман Абдураманович	0	12.07.2																																																																																																																										
584-05	Зуляев Константин Антонович	0	12.07.2																																																																																																																										
0014-03	Вологов Владимир Михайлович	0	12.07.2																																																																																																																										
0006-02	Чернышев Андрей Николаевич	0	12.07.2																																																																																																																										
0030-03	Кликин В.И.	0	12.07.2																																																																																																																										
0030-03	Терехов Г.Р.	0	12.07.2																																																																																																																										
0074-06	Полынов Александр Васильевич	0	12.07.2																																																																																																																										
075-03 вкл.	ГОУ СПО "ИИП"	0	12.07.2																																																																																																																										
851-03	Ковалев Александр Иванович	3,0	12.07.2																																																																																																																										
850-03	Беляев Александр Радикович	3,0	12.07.2																																																																																																																										
865-04	Бийчиев Хан	3,0	12.07.2																																																																																																																										
871-04	Рудкович И.А.	3,0	12.07.2																																																																																																																										
882-05	Голованов Владимир Евгеньевич	3,0	12.07.2																																																																																																																										
897-05	Курапов Иван Михайлович	0	12.07.2																																																																																																																										
0009-03	Мильников Илья Викторович	0	12.07.2																																																																																																																										
0013-03	Харисов Марат Хабирович	0	12.07.2																																																																																																																										
Итого:		18,0																																																																																																																											

3. Отчет по расходам

В целом идентичен отчёту по платежам, но выводит не платежи, а расходы, занесённые на договоры.

4. Отчет по наработке

Выводит наработку договоров выбранных групп за определённый месяц.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид

Отчет по наработке

Параметры

← Июль 2005 →

Группы

- A
- AA
- AD
- AL
- AO
- B
- IP2
- Local
- N
- Network
- NK
- NT
- R

Отчет по наработке
год: 2005 месяц: июль

Услуга	Сумма
Ежемесячная аб-я плата (тип 1)	10,00
Ежемесячная аб-я плата (тип 11)	1,67
(IPN) Внешний трафик (входящий)	12,94
(IPN) Внешний трафик (исходящий)	
(IPN) Локальный трафик (входящий)	1,08
(IPN) Локальный трафик (исходящий)	
(IPN) Почтовый сервер POP3.UFANET.RU	54
(IPN) Прокси IP1(входящий)	1,09

5. Детализированный отчет по наработке

Выводит наработку по договорам из выбранных групп.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																																
<p>Детализированный отчет по наработке</p> <p>Параметры</p> <p>← Январь 2006 →</p> <p>Группы</p> <ul style="list-style-type: none"> <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> E <input type="checkbox"/> L <input type="checkbox"/> M <input type="checkbox"/> S <input type="checkbox"/> V <input type="checkbox"/> VIP-клиенты <input type="checkbox"/> x <p style="text-align: center;"> <input type="button" value="Все"/> <input type="button" value="Сброс"/> <input type="button" value="Инверт."/> </p> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Сгенерировать"/></p>	<div style="text-align: center;"> <p>Отчет по наработке год: 2006 месяц: январь</p> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #00FF00;"> <th colspan="2">Договор: x0000</th> </tr> <tr style="background-color: #00FF00;"> <th>Услуга</th> <th>Наработка</th> </tr> </thead> <tbody> <tr><td>Сл-ф(время)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,00</td></tr> <tr><td>Абонплата</td><td>30,00</td></tr> <tr style="background-color: #00FF00;"> <td>Итого:</td> <td>30,00</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #00FF00;"> <th colspan="2">Договор: x0002</th> </tr> <tr style="background-color: #00FF00;"> <th>Услуга</th> <th>Наработка</th> </tr> </thead> <tbody> <tr><td>Сл-ф(время)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,00</td></tr> <tr><td>Сл-ф(исходящий)</td><td>0,26</td></tr> <tr style="background-color: #00FF00;"> <td>Итого:</td> <td>0,26</td> </tr> </tbody> </table> <p style="background-color: #00FF00; padding: 2px;">Итого по договорам: 30,26</p>	Договор: x0000		Услуга	Наработка	Сл-ф(время)	0,00	Сл-ф(исходящий)	0,00	Сл-ф(исходящий)	0,00	Сл-ф(исходящий)	0,00	Сл-ф(исходящий)	0,00	Абонплата	30,00	Итого:	30,00	Договор: x0002		Услуга	Наработка	Сл-ф(время)	0,00	Сл-ф(исходящий)	0,00	Сл-ф(исходящий)	0,00	Сл-ф(исходящий)	0,26	Итого:	0,26
Договор: x0000																																	
Услуга	Наработка																																
Сл-ф(время)	0,00																																
Сл-ф(исходящий)	0,00																																
Сл-ф(исходящий)	0,00																																
Сл-ф(исходящий)	0,00																																
Сл-ф(исходящий)	0,00																																
Абонплата	30,00																																
Итого:	30,00																																
Договор: x0002																																	
Услуга	Наработка																																
Сл-ф(время)	0,00																																
Сл-ф(исходящий)	0,00																																
Сл-ф(исходящий)	0,00																																
Сл-ф(исходящий)	0,26																																
Итого:	0,26																																

5. Отчет по тарифам

Выводит договоры, сгруппированные по тарифам . Доступен фильтр по договорам и дате тарифного плана.

Таблица. Фильтр и внешний вид

Фильтр

Параметры

Дата

18.02.2009

Тарифы

- Домен (физ. лица)
- Домен (юр. лица)
- Услуги (физ. лица)
- Хостинг

Все

Сброс

Инверт.

Внешний вид

Отчет по тариф

За дату: 19.02.2009

Тариф: Домен (юр. лица)

Договор	Договор
DU036-07 (ИП Муллаянов Р.Р. - { autokorufa.ru })	DU019-07
DU081-08 (ООО "Сервисно-производственный")	DU072-08
DU037-08 (ООО "РТБ-Софт" - { rtb-soft.ru })	DU096-08
DU011-08 (МУП Парк культуры и отдыха)	DU047-08
DU082-08 (ДП "Башкирская машино-")	DU071-08
DU097-08 (ООО "Золотое кольцо" - { ringtour.ru })	DU129-08
DU109-07 (ООО "Авто-Моторс" - { avto-motors.ru })	DU048-07
DU073-08 (ООО "Спецнефтехимавтоматика" - { })	

Итого договоров:

Тариф: Хостинг

Договор	Договор
NF121-07 (Вахитов Ильдар Фларитович - { })	NF703-07
HU006-08 (ООО "СУ-33" - { su33-ufa.ru })	NF018-08
NF019-08 (Крашенинников Денис Николаевич - { })	NF059-08
HU033-08 (ООО "Инвестиционно-торговая")	NF060-08
HU048-08 (ООО "Промэнергострой" - { promes-})	HU001-08
NF059-08 (Тименко Владимир Сергеевич - { st-})	HU005-08
HU068-08 (ООО "УралРегионСервис" - { urs-kkm.ru })	HU068-08
NF080-08 (Хузяттов Артур Арсланович - { kvadart.ru })	HU091-08
HU091-08 (ООО "Автошкола XXI век" - { taksi-})	NF102-08
NF100-07 (Хватов Максим Владимирович - { })	NF096-07
NF133-07 (Плеханов Павел Игоревич - { })	HU005-07
HU012-08 (ООО "Издательство "Диалог" - { })	HU026-08
HU021-08 (ООО "РБК-Инвест" - { rbk-invest.ru })	HU022-08
HU037-08 (Управление МИР - { centermir.ru }) (не)	NF040-08
HU013-04 (ООО "Башкирская инвестиционно-")	NF061-04
HU056-05 (ООО "Геострой" - { geostroy-ufa.ru })	HU071-05
N107-08 ()	NF112-08
HU099-08 (ЗАО "Бизнес-школа Альтернатива" - { })	HU016-08
HU001-08 (ООО "фирма "Спецоснастка МК" - { })	NF011-08
HU015-08 (МУП "Парк культуры и отдыха")	HU055-08
HU028-08 (АНО "Издательский дом "Республика"")	HU005-08
HU044-08 (ООО "Инвестиционная компания")	HU085-08
NF054-08 (Крупский Андрей Викторович - { })	HU020-08
HU059-08 (МУ "Централизованная система")	NF039-08
HU083-07 (ИП "Кинзягулов Руслан Фанильевич" - { })	HU132-07
NF707-09 ()	HU118-09
NF045-07 (Хабиров Р.Г. - { ufaexpro.ru })	NF003-07
HU017-07 ("КП РБ "Редакция газеты "Йвншишма"")	NF018-07
HU058-07 ("Региональная детско-юношеская")	NF036-07
HU061-06 (ФГУ "Центр лабораторного анализа и")	HU050-06
HU068-06 (ИП "Интересов В.В." - { teebokufa.ru })	HU058-06

Документ подготовлен системой "СБ-888.ru" © ООО "Белит" http://www.b888.ru

6.Наработка по тарифам

Выводит наработку по тарифам за выбранный месяц.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид

Параметры

Месяц

◀ Февраль ▼ 2009 ▼ ▶

Нарботка по та
февраль 2009

ID	Тарифный план
60	111111
71	2222
72	3333
59	4444
56	abon
135	den4
131	dialup
74	hlp
58	New tariff plan
47	New tariff plan
134	New tariff plan
45	New tariff plan 1
75	nnnp
70	no lim it rтс с апус для асвт-нн
43	Sovintel
65	T1_Телефония местная
63	T1_Телефония Совинтел
133	test
49	Test DSI Bug
57	TEST1
61	TestForumBug
40	VOIP
27	voip_chita7
32	voip_ufanet
67	A10

Клик мышки по строке выводит детализацию по конкретному тарифу:

Отчет по тарифу: Т1_Телефония+местная
февраль 2009

ID	Договор	Комментарий	Наработка за период
32543	T1	Гайдеко Илья Валерьевич	
352720	Переоформлен 1800000011	Гичев Денис	
352639	1800000004	Козырь Денис	
352640	1800000005	Афони娜 Нина Игоревна	
352641	1800000006	Шушков Сергей	
352642	1800000007	Астро	
352643	1800000008	Николайчук Денис	
352644	1800000009	Кострулев Сергей Алек сандрович	
352645	1800000010	Степанов Никита Петрович	
352646	1800000011	Гичев Денис	
352647	1800000012	Шнейдер Вячеслав	
352648	1800000013	Шабуневич Василий	
			12

7.Отчет по должникам

Отчет выводит должников за определённый период. Фильтр и внешнее представление имеют вид:

Таблица. Отчет по должникам

Фильтр	Внешний вид																																																																																																																			
	Отчет по должникам Входящее сальдо на: 01.02.; Итоговое сальдо на: 19.02.2																																																																																																																			
	<table border="1"> <thead> <tr> <th>№№</th> <th>Номер телефона абонента</th> <th>Номер договора</th> <th>Ф.И.О. абонента</th> <th>Адрес уст</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Тандем Телеком</td> <td>8001-08</td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Интегриро данная транспортн ая сеть (ИТС)</td> <td>8004-06</td> <td>Власюк Леонид Яковлевич</td> <td></td> </tr> <tr> <td>3</td> <td>Волков- Онлайн</td> <td>8025-06</td> <td>Слонов Е.Г.</td> <td></td> </tr> <tr> <td>4</td> <td>Комплексн ые ТелеСисте мы</td> <td>8116-08</td> <td>Дежурный Евгений Геннадьевич</td> <td></td> </tr> <tr> <td>5</td> <td>Ариадна- Ленин</td> <td>8132-08</td> <td>Паршутин Александр Степанович</td> <td></td> </tr> <tr> <td>6</td> <td>ТОО "Радиобайл анис"</td> <td>8133-08</td> <td>Кока Марина Анатольевна</td> <td></td> </tr> <tr> <td>7</td> <td>Сибком</td> <td>8146-09</td> <td></td> <td></td> </tr> <tr> <td>8</td> <td>Алик</td> <td>8150-09</td> <td>Кушнарченко Сергей Николаевич</td> <td></td> </tr> <tr> <td>9</td> <td></td> <td>85001-06</td> <td></td> <td></td> </tr> <tr> <td>10</td> <td></td> <td>85002-08</td> <td></td> <td></td> </tr> <tr> <td>11</td> <td></td> <td>D101-07</td> <td></td> <td></td> </tr> <tr> <td>12</td> <td></td> <td>DF001-05</td> <td></td> <td></td> </tr> <tr> <td>13</td> <td></td> <td>DF001-07</td> <td></td> <td></td> </tr> <tr> <td>14</td> <td></td> <td>DF003-08</td> <td></td> <td></td> </tr> <tr> <td>15</td> <td></td> <td>DF005-08</td> <td></td> <td></td> </tr> <tr> <td>16</td> <td></td> <td>DF006-04</td> <td></td> <td></td> </tr> <tr> <td>17</td> <td></td> <td>DF009-06</td> <td></td> <td></td> </tr> <tr> <td>18</td> <td></td> <td>DF009-08</td> <td></td> <td></td> </tr> <tr> <td>19</td> <td></td> <td>DF010-05</td> <td></td> <td></td> </tr> <tr> <td>20</td> <td></td> <td>DF010-08</td> <td></td> <td></td> </tr> <tr> <td>21</td> <td></td> <td>DF011-05</td> <td></td> <td></td> </tr> <tr> <td>22</td> <td></td> <td>DF012-08</td> <td></td> <td></td> </tr> </tbody> </table>	№№	Номер телефона абонента	Номер договора	Ф.И.О. абонента	Адрес уст	1	Тандем Телеком	8001-08			2	Интегриро данная транспортн ая сеть (ИТС)	8004-06	Власюк Леонид Яковлевич		3	Волков- Онлайн	8025-06	Слонов Е.Г.		4	Комплексн ые ТелеСисте мы	8116-08	Дежурный Евгений Геннадьевич		5	Ариадна- Ленин	8132-08	Паршутин Александр Степанович		6	ТОО "Радиобайл анис"	8133-08	Кока Марина Анатольевна		7	Сибком	8146-09			8	Алик	8150-09	Кушнарченко Сергей Николаевич		9		85001-06			10		85002-08			11		D101-07			12		DF001-05			13		DF001-07			14		DF003-08			15		DF005-08			16		DF006-04			17		DF009-06			18		DF009-08			19		DF010-05			20		DF010-08			21		DF011-05			22		DF012-08		
№№	Номер телефона абонента	Номер договора	Ф.И.О. абонента	Адрес уст																																																																																																																
1	Тандем Телеком	8001-08																																																																																																																		
2	Интегриро данная транспортн ая сеть (ИТС)	8004-06	Власюк Леонид Яковлевич																																																																																																																	
3	Волков- Онлайн	8025-06	Слонов Е.Г.																																																																																																																	
4	Комплексн ые ТелеСисте мы	8116-08	Дежурный Евгений Геннадьевич																																																																																																																	
5	Ариадна- Ленин	8132-08	Паршутин Александр Степанович																																																																																																																	
6	ТОО "Радиобайл анис"	8133-08	Кока Марина Анатольевна																																																																																																																	
7	Сибком	8146-09																																																																																																																		
8	Алик	8150-09	Кушнарченко Сергей Николаевич																																																																																																																	
9		85001-06																																																																																																																		
10		85002-08																																																																																																																		
11		D101-07																																																																																																																		
12		DF001-05																																																																																																																		
13		DF001-07																																																																																																																		
14		DF003-08																																																																																																																		
15		DF005-08																																																																																																																		
16		DF006-04																																																																																																																		
17		DF009-06																																																																																																																		
18		DF009-08																																																																																																																		
19		DF010-05																																																																																																																		
20		DF010-08																																																																																																																		
21		DF011-05																																																																																																																		
22		DF012-08																																																																																																																		

Параметры

Режим

Текущий баланс ▼

Период

с по

Вывести

Должники ▼

Группы договоров

- Банки
- Биллинг
- Договор отправлен в эл. виде
- Договора нет
- Должники
- Должны мы
- Домен
- Техподдержка
- Услуги
- Хостинг
- Хостинг (FTP)
- Хостинг (VDS)
- Хостинг (ТП1)
- Хостинг (ТП2)

Все Сброс Инверт.

Этот отчет не является встроенным в BGBilling и может быть изменён . Файлы отчёта - **kernel_contract_debtor.rep.xml** и **kernel_contract_debtor.jrxml**. О том, как создавать свои собственные отчёты читайте [здесь](#). Тут возможны режимы: Сальдо по периоду и Текущий баланс.

В режиме **Текущего баланса** анализируется входящий остаток на начало месяца и исходящий остаток на конец месяца. При это должником считается тот, у кого исходящий остаток отрицательный. Оплатившим считается тот, у кого входящий остаток на начало месяца отрицательный, а исходящий остаток на конец месяца не отрицательный . В столбце **Входящее сальдо** показывается входящий остаток на начало месяца. В столбце **Итоговое сальдо** указывается исходящий остаток на конец месяца. В столбце **Приход** указывается сумма всех платежей (кроме платежей типа ГОРОД) с начала месяца по вторую дату (если на указана, то по текущее число). В столбце **Приход (Город)** указывается сумма всех платежей типа ГОРОД с начала месяца по вторую дату (если на указана, то по текущее число). Если договор является супердоговором, то при учёте платежей ему добавляются все платежи субдоговоров.

В режиме **Сальдо по периоду** анализируется входящий остаток на начало месяца и сальдо на конец месяца. Сальдо вычисляется как сумма входящего остатка на начало месяца и платежей в указанный период. Введём следующие обозначения:

incoming - входящий остаток на начало месяца;

payment_1_date1 - сумма всех платежей с начала до первой даты указанного периода (не включая её);

payment_date1_date2 - платежи в указанный период (включая границы);

saldo = incoming + payment_date1_date2 - сальдо.

Должником считается тот, у кого сальдо на конец периода отрицательное (saldo < 0). Оплатившим считается тот, у кого incoming + payment_1_date1 < 0, а saldo >= 0 (т.е. сумма входящего остатка на начало месяца и платежей до первой даты указанного периода (не включая её) отрицательное, а сальдо не отрицательное). В столбце **Входящее сальдо** показывается incoming + payment_1_date1. В столбце **Итоговое сальдо** указывается сальдо. В столбце **Приход** указывается payment_date1_date2 - (платежи типа ГОРОД в указанный период). В столбце **приход (Город)** указывается сумма всех платежей типа ГОРОД в указанный период. Если договор является супердоговором, то при учёте платежей ему добавляются все платежи субдоговоров. В этом режиме нужно обязательно указать обе даты, иначе результат будет пустым.

В обоих режимах, если указано выводить всех, то выводятся все договора независимо от того оплатили они или являются должниками.

Для этого отчёта требуется настройка в конфигурации модуля (**Модули->Отчёты->Конфигурация модуля**):

```
#
report.contract_debtor.fio.pid=1
#
report.contract_debtor.phone.pid=2
#
report.contract_debtor.address.pid=3
# . 0, .
report.contract_debtor.gorod.pt=4
# ( ). , ().
#report.contract_debtor.address.cut=.\s*(.*)
```

Эта настройка привязывает столбцы отчёта к реальным параметрам договоров и задаёт код типа платежа Город.

Отчёты модуля DialUP

1. Отчёт по услугам

Выводит наработку договоров по выбранным услугам в часах, либо мегабайтах.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид						
<p>Параметры</p> <p>Вывести в: <input checked="" type="radio"/> МВ <input type="radio"/> часы</p> <p>Период</p> <p>◀ Январь 2006 ▶</p> <p>дни: с <input type="text"/> до <input type="text"/></p> <p>Часы</p> <p>часы с: <input type="text"/> по: <input type="text"/></p> <p>Услуги Группы договоров</p> <p>Услуги</p> <ul style="list-style-type: none"><input type="checkbox"/> VPN - время<input type="checkbox"/> VPN - время (local)<input type="checkbox"/> VPN - время (voip)<input type="checkbox"/> VPN - входящий<input type="checkbox"/> VPN - входящий (local)<input type="checkbox"/> VPN - входящий (radio)<input type="checkbox"/> VPN - входящий (voip)<input type="checkbox"/> VPN - исходящий<input type="checkbox"/> VPN - исходящий (local)<input type="checkbox"/> VPN - исходящий (radio)<input type="checkbox"/> VPN - исходящий (voip) <p>Все Сброс Инверт.</p>	<p>Отчет по трафику</p> <table border="1"><thead><tr><th>Услуга</th><th>Объем (МБ / часы)</th></tr></thead><tbody><tr><td>VPN - входящий</td><td>192 928</td></tr><tr><td>VPN - входящий (local)</td><td>7 649</td></tr></tbody></table>	Услуга	Объем (МБ / часы)	VPN - входящий	192 928	VPN - входящий (local)	7 649
Услуга	Объем (МБ / часы)						
VPN - входящий	192 928						
VPN - входящий (local)	7 649						

2. Отчёт Ситно 1

Показывает логины, тарифы и наработку в единицах услуги, либо денег в разрезе по группам. В фильтре должны быть обязательно установлены группы договоров и услуги.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид

Параметры

Режим

Услуга деленая на: Наработка

Месяц

◀ ▼ ▼ ▶

дни: с до

Услуги

Группы договоров

Группы

- A
- AA
- AD
- AL
- AO
- AS
- B
- IP2
- Local
- N
- Network
- NK
- NT

Отчет по наработке

Услуги: VPN - входящий / VPN - входящий (local) / VPN - вход

Период: 01.01.2006 - 31.01.2006

A

Договор	Тариф	Логин
A6297-03 (Грегориц Рита Аделия Браво)	ДС - частники (2.40 НДС	283 ()
A6298-02 (Фунг Ксак Хоан)	ДС - основной (1.80)	282 ()
A6299-02 (Манул Жорж е Жоау)	ДС - частники (2.40 НДС	284 ()
A6300-03 (Динь Хай Ха)	ДС - основной (1.80)	285 ()
A6567-03 (Клаудио Ткалю Внера Днас)	ДС - частники (2.40 НДС	281 ()
A6580-04 (Айемоба Абдулрасак Абно дуа)	ДС - частники (2.40 НДС	858 ()

Итого:

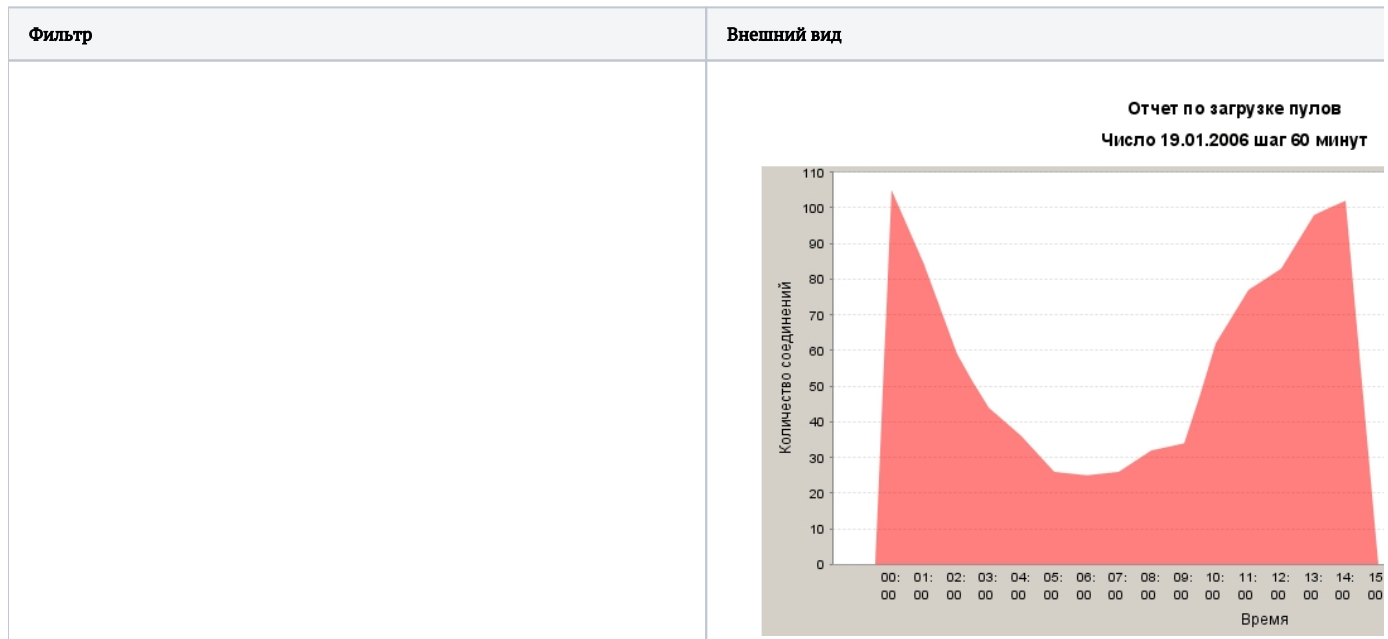
VK

Договор	Тариф	Логин
VK0004-04 (Фирма Теклайн (был V9028-01)Хстал	VK ДС - 1.25 руб	144 ()

3. Отчёт Dolphin 1

Позволяет посмотреть загрузку модемного, либо VPN-пула по количеству одновременных соединений с заданным шагом.

Таблица. Фильтр и внешний вид



Отчет

Отчет Dolphin 1

Параметры

Дата

19.01.2006

Шаг

60

минут

NAS ы

- drs1.ufanet.ru (Ф.Мир, кварталы 92, 177)
- drs2.ufanet.ru (Айская 63)
- drs3.ufanet.ru (К.Маркса 55)
- drs4.ufanet.ru (Ф.Мир)
- drs5.ufanet.ru (Цюрупа 130)
- drs6.ufanet.ru (8-ми этажка)
- drs7.ufanet.ru ()
- drs8.ufanet.ru ()
- fmir.ufanet.ru (тестовая машина)
- lgw.ufanet.ru ()

Все

Сброс

Инверт.

4. Отчёт по сессиям

Позволяет просмотреть dialup-сессии клиентов. Для фильтрации по IP-адресу его нужно вводить полностью

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																																										
	<p style="text-align: center;">Отчет по сессиям год: 2006 месяц: февраль</p> <p>Договор: TEST_ULUP</p> <table border="1"><thead><tr><th>Логин</th><th>Время начала</th><th>Время окончания</th><th>Длительность</th><th>Стоимость</th><th>С номера</th><th>На номер</th></tr></thead><tbody><tr><td>5</td><td>03.02.2006 19:59:48</td><td>03.02.2006 19:59:58</td><td>00:00:10 [10]</td><td>0.00000</td><td>5-48228</td><td></td></tr><tr><td>5</td><td>04.02.2006 22:27:30</td><td>04.02.2006 22:27:30</td><td>00:00:10 [10]</td><td>0.10881</td><td>5-48228</td><td></td></tr><tr><td>5</td><td>09.02.2006 20:28:44</td><td>09.02.2006 20:28:54</td><td>00:00:10 [10]</td><td>0.10881</td><td>5-48228</td><td></td></tr><tr><td>5</td><td>10.02.2006 20:24:59</td><td>10.02.2006 20:25:09</td><td>00:00:10 [10]</td><td>0.00000</td><td>5-48228</td><td></td></tr><tr><td>5</td><td>10.02.2006 20:30:03</td><td>10.02.2006 20:30:13</td><td>00:00:10 [10]</td><td>0.00000</td><td>5-48228</td><td></td></tr></tbody></table>	Логин	Время начала	Время окончания	Длительность	Стоимость	С номера	На номер	5	03.02.2006 19:59:48	03.02.2006 19:59:58	00:00:10 [10]	0.00000	5-48228		5	04.02.2006 22:27:30	04.02.2006 22:27:30	00:00:10 [10]	0.10881	5-48228		5	09.02.2006 20:28:44	09.02.2006 20:28:54	00:00:10 [10]	0.10881	5-48228		5	10.02.2006 20:24:59	10.02.2006 20:25:09	00:00:10 [10]	0.00000	5-48228		5	10.02.2006 20:30:03	10.02.2006 20:30:13	00:00:10 [10]	0.00000	5-48228	
Логин	Время начала	Время окончания	Длительность	Стоимость	С номера	На номер																																					
5	03.02.2006 19:59:48	03.02.2006 19:59:58	00:00:10 [10]	0.00000	5-48228																																						
5	04.02.2006 22:27:30	04.02.2006 22:27:30	00:00:10 [10]	0.10881	5-48228																																						
5	09.02.2006 20:28:44	09.02.2006 20:28:54	00:00:10 [10]	0.10881	5-48228																																						
5	10.02.2006 20:24:59	10.02.2006 20:25:09	00:00:10 [10]	0.00000	5-48228																																						
5	10.02.2006 20:30:03	10.02.2006 20:30:13	00:00:10 [10]	0.00000	5-48228																																						

Отчет

Отчет по сессиям

Параметры

IP-адрес:

телефон:

Период

◀ Февраль 2006 ▶

дни: с до

Часы

часы с: по:

5. Отчёт Shturman 1

Позволяет посмотреть наработку по dialup + произвольным услугам

```
#
# dialup-:
#reports.shturman_1.sids=23:1048576:;24:1024:
reports.shturman_1.sids=
#
#      ,   " , ( , )
#1,2,3
reports.shturman_1.addsids=
```

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																																																																								
<p>Отчеты: Конфигурация модуля</p> <p>Модель: Dial-Up</p> <p>Отчет: Shturman 1</p> <p>Параметры: Март 2009</p> <p>Группы:</p> <ul style="list-style-type: none"> <input type="checkbox"/> #01 <input type="checkbox"/> #02 <input type="checkbox"/> С <input type="checkbox"/> #0000 <input type="checkbox"/> E <input type="checkbox"/> #0000Высказал - не ставить просто так! <input type="checkbox"/> L <input type="checkbox"/> #0 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> V <input type="checkbox"/> VIP-клиенты <input type="checkbox"/> X <input type="checkbox"/> A1 <input type="checkbox"/> A2 <input type="checkbox"/> A3 <input type="checkbox"/> A4 <input type="checkbox"/> A5 <input type="checkbox"/> группа 25 <input type="checkbox"/> Дольщики/дого <input type="checkbox"/> Загруженные <input type="checkbox"/> Зарытые <input type="checkbox"/> импорт <input type="checkbox"/> карты - подарок <input type="checkbox"/> Календарь <input type="checkbox"/> Организация <input type="checkbox"/> Сотрудник <input type="checkbox"/> Физ. лица <input type="checkbox"/> Хостинг <input type="checkbox"/> Частные лица <p>Все Обнос Импорт</p> <p>Сгенерировать</p> <p>X Модель Отчеты X SQLEdit X Поиск договоров X x88</p>	<p>Отчет по наработке</p> <p>год 2009 месяц март</p> <p>Договор: x00 (zsdffsd)</p> <table border="1"> <thead> <tr> <th>Услуга</th> <th>Стоимость (руб)</th> <th>Объем</th> <th>Единица</th> </tr> </thead> <tbody> <tr><td>Dial-Up(аренд)</td><td>0,00</td><td>78,00</td><td></td></tr> <tr><td>Dial-Up(исходящий)</td><td>39,08</td><td>29,54</td><td>ME</td></tr> <tr><td>Dial-Up(исходящий)</td><td>0,00</td><td>10,94</td><td>ME</td></tr> <tr><td>Dial-Up(исходящий)</td><td>39,08</td><td></td><td></td></tr> <tr><td>Итого:</td><td>191518,50</td><td></td><td></td></tr> <tr><td>Итого:</td><td>191596,66</td><td>5</td><td></td></tr> </tbody> </table> <p>Договор: Pech1118 ()</p> <table border="1"> <thead> <tr> <th>Услуга</th> <th>Стоимость (руб)</th> <th>Объем</th> <th>Единица</th> </tr> </thead> <tbody> <tr><td>Dial-Up(аренд)</td><td>0,00</td><td>21,00</td><td></td></tr> <tr><td>Dial-Up(исходящий)</td><td>0,00</td><td>0,57</td><td>ME</td></tr> <tr><td>Dial-Up(исходящий)</td><td>0,00</td><td>1,95</td><td>ME</td></tr> <tr><td>Итого:</td><td>0,00</td><td>3</td><td></td></tr> </tbody> </table> <p>Договор: Pech1119 ()</p> <table border="1"> <thead> <tr> <th>Услуга</th> <th>Стоимость (руб)</th> <th>Объем</th> <th>Единица</th> </tr> </thead> <tbody> <tr><td>Dial-Up(аренд)</td><td>0,00</td><td>584,00</td><td></td></tr> <tr><td>Dial-Up(исходящий)</td><td>0,00</td><td>1,62</td><td>ME</td></tr> <tr><td>Dial-Up(исходящий)</td><td>0,00</td><td>195,31</td><td>ME</td></tr> <tr><td>Итого:</td><td>0,00</td><td>3</td><td></td></tr> <tr><td>Итого по договорам:</td><td>191596,66</td><td></td><td></td></tr> </tbody> </table> <p>Страница 1 из 1</p>	Услуга	Стоимость (руб)	Объем	Единица	Dial-Up(аренд)	0,00	78,00		Dial-Up(исходящий)	39,08	29,54	ME	Dial-Up(исходящий)	0,00	10,94	ME	Dial-Up(исходящий)	39,08			Итого:	191518,50			Итого:	191596,66	5		Услуга	Стоимость (руб)	Объем	Единица	Dial-Up(аренд)	0,00	21,00		Dial-Up(исходящий)	0,00	0,57	ME	Dial-Up(исходящий)	0,00	1,95	ME	Итого:	0,00	3		Услуга	Стоимость (руб)	Объем	Единица	Dial-Up(аренд)	0,00	584,00		Dial-Up(исходящий)	0,00	1,62	ME	Dial-Up(исходящий)	0,00	195,31	ME	Итого:	0,00	3		Итого по договорам:	191596,66		
Услуга	Стоимость (руб)	Объем	Единица																																																																						
Dial-Up(аренд)	0,00	78,00																																																																							
Dial-Up(исходящий)	39,08	29,54	ME																																																																						
Dial-Up(исходящий)	0,00	10,94	ME																																																																						
Dial-Up(исходящий)	39,08																																																																								
Итого:	191518,50																																																																								
Итого:	191596,66	5																																																																							
Услуга	Стоимость (руб)	Объем	Единица																																																																						
Dial-Up(аренд)	0,00	21,00																																																																							
Dial-Up(исходящий)	0,00	0,57	ME																																																																						
Dial-Up(исходящий)	0,00	1,95	ME																																																																						
Итого:	0,00	3																																																																							
Услуга	Стоимость (руб)	Объем	Единица																																																																						
Dial-Up(аренд)	0,00	584,00																																																																							
Dial-Up(исходящий)	0,00	1,62	ME																																																																						
Dial-Up(исходящий)	0,00	195,31	ME																																																																						
Итого:	0,00	3																																																																							
Итого по договорам:	191596,66																																																																								

Промежуточные итоги по договорам выводят количество записей для данного договора.

6. Отчёт Телекей 1

Выводит наработку с группировкой по договорам и услугам . Он имеет 2 части, вначале выводится информация по услугам для каждого договора, потом информация по услугам для всех договоров.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид								
<div data-bbox="159 380 289 405"> Параметры </div> <div data-bbox="276 415 795 464"> ← Февраль ▼ 2009 ▼ → </div> <div data-bbox="167 468 272 493"> Услуга в </div> <div data-bbox="337 520 730 548"> <input checked="" type="radio"/> КБ <input type="radio"/> МБ <input type="radio"/> ГБ <input type="radio"/> Часы </div> <div data-bbox="167 579 256 604"> Услуги </div> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Dial-Up(время) <input checked="" type="checkbox"/> Dial-Up(входящий) <input checked="" type="checkbox"/> Dial-Up(исходящий) <input checked="" type="checkbox"/> Dial-Up(локальный/время) <input checked="" type="checkbox"/> Dial-Up(локальный/входящий) <input checked="" type="checkbox"/> Dial-Up(локальный/исходящий) <input checked="" type="checkbox"/> Dial-UP(максимальный) <input checked="" type="checkbox"/> Dial-Up(превалирующий) <input checked="" type="checkbox"/> Dial-Up(суммарный) <input checked="" type="checkbox"/> WiFi_128 (по времени) <input checked="" type="checkbox"/> WiFi_256(по времени) <input checked="" type="checkbox"/> Абонплата1 <input checked="" type="checkbox"/> Тестовая WIFI (по времени) <input checked="" type="checkbox"/> Тестовая WIFI (по трафику) 	<div data-bbox="1284 401 1498 426" style="text-align: right;"> Отчет по трафик </div> <div data-bbox="1284 436 1498 462" style="text-align: right;"> год: 2009 месяц: февра </div> <div data-bbox="967 499 1027 525" style="text-align: right;"> x00 0 </div> <table border="1" data-bbox="967 531 1498 583"> <thead> <tr style="background-color: #00FF00;"> <th>Услуга</th> <th>Объем</th> </tr> </thead> <tbody> <tr> <td>Dial- Up(входящий)</td> <td style="text-align: right;">16640</td> </tr> </tbody> </table> <div data-bbox="967 615 1060 640" style="text-align: right;"> Итого: 0 </div> <table border="1" data-bbox="967 646 1498 699"> <thead> <tr style="background-color: #00FF00;"> <th>Услуга</th> <th>Объем</th> </tr> </thead> <tbody> <tr> <td>Dial- Up(входящий)</td> <td style="text-align: right;">16640</td> </tr> </tbody> </table>	Услуга	Объем	Dial- Up(входящий)	16640	Услуга	Объем	Dial- Up(входящий)	16640
Услуга	Объем								
Dial- Up(входящий)	16640								
Услуга	Объем								
Dial- Up(входящий)	16640								

Отчёты модуля IPN

1. Отчет по трафику

Позволяет вывести объем трафика за выбранный месяц по группам договоров. Он имеет 2 части: вначале выводится информация по услугам для каждого договора, потом информация по услугам для всех договоров. **Таблица. Фильтр и внешний вид**

Фильтр	Внешний вид																
<p>Параметры</p> <p>Период</p> <p>◀ Январь 2006 ▶</p> <p>дни: с <input type="text"/> до <input type="text"/></p> <p>Часы</p> <p>часы с: <input type="text"/> по: <input type="text"/></p> <p>Группы</p> <ul style="list-style-type: none"><input type="checkbox"/> A<input type="checkbox"/> AA<input type="checkbox"/> AD<input type="checkbox"/> AL<input type="checkbox"/> AO<input type="checkbox"/> AS<input type="checkbox"/> B<input type="checkbox"/> IP2<input type="checkbox"/> Local	<p style="text-align: center;">Отчет по трафику</p> <table border="1"><thead><tr><th>Услуга</th><th>Объем (МБ)</th></tr></thead><tbody><tr><td>(IPN) Локальный трафик (исходящий)</td><td>1 500</td></tr><tr><td>(IPN) Внешний трафик (исходящий)</td><td>1</td></tr><tr><td>(IPN) Внешний трафик (входящий)</td><td>1 500</td></tr><tr><td>(IPN) Локальный трафик (входящий)</td><td>1 500</td></tr><tr><td>(IPN) Почтовый сервер POP3.UFANET.RU</td><td>4</td></tr><tr><td>(IPN) Прокси IP1(входящий)</td><td>1</td></tr><tr><td>(IPN) Компьютерная телефония</td><td>0</td></tr></tbody></table>	Услуга	Объем (МБ)	(IPN) Локальный трафик (исходящий)	1 500	(IPN) Внешний трафик (исходящий)	1	(IPN) Внешний трафик (входящий)	1 500	(IPN) Локальный трафик (входящий)	1 500	(IPN) Почтовый сервер POP3.UFANET.RU	4	(IPN) Прокси IP1(входящий)	1	(IPN) Компьютерная телефония	0
Услуга	Объем (МБ)																
(IPN) Локальный трафик (исходящий)	1 500																
(IPN) Внешний трафик (исходящий)	1																
(IPN) Внешний трафик (входящий)	1 500																
(IPN) Локальный трафик (входящий)	1 500																
(IPN) Почтовый сервер POP3.UFANET.RU	4																
(IPN) Прокси IP1(входящий)	1																
(IPN) Компьютерная телефония	0																

2. Отчет Телекей 1.

Выводит наработку по трафику с группировкой по договорам и услугам.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид								
	<p style="text-align: center;">Отчет по трафику год: 2009 месяц: февра</p> <p>x0000 (Иванов Петр Сидорович)</p> <table border="1"><thead><tr><th>Услуга</th><th>Объем</th></tr></thead><tbody><tr><td>Внешний вх.</td><td>18570</td></tr></tbody></table> <p>Итого: 0</p> <table border="1"><thead><tr><th>Услуга</th><th>Объем</th></tr></thead><tbody><tr><td>Внешний вх.</td><td>18570</td></tr></tbody></table>	Услуга	Объем	Внешний вх.	18570	Услуга	Объем	Внешний вх.	18570
Услуга	Объем								
Внешний вх.	18570								
Услуга	Объем								
Внешний вх.	18570								

параметры

Февраль ▼ 2009 ▼

Услуга в

КБ МБ ГБ Часы

Услуги

- Внешний вх.
- Входящий POP3
- Исходящий внутренний
- Локальный вх.
- Локальный исх.
- Максимальный IPN

Отчёты модуля IP Телефония

1. Отчет по сессиям

Позволяет выбрать сессии за определённый период с фильтром по причине завершения (disconnect-cause). Этот отчёт предназначен, прежде всего, для мониторинга качества связи. Несколько DC могут быть введены через запятую.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																																																																																																												
<p>Отчеты </p> <p>Модуль</p> <p>IP телефония</p> <p>Отчет</p> <p>Отчет по сессиям</p> <p>Параметры</p> <p>Период</p> <p>с 13.07.2005 по 13.07.2005</p> <p>Фильтр по DC</p> <p><input checked="" type="radio"/> включить <input type="radio"/> исключить</p> <p>10</p>	<table border="1"> <thead> <tr> <th>Договор</th> <th>Вр. входа</th> <th>Длнк.</th> <th>Ст.-ль</th> <th>С номера / на номер</th> <th>DC</th> <th>NAS</th> <th>Reason #</th> <th>connect time</th> </tr> </thead> <tbody> <tr> <td>XW0013-01</td> <td>13.07.2005 00:00:05</td> <td>0</td> <td>0.00000</td> <td>555557909352371</td> <td>10</td> <td>81.30.206.222</td> <td>194.67.26.235</td> <td>00:00:06.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0012-04</td> <td>13.07.2005 00:00:13</td> <td>0</td> <td>0.00000</td> <td>90144009031714529336</td> <td>10</td> <td>81.30.206.222</td> <td>217.151.64.138</td> <td>00:00:13.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0038-04</td> <td>13.07.2005 00:00:16</td> <td>0</td> <td>0.00000</td> <td>4473472778493</td> <td>10</td> <td>81.30.206.222</td> <td>80.86.166.248</td> <td>00:00:17.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0013-01</td> <td>13.07.2005 00:00:18</td> <td>0</td> <td>0.00000</td> <td>5555579174754206</td> <td>10</td> <td>81.30.206.222</td> <td>194.67.26.234</td> <td>00:00:18.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0013-01</td> <td>13.07.2005 00:00:25</td> <td>17</td> <td>0.00368</td> <td>555557904255274</td> <td>10</td> <td>81.30.206.222</td> <td>194.67.26.235</td> <td>00:00:26.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0019-04</td> <td>13.07.2005 00:00:29</td> <td>0</td> <td>0.00000</td> <td>738323392375273222938</td> <td>10</td> <td>81.30.206.222</td> <td>193.124.20.58</td> <td>00:00:30.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0011-04</td> <td>13.07.2005 00:00:35</td> <td>307</td> <td>0.05117</td> <td>980048873472353193</td> <td>10</td> <td>81.30.206.222</td> <td>82.198.166.14</td> <td>00:00:36.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0029-03</td> <td>13.07.2005 00:00:39</td> <td>168</td> <td>0.70000</td> <td>347299022273472243847</td> <td>10</td> <td>81.30.206.222</td> <td>212.53.35.34</td> <td>00:00:39.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0014-04</td> <td>13.07.2005 00:00:46</td> <td>787</td> <td>0.13117</td> <td>347623234073472350260</td> <td>10</td> <td>81.30.206.222</td> <td>83.102.205.7</td> <td>00:00:48.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0013-01</td> <td>13.07.2005 00:00:49</td> <td>0</td> <td>0.00000</td> <td>5555579174866759</td> <td>10</td> <td>81.30.206.222</td> <td>194.67.26.235</td> <td>00:00:50.000 YEKST Wed Jul 13</td> </tr> <tr> <td>XW0013-01</td> <td>13.07.2005 00:01:16</td> <td>0</td> <td>0.00000</td> <td>5555579177343483</td> <td>10</td> <td>81.30.206.222</td> <td>194.67.26.235</td> <td>00:01:04.000 YEKST Wed Jul 13</td> </tr> </tbody> </table>	Договор	Вр. входа	Длнк.	Ст.-ль	С номера / на номер	DC	NAS	Reason #	connect time	XW0013-01	13.07.2005 00:00:05	0	0.00000	555557909352371	10	81.30.206.222	194.67.26.235	00:00:06.000 YEKST Wed Jul 13	XW0012-04	13.07.2005 00:00:13	0	0.00000	90144009031714529336	10	81.30.206.222	217.151.64.138	00:00:13.000 YEKST Wed Jul 13	XW0038-04	13.07.2005 00:00:16	0	0.00000	4473472778493	10	81.30.206.222	80.86.166.248	00:00:17.000 YEKST Wed Jul 13	XW0013-01	13.07.2005 00:00:18	0	0.00000	5555579174754206	10	81.30.206.222	194.67.26.234	00:00:18.000 YEKST Wed Jul 13	XW0013-01	13.07.2005 00:00:25	17	0.00368	555557904255274	10	81.30.206.222	194.67.26.235	00:00:26.000 YEKST Wed Jul 13	XW0019-04	13.07.2005 00:00:29	0	0.00000	738323392375273222938	10	81.30.206.222	193.124.20.58	00:00:30.000 YEKST Wed Jul 13	XW0011-04	13.07.2005 00:00:35	307	0.05117	980048873472353193	10	81.30.206.222	82.198.166.14	00:00:36.000 YEKST Wed Jul 13	XW0029-03	13.07.2005 00:00:39	168	0.70000	347299022273472243847	10	81.30.206.222	212.53.35.34	00:00:39.000 YEKST Wed Jul 13	XW0014-04	13.07.2005 00:00:46	787	0.13117	347623234073472350260	10	81.30.206.222	83.102.205.7	00:00:48.000 YEKST Wed Jul 13	XW0013-01	13.07.2005 00:00:49	0	0.00000	5555579174866759	10	81.30.206.222	194.67.26.235	00:00:50.000 YEKST Wed Jul 13	XW0013-01	13.07.2005 00:01:16	0	0.00000	5555579177343483	10	81.30.206.222	194.67.26.235	00:01:04.000 YEKST Wed Jul 13
Договор	Вр. входа	Длнк.	Ст.-ль	С номера / на номер	DC	NAS	Reason #	connect time																																																																																																					
XW0013-01	13.07.2005 00:00:05	0	0.00000	555557909352371	10	81.30.206.222	194.67.26.235	00:00:06.000 YEKST Wed Jul 13																																																																																																					
XW0012-04	13.07.2005 00:00:13	0	0.00000	90144009031714529336	10	81.30.206.222	217.151.64.138	00:00:13.000 YEKST Wed Jul 13																																																																																																					
XW0038-04	13.07.2005 00:00:16	0	0.00000	4473472778493	10	81.30.206.222	80.86.166.248	00:00:17.000 YEKST Wed Jul 13																																																																																																					
XW0013-01	13.07.2005 00:00:18	0	0.00000	5555579174754206	10	81.30.206.222	194.67.26.234	00:00:18.000 YEKST Wed Jul 13																																																																																																					
XW0013-01	13.07.2005 00:00:25	17	0.00368	555557904255274	10	81.30.206.222	194.67.26.235	00:00:26.000 YEKST Wed Jul 13																																																																																																					
XW0019-04	13.07.2005 00:00:29	0	0.00000	738323392375273222938	10	81.30.206.222	193.124.20.58	00:00:30.000 YEKST Wed Jul 13																																																																																																					
XW0011-04	13.07.2005 00:00:35	307	0.05117	980048873472353193	10	81.30.206.222	82.198.166.14	00:00:36.000 YEKST Wed Jul 13																																																																																																					
XW0029-03	13.07.2005 00:00:39	168	0.70000	347299022273472243847	10	81.30.206.222	212.53.35.34	00:00:39.000 YEKST Wed Jul 13																																																																																																					
XW0014-04	13.07.2005 00:00:46	787	0.13117	347623234073472350260	10	81.30.206.222	83.102.205.7	00:00:48.000 YEKST Wed Jul 13																																																																																																					
XW0013-01	13.07.2005 00:00:49	0	0.00000	5555579174866759	10	81.30.206.222	194.67.26.235	00:00:50.000 YEKST Wed Jul 13																																																																																																					
XW0013-01	13.07.2005 00:01:16	0	0.00000	5555579177343483	10	81.30.206.222	194.67.26.235	00:01:04.000 YEKST Wed Jul 13																																																																																																					

2. Отчет Чита-Он-Лайн 1

Позволяет просмотреть voip-сессии с фильтром по типу и группам договоров. В отличие от предыдущего отчёта техническая информация не отображается.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																																																		
	<table border="1"> <thead> <tr> <th>Договор</th> <th>Логин</th> <th>Тип</th> <th>С номера / на номер</th> <th>Направление</th> </tr> </thead> <tbody> <tr> <td>K186278-05 ()</td> <td>186278 ()</td> <td><<</td> <td>-80955168825</td> <td>Москва</td> </tr> <tr> <td>K259482-06 ()</td> <td>259482 ()</td> <td><<</td> <td>-810380546851423</td> <td>Узбекистан</td> </tr> <tr> <td>K259482-06 ()</td> <td>259482 ()</td> <td><<</td> <td>-810380542217454</td> <td>Узбекистан</td> </tr> <tr> <td>K411725-06 ()</td> <td>411725 ()</td> <td><<</td> <td>-81012547293336</td> <td>США,</td> </tr> <tr> <td>K259482-06 ()</td> <td>259482 ()</td> <td><<</td> <td>-810380542217454</td> <td>Узбекистан</td> </tr> <tr> <td>K324411-06 ()</td> <td>324411 ()</td> <td><<</td> <td>-89166266804</td> <td>Москва и Моск.</td> </tr> <tr> <td>K555112-05 ()</td> <td>555112 ()</td> <td><<</td> <td>-88617617853</td> <td>Краснодар</td> </tr> <tr> <td>K324411-06 ()</td> <td>324411 ()</td> <td><<</td> <td>-80951058646</td> <td>Москва</td> </tr> <tr> <td>K259482-06 ()</td> <td>259482 ()</td> <td><<</td> <td>-810380542217454</td> <td>Узбекистан</td> </tr> </tbody> </table>	Договор	Логин	Тип	С номера / на номер	Направление	K186278-05 ()	186278 ()	<<	-80955168825	Москва	K259482-06 ()	259482 ()	<<	-810380546851423	Узбекистан	K259482-06 ()	259482 ()	<<	-810380542217454	Узбекистан	K411725-06 ()	411725 ()	<<	-81012547293336	США,	K259482-06 ()	259482 ()	<<	-810380542217454	Узбекистан	K324411-06 ()	324411 ()	<<	-89166266804	Москва и Моск.	K555112-05 ()	555112 ()	<<	-88617617853	Краснодар	K324411-06 ()	324411 ()	<<	-80951058646	Москва	K259482-06 ()	259482 ()	<<	-810380542217454	Узбекистан
Договор	Логин	Тип	С номера / на номер	Направление																																															
K186278-05 ()	186278 ()	<<	-80955168825	Москва																																															
K259482-06 ()	259482 ()	<<	-810380546851423	Узбекистан																																															
K259482-06 ()	259482 ()	<<	-810380542217454	Узбекистан																																															
K411725-06 ()	411725 ()	<<	-81012547293336	США,																																															
K259482-06 ()	259482 ()	<<	-810380542217454	Узбекистан																																															
K324411-06 ()	324411 ()	<<	-89166266804	Москва и Моск.																																															
K555112-05 ()	555112 ()	<<	-88617617853	Краснодар																																															
K324411-06 ()	324411 ()	<<	-80951058646	Москва																																															
K259482-06 ()	259482 ()	<<	-810380542217454	Узбекистан																																															

Параметры

Период

◀ Январь ▼ 2006 ▼ ▶

дни: с до

Фильтр

Вывести: только платные

Группы | Направления

- DialUP
- IPN
- VOIP
- VPN
- VPNU
- Карточный

Отчёты плагина CRM

1. Отчет по проблемам

Позволяет отслеживать количество проблем, решённых различными исполнителями, а также количество проблем по категориям и типам. **Таблица. Фильтр и внешний вид**

Фильтр	Внешний вид																								
<p>Параметры</p> <p>Период</p> <p>с <input type="text" value="13.01.2003"/> по <input type="text" value="13.01.2010"/></p> <p>Группы Категории Исполнители</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Диспетчера<input checked="" type="checkbox"/> Оптики <p><input type="button" value="Все"/> <input type="button" value="Сброс"/> <input type="button" value="Инверт."/></p>	<p>Отчет г</p> <table border="1"><thead><tr><th>Договор</th><th>Код</th><th>Вес</th><th>Открыта</th><th>Закрыта</th><th></th></tr></thead><tbody><tr><td>-</td><td>61</td><td>10.05</td><td>19:29</td><td>10.05 19:30</td><td>К</td></tr><tr><td>-</td><td>102</td><td>11.06</td><td>15:03 (123)</td><td>01.07 14:27 (123)</td><td>К</td></tr><tr><td>-</td><td>106</td><td>11.07</td><td>17:28 (123)</td><td></td><td>Г</td></tr></tbody></table> <p>Итого: 3</p> <p><small>Документ подготовлен системой "BC-Billing" © ООО "Билет" http://www.bilet.ru</small></p>	Договор	Код	Вес	Открыта	Закрыта		-	61	10.05	19:29	10.05 19:30	К	-	102	11.06	15:03 (123)	01.07 14:27 (123)	К	-	106	11.07	17:28 (123)		Г
Договор	Код	Вес	Открыта	Закрыта																					
-	61	10.05	19:29	10.05 19:30	К																				
-	102	11.06	15:03 (123)	01.07 14:27 (123)	К																				
-	106	11.07	17:28 (123)		Г																				

Отчёты модуля Телефония (Phone)

1. Отчет Dect 1.

Отчет выводит статистические данные по услугам телефонии. Для каждой услуги он выводит суммарное количество соединений, суммарную длительность соединений и сумму за указанный период. Доступна фильтрация по услугам и группам договоров.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид									
<p>Параметры</p> <p>Период</p> <p>← Август 2008 →</p> <p>дни: с <input type="text"/> до <input type="text"/></p> <p>Часы</p> <p>часы с: <input type="text"/> по: <input type="text"/></p> <p>Услуги Группы договоров</p> <p>Услуги</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Зоновая опер 2 преселект<input checked="" type="checkbox"/> Местная связь<input checked="" type="checkbox"/> Местное завершение 1<input checked="" type="checkbox"/> Совинтел МГ<input checked="" type="checkbox"/> Совинтел МН	<p>Отчет по услугам год: 2008 месяц: август</p> <table border="1"><thead><tr><th>Услуга</th><th>Кол-во соединений (шт)</th><th>Длгт. соединений (мин)</th></tr></thead><tbody><tr><td>Совинтел МГ</td><td>27</td><td>20</td></tr><tr><td>Итого:</td><td>27</td><td>20</td></tr></tbody></table>	Услуга	Кол-во соединений (шт)	Длгт. соединений (мин)	Совинтел МГ	27	20	Итого:	27	20
Услуга	Кол-во соединений (шт)	Длгт. соединений (мин)								
Совинтел МГ	27	20								
Итого:	27	20								

2. Отчет по направлениям.

Отчет по направлениям за месяц. Для каждого направления выводится суммарное количество соединений, суммарную длительность соединений и сумму. Доступна фильтрация по услугам и группам договоров, префиксу и типу лица договора (физ.лицо, юр.лицо, все).

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид						
	<p>Отчет по направлению Август 2008</p> <table border="1"><thead><tr><th>Направление</th><th>Количество сессий</th></tr></thead><tbody><tr><td>Sataga</td><td>27</td></tr><tr><td>Итого:</td><td>27</td></tr></tbody></table>	Направление	Количество сессий	Sataga	27	Итого:	27
Направление	Количество сессий						
Sataga	27						
Итого:	27						

Параметры

Месяц

◀ Август ▼ 2008 ▼ ▶

Префикс

Режим

Любые ▼

Группы договоров

Услуги

Услуги

- Зоновая опер 2 преселект
- Местная связь
- Местное завершение 1
- Совинтел МГ
- Совинтел МН

Отчёты модуля Карточки

1. Отчет по дилерам.

Выводит информацию по дилерам: количество карт на начало периода (Входящий остаток), количество карт? взятых дилером за период, количество активированных карт за период, количество карт на конец периода. Доступна фильтрация по периоду, стоимости карт (сумма), услугам, дилерам.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																		
<p>Параметры</p> <p>Период</p> <p>с <input type="text" value="18.02.2009"/> по <input type="text" value="18.02.2009"/></p> <p>Сумма</p> <p>с <input type="text"/> по <input type="text"/></p> <p>Услуги Дилеры</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Агентское вознаграждение<input checked="" type="checkbox"/> Активация DialUp карт<input checked="" type="checkbox"/> Карточки<input checked="" type="checkbox"/> Платежи	<p>Отчет по дилерам</p> <p>Период: 18.02.09 - 18.02.09</p> <table border="1"><thead><tr><th>Дилер</th><th>Вх. ост.</th><th>Взял э/п</th></tr></thead><tbody><tr><td>111 sdfs</td><td>0</td><td>0</td></tr><tr><td>fsdfs d</td><td>0</td><td>0</td></tr><tr><td>pech</td><td>2</td><td>0</td></tr><tr><td>sdfs d</td><td>1</td><td>0</td></tr><tr><td>test</td><td>0</td><td>0</td></tr></tbody></table>	Дилер	Вх. ост.	Взял э/п	111 sdfs	0	0	fsdfs d	0	0	pech	2	0	sdfs d	1	0	test	0	0
Дилер	Вх. ост.	Взял э/п																	
111 sdfs	0	0																	
fsdfs d	0	0																	
pech	2	0																	
sdfs d	1	0																	
test	0	0																	

Отчёты модуля RSCM

1. Отчет "Разовые услуги RSCM"

Отчет выводит начисленные разовые услуги по всем договорам за период. Отчет не является встроенным и может быть изменен. Файлы отчёта - rscm_contract_telenettv_rscm.jrxml и rscm_contract_telenettv_rscm.rep.xml.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид																
<p>Параметры</p> <p>Месяц и дни</p> <p>◀ Февраль 2009 ▶</p> <p>дни: с <input type="text"/> до <input type="text"/></p>	<p>Разовые услуги RSC</p> <table border="1"><thead><tr><th>№№</th><th>Номер договора</th><th>Дата</th><th>Уп</th></tr></thead><tbody><tr><td>1</td><td>king</td><td>19.02.09 0:00</td><td>Вызов ма</td></tr><tr><td>2</td><td>king</td><td>11.02.09 0:00</td><td>Получение пакета обнов</td></tr><tr><td colspan="4">Итого: 2</td></tr></tbody></table>	№№	Номер договора	Дата	Уп	1	king	19.02.09 0:00	Вызов ма	2	king	11.02.09 0:00	Получение пакета обнов	Итого: 2			
№№	Номер договора	Дата	Уп														
1	king	19.02.09 0:00	Вызов ма														
2	king	11.02.09 0:00	Получение пакета обнов														
Итого: 2																	

Отчёты модуля CerberCrypt

Отчет "Количество абонентов"

Выводит количество абонентов по пакетам.

Количество абонентов

2009/03/18

Пакет	Количество
Пакет 1	1
Пакет 2-3	0
Пакет 4-5	0
Виртуальный кинозал	0

Отчет "Наработка пакетов"

Выводит общую наработку по пакетам за определённый период месяца.

Таблица. Фильтр и внешний вид

Фильтр	Внешний вид								
<p>Отчет</p> <p>Наработка пакетов ▼</p> <p>Параметры</p> <p>Месяц и дни</p> <p>◀ Декабрь ▼ 2008 ▼ ▶</p> <p>дни: с 1 до 15</p>	<p>Наработка пак</p> <p>2008/12/01 - 2008</p> <table border="1"><thead><tr><th data-bbox="980 428 1503 453">Пакет</th><th data-bbox="980 453 1503 478">С</th></tr></thead><tbody><tr><td data-bbox="980 478 1503 504">Пакет 1</td><td data-bbox="980 504 1503 529">:</td></tr><tr><td data-bbox="980 504 1503 529">Пакет 2-3</td><td data-bbox="980 529 1503 554">:</td></tr><tr><td data-bbox="980 529 1503 554">Пакет 4-5</td><td data-bbox="980 554 1503 579">:</td></tr></tbody></table>	Пакет	С	Пакет 1	:	Пакет 2-3	:	Пакет 4-5	:
Пакет	С								
Пакет 1	:								
Пакет 2-3	:								
Пакет 4-5	:								

Создание собственных отчетов

Для модуля отчетов возможно создание шаблонов двух видов: в формате JasperReports с помощью программы iReport и создание табличных отчетов.

Фильтр отчетов создаётся единым образом, меняется только их вывод.

Для того, чтобы сервер не кэшировал шаблоны отчетов добавьте в конфигурацию сервера (не модуля) параметр **reports.cache=0**.

Для того, чтобы скрыть "ненужные" отчеты из выпадающего списка с отчетами, необходимо удалить поле **title** в файле *.rep.xml (см. ниже).

Есть возможность добавить дополнительные группы отчетов в списке модулей отчетов, для группировки, если своих (или переделанных) отчетов много. После этого имена шаблонов надо начинать с соответствующего префикса.

```
# .
reports.custom.groups=custom:test:
```

Настройка фильтра

Создайте в директории сервера биллинга директорию reports. Создайте файл **<модуль>_<id>.rep.xml**, где <модуль> - название модуля, для которого создаётся отчёт: kernel (ядро, основной модуль), dialup, voiceip, bill, mps и т.д., а <id> - уникальное имя отчёта (рекомендуется добавлять суффикс, чтобы идентификатор не совпал с идентификаторами отчётов, которые входят/войдут в будущем в дистрибутив модуля).

Структура файла *.rep.xml:

```

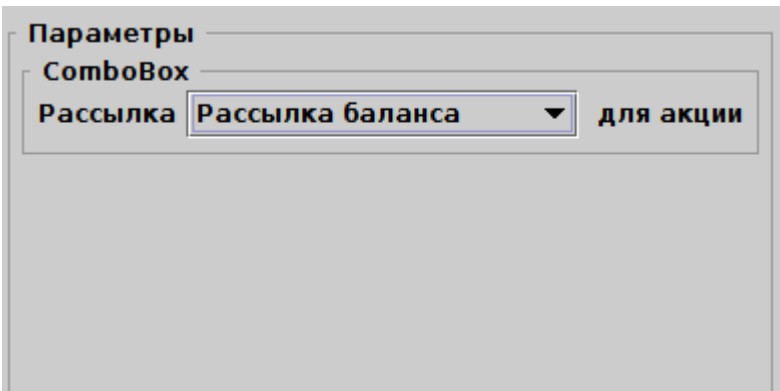
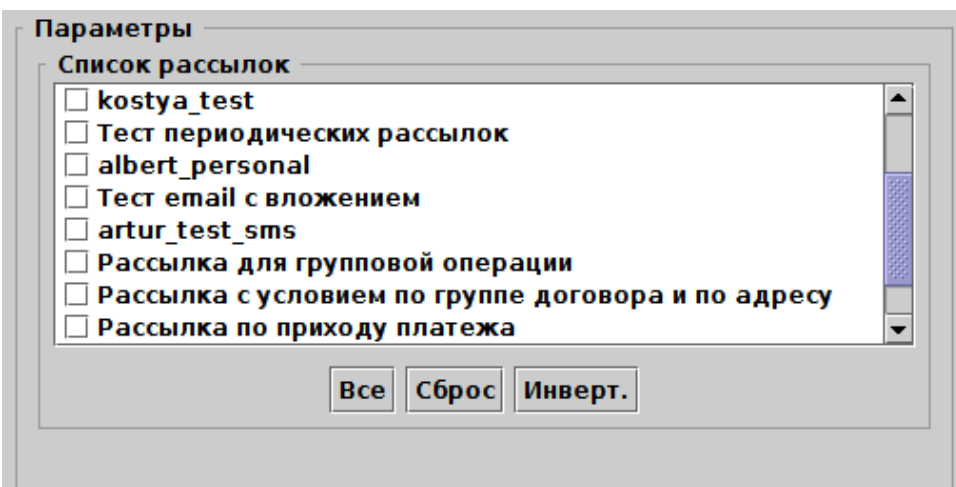
Структура файла rep.xml

<?xml version="1.0" encoding="UTF-8"?>
<report title=" " >
    <_>
</report>
    
```

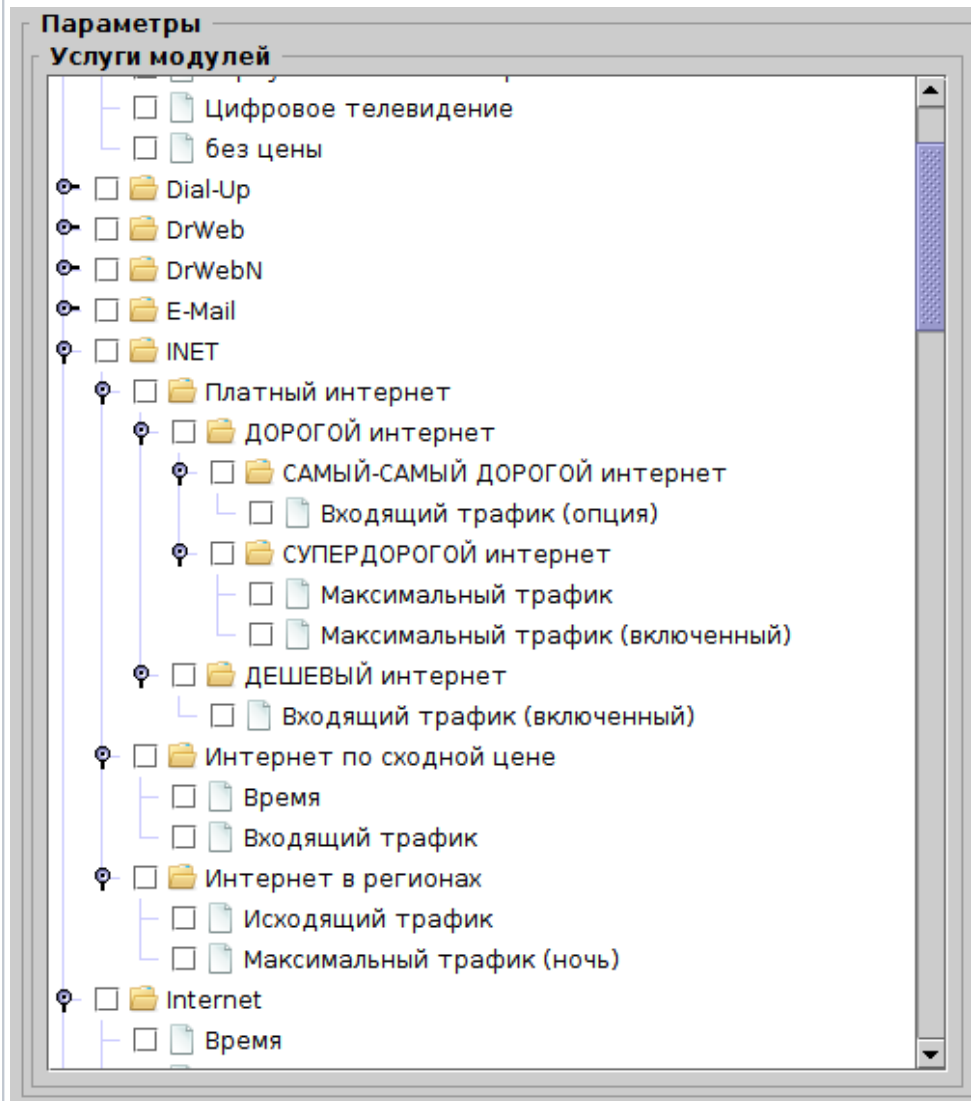
В *.rep.xml хранится название отчёта и описание фильтра в виде xml-файла. **report/@title** - отображаемое в списке название отчёта (если имя пустое или отсутствует, то отчет не попадает в выпадающий список с перечнем отчетов).

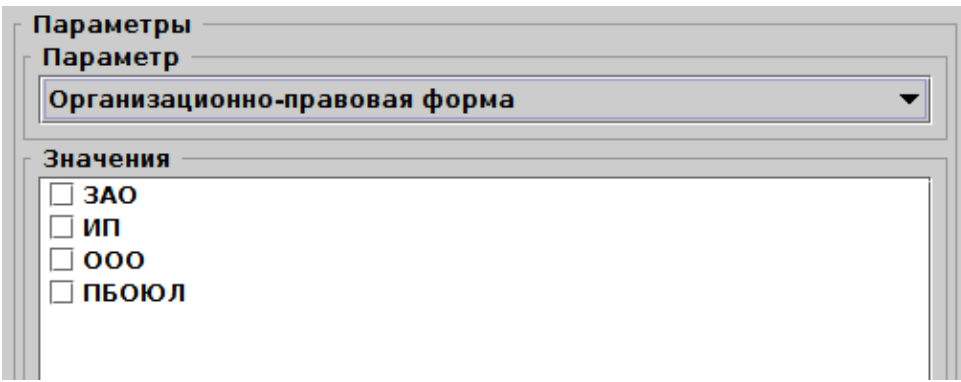
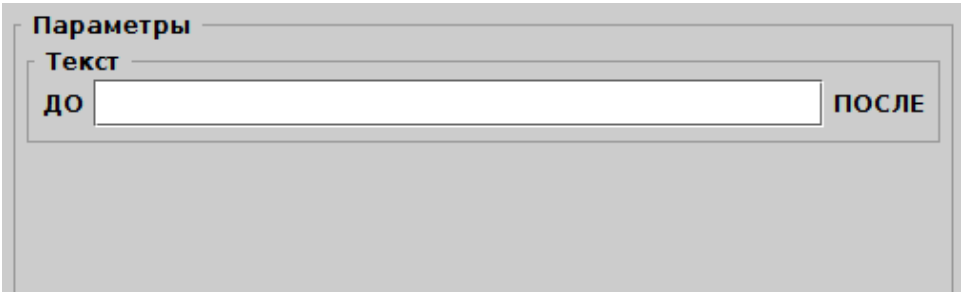
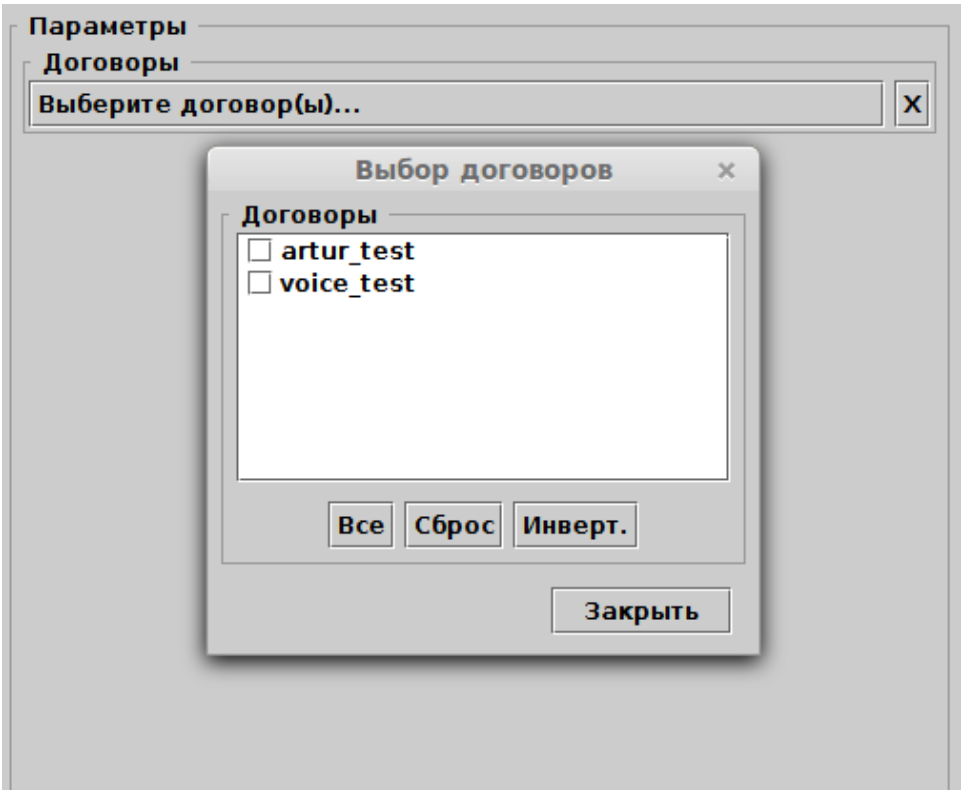
Внутри report возможно использовать следующие фильтры:

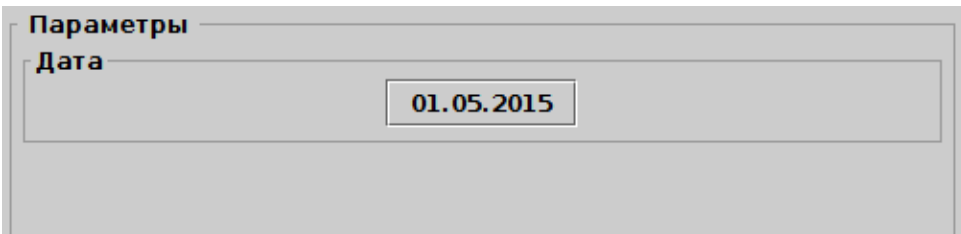
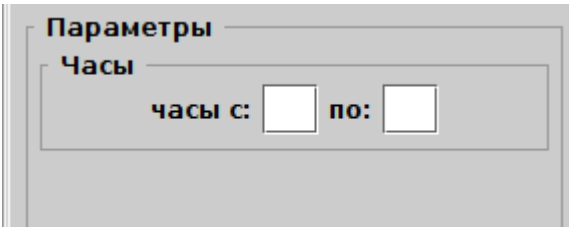
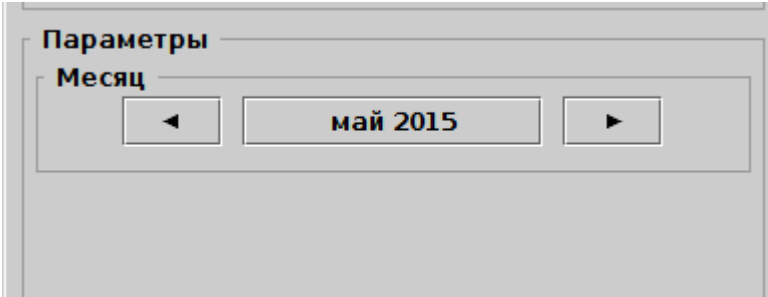
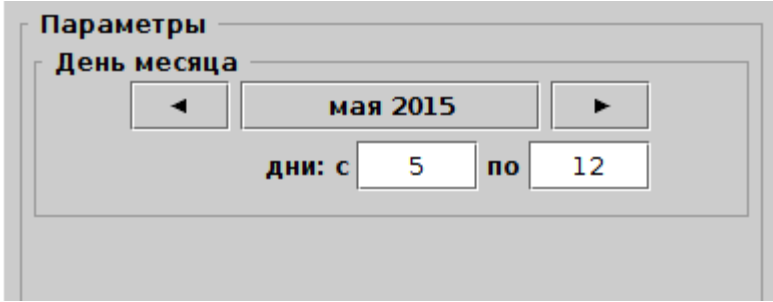
Имя	Параметры	Описание	Пример использования	Внешний вид
hide	title - название панели	Панель-контейнер, которая показывает /скрывает свое содержимое по щелчку на заголовке.	<pre> <hide title=" Месяцы"> <month name="m1" title=" Месяц с"/> <month name="m2" title=" Месяц по" /> </hide> </pre>	
combobox	name - имя переменной, по которой будет доступно значение данного	Выпадающий список с возможностью выбора значений. Заполнение значений списка осуществляется	<pre> <combo name=" combo" title=" ComboBox" textBefore=" "Рассылка" textAfter=" для акции" </pre>	

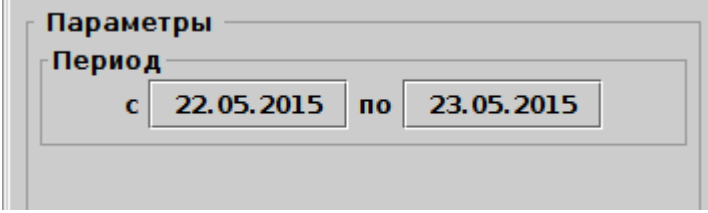
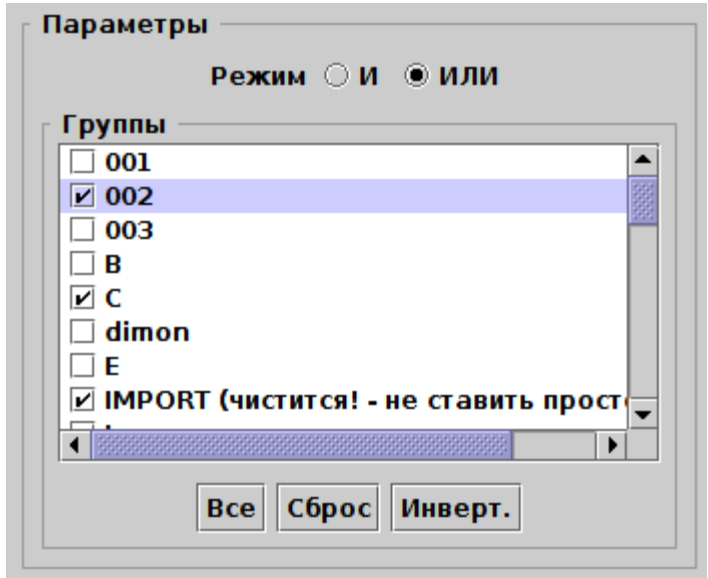
<p>фильтра в коде bgbs-скрипта;</p> <p>title - название для выпадающего списка;</p> <p>textBefore - текст до выпадающего списка;</p> <p>textAfter - текст после выпадающего списка;</p> <p>sql - select-запрос, который должен возвращать список значений вида "id->title", которыми заполнится выпадающий список.</p> <p>Параметры для item:</p> <p>id - уникальный код элемента;</p> <p>title - название элемента.</p>	<p>с помощью вложенного элемента item и /или sql-запроса.</p>	<pre>sql="select id,title from dispatch_dispatch"> <item id="0" title="-" /> <item id="1" title="Value 1" /> </combo></pre>	
<p>list name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта;</p> <p>title - название списка;</p> <p>sql - select-запрос, который должен возвращать список значений вида "id->title", которыми заполнится выпадающий список;</p> <p>tabbed - перенести список в закладочницу.</p> <p>Параметры для item:</p> <p>id - уникальный</p>	<p>Список с возможностью выбрать несколько элементов одновременно. Заполнение значений списка осуществляется с помощью вложенного элемента item и /или sql-запроса.</p>	<pre><list name="name" title="Список рассылок" sql="select id, title from dispatch_dispatch" tabbed="1"> <item id="0" title="Не выбрано" /> </list></pre>	

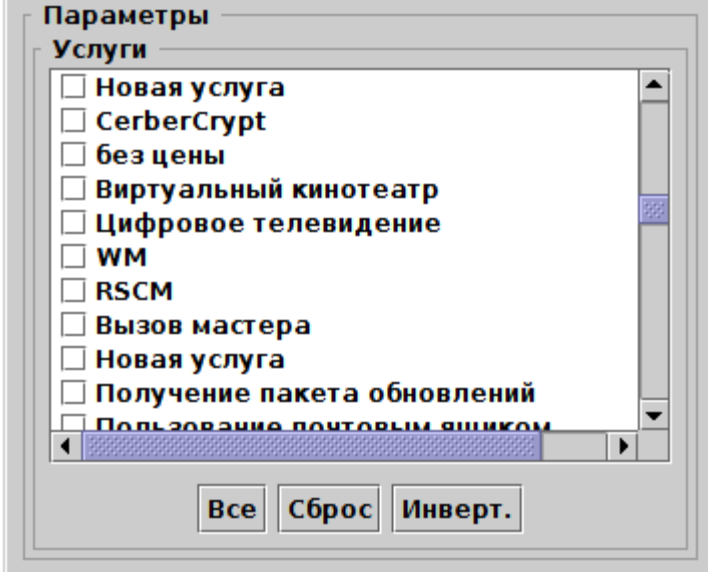
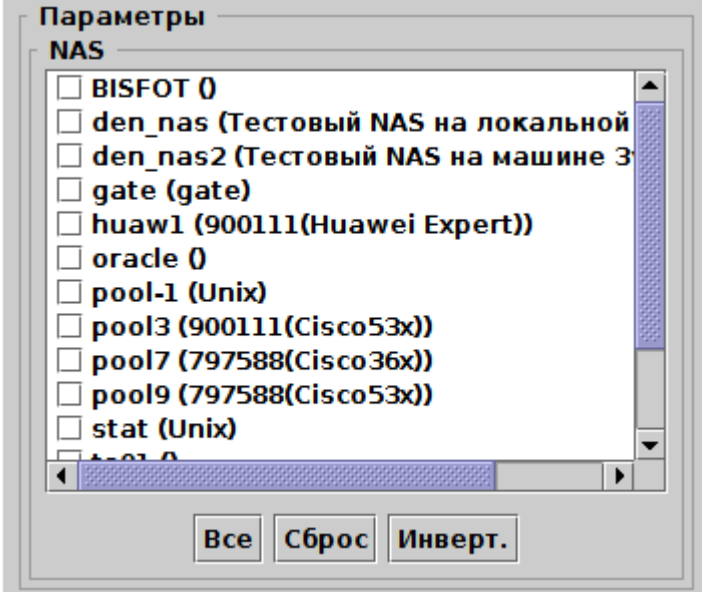
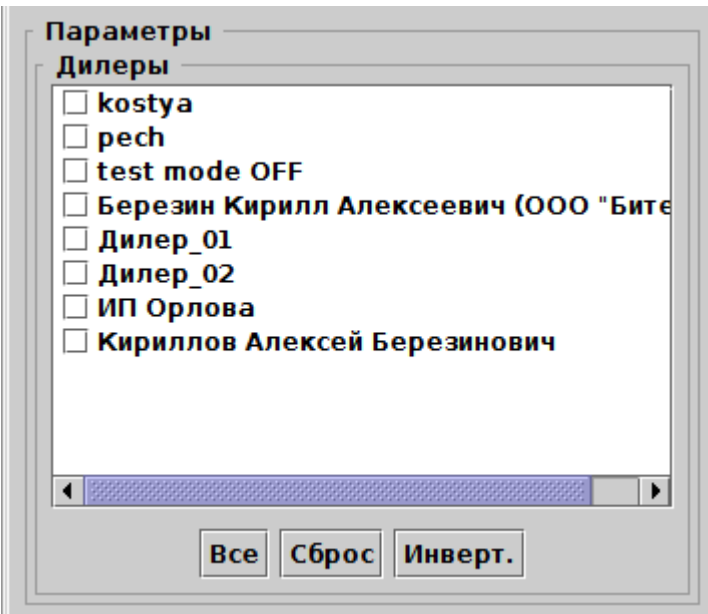
	код элемента;		
	title - название элемента.		
module	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта;</p> <p>title - название панели.</p> <p>Параметры service_set:</p> <p>mid - код модуля, для которого нужно каталогизировать услуги.</p> <p>Параметры folder:</p> <p>title - название каталога;</p> <p>items - список услуг через запятую. Также возможно указать ключевое слово others и будут отображены в данном каталоге ранее не указанные услуги.</p>	<p>Список услуг модулей с возможностью произвольной каталогизации. Выводятся только те модули, для которых заведены услуги. Если необходима каталогизация услуг для определенного модуля, то необходимо использовать теги service_set с вложенными тегами folder, которые, в свою очередь, также могут содержать folder.</p> <p>Внимание! Если для модуля не указан service_set, то отобразится линейный список всех услуг модуля.</p>	<pre> <module title=" Услуги модулей" name="sids" > <service_set mid="179"> <folder title=" Платный интернет"> <folder title=" ДОРОГОЙ интернет"> <folder title=" САМЫЙ- САМЫЙ ДОРОГОЙ интернет" items=" 208" /> <folder title=" СУПЕРДО РОГОЙ интернет" items=" 209,210" /> </folder> <folder title=" ДЕШЕВЫЙ интернет" items=" 207" /> </folder> <folder title=" Интернет по сходной цене" items=" 205,203" /> <folder title=" Интернет в регионах" items=" other" /> < /service_set> </module> </pre>
lists	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта;</p> <p>title - название для панели с списковыми</p>	<p>Имеющиеся в системе списковые параметры с значениями.</p>	<pre> <listParams name="par" title=" Списковые параметры " /> </pre>

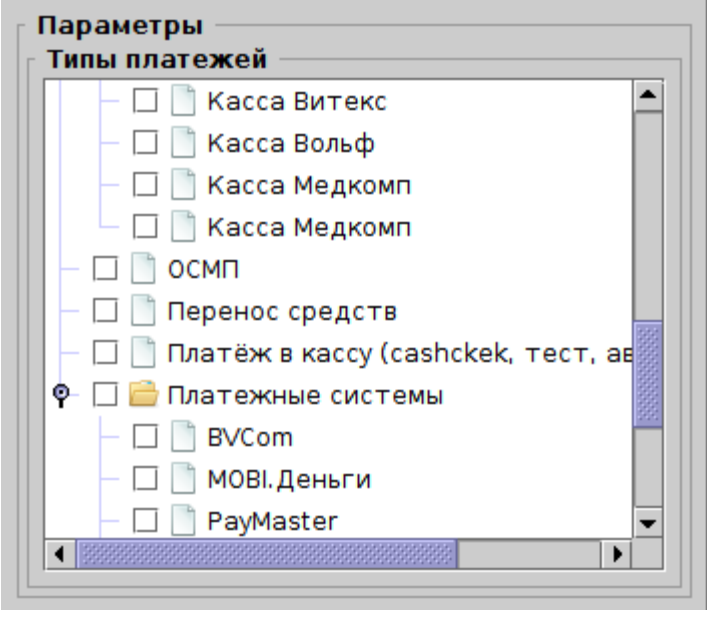
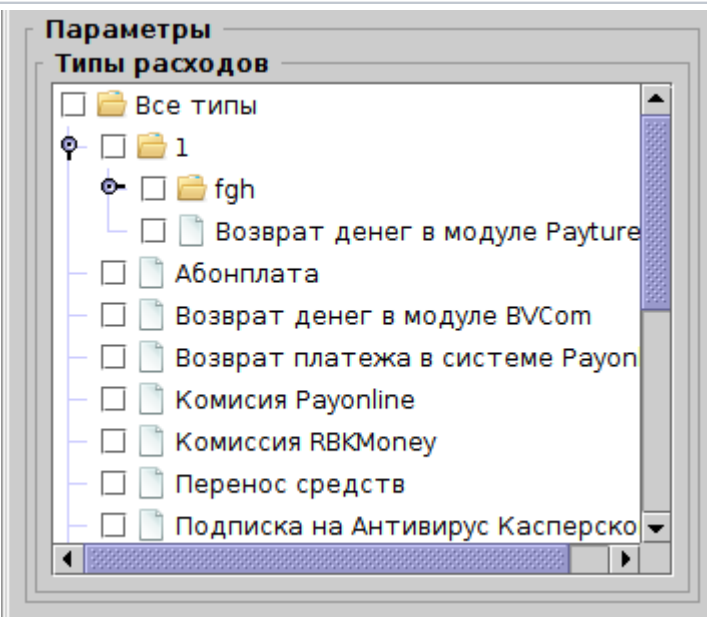


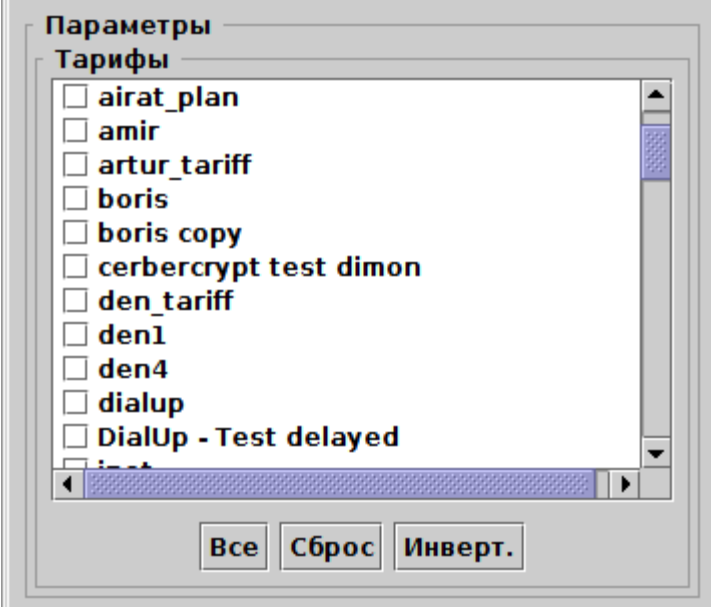
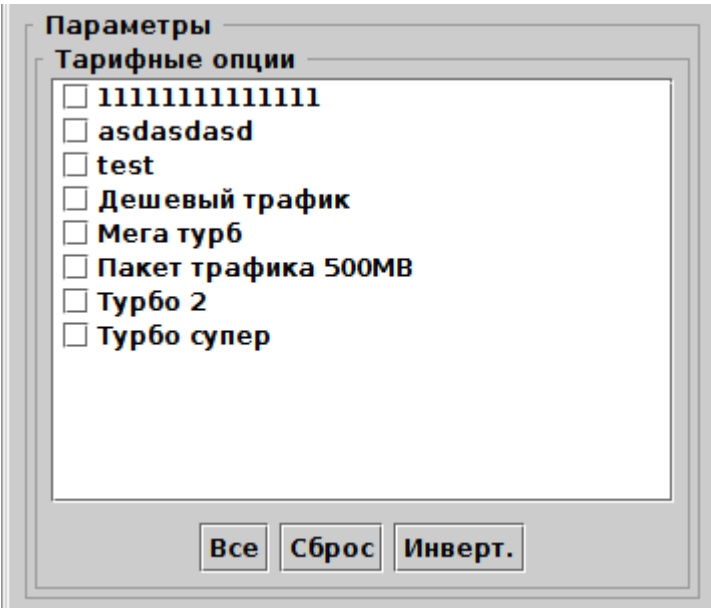
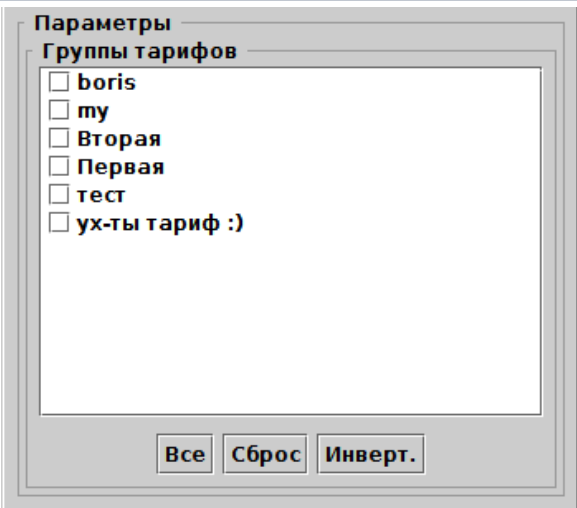
	параметрами договора.		
textField	name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта; title - название для текстового поля; textBefore - текст до поля; textAfter - текст после поля.	Обычное текстовое поле	<pre><textField name="Текстовое поле" textBefore="ДО" textAfter="ПОСЛЕ" /></pre> 
contracts	name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка кодов договоров через запятую.	Выбор одного и более договора, которые в данный момент открыты в клиенте биллинга.	<pre><contracts name="contracts" /></pre> 
date	name - имя переменной, по которой будет доступно	Выбор даты в календаре.	<pre><date name="dt" title="Дата" default=""/></pre>

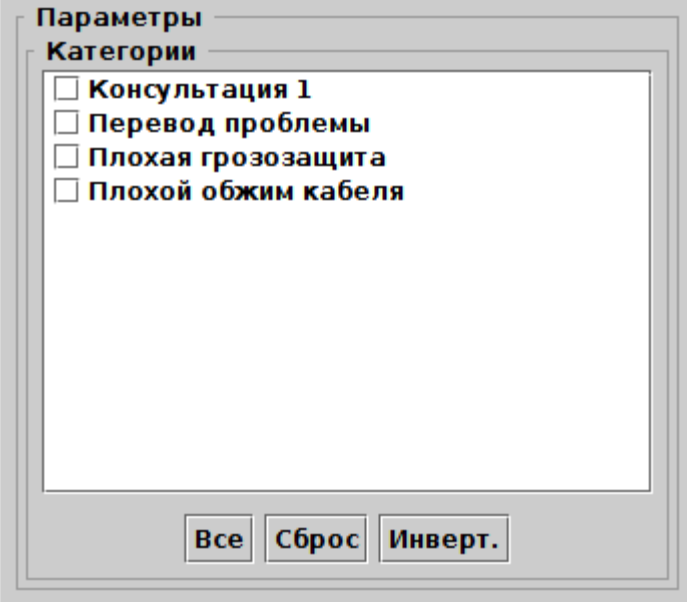
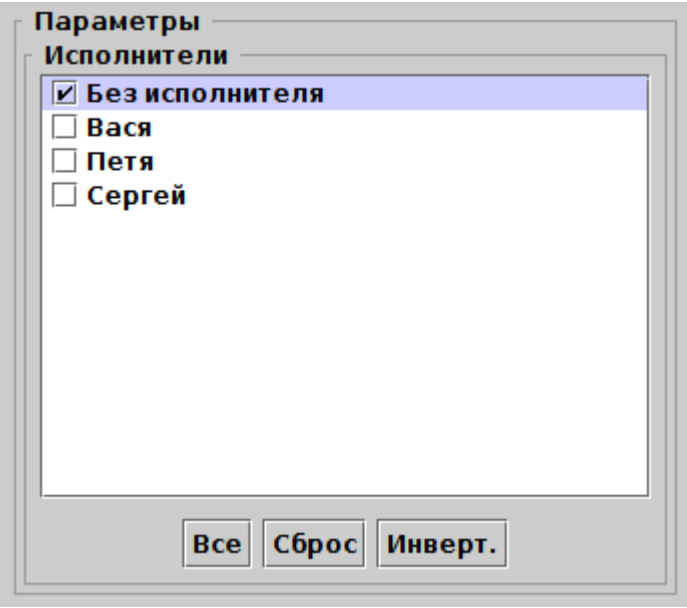
	<p>значение данного фильтра в коде bgbs-скрипта;</p> <p>title - название для каледаря;</p> <p>default - значение, установленное по умолчанию. Возможны варианты:</p> <p><i>today</i> - сегодня, <i>yesterday</i> - вчера, <i>tomorrow</i> - завтра, <i>first_day_of_month</i> - первый день месяца, <i>last_day_of_month</i> - последний день месяца</p>	<pre>first_day_of_month"/></pre>	
<p>hours</p>	<p>name1 и name2 - имена переменных, по которым будут доступны значения данного фильтра в коде bgbs-скрипта.</p>	<p>Выбор временного интервала "с - по".</p> <pre><hours name1="h1" name2="h2" /></pre>	
<p>month</p>	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта;</p> <p>title - название для панели с выбором месяца.</p>	<p>Выбор конкретного месяца в календаре.</p> <pre><month name="h1" title="Месяц"/></pre>	
<p>monthAndDays</p>	<p>name1 и name2 - имена переменных, по которым будут доступны значения данного фильтра в коде bgbs-скрипта;</p> <p>title - название для календаря.</p>	<p>Выбор интервала дней в определенном месяце</p> <pre><monthAndDays name1="h1" name2="h2" title="День месяца"/></pre>	
<p>period</p>	<p>name1 и name2 - имена переменных, по которым будут доступны значения данного фильтра в</p>	<p>Выбор периода с определенной даты до определенной даты.</p> <pre><period name1="h1" name2="h2" title="Период" default1="today" default2="</pre>	

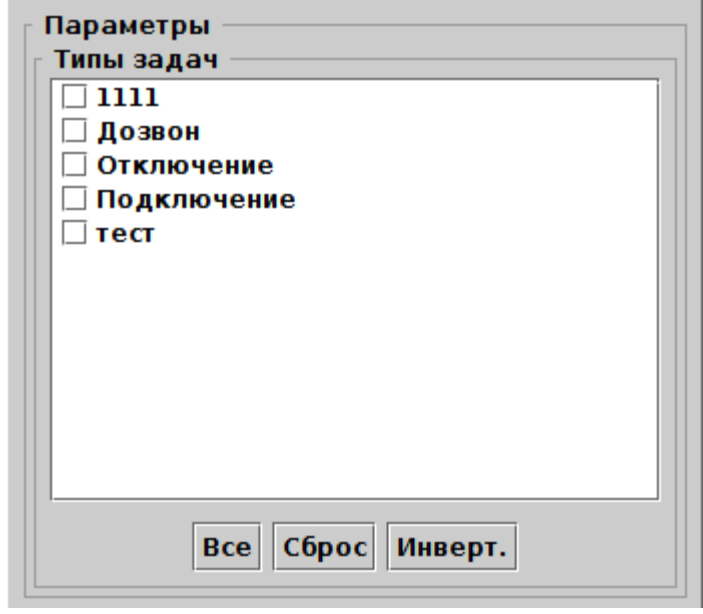
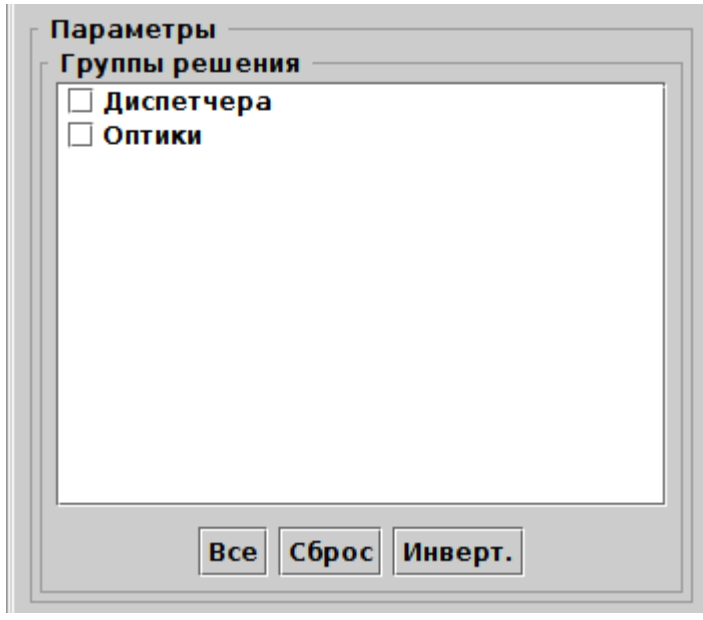
	<p>коде bgbs-скрипта;</p> <p>title - название для панели периода;</p> <p>default1 и default2 - значения по умолчанию для начала и конца периода.</p> <p>Возможны варианты:</p> <p><i>today</i> - сегодня, <i>yesterday</i> - вчера, <i>tomorrow</i> - завтра, <i>first_day_of_month</i> - первый день месяца, <i>last_day_of_month</i> - последний день месяца</p>	<pre>tomorrow "/></pre>	
<p>contract_groups</p>	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта;</p> <p>title - название для списка;</p> <p>select_name - имя переменной, по которой будет доступен выбранный режим "И/ИЛИ" в коде bgbs-скрипта в виде строки "and"/"or".</p>	<p>Список групп договоров. Возможность выбора режима И/ИЛИ появляется при указании в параметрах фильтра атрибута select_name.</p> <pre><contractGroups name="gr" title=" Группы" select_name="andOr"/></pre>	
<p>contract_labels</p>	<p>Дерево с метками договора</p>	<pre><contractLabels name=" contractLabels" title=" " /> id filter. getStringParam ("contractLabels")</pre>	
<p>services</p>	<p>Список услуг. Если данный фильтр будет добавлен в отчеты ядра, отобразится весь список услуг в системе, а иначе только того модуля, для которого строится отчет.</p>	<pre><services name="srv" title=" Услуги" /></pre>	

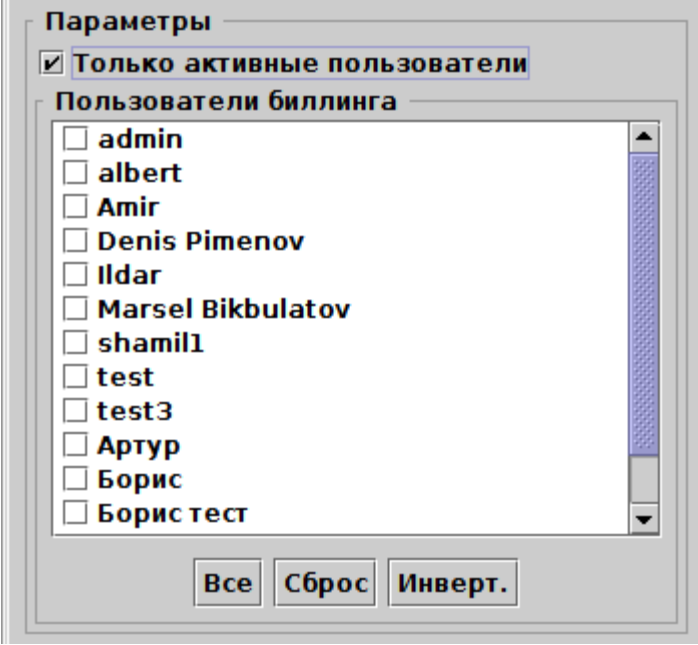
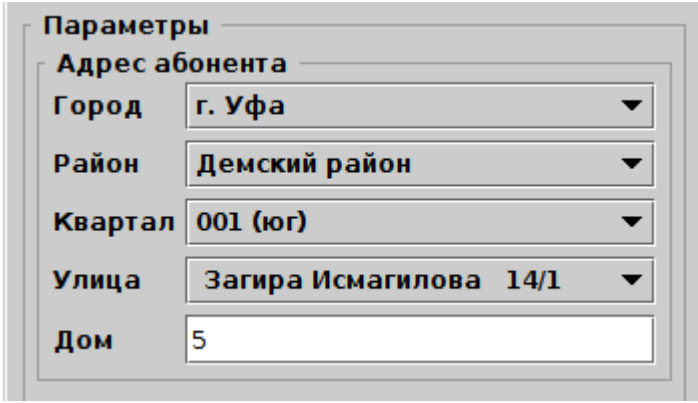
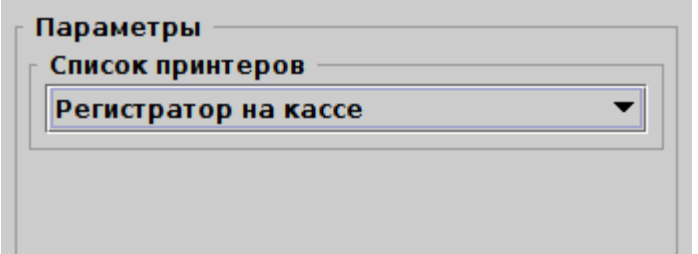
			
<p>na ses name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка nas через запятую;</p> <p>title - название для списка.</p>	<p>Список NAS модулей.</p>	<pre><nases name=" nas" title=" NAS" /></pre>	
<p>dealers name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка дилеров через запятую;</p> <p>title - название для списка.</p>	<p>Список дилеров.</p>	<pre><dealers name="d" title=" Дилеры" /></pre>	

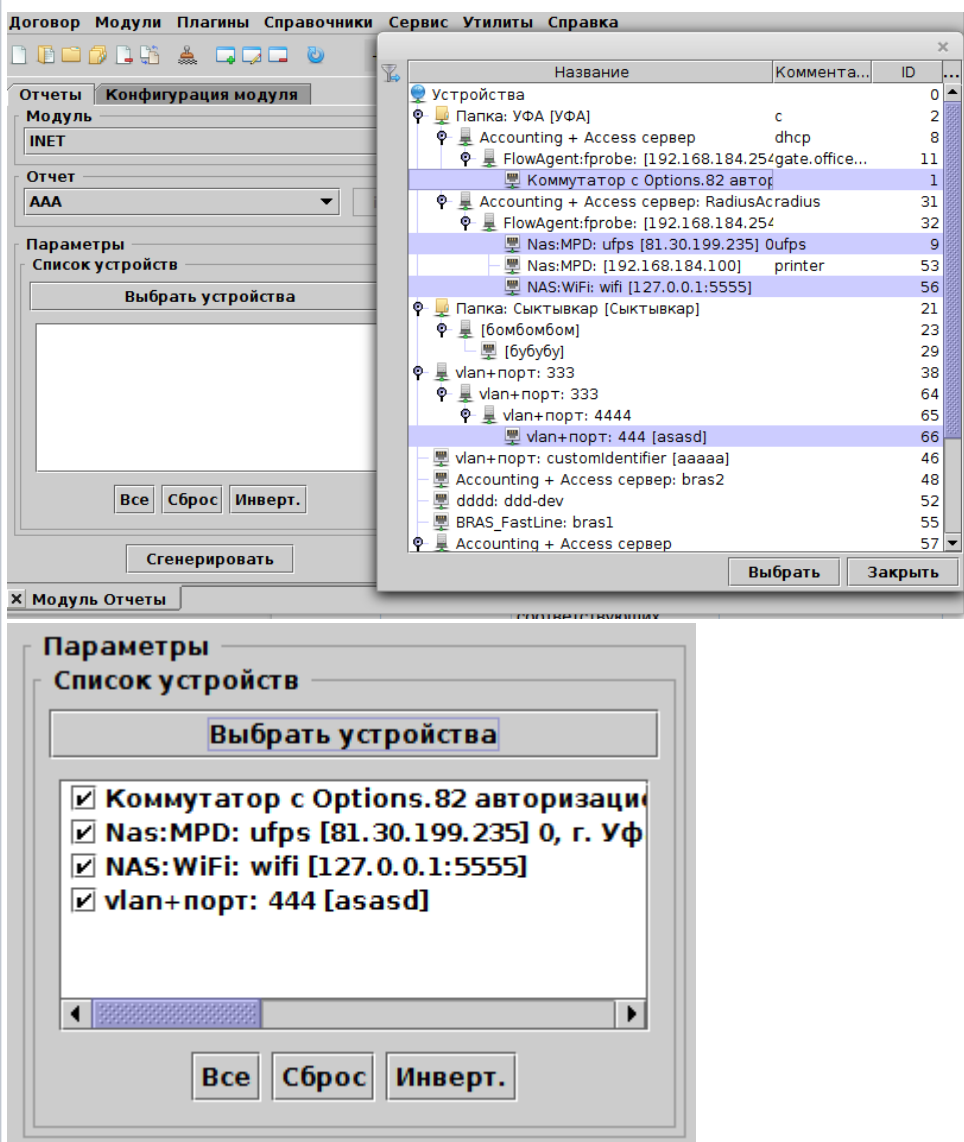
<p>paymentTypes</p>	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка типов платежей через запятую;</p> <p>title - название для списка.</p>	<p>Список типов платежей.</p>	<pre><paymentTypes name="pt" title=" Типы платежей" /></pre>	
<p>charges</p>	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка типов расходов через запятую;</p> <p>title - название для списка.</p>	<p>Список типов расходов.</p>	<pre><chargeTypes name="ct" title=" Типы расходов" /></pre>	
<p>tariffs</p>	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка тарифов через запятую;</p> <p>title - название для списка.</p>	<p>Список тарифов в системе.</p>	<pre><tariffs name="tariff" title=" Тарифы" /></pre>	

				
tarifoptions	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка тарифных опций через запятую;</p> <p>title - название для списка.</p>	Список тарифных опций.	<pre><tariffOptions name="tariffOptions" title="Тарифные опции" /></pre>	
tarifgroups	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка тарифных групп через запятую;</p> <p>title - название для списка.</p>	Список групп тарифов.	<pre><tariffGroups name="tariffGroups" title="Группы тарифов" /></pre>	
register	<p>name - имя переменной, по которой будет</p>	Список категорий проблем плагина CRM.	<pre><registerCategories name="rc" title="</pre>	

Cat eg or ies	<p>доступно значение данного фильтра в коде bgbs-скрипта в виде списка категорий через запятую;</p> <p>title - название для списка.</p>		Категории" />	
re gi st er Ex ec ut ors	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка исполнителей через запятую;</p> <p>title - название для списка;</p> <p>blank - добавлять ли пункт меню "Без исполнителей". Возможные значения: 1 - добавлять, 0 - не добавлять.</p>	Список исполнителей плагина CRM.	<registerExecutors name="re" title="Исполнители" blank="1" />	
re gi st er T as k T ype	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка категорий через запятую;</p> <p>title - название для списка.</p>	Список типов задач плагина CRM.	<registerTaskType name="rtt" title="Типы задач" />	

				
registerGroups	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка групп решения через запятую;</p> <p>title - название для списка.</p>	Список групп решения плагина CRM.	<pre><registerGroups name="rg" title=" Группы решения" /></pre>	
users	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка пользователей через запятую;</p> <p>title - название для списка.</p>	Список пользователей в системе с возможностью фильтрацией по активности.	<pre><users name=" users" title=" Пользовате ли биллинга" /></pre>	

			
address	<p>title - название для фильтра;</p> <p>cityName, streetName, areaName, quarterName - названия переменных, по которым будут доступны коды соответствующих адресных компонент;</p> <p>houseName - введенный номер дома.</p>	<p>Выбор адресных компонент: город, улица, район, квартал, дом.</p>	<pre><address cityName=" cn" streetName="sn" areaName="an" quarterName="cn" houseName="hn" title=" Адрес абонента" /></pre>
cashcheckKM	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде кода принтера;</p> <p>title - название для выпадающего списка.</p>	<p>Выбор одного из ККМ, установленных в модуле Cashcheck.</p>	<pre><cashcheck ККМ name=" kkm" title=" Список принтеров" /></pre>
inDevice	<p>name - имя переменной, по которой будет доступно значение данного фильтра в коде bgbs-скрипта в виде списка устройств через запятую;</p>	<p>Выбор устройств модуля Inet с фильтрацией.</p>	<pre><inetDevice s name=" dev" title=" Список устройств" /></pre>
			
			

<p>title - название для списка;</p> <p>mid - код экземпляра модуля Inet, для которого необходимо вывести устройства.</p>		
<p>checkbox</p> <p>name - имя переменной;</p> <p>title - название галки;</p> <p>default - значение по умолчанию (имеет смысл true для установки галочки заранее)</p>	<p>чекбокс</p>	<pre><checkbox name=" inverted" title=" Инвертировано" default=" true" /></pre>

Если фильтр большой и не помещается в высоту, то можно установить в элемент report атрибуты высоты и использования скроллинга (атрибуты **height** и **scroll** соответственно). По умолчанию элементы фильтра располагаются друг под другом, последними размещаются фильтры-вкладки. Однако, можно изменять положение элементов фильтра, их размер и занимаемое пространство. За это отвечают атрибуты **x**, **y**, **width**, **height**, **xweight**, **yweight**, **fill**, **anchor**. Эти атрибуты соответствуют параметрам размещения менеджера раскладки GridBagLayout, который используется в клиенте биллинга (параметры **gridx**, **gridy**, **gridwidth**, **gridheight**, **weightx**, **weighty**, **fill**, **anchor** соответственно). Подробнее можно почитать на [официальной странице документации Oracle](#), в [обучающем уроке от Oracle](#), на [русскоязычной странице одного пользователей интернет](#).

Код генерации отчёта имеет доступ к:

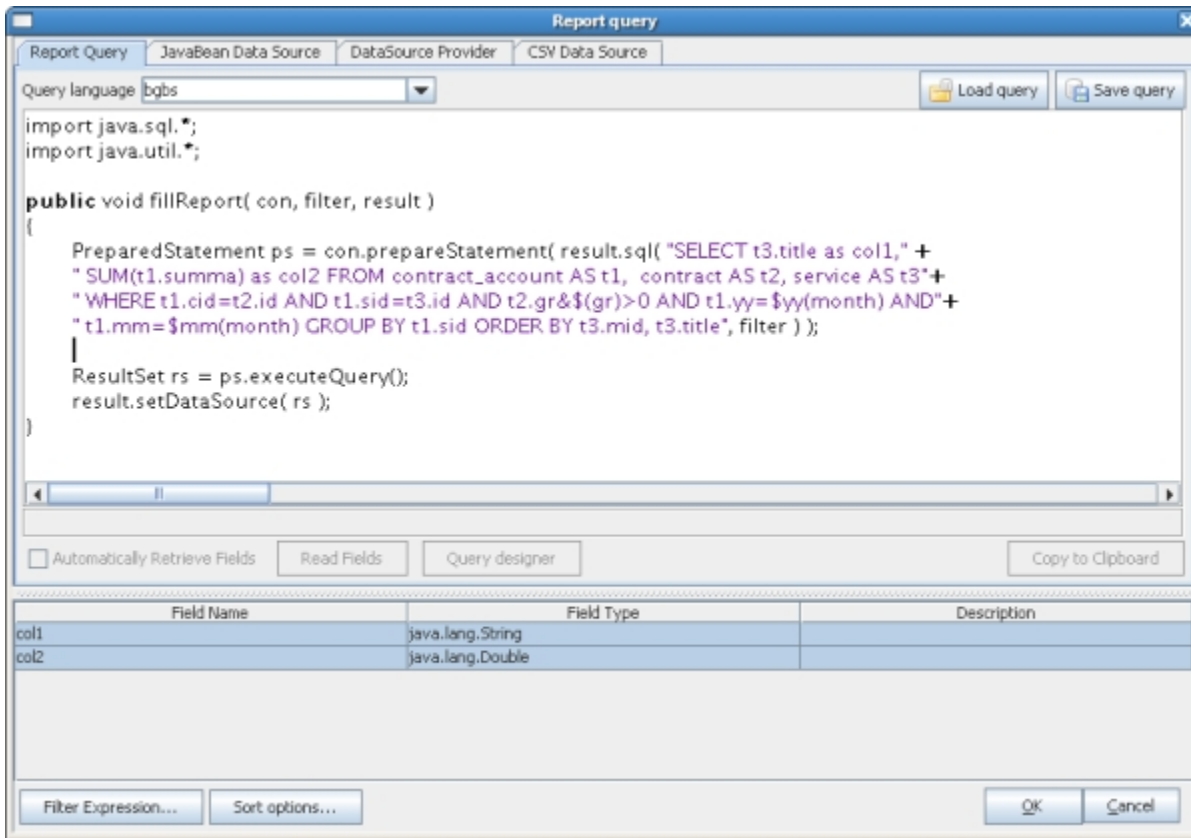
- значениям фильтра с использованием атрибута **name**;
- переменным конфигурации модуля, используя название переменной конфигурации, также можно использовать для передачи в отчёт кодов параметров и т.п.;

- переменной mid - коду текущего выбранного экземпляра модуля, для которого делается отчёт (0 - для ядра).

Отчёты JasperReports.

В iReport создайте шаблон и сохраните его в директорию reports как <модуль>_<id>.jrxml. Откройте его в iReport.

Для отображения отчёта необходимо получить данные (datasource). Получение данных описывается в шаблоне отчёта, для iReport - это пункт меню **Data->Report Query** (Запрос Отчёта).



На данный момент возможны два метода получения datasource - SQL и BGBS. В параметре Query language необходимо указать используемый.

SQL-метод используется для отчётов, данные которых можно получить одним запросом.

При этом значения фильтров, переменные конфигурации подставляются в запрос макросами:

\$(name) - подставляет текстовое значение, где name - имя фильтра, либо переменная конфигурации модуля;

\$mm(name) - подставляет значение месяца вида ММ, январь - 01;

\$yy(name) - подставляет год вида уууу;

\$date(name) - подставляет дату в формате sql;

\$time(name) - подставляет время в формате sql;

\$module_month_table(table, name) - подставляет имя таблицы в зависимости от текущего модуля, указанного имени таблицы и даты - параметра фильтра,

т.е \$module_month_table(log_session , date) , если фильтр date="01.01.2008" , а текущий модуль 1 подставит log_session_1_200801;

\$month_table - тоже самое без идентификатора модуля, т.е, например log_session_200801;

При построении запроса можно установить соединение с базой и, используя статические данные вместо макросов значений фильтров, проверить запрос и получить имена и типы полей отчёта. Промежуточные и итоговые значения (сумма, счётчик, максимум, минимум) можно указать для подсчёта в variables - параметр Calculation Type, а в Variable Expression указать макрос поля (например \${col2}). Если одного запроса для построения отчёта не достаточно? то можно воспользоваться скриптом - в Query language укажите bgbs, а в поле запроса должна быть конструкция вида

```
import java.sql.*;
import java.util.*;
```

```

public void fillReport( con, filter, result )
{
    // datasource

    result.setParams( params ); // , (Parameters) - Map
    result.setDataSource( datasource ); // datasource, Collection<Map<String, Object>>, ResultSet
}

```

con - объект класса java.sql.Connection - соединение с базой данных;

filter - объект класса bitel.billing.server.admin.reports.BGReportFilter, содержащий параметры фильтра, переменные конфигурации модуля отчётов. Краткий перечень доступных функций (полный - в API-документации для разработки скриптов BGBS):

```

String getStringParam( String name )

int getIntParam( String name )
int getIntParam( String name, int def )

long getLongParam( String name )
long getLongParam( String name, long def )

Calendar getCalendarParam( String name )
Date getDateParam( String name )

```

result - объект, в который необходимо передать параметры отчёта и datasource. У него имеются функции:

```

void setDataSource( JRDataSource dataSource )
void setDataSource( Collection<Map<String, Object>> dataSource )
void setDataSource( ResultSet dataSource )

void setParams( Map params )

JRDataSource createDataSource( ResultSet dataSource ) // datasource ResultSet
JRDataSource createDataSource( Collection<Map<String, Object>> dataSource ) // datasource Map

String sql( String sql, BGReportFilter filter ) //
// ( sql )

```

В параметрах (Parameters) отчёта должны быть определены параметры _filter и _months с соответствующими снимку экрана ниже типами:

JPA_ENTITY_MANAGER	javax.persistence.EntityManager	no	yes
JPA_QUERY_HINTS_MAP	java.util.Map	no	yes
MONDRIAN_CONNECTION	mondrian.olap.Connection	no	yes
_filter	bitel.billing.server.admin.reports.BGReportFilter	no	no
_months	java.util.Map	no	no

Пример шаблона отчёта "Отчет наработки по услугам":

Файл kernel_report.rep.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<report title=" ">
  <month name="month" title="" />
  <contractGroups name="gr" title=" " />
</report>

```

При использовании типа запроса SQL: Query language - SQL, сам запрос:

```

SELECT t3.title as col1, SUM(t1.summa) as col2 FROM contract_account AS t1, contract AS t2, service AS t3
WHERE t1.cid=t2.id AND t1.sid=t3.id
AND t2.gr&$(gr)>0
AND t1.yy=$yy(month) AND t1.mm=$mm(month)
GROUP BY t1.sid ORDER BY t3.mid, t3.title

```

Здесь макросы \$() будут заменены на соответствующие значения фильтра, т.е month и gr.

При использовании типа запроса BGBS: Query language - BGBS, запрос:

```

import java.sql.*;
import java.util.*;

public void fillReport( con, filter, result )
{
    PreparedStatement ps = con.prepareStatement( result.sql( "SELECT t3.title as col1," +
    " SUM(t1.summa) as col2 FROM contract_account AS t1, contract AS t2, service AS t3"+
    " WHERE t1.cid=t2.id AND t1.sid=t3.id AND t2.gr&$(gr)>0 AND t1.yy=$yy(month) AND"+
    " t1.mm=$mm(month) GROUP BY t1.sid ORDER BY t3.mid, t3.title", filter ) );

    ResultSet rs = ps.executeQuery();
    result.setDataSource( rs );
}

```

запрос использующий Collection<Map>:

Код

```

import java.sql.*;
import java.util.*;

public void fillReport( con, filter, result )
{
    PreparedStatement ps = con.prepareStatement( result.sql( "SELECT t3.title as col1," +
    " SUM(t1.summa) as col2 FROM contract_account AS t1, contract AS t2, service AS t3"+
    " WHERE t1.cid=t2.id AND t1.sid=t3.id AND t2.gr&$(gr)>0 AND t1.yy=$yy(month) AND"+
    " t1.mm=$mm(month) GROUP BY t1.sid ORDER BY t3.mid, t3.title", filter ) );

    ResultSet rs = ps.executeQuery();

    double total = 0;
    List res = new ArrayList();
    while(rs.next())
    {
        Map map = new HashMap();
        map.put("col1", rs.getString(1));
        double val = rs.getDouble(2);
        map.put("col2", val);
        total += val;

        res.add(map);
    }

    Map params = new HashMap();
    params.put( "total", total );

    result.setDataSource( res );
    result.setParams( params );
}

```

Для отображения в отчёте даты отчёта добавляем textfield с Expression Class: java.util.Date, Pattern: год: уууу месяц: МММММ, Expression: \$P {filter}.getDateParam("month").

Подсчёт итогового значения предоставим отчёту, добавив variable sum с Class Type java.lang.Double, Calculation Type Sum, Variable Expression \$F {col2}.

Добавим текстовое поле для его отображения, Expression Class java.lang.Double, Pattern # ##0.00, Text Field Expression \$V{sum}.

iReport вместе с вариантом данного примера: ftp://bgbilling.ru/pub/bgbilling/reports/iReport_1.3.2.zip

Другие примеры отчётов доступны на [Wiki](#). Наиболее простой способ разработки собственного отчёта - модификация существующего. Также некоторые отчёты, идущие в стандартной поставке (например, отчёт по должникам) могут быть изменены.

Имеется возможность в отчёте использовать гиперссылки. Это позволяет из одного отчёта быстро открывать другие связанные отчёты или разные сущности системы, например, договоры. Можно использовать отчёты как "универсальный поиск" - формируем любые фильтры и логику поиска, связываем строки результата с отрываемыми договорами и можно получить список договоров по любому критерию с возможностью открыть нужный. На данный момент поддерживаются следующие типы ссылок:

- **bgbilling:reports://bitel.billing.module.services.reports.BGReportsPanel?param1=value1¶m2=value2&...** для открытия любого другого отчёта. Для этого нужно знать соответствующие параметры метода Report модуля reports (код отчёта и т. д.).
- **bgbilling:reports://bitel.billing.module.contract.ContractEditor?<cid>** для открытия вкладки с соответствующим договором.

Можно указывать глобальные настройки для библиотеки jasperreports в файле jasperreports.properties, путь к которому можно указать с помощью параметра запуска сервера (server.sh/server.bat):

```
-Dnet.sf.jasperreports.properties=/path/jasperreports.properties
```

Значения опции вы можете узнать в [документации библиотеки jasperreports](#).

Табличные отчёты

Это отчёты, выполняемые с помощью скрипта. С возможностью сохранения в csv.

Фильтр отчёта настраивается так, как описано выше. Кроме того, нужно сделать следующие модификации: в файле *.rep.xml нужно поместить атрибут type=""java"":

```
<report title="Test" type="java">
  <address title="" cityName="cityId" streetName="streetId" houseName="house"/>
  <combo name="status" title="">
    <item id="-1" title=""/>
    <item id="0" title=""/>
    <item id="4" title=""/>
    <item id="3" title=""/>
  </combo>
  <fields>
    <item id="title" title=""/>
    <item id="comment" title=""/>
  </fields>
</report>
```

В теге fields указываются заголовки полей данного отчёта .

С верси 8.2201+ доступны дополнительные параметры: **align** (выравнивание данных в столбце); **wmin**, **wpref**, **wmax** (минимальная, предпочтительная, максимальная ширина столбца); **type** (тип данных в столбце)

```
<report title="Test" type="java" json="true|false">
<fields>
  <item id="title" title="" align="-1|0|1" wmin="-1|0|x" wpref="-1|0|y" wmax="-1|0|z" type="
string|integer|decimal|date"/>
  <item id="comment" title=""/>
</fields>
```

Создайте файл и сохраните его в директории reports как <модуль>_<id>.java. Пример файла :

```
public void fillReport( con, filter, ru.bitel.bgbilling.modules.reports.server.report.BGJsonReport.ReportResult
result )
{
  query = " SELECT * from contract ";
  query += " LIMIT " + ((pageIndex - 1) * pageSize )+ "," + pageSize;

  ps = con.prepareStatement( query );
  data = new ArrayList( 1000 );

  rs = ps.executeQuery();

  while( rs. next() )
  {
    title = rs.getString("title");
    comment = rs.getString("comment");

    map = new HashMap();
    map.put( "title", title );
    map.put( "comment", comment );
    data.add( map );
  }

  result.setData( data );
}
```

con - объект класса java.sql.Connection - соединение с базой данных;

filter - объект класса bitel.billing.server.admin.reports.BGReportFilter, содержащий параметры фильтра, переменные конфигурации модуля отчётов;

result - объект класса `bitel.billing.server.reports.BGCSVReport.ReportResult`, в который необходимо передать параметры отчёта и `datasource`;

pageSize - размер страницы;

pageIndex - номер страницы.

Пример внешнего вида отчёта :

The screenshot shows a web application interface with a sidebar on the left and a main content area on the right. The sidebar has tabs for 'Отчеты' and 'Конфигурация модуля'. Under 'Конфигурация модуля', there are sections for 'Модуль' (set to 'Основной модуль'), 'Отчет' (set to 'Список клиентов в CSV(Теленет)'), and 'Параметры'. The 'Параметры' section includes fields for 'Адрес', 'Город' (set to 'Любой'), 'Улица' (set to 'Любой'), 'Дом', and 'Статус' (set to 'любой'). The main content area has a toolbar with a save icon, navigation arrows, and a page indicator '3 из 18 [20]'. Below the toolbar is a table with columns: 'Имя', 'Комментарий', 'Адрес', and 'Телефон'. The table contains 18 rows of data, including client IDs, names, and phone numbers.

Имя	Комментарий	Адрес	Телефон
2800000014	ООО "АФРО"		36-29-7...
2800000015	ООО "АФРО"		
2800000016	Ханин Александр		
2800000017	ООО "Аудит"		
2800000021	ООО "Авиа-плюс"		36-39-0...
2800000028	ОАО ПИ "Братскэ...		
2800000693			
3087			
3800000007	Еловский Дмитр...		
3800000008	Ушакова Надежд...		
3800000009	Ивахненко Нина ...		
3800000010	Кафаров Альбер...		
3800000011	Кафаров Альбер...		35-27-8...
3800000012	Старокожко Мар...		
3800000013	Никитина Ирина ...		
3800000014	Панарин Виктор ...		
3800000015	Андрей Петров		
3800000016	Савченко Андрей		
444			
5487			

С помощью кнопки "сохранить" можно сохранить данные отчёта в csv-файл.

Примеры отчётов доступны на [Wiki](#).

Есть возможность спрятать столбец в итоговой таблице, для этого надо в файле с описанием фильтров его значение `title` предворить символом '#' или с версии 8.2201+ задать нулевые значения параметров `wmin`, `wpref` и `wmax`:

```
<fields>
...
<item id="cid" title="#cid"/>
...
```

После этого столбец станет невидимым (и его настоящее заглавие останется с решёткой). Это может понадобиться, например, для нижеописанной возможности.

При клике на строки таблицы могут выполняться некоторые действия в зависимости от содержимого таблицы и содержимого в каждой строке. В данный момент имеются следующие возможности:

- Если в таблице присутствует столбец с заголовком `cid` (или же невидимый `#cid`), то при клике на строку откроется соответствующий содержимому столбца `cid` договор. Как и в случае с `jasper`-отчётами, это можно использовать для организации произвольного поиска договоров.

Мобильные отчеты

Модуль отчетов позволяет просматривать отчеты на мобильных устройствах(пока только для ios) в виде интерактивных графиков. Где бы вы не находились вы сможете получать свежую информацию по состоянию дел вашего биллинга, соотносить данные на разных промежутках времени и анализировать их. Например кол-во приходов и их сумма, кол-во и активность абонентов и в обще любую информацию которую вы хотите визуализировать в виде графиков, диаграмм и т.д.(с помощью дин. классов).

Для установки программы перейдите по [ссылке](#) в магазин приложений Apple. Если вы это сделаете с мобильного устройства у вас автоматически запустится App Store, иначе iTunes, и приобрести бесплатно приложение. Если вы это сделали через iTunes, то вам надо будет в мобильном устройстве зайти в историю ваших покупок и установить от туда, либо синхронизировать с iTunes.

Настройка сервера

Для обмена данными между приложением и сервером вы должны открыть доступ к reportsexecuter и предоставить пользователям приложения адрес который они должны вводить в приложении BGBilling Top Reports. Для работы приложения сервер должен знать код модуля отчетов, его можно передать как в запросе к серверу (../reportsexecuter?mid=12) или для удобства прописать его в настройках сервера(**Сервис ->Настройка -> Конфигурация**) под параметром **reports.defaultCodeModuleForMobileReports** где в качестве значения будет код модуля отчетов. При этом возможность указания кода модуля при запросе сохраниться и будет иметь более высокий приоритет.

Далее вам следует настроить модуль отчетов. Для этого пропишите метод входа пользователей под параметром **reports.mobileLoggingMethod**, где в качестве значения можете использовать 1 или 2. Соответственно 1 - это свободный вход, 2 - вход по логину/паролю пользователя BGBilling

```
reports.mobileLoggingMethod=2
```

Рекомендуется использование защищенного протокола https. Приложение поддерживает работу с само подписанными сертификатами.

Собственные отчеты.

Вы можете создавать собственные отчеты для приложения BGBilling Top Reports. Для этого создайте дин. класс потомок от MobileReport

В методе getReportType вам нужно передать параметры отчета по которым пользователь сможет производить выборку. В getData вам нужно вернуть данные для построения отчета. В getTypeGraf вам нужно вернуть тип графика который будет строиться по вашим данным(гистограмма, линейный график, круговой график или табличный отчет). В методе getOptions можно вернуть дополнительные данные которые будут отображаться при построении графика или указывать способ отображения. Ниже примеры отчетов.

Круговой отчет

Код

```
import java.awt.Color; import
import java.awt.Color;
import java.sql.Connection;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import ru.bitel.bgbilling.common.BGIllegalArgumentException;
import bitel.billing.server.reports.mobile.MobileParamType;
import bitel.billing.server.reports.mobile.MobileReport;
import bitel.billing.server.reports.mobile.MobileReportType;
import bitel.billing.server.reports.mobile.Slice;

public class Test2 extends MobileReport
{
    @Override
    public MobileReportType getReportType( Connection con )
    {
        List<MobileParamType> list = new ArrayList<>();
        //      .
        //
        list.add( new MobileParamType( "Date1", MobileParamType.DateType, " ", "11.12.2010" ) );
        list.add( new MobileParamType( "Date2", MobileParamType.DateType, " ", "03.03.2014" ) );
        list.add( new MobileParamType( "flag1", MobileParamType.BooleanType, " ", "false" ) );
        return new MobileReportType( "Test2", list );
    }

    @Override
    public List<Object> getData()
        throws BGIllegalArgumentException
    {
        ArrayList<Object> list = new ArrayList<>();
        //      .      Slice.
        // : 1- , 2 - , 3 - -.
        list.add( new Slice( "", new Color( 101, 214, 86 ), 23 ) );//
        list.add( new Slice( "", getColor(), 34 ) );//
        list.add( new Slice( "", getColor(2), 45 ) );//
        grafType = PieChartType;//      (PieChartType,HistogramChartType,LinearChartType,TableType)
        return list;
    }

    @Override
    public Map<String, Object> getOptions()
    {
        return null;//      .
    }
}
```

Гистограммный отчет.

Код

```

public class Test1 extends MobileReport
{
    @Override
    public MobileReportType getReportType( Connection con )
    {
        List<MobileParamType> list = new ArrayList<>();
        list.add( new MobileParamType( "flag1", MobileParamType.BooleanType, " ", "false" ) );
        list.add( new MobileParamType( "Date1", MobileParamType.DateType, " ", "01.09.2014" ) );
        list.add( new MobileParamType( "Date2", MobileParamType.DateType, " ", "07.09.2014" ) );
        return new MobileReportType( "( )", list );
    }

    @Override
    public List<Object> getData() throws BGIllegalArgumentException
    {
        grafType = HistogramChartType;
        try
        {
            boolean flag1 = getBooleanParameter( "flag1", false );

            Date date1 = getDateParameter( "Date1", new Date() );// .
            Date date2 = getDateParameter( "Date2", new Date() );//
            if( date1 == null || date2 == null && TimeUtils.dateBefore( date1, date2 ) )
            {
                throw new BGIllegalArgumentException();
            }

            StringBuilder query = new StringBuilder( "SELECT SUM(summa), DATE_FORMAT(dt, ' ' );
            String periodParam = getParameter( "period", "w" );
            if( periodParam != null )
            {
                String format = "";
                Calendar time = Calendar.getInstance();
                flag1 = true;
                switch ( periodParam )
                {
                    case "w":
                        time.add( Calendar.DAY_OF_YEAR, -7 );
                        format = "%d.%m";
                        break;
                    case "m":
                        time.add( Calendar.MONTH, -1 );
                        format = "%d.%m.%y";
                        break;
                    case "q":
                        time.add( Calendar.MONTH, -3 );
                        format = "%d.%m.%y";
                        break;
                    case "y":
                        flag1 = false;
                        time.add( Calendar.YEAR, -1 );
                        format = "%m.%Y";
                        break;
                }
                query.append( format );
                date1 = time.getTime();
                date2 = new Date();
            }
            else
            {
                query.append( flag1 ? "%d.%m" : "%m %Y" );// , .
            }
            query.append( " ), COUNT(id) FROM contract_payment WHERE dt>=? AND dt<? GROUP BY YEAR(dt),MONTH
(dt)" );

            query.append( flag1 ? ",DAY(dt)" : "" );
            query.append( " ORDER BY dt ASC" );
            PreparedStatement ps = con.prepareStatement( query.toString() );// .

            ps.setDate( 1, TimeUtils.convertDateToSqlDate( date1 ) );
            ps.setDate( 2, TimeUtils.convertDateToSqlDate( date2 ) );

```

```

        ArrayList<Object> dataList = new ArrayList<>();
        ResultSet rs = ps.executeQuery();
        int count = 1;
        while( rs.next() )
        {
            dataList.add( new Slice( rs.getString( 2 ), getColor(), rs.getDouble( 1 ) ) );
        }
        return dataList;
    }
    catch( SQLException e )
    {
        e.printStackTrace();
    }
    return null;
}

@Override
public Map<String, Object> getOptions()
{
    //
    //
    // id - ( - getParameter( "period", "w" ) ).
    // values - / , .
    // titles - .
    HashMap<String, Object> map = new HashMap<>(), dateMap = new HashMap<>();
    dateMap.put( "id", "period" );
    dateMap.put( "values", "w,m,q,y" );
    dateMap.put( "titles", ",,," );
    //
    // dataToolbar.
    map.put( "dataToolbar", dateMap );
    //
    // changeGrafHide, .
    //
    map.put( "changeGrafHide", "1" );
    return map;
}
}

```

Линейный график.

Код

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import ru.bitel.bgbilling.common.BGIllegalArgumentException;
import bitel.billing.common.TimeUtils;
import bitel.billing.server.reports.mobile.MobileParamType;
import bitel.billing.server.reports.mobile.MobileReport;
import bitel.billing.server.reports.mobile.MobileReportType;

public class PaymentsMobileReport extends MobileReport
{
    private ArrayList<String> descriptionList = null;
    private ArrayList<String> titleList = null;

    @Override
    public MobileReportType getReportType( Connection con )
    {
        List<MobileParamType> list = new ArrayList<>();
        list.add( new MobileParamType( "Date1", MobileParamType.DateType, " ", "11.12.2010" ) );
        list.add( new MobileParamType( "Date2", MobileParamType.DateType, " ", "03.03.2014" ) );
    }
}

```

```

        list.add( new MobileParamType( "flag1", MobileParamType.BooleanType, " ", "false" ) );
        return new MobileReportType( "( )", list );
    }

    @Override
    public List<Object> getData() throws BGIllegalArgumentException
    {
        //      double .      .
        grafType = LinearChartType;
        try
        {
            boolean flag1 = getBooleanParameter( "flag1", false );
            StringBuilder query = new StringBuilder( "SELECT SUM(summa), DATE_FORMAT(dt, ' ' );
            query.append( flag1 ? "%d.%m" : "%m %Y" );//      ,      .
            query.append( "''), COUNT(id) FROM contract_payment WHERE dt>=? AND dt<? GROUP BY YEAR(dt),
MONTH(dt)" );

            query.append( flag1 ? ",DAY(dt)" : "" );
            query.append( " ORDER BY dt ASC" );
            PreparedStatement ps = con.prepareStatement( query.toString() );

            Date datel = getDateParameter( "Date1", new Date() );
            Date date2 = getDateParameter( "Date2", new Date() );
            if( datel == null || date2 == null && TimeUtils.dateBefore( datel, date2 ) )
            {
                throw new BGIllegalArgumentException();
            }
            ps.setDate( 1, TimeUtils.convertDateToSqlDate( datel ) );
            ps.setDate( 2, TimeUtils.convertDateToSqlDate( date2 ) );

            ArrayList<Object> dataList = new ArrayList<>();
            titleList = new ArrayList<>();
            descriptionList = new ArrayList<>();
            ResultSet rs = ps.executeQuery();
            while( rs.next() )
            {
                dataList.add( rs.getDouble( 1 ) );
                titleList.add( rs.getString( 2 ) );//
                descriptionList.add( "- : " + rs.getInt( 3 ) );//      .
            }
            return dataList;
        }
        catch( SQLException e )
        {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    public Map>String, Object> getOptions()
    {
        HashMap<String, Object> map = new HashMap<>();
        //
        if( descriptionList != null && descriptionList.size() > 0 )
            map.put( "description", descriptionList );
        //      description titles .
        if( titleList != null && titleList.size() > 0 )
            map.put( "titles", titleList );
        return map.size() > 0 ? map : null;
    }
}

```

Табличный отчет.

Код

@Override

```

public List<Object> getData() throws BGIllegalArgumentException
{
    //
    //      ,      3 ,      3.
    grafType = TableType;
    ArrayList<Object> list = new ArrayList<>();
    ArrayList<String> rowList1 = new ArrayList<>();
    ArrayList<String> rowList2 = new ArrayList<>();
    rowList1.add(" 1 1");
    rowList1.add(" 2 1");
    rowList1.add(" 3 1");
    rowList2.add(" 1 2");
    rowList2.add(" 2 2");
    rowList2.add(" 3 2");
    list.add(rowList1);
    list.add(rowList2);
    return list;
}

@Override
public Map<String, Object> getOptions()
{
    // titles
    HashMap<String, Object> map = new HashMap<>();
    ArrayList<String> titles = new ArrayList<>();
    titles.add(" 1 ");
    titles.add(" 2 ");
    titles.add(" 3 ");
    map.put( "titles", titles );
    return map;
}

```

Рассмотрим параметры отчетов, по которым пользователь может регулировать выборку данных. Существует 5 типов параметров: Дата (DateType), Список(ListType), Число(NumberType), Список с возможностью выбрать несколько значений(MultiSelectListType), Логический (BooleanType).Ниже приведены примеры их создания в методе getReportType.

Код

```

List<MobileParamType> list = new ArrayList<>();
//
Calendar calendar = Calendar.getInstance();
list.add( new MobileParamType( "d1", MobileParamType.DateType, " ", TimeUtils.format( calendar, TimeUtils.
DATE_FORMAT_PATTERN_DDMYYYY ) ) );
//
LinkedHashMap<Integer, String> map = new LinkedHashMap<>();
map.put( 1, "10" );
map.put( 2, "100" );
map.put( 3, "1000" );
list.add( new MobileParamType( "limit", MobileParamType.ListType, ". .", "3", map ) );
//
list.add( new MobileParamType( "Num1", MobileParamType.NumberType, " ", "50" ) );
//
LinkedHashMap<Integer, String> fields = new LinkedHashMap<>();
fields.put( 1, "" );
fields.put( 2, "" );
fields.put( 3, "" );
fields.put( 4, "" );
fields.put( 5, "" );
fields.put( 6, " " );
fields.put( 7, "" );
list.add( new MobileParamType( "fields", MobileParamType.MultiSelectListType, "", "1,2,3,4,5,6", fields ) );
//
list.add( new MobileParamType( "flag1", MobileParamType.BooleanType, " ", "false" ) );

```

После создания класса отчета вам необходимо прописать ваш класс в настройках вашего модуля.

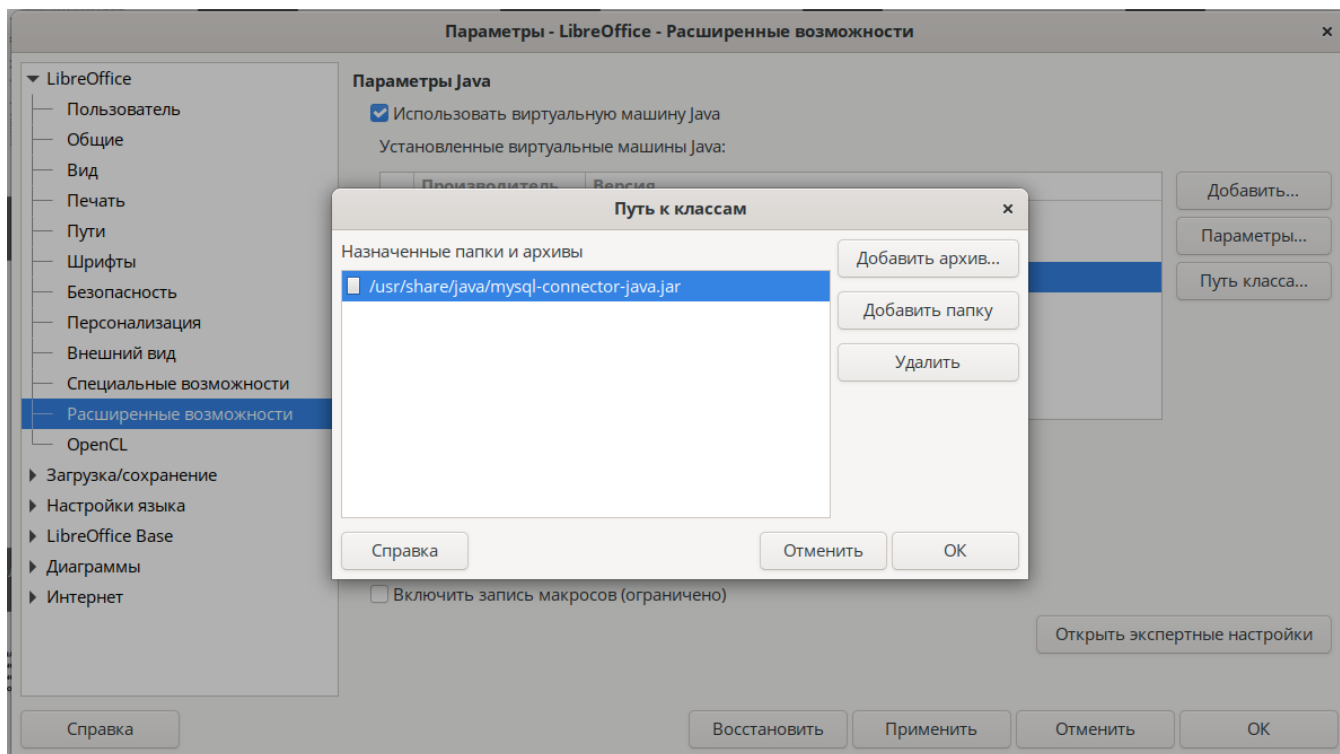

```
reports.mobile.dinamic.1.title=  
reports.mobile.dinamic.1.classes=mobileReports.Test2,mobileReports.Test1
```

В параметре `reports.mobile.dinamic.1.title` должно содержаться название категории отчетов, а в `reports.mobile.dinamic.1.classes` через запятую полное название ваших классов. Если вы решите добавить еще одну категорию, то измените 1 на 2 и т.д.

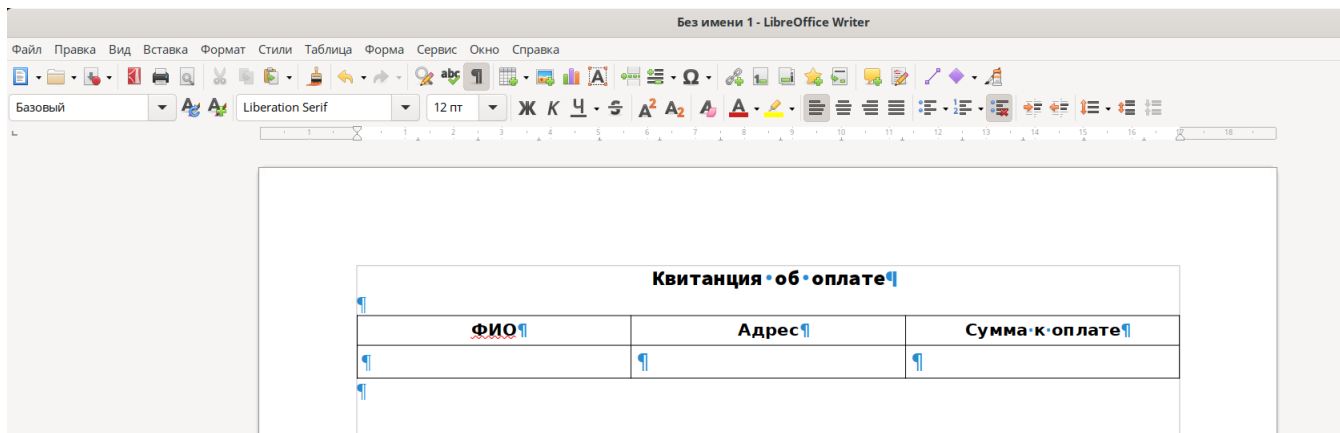
Печать квитанций из LibreOffice

Для массовой печати например квитанций можно использовать функционал слияния документов из офисного пакета LibreOffice (OpenOffice)

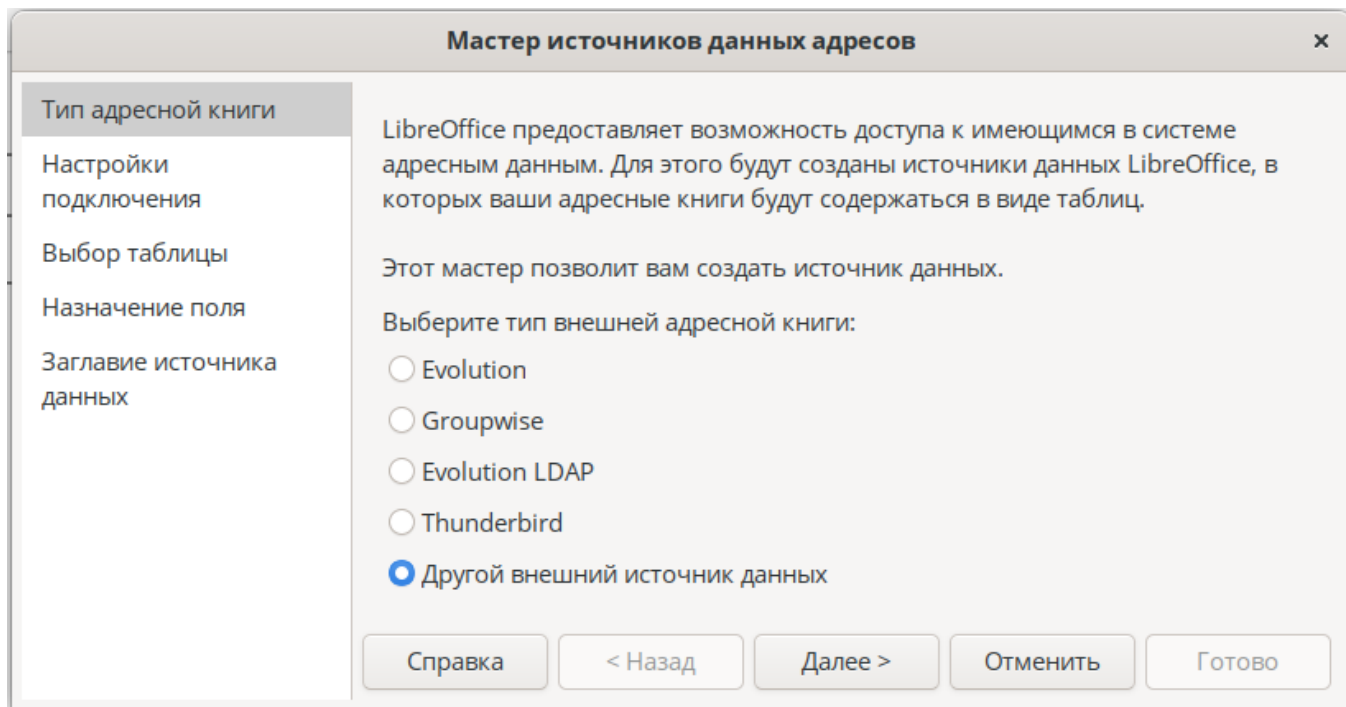
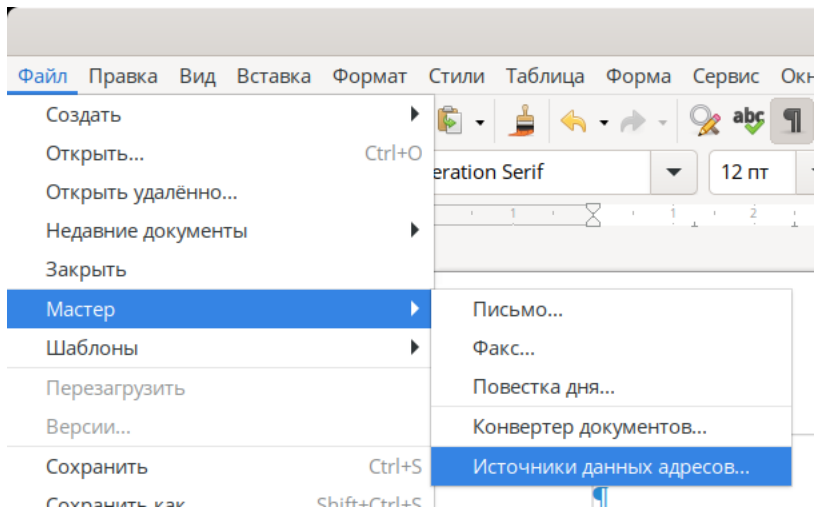
Для этого для получения данных из биллинга можно подключиться к БД биллинга используя jdbc драйвер. В зависимости на какой ОС будет запускать офис, драйвер можно установить из репозитория (Linux (`dnf install mysql-connector-java.noarch`)) или загрузить архив с сайта <https://dev.mysql.com/downloads/connector/j/> (Windows, Linux). Распаковать архив и далее в офисе указать путь к Jar файлу.



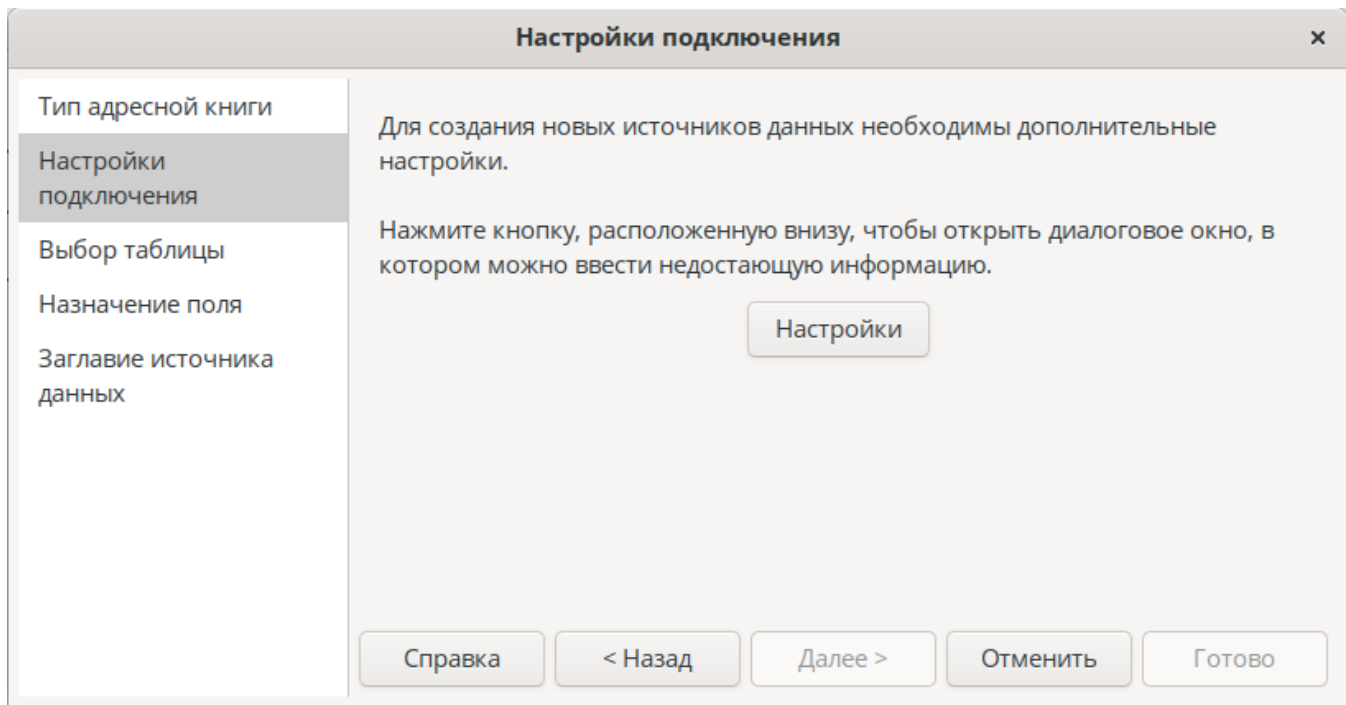
Далее создаем шаблон квитанции или используем созданный ранее



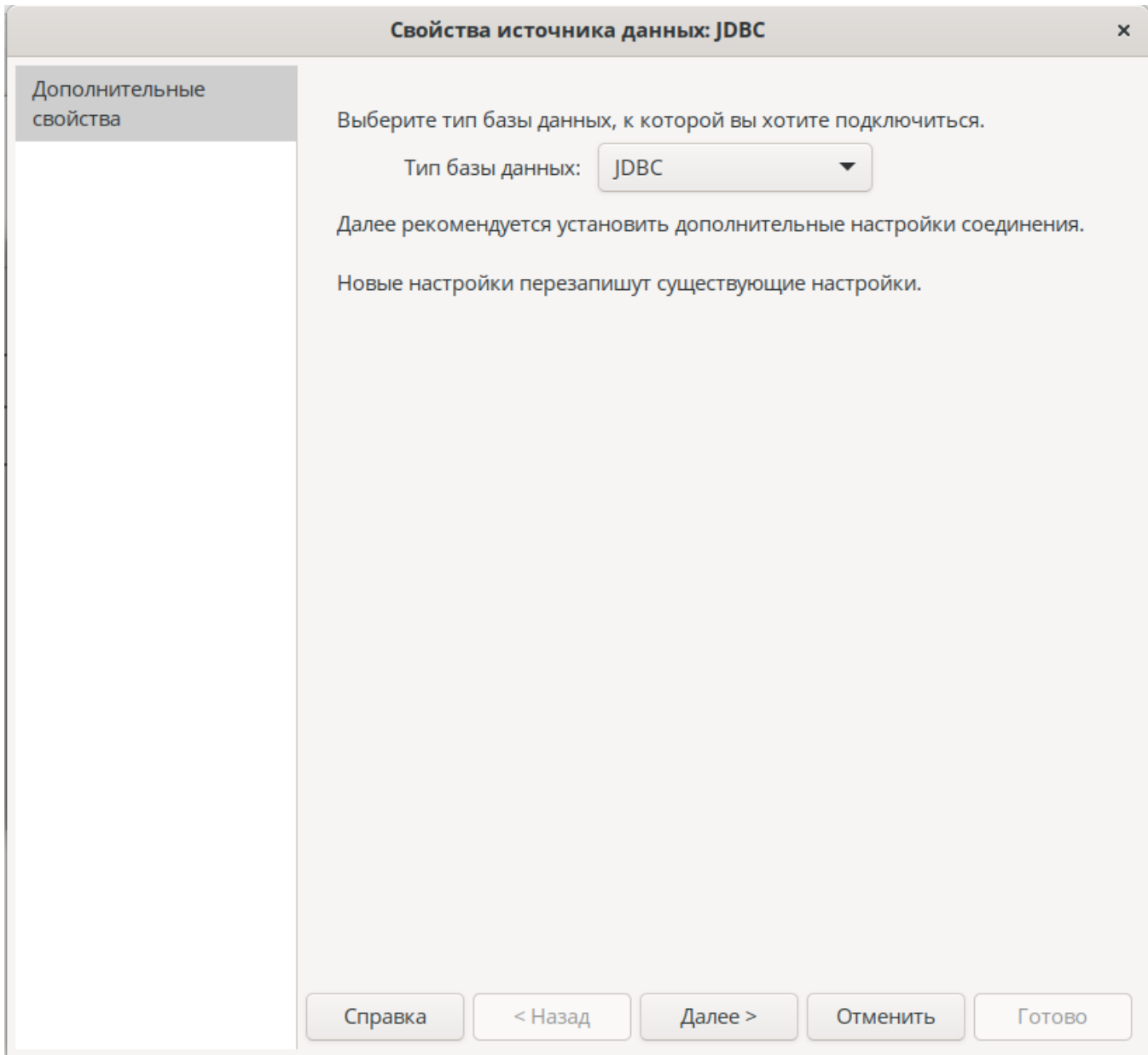
Далее подключаем источник данных для заполнения шаблона



Выбираем "Другой внешний источник данных" и жмем **Далее**



Жмем "Настройки"



Выбираем Тип базы данных "JDBC", жмем **Далее**

Настройки подключения ×

Дополнительные свойства

Настройки подключения

Общие

URL источника данных (т.е. host=\$host:\$port dbname=\$database)

jdbc:

Аутентификация пользователя

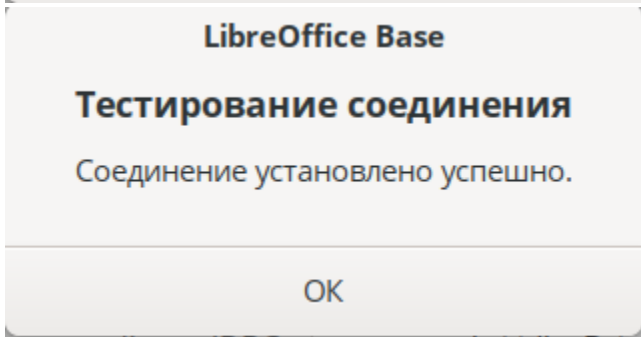
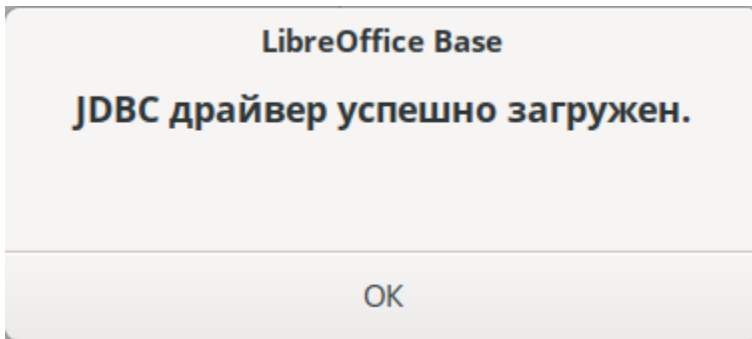
Имя пользователя:

Требуется пароль

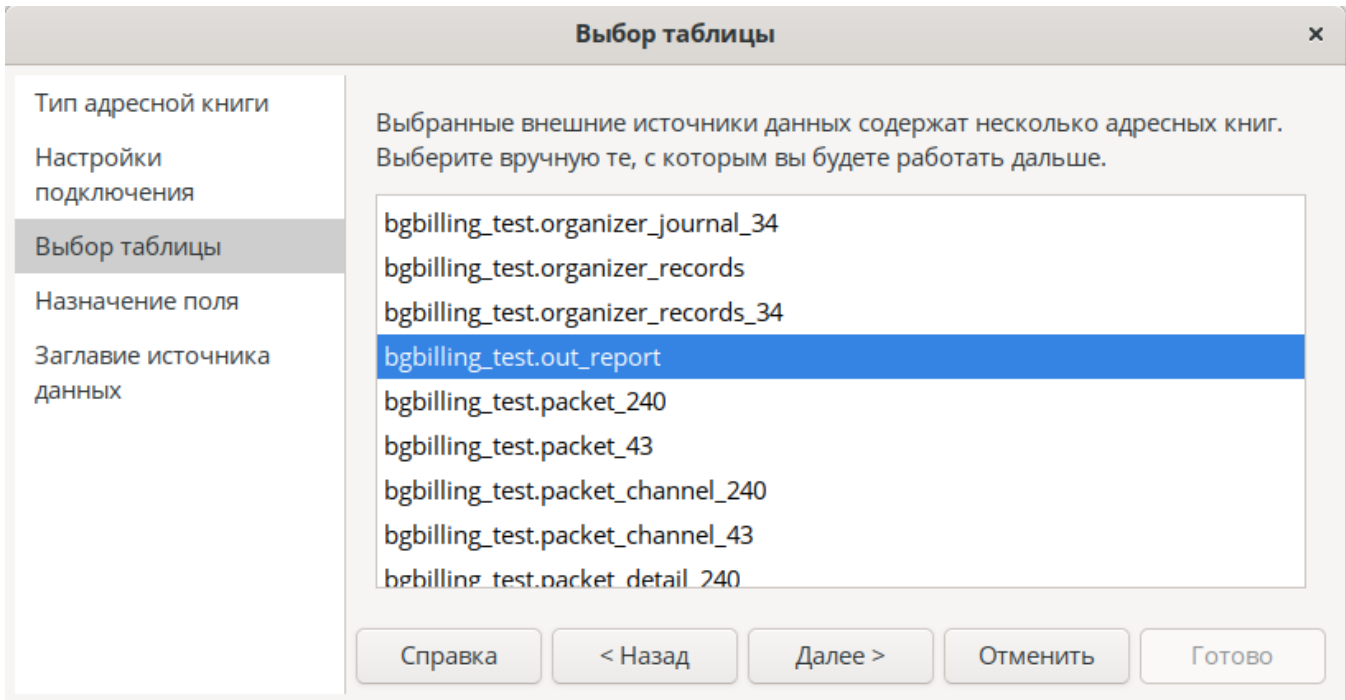
Свойства JDBC

Класс драйвера JDBC:

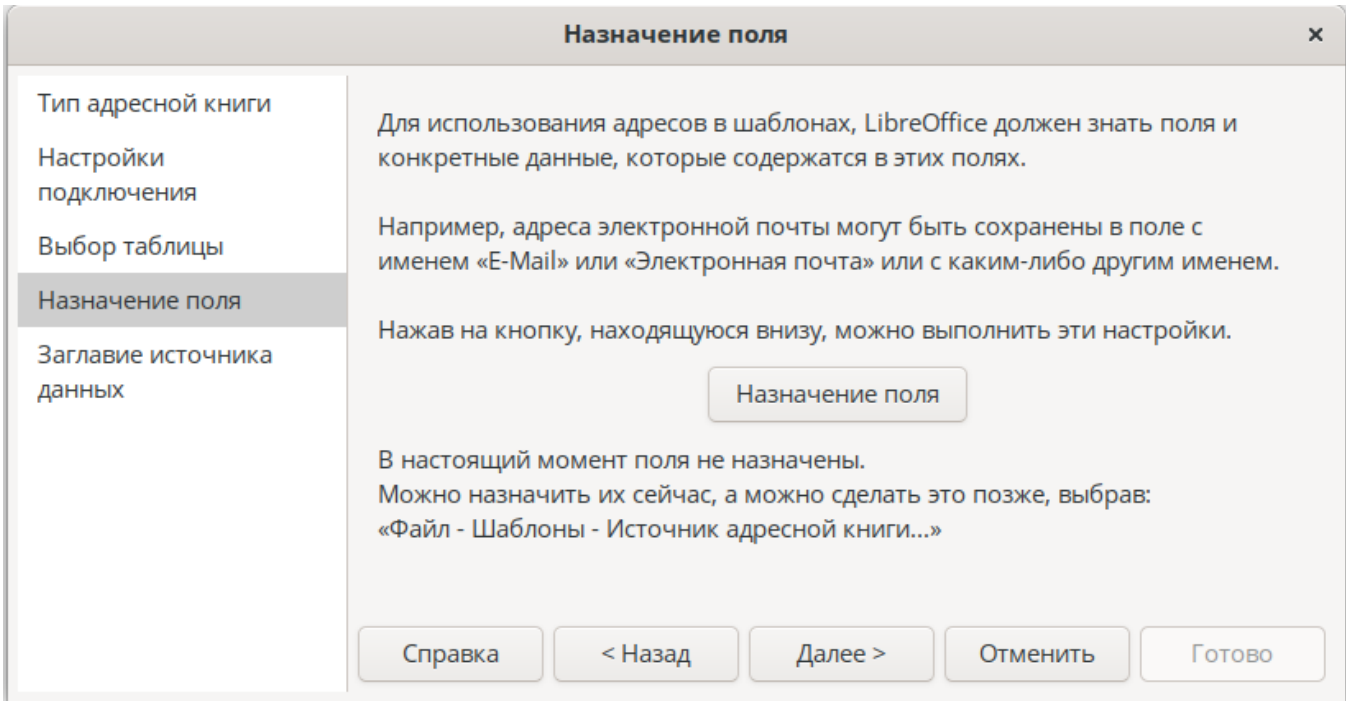
Вводим адрес mysql mysql сервер, порт и название БД, указываем пользователя и название драйвера "**com.mysql.cj.jdbc.Driver**", нажимаем на **Тест класса** и **Тест подключения**



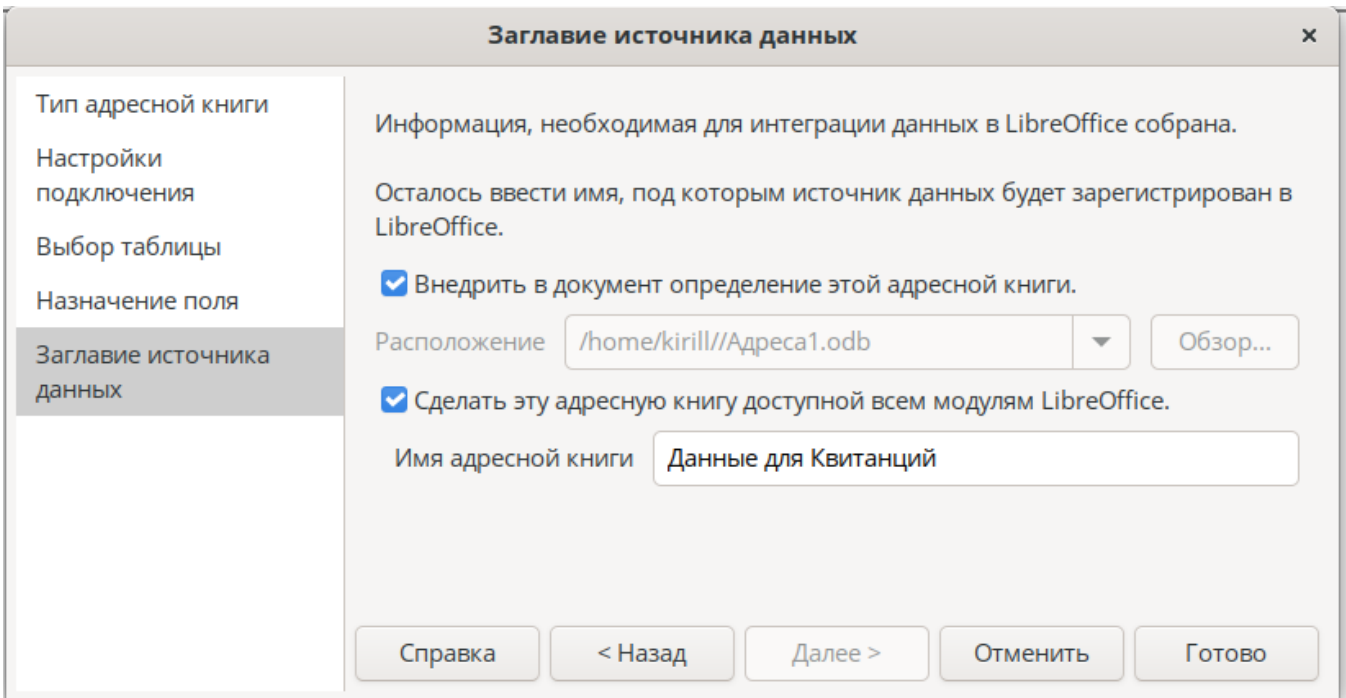
Нажимаем "**Готово**" Ждем загрузки данных из БД (если у вас в БД много таблиц может зависнуть на минуту-другую)



Выделяем нужную таблицу (как ее создать описано ниже), нажимаем **Далее**



Нажимаем **Далее**



Указываем **Имя адресной книги** и нажимаем **Готово**

Модуль Mobile

Назначение модуля

Модуль предназначен для работы с мобильным приложением на [iOS](#) и [Android](#).

Настройка модуля

Установите модуль на сервер, используя скрипт **update.sh**, обновите клиент биллинга. Затем создайте экземпляр модуля, назвав его произвольным образом (например, **Mobile**).

Создайте конфигурацию модуля и установите ее. В конфигурации обязательно должен быть параметр **token.signature.key**, его значением подписывается токен, его минимальная длина 32 байта, от него будет зависеть какой алгоритм будет использован (HMAC 256/384/512 итд), размерность можете посчитать в любом online калькуляторе ([например](#))

Далее вам нужно будет предоставить приложению доступ до модуля, например из вашей сети он может выглядеть так <http://12.34.56.100:8080/bgbilling/mobapi/13> (где 13 код модуля), но так как вы скорее всего находитесь за nginx-ом, то нужно будет настроить путь с наружи (например - <https://provider.ru/mobile>)

Со всеми параметрами конфигурации и их описанием вы можете ознакомиться в шаблоне модуля, а ниже представлены некоторые параметры с описанием:

Конфигурация

```
# . . . . . ( ) , . . . . .
token.signature.key=d%°@°°...`°@21
#
# - , - , ( 1, 10, 2 )
token.max.active=2
# ( 1, 180, 30 )
token.refresh.lifetime.days=30
# ( ) . , , . ( 10, 1440(2) , 60 )
token.access.lifetime.minutes=10
#
# . [title= ].
#menu.items=main; balance[title=]; tariff; tariff_options; limit; status; helpDesk; notification; help; exit
#
# push ( )
#push.url=
#push.login=
#push.pswd=
#
# ( )
# Mobile , .
showSubContracts=true
#
# help. web , , email.
#about.items=url[value=https://bgbilling.ru/mobile_terms_and_conditions.html, title=U+1F310 ] ; email
[value=support@test.ru, title=U+1F4E7 , subject= :@contractTitle] ; phone[value=+79171234567, title=U+260E
+79171234567]
#
#payment.modules=yamoney[moduleId=157,title=.];sberdeeplink[title= - ];vepay[moduleId=2,title=VePay,minSum=1,
maxSum=15000];sberbank[moduleId=23,minSum=1,maxSum=5000,title= ]
#####
#
#push.notification.payment=true
#
#push.notification.payment.allow.ids=1,2,4,5
# : @contractTitle - , @sum - , @balance -
#push.notification.payment.text= @contractTitle @sum. : @balance.

# / .
#push.notification.charge=true
# / , ,
#push.notification.charge.allow.ids=1,2,3
# : @contractTitle - , @sum - , @balance -
#push.notification.charge.text= @contractTitle @sum. : @balance.
```

```

#push.notification.payback.text= @contractTitle @sum. : @balance.

#
#push.notification.status.change=true
# , ,
#push.notification.status.change.allow.codes=1,2,3,4,5,6
# , , . 0 - ( , ). onlyManagers - ( > 0)
#push.notification.status.change.allow.users=0,1,2,3
# : @contractTitle - , @balance - , @status - .
#push.notification.status.change.text= @contractTitle: - @status.
#
#push.notification.status.change.<status_code>.text=

# ( , , , HelpDesk) ( ->-> - ). ,
#push.notification.allow.groupMask

```

Авторизация в мобильном приложении

До версии 8.2110 авторизация доступна только по номеру договора и паролю

С версии 8.2110+ авторизация использует параметр **web.auth.modes** заданный в конфиге ядра (используются только авторизации относящиеся к ядру, авторизации по модулям игнорируются)

Платежи

Приложение позволяет пополнение баланса договора. Для этого должен быть один из поддерживающих оплату в моб. приложении модуль и настроен параметр `payment.modules`.

В параметре `payment.modules` через точку с запятой можно указать несколько способов оплаты и в квадратных скобках параметры для каждого из них, такие как `title` - название, `moduleId` - код экземпляра модуля, `minSum` - минимальная сумма, `maxSum` - максимальная сумма. Параметры `minSum` и `maxSum` не переопределяют соответствующие параметры в самих модулях, а лишь добавляют дополнительные ограничения при оплате в моб. приложении.

Если договор не имеет экземпляра платежного модуля указанный в `payment.modules`, то ему не будет предложен вариант такой оплаты.

Доступные способы(модули), в скобках указан их ключ для параметра :

- Яндекс.Касса(yamoney) - поддерживает оплату банковской картой, кошельком, Apple Pay, Google Pay и сбербанк онлайн.
- Paymaster(paymentmaster)
- Сбербанк(sberbank) - банковской картой, Apple Pay и в приложении Сбербанка.
- VePay(vepay)
- Moneta.ru(moneta) с версии 8.2110+
- Диплинк на приложение сбербанка(sberdeeplink) - позволяет перенаправить абонента на оплату в приложение сбербанка(за диплинком обращайтесь к сбербанку)

Ошибки

"Доступ к мобильному ЛК ограничен, обратитесь к тех. поддержке провайдера":

- договор не содержит экземпляра модуля `mobile`. Убедитесь, чтобы все необходимые договора содержали модуль.
- не корректно задан код модуля в параметрах `nginx`. (Модули Редактор модулей и услуг поле `Id` нашего модуля)

"Не получен код доступа": - возникла исключительная ситуация на сервере, смотрите логи сервера.

- если в логах ошибка `io.jsonwebtoken.security.WeakKeyException`, значит у вас слишком короткий код - **token.signature.key**

Модуль Smarthouse

Модуль предназначен для интеграции с системами видеонаблюдения и домофонами.

Система видеонаблюдения Forpost (Форпост)

После установки модуля, создать экземпляр модуля, создать конфигурацию модуля, сделать ее активной.

В конфигурации как минимум заполнить поля:

```
forpost.server.url=http[s]://<__>  
forpost.admin.login=  
forpost.admin.password=
```

После перейти в клиенте на закладку Форпост-Справочники и попробовать синхронизировать их нажав на соответствующие кнопки.

Если синхронизация проходит успешно, можно перейти в нужный договор добавить модуль на договор и на закладке Форпост попробовать привязать существующий аккаунт из форпоста к договору в биллинге или создать новый. После создания аккаунта можно создавать и менять параметры других объектов (камеры, группы и т.д.)

Модуль СОРМ

Модуль предназначен для:

- 1) Формирования файлов, исходя из документации предоставленной производителями решений для реализации требований к СОРМ-3
- 2) Выгрузки сформированных файлов локально и на удалённый FTP сервер.

На данный момент модуль поддерживает формирование файлов для следующего списка поставщиков:



1. МФИ-Софт
2. Vas Experts
3. Специальные технологии
4. Норси-Транс

(Если Вы не нашли реализацию для Вашего поставщика оборудования - обратитесь к разработчикам)

Для начала работы убедитесь, что у всех абонентов, информацию по которым необходимо сформировать, присутствует модуль на договоре.

Перед началом конфигурации модуля советуем ознакомиться с [рекомендациями](#) по организации хранения пользовательских данных для настройки взаимодействия с СОРМ-3.

На данный момент модуль не выгружает CDR файлы телефонии. Данный функционал реализуется дополнительными скриптами, разработка которых не входит в стоимость модуля.

Рекомендации по СОРМ 3

В данной статье речь пойдет о том, как наилучшим образом организовать хранение данных абонентов, оборудования, ресурсов с точки зрения их дальнейшей выгрузки в информационные системы СОРМ 3. Все, что вы прочитаете далее, носит исключительно рекомендательный характер. Все положения основаны на реальном опыте взаимодействия с решениями СОРМ 3 от ведущих поставщиков соответствующих решений.

Справочник абонентов

Первый вопрос, который необходимо осветить в контексте разговора о СОРМ3 - организация хранения персональных данных абонентов. Справочник параметров договора биллинга позволяет гибко настроить тот набор, который необходим оператору для организации работы с абонентом. Однако понятие минимального набора обязательных параметров никак биллингом не регламентируется. Это приводит к тому, что нет унифицированного способа получения обязательных данных по абоненту. Об обязательных параметрах мы поговорим далее.

В технической документации к решениям СОРМ 3 есть так называемые структурированные и неструктурированные поля для выгрузки данных. Первые, как следует из названия, разделены по смысловой нагрузке на атомарные единицы, хранящие в себе строго определенные данные. Вторые - это большие строки произвольной длины, в которых, в общем случае, может быть записано что угодно и как угодно. Да, формально, можно сказать, что необязательно хранить данные в биллинге структурированно, ведь есть поддержка неструктурированных данных. Но по опыту работы с УФСБ разных регионов операторам приходится приводить свои данные в структурированный вид. К таким данным можно отнести следующее:

1. ФИО абонента
2. Даты (дата рождения, заключения договора, выдачи паспорта и т.д.)
3. Сведения о документе, удостоверяющем личность
4. Адрес регистрации (для физических лиц)
5. Юридический адрес (для юридических лиц)
6. Адрес установки конечного абонентского оборудования
7. Почтовый адрес (для юридических лиц)
8. Адрес доставки счетов (для юридических лиц)

Рассмотрим каждый вид более подробно.

ФИО абонента.

Рекомендуется завести 3 отдельных текстовых поля, хранящих отдельно Фамилию, Имя, Отчество. Для удобства работы, если в качестве комментария договора выступает не ФИО, а какой-либо шаблонный номер, можно использовать дополнительное четвертое текстовое поле, в котором аккумулировать результат структурированных полей. Правда, для этого необходимо написать скрипт, который бы при наступлении события изменения параметра договора, формировал данные в этом сборном поле.

Даты

Параметры договора, которые предполагается использовать под хранение дат, рекомендуется создавать с типом Дата. Также необходимо следить за тем, чтобы вбивались корректные даты (речь в основном идет о годах, т.к. часто вбивают по ошибке даты либо далеко в прошлом, либо в будущем). Помимо обучения сотрудников можно создать скрипты поведения, которые срабатывали бы на изменение параметра договора и в случае некорректного значения выдавали бы сообщение оператору. Использование параметра типа Дата предпочтительно вместо текстового параметра, т.к. в выгрузках даты должны фигурировать в определенном формате. А при использовании текстового параметра в общем случае невозможно гарантировать корректный разбор значения с целью переформатирования.

Сведения о документе, удостоверяющем личность

К документам, удостоверяющим личность можно отнести следующие документы:

- Паспорт гражданина РФ
- Заграничный паспорт
- Паспорт СССР
- Удостоверение военнослужащего
- Паспорт иностранного гражданина
- и другие, установленные законодательством РФ

Важное замечание! Водительское удостоверение не является документом, удостоверяющим личность.

Характеристики документов, удостоверяющих личность, которые рекомендуется хранить по отдельности:

- серия документа, удостоверяющего личность
- номер документа, удостоверяющего личность
- дата выдачи документа, удостоверяющего личность
- кем выдан документ, удостоверяющий личность

Рекомендуем завести отдельный списковый параметр, в котором перечислить все используемые оператором типы документов, удостоверяющих личность. Даже если в этом списковом параметре будет лишь одно значение - Паспорт РФ, это значительно поможет

унифицировать и автоматизировать выгрузку справочника типов документов, удостоверяющих личность, потому что при добавлении нового значения не придется править скрипты для выгрузки.

Адреса регистрации для юридических и физических лиц, а также адреса установки конечного абонентского оборудования, почтовый адрес и адрес доставки счетов

Рекомендуется хранить в структурированном виде с использованием параметра типа Адрес. Список обязательных полей для адреса:

- почтовый индекс
- страна

- область/регион
- район, муниципальный округ
- город/посёлок/деревня/аул
- улица
- номер дома
- корпус адреса
- квартира/комната/офис/помещение

Некоторые параметры адреса (например, регион, индекс) возможно хранить в качестве доп. параметров адресного параметра. Обращаем внимание, что не нужно при заполнении адресного справочника использовать сокращенные названия структурных единиц, которые обозначают данную часть адреса. Например: "г.", "ул.", "д." и т.п.

Список обязательных полей

Для физических лиц рекомендуется хранить следующий минимальный набор параметров

1. ФИО (структурированно)
2. Данные о документе, удостоверяющем личность (структурированно)
3. Адрес регистрации (структурированно)
4. Адрес установки оконечного оборудования (структурированно)
5. Дата рождения (поле типа дата)

Для юридических лиц рекомендуется хранить следующий минимальный набор параметров

1. Полное наименование юридического лица (текстовое поле)
2. Юридический адрес (структурированно)
3. Почтовый адрес (структурированно)
4. Адрес доставки счетов (структурированно)
5. Адрес установки абонентского оборудования (структурированно)
6. ИНН юридического лица
7. Банк, в котором у юридического лица расположен расчетный счет для взаиморасчетов с оператором (текстовое поле)
8. Номер расчетного счета
9. ФИО контактного лица (текстовое поле, структурировать не нужно)
10. Контактные данные контактного лица (текстовое поле, содержащее телефон)

Рекомендуем все параметры, содержащие в себе дату (дата рождения, дата выдачи документа и т.д.), хранить в параметре типа Дата, а не в текстовом поле. Это связано с тем, что в выгрузках требуется выгружать даты в определенном формате, а при использовании текстовой строки в общем случае вообще невозможно распознать хранящуюся там дату.

Платежи абонентов

По-возможности рекомендуется максимально подробно сохранять дополнительную информацию по платежу в комментариях: номера счетов, коды транзакций, реквизиты документов и т.д.

Справочные данные

К справочным данным можно отнести следующую информацию:

1. Номенклатура услуг оператора
2. Диапазоны IP-ресурсов, принадлежащих оператору
3. Коммутаторы (к которым подключаются абоненты и получают адреса) и шлюзы (через которые происходит выход в глобальную сеть)
4. Номерная телефонная емкость, принадлежащая оператору, согласно реестру Россвязи.
5. Пучки (транки) телефонии и карта их подключения
6. Типы вызовов
7. Коды причин завершения вызовов
8. Сигнальные коды SS7

Далее рассмотрим некоторые особенности.

IP-ресурсы

Рекомендуется заполнять периоды действия для ресурсов - по мере добавления новых диапазонов добавлять дату добавления.

Также рекомендуется не смешивать в одной категории белые и серые ip-ресурсы.

Коммутаторы

В данном случае под коммутаторами понимаются устройства модуля Inet, nas'ы модуля Dialup, Voiceip, Phone. Аналогично ip-ресурсам рекомендуется выставлять дату начала действия устройства.

Зачастую требуется выгружать физический адрес установки коммутатора. В случае использования модуля Inet, например, можно использовать параметры устройства, в котором завести адресный параметр. Для остальных модулей можно использовать поле комментариев.

Выгрузки принадлежности абонентов

Под принадлежностью понимаются все идентификаторы, которые выдаются абоненту для работы. Сюда можно отнести логины (интернет и телефония), ip-адреса, номера телефонов, sip-аккаунты.

Зеркалирование и генерация Radius

Решения СОРМ 3 требуют зеркалировать nat/flow/radius (как правило, требуется Access-Request/Access-Response, Accounting-start, accounting-stop) потоки на их железо для дальнейшей привязки к выгруженным справочникам данных. Это можно делать напрямую с оборудования оператора. Необходимо следить за тем, чтобы в radius/flow-данных были те идентификаторы (логины, адреса, телефоны), которые выгружаются в справочниках, чтобы решение СОРМ 3 могло соотнести между собой эти данные.

Модуль Inet поддерживает зеркалирование Radius-трафика от абонента с возможностью подмены некоторых полей (например, User-Name), что может требоваться при использовании некоторых схем подключения, где нет четкого идентификатора, который. Для схем, где нет radius-трафика (netflow/dhcp) модуль Inet умеет генерировать Radius start/stop пакеты, сформированные путем форматирования строки с макросами. Это будет происходить в момент начала сессии в биллинге и в момент ее закрытия по таймауту. Для этих целей можно воспользоваться сервис активатором RadiusFanoutServiceActivator. Более подробно см. в описании данного класса в динамическом коде.

Начало работы. Общая конфигурация.

В главном меню модуля на панели "Проекты СОРМ" после клика на  для создания нового проекта откроется панель для настройки всего проекта выгрузки.

ID	Название	Комментарий	Пользователь	Дата создания	Дата изменения	Тип проекта
5	Тестовый проект выгрузки	Доступ к FTP на данный момент не актуален	Валерий Зернов	20.02.2020	22.09.2020	Специальные технологии
7	Второй тест		Валерий Зернов	26.08.2020	22.09.2020	СОРМ3 (ВасЭкспертс)

Х: Модуль СОРМ

В левой части расположена панель для общих настроек.

Где необходимо указать:

- Название проекта выгрузки и комментарий.
- Хост, порт, логин и пароль для доступа к удалённому FTP серверу куда должна производиться выгрузка сформированных файлов.
- Директорию для файлов на локальном сервере
- При необходимости включить функцию удаления локальных файлов после выгрузки файлов на FTP
- Email адрес для отправки отчета об ошибках при выгрузке. Опционально, но рекомендуется для своевременного обнаружения ошибок и более оперативного их исправления
- Поставщика оборудования по формату которого должны быть сформированы выгружаемые файлы
- Выбрать модули, данные по которым должны участвовать в выгрузке

ID проекта: 5

Название проекта:

Тестовый проект выгрузки

Комментарий:

Доступ к ФТП на данный момент не актуален

Хост FTP

10.0.0.111

Порт FTP

21

FTP логин

fsb_remote

FTP пароль

.....

Кодировка

UTF-8

Локальный каталог на сервере

local/tmp/Sorm

Удалить локальные файлы после выгрузки на FTP?

Каталог на FTP сервере ?

/sorm_test/

Email (для отправки отчёта об ошибках)

admin.email@yandex.ru

Поставщик оборудования

Специальные технологии

Выбрать модули для выгрузки

Название экземпляра	Название модуля	<input type="checkbox"/>	...
INET	inet	<input checked="" type="checkbox"/>	▲
Internet	inet	<input type="checkbox"/>	
IPN	ipn	<input type="checkbox"/>	
Dial-Up	dialup	<input type="checkbox"/>	
Phone	phone	<input type="checkbox"/>	
VoIP	voiceip	<input type="checkbox"/>	
Voice	voice	<input checked="" type="checkbox"/>	▼

Сохранить конфигурацию

Закреть

В процессе настройки выгрузки рекомендуется периодически сохранять текущие настройки во избежание потери

Далее на панели с конфигурированием выгрузки абонентов необходимо заполнить данные об операторе связи:

- Полное название организации
- Неструктурированный адрес месторасположения оператора
- Структурированный адрес месторасположения оператора
- Стандарт связи для абонентов интернета (если есть)
- Стандарт связи для абонентов телефонии (если есть)

Стандарт связи можно оставить как "Неконкретизированный", если оператор не предоставляет соответствующую услугу

Общее

Включить данную выгрузку

Данные об операторе связи

Полное название организации оператора
ООО Бител

**Местоположение оператора связи
(Пример: г.Уфа, Республика Башкортостан)**
г.Уфа, Республика Башкортостан

Структурированный адрес оператора

Индекс	Город
440018	Уфа

Улица
Комсомольская

Дом	Номер дома
221	13

Стандарт связи абонентов интернета
Неконкретизированный стандарт ▼

Стандарт связи абонентов телефонии
Мобильная сеть передачи данных ▼

При наличии установленного модуля Бухгалтерия будет отображена панель для взаимодействия модуля СОПМ и модуля Бухгалтерия. В данной панели при необходимости можно указать от куда нужно брать параметры о юридических лицах при формировании файлов.

Выбрать параметры из модуля Бухгалтерия

Укажите соответствие параметров с атрибутами модуля

Полное название организации

address ▼

Адрес

org_name ▼

ИНН

address ▼

БИК

kpp ▼

Название банка

pol_org_name ▼

Расчётный счёт

address ▼

Отмена

Далее необходимо указать из каких полей параметров договора стоит брать те или иные контактные данные абонентов для формирования выгружаемых файлов.

Физическое лицо

Ф.И.О.

Фамилия

Фамилия ▼

Имя

Имя ▼

Отчество

Отчество ▼

ИНН абонента (неструктурированный)

ИНН ▼

Получение

Страна

Страна ▼

Номер

Номер ▼

Кем выдан

Кем выдан ▼

Вид документа

Вид документа ▼

структурированный

структурированный ▼

Дата рождения

Дата рождения ▼

М.И.О. руководителя

М.И.О. руководителя ▼

Контактные данные

Телефон

Телефон (новый) ▼

Факс

Факс ▼

Адрес регистрации

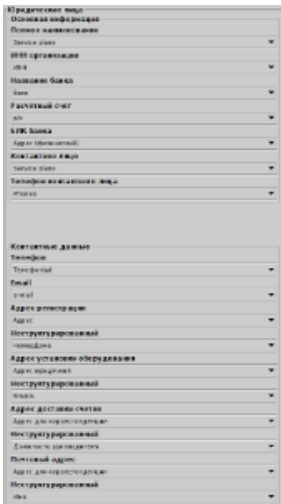
Адрес регистрации ▼

структурированный

структурированный ▼

Адрес: установка оборудования

Адрес ▼



Указание региона и области в адресах



В выгружаемых абонентских данных, в частности адресов, поставщиками оборудования COPM рекомендуется указывать регион и область для указанного адреса. На данный момент указание этих параметров не предусмотрено при заведении адресов в BGBilling, но есть возможность указать регион и область как дополнительные параметры к городу.

Для этого в конфигурации сервера нужно прописать ключи доп.параметров:

address.city=region, zone

address.city.region.title=Регион для выгрузки COPM

address.city.region.type=string

address.city.zone.title=Район для выгрузки COPM

address.city.zone.type=string

После чего появляется возможность для каждого города указать регион и область, которые будут указаны при выгрузке данных.

Указание региона и области в выгружаемых данных возможна только для адресов, которые были заведены в BGBilling

Исходя из требований к выгрузке нужно выгружать так же "внутренних пользователей". Внутренние пользователи - сотрудники юр.лиц, которые имеют доступ к интернету и выходят в сеть с тем же адресом/логином, что и юр.лицо.

Добавить список таких сотрудников к юр.лицу можно используя [Объекты](#). Заведите тип объекта :

11 ЮЛ - Сотрудник	{{text:37}} {{text:40}} {{text:43}}	Проверка для COPM. Выгрузка сотрудников юр.лиц у которых есть доступ к ине...
-------------------	-------------------------------------	---

затем список параметров объекта:

11	1	ЮЛ - Сотрудник - Адрес прописки (строка)
12	3	ЮЛ - Сотрудник - Дата выдачи
13	3	ЮЛ - Сотрудник - Дата рождения
14	1	ЮЛ - Сотрудник - Имя
15	1	ЮЛ - Сотрудник - Кем выдан
17	1	ЮЛ - Сотрудник - Номер документа
22	1	ЮЛ - Сотрудник - Номер телефона
18	1	ЮЛ - Сотрудник - Отчество
19	1	ЮЛ - Сотрудник - Серия документа
23	2	ЮЛ - Сотрудник - Тип документа
21	1	ЮЛ - Сотрудник - Фамилия

где "тип документа" - параметр типа список, "дата выдачи" и "дата рождения" - параметры типа дата, остальные - текстовые параметры.

Далее на вкладке "Справочники" "Объекты" "Параметры типов" установите созданные параметры как параметры к типу созданного объекта.

В модуле COPM, на панели настройки юр.лиц выберите тип объекта и настройте соответствующие параметры к параметрам типа объекта

Сотрудники юр.лиц с доступом к сети интернет



Выберите тип объекта

ЮЛ - Сотрудник ▼

Фамилия сотрудника

ЮЛ - Сотрудник - Фамилия ▼

Имя сотрудника

ЮЛ - Сотрудник - Имя ▼

Отчество сотрудника

ЮЛ - Сотрудник - Отчество ▼

Дата рождения

ЮЛ - Сотрудник - Дата рождения ▼

Тип документа

ЮЛ - Сотрудник - Тип документа ▼

Серия

ЮЛ - Сотрудник - Серия документа ▼

Номер

ЮЛ - Сотрудник - Номер документа ▼

Кем выдан

ЮЛ - Сотрудник - Кем выдан ▼

Когда выдан

ЮЛ - Сотрудник - Дата выдачи ▼

Адрес регистрации(неструктурированный)

ЮЛ - Сотрудник - Адрес прописки (строка) ▼

Контактный номер телефона

ЮЛ - Сотрудник - Номер телефона ▼

Настройка выгрузки справочных данных

На панели "Справочники" предлагается выбрать справочные данные не касающиеся непосредственно абонентских данных.

Каждая логически обособленная справочная информация находится в отдельной панели и первоначально скрыта для настройки.

Чтобы приступить к настройке, нужно "включить" панель путем нажатия на галочку рядом с названием панели,

например, выгрузка ip-ресурсов: **Включить выгрузку IP-нумерации оператора**, после чего содержимое будет доступно для изменений и конфигурации.

В панели с настройкой интернета и телефонии участвуют только те модули, которые были выбраны в таблице с общей настройкой:

Выбрать модули для выгрузки

Название экземпляра	Название модуля	<input type="checkbox"/>	...
INET	inet	<input checked="" type="checkbox"/>	▲
Internet	inet	<input type="checkbox"/>	
IPN	ipn	<input type="checkbox"/>	

Для настройки конкретного модуля интернета или телефонии, он должен быть выбран для выгрузки путём нажатия галочки, например:

Модуль не включен для выгрузки

Включить выгрузку информации о коммутаторах телефонии
Выберите устройства, которые будут выгружены в качестве коммутаторов
 Модуль: Voice , id=217

Модуль включен

Включить выгрузку информации о коммутаторах телефонии
Выберите устройства, которые будут выгружены в качестве коммутаторов
 Модуль: Voice , id=217

ATC Si2000
 third: test_script

Выгрузка общих справочников.

Выгрузка типов платежей.

В данной настройке необходимо выбрать справочник с типами платежей оператора, которые будут указаны в выгрузке и на которые будут ссылаться выгружаемые данные о платежах абонентов.

Справочники

Включить выгрузку типов платежей

Выберите типы платежей, которые будут выгружены

- Chronopay
- Click
- CyberPlat
- EPort
- IMPORT
- MOBI. Деньги
- MobiCash
- Moneta

Все Сброс Инверт.

Выгрузка услуг модулей Абонентских плат(Npay).

Панель с выбором услуг делится на две панели: левая - список установленных экземпляров модулей Npay, правая - список услуг, заведённых для данного модуля.

Необходимо выбрать все услуги, которые должны быть выгружены, как услуги оператора. Исходя из этих данных будет сформирован и выгружен справочник соответствия конкретного абонента и всех его услуг.

Включить выгрузку услуг(тарифов)

Выберите услуги, которые будут выгружены
(по двойному клику будут выбраны все услуги в модуле)

Новые абонплаты	ID	Название	Используется	Выбрать ...
Новые абонплаты2 для СОРМ	120	qqq (Abonplata)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	198	Wut?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	79	Абонплата	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	75	Абонплата 60 в месяц	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	76	Абонплата 70 дневная	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	230	Абонплата за пользование сервисом	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	103	Абонплата за телефоны	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	114	Абонплата за трафик	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Выгрузка типов документов.

Необходимо выбрать параметр договора с типом список, в котором указан тип документа, который заполнен у абонентов оператора. (Обычно в данном поле содержится параметр "Паспорт РФ")

Включить выгрузку типов документов

Выберите параметр договора с указанием типа документа, который будет фигурировать в выгрузке

Список

Выгрузка справочника регионов.

В данной выгрузке указывается код и название региона, в котором работает оператор.

Включить выгрузку справочника регионов

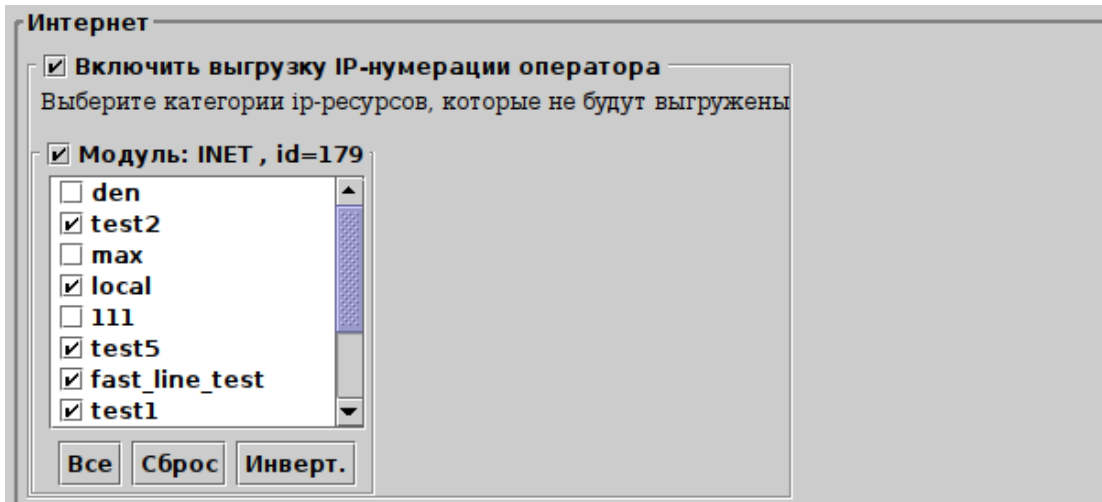
Код региона: 102

Название региона: Республика Башкортостан

Настройки выгрузке данных интернета.

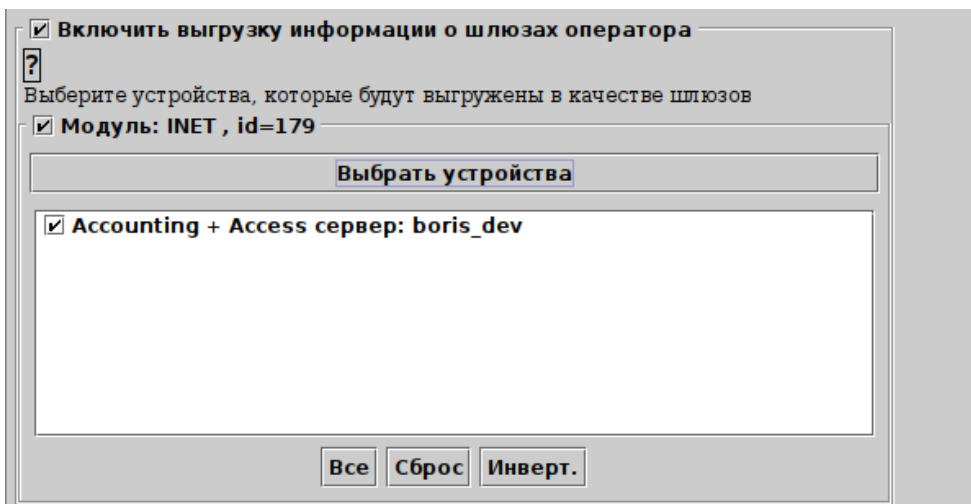
Выгрузка ip-нумерации оператора.

По умолчанию ip-нумерация выгружается полностью, но есть возможность выбрать категории, которые не будут выгружены.



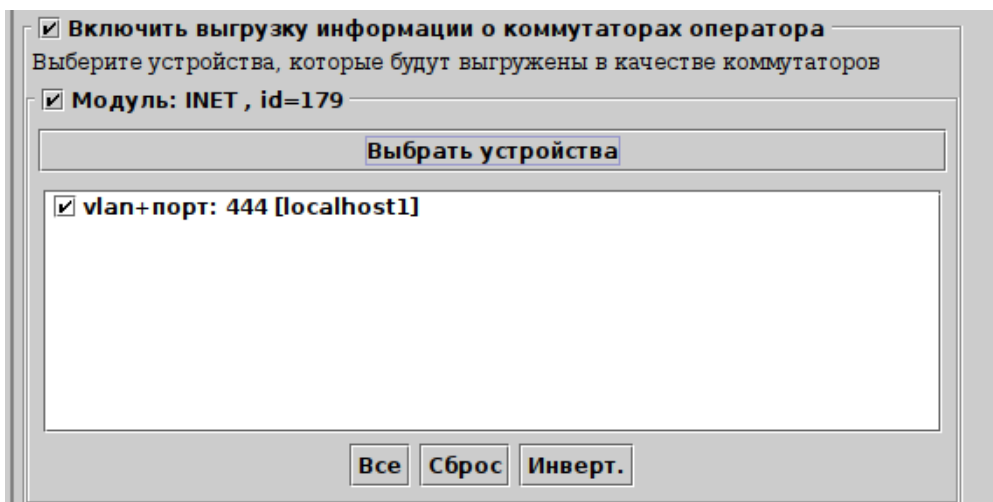
Выгрузка информации о шлюзах оператора.

В контексте СОРМа под шлюзами оператора понимается любые пограничные коммутаторы.



Выгрузка информации о коммутаторах оператора.

Необходимо выбрать все устройства, которые должны быть выбраны в качестве коммутаторов оператора.



В выгружаемых файлах должны быть заполнены адреса устройств. Их можно добавить воспользовавшись атрибутами устройств. Про атрибуты можно прочитать на странице [Устройства](#). Для устройств телефонии модуля Phone единственный способ указать адрес - прописать его в комментарии в неструктурированном виде

Настройки выгрузки данных телефонии.

Выгрузка данных о коммутаторах телефонии оператора.

Аналогично коммутаторам интернета, необходимо выбрать устройства, которые будут выгружены в качестве коммутаторов телефонии.

Включить выгрузку информации о коммутаторах телефонии

Выберите устройства, которые будут выгружены в качестве коммутаторов

Модуль: Voice , id=217

Выбрать устройства

ATC Si2000
 third: test_script

Все Сброс Инверт.

Выгрузка номерной ёмкости оператора.

По умолчанию, в качестве номерной ёмкости будут выгружены все категории номеров. Необходимо отметить категории номеров, которые не должны быть выгружены.

Включить выгрузку номерной ёмкости

Категории номеров, которые не нужно выгружать

Модуль: Phone id=73

Модуль: Voice id=217

основная категория
 вторая категория
 2222
 3333




Все Сброс Инверт.

Все Сброс Инверт.


Выгрузка пучков телефонии.

При необходимости в данной настройке можно задать список пучков оператора

Включить выгрузку пучков оператора

ID	Название	Направленность	Описание	...
1	пучок	двунаправленный	тестовое описание	▲

При клике на  для добавления нового пучка, открывается панель для создания.

Где "дополнительная информация" является необязательной, но желательной к заполнению.

Название		Направленность	
<input type="text"/>		входящий ▼	
Описание			
<input type="text"/>			
Дополнительная информация ?			
MAC-адрес пучка	<input type="text"/>		
ID коммутатора	<input type="text"/>		
Период	с <input type="text"/> по <input type="text"/>		
			<input type="button" value="Ок"/> <input type="button" value="Отмена"/>

Выгрузка. Настройка расписания выгрузки

Выгрузка файлов делится на 2 типа.

1- полная выгрузка, когда выполняется выгрузка всех абонентов чьи договора актуальны или были закрыты в течении последних 3-х лет от даты начала выгрузки, а так же выгружаются все их данные, и данные о платежах и прочая информация.

2- инкрементальная выгрузка в процессе работы которой находится последний выгруженный договор, последний выгруженный платёж и исходя из этих данных составляется список абонентов, которые ещё не были "выгружены" и по ним происходит формирование файлов.

Запустить выгрузку можно единоразово из главной панели модуля, в таблице с проектами СОРМ, где нужно выбрать проект для выгрузки, убедиться, что он находится в статусе "включен", после чего кликнуть правой кнопкой мыши и выбрать "Запустить выгрузку" (будет запущена инкрементальная выгрузка) или "Запустить полную выгрузку" (будет выполнена полная выгрузка).

Периодическая выгрузка файлов, предполагает автоматическое формирование и отправку файлов по расписанию.

Для этого нужно зайти в Планировщик, где создать 2 задачи для модуля СОРМ:

1- Задача на выгрузку общих данных.

2- Задача на выгрузку платежей абонентов.

Рекомендуется установить интервал работы задач не менее 20 минут.

По умолчанию выгрузка задач по расписанию через планировщик подразумевает инкрементальную выгрузку(только изменения абонентов и новые абоненты). Для того, чтобы включить полную выгрузку в планировщике необходимо в конфигурации задачи указать **typeUpload.X=1**, где X - id проекта сорм, а 1 - флаг, что должна выполняться полная выгрузка. Для того, чтобы вернуться к выполнению инкрементальных выгрузок нужно указать 0 или удалить данную настройку. То же самое работает и для выгрузки платежей. Если 1-будут выгружены все платежи за последние 3 года, иначе только новые платежи с момента последней выгрузки

Задача очистки статистики.

Со временем кол-во записей в истории выгрузок копится и для того, чтобы не очищать в ручную из модуля, есть возможность создать задачу в планировщике. Например, на запуск очистки статистики раз в месяц.

В конфигурации задачи можно указать ID проекта сорм, для которого необходимо очищать статистику. Если ID не будет указано, то очистится вся история для всех проектов. **sormProjectId=1** - будет очищена история для проекта с ID = 1

Так же в конфигурации можно указать кол-во дней за которые не надо удалять статистику. **doNotDeleteLastDays** (Т.е. например, запуск задачи настроен на 20-е число месяца, и установлено doNotDeleteLastDays=10, в таком случае будет удалена вся статистика до 10 числа данного месяца)

Статистика выгрузок

Рядом с главной вкладкой модуля располагается вкладка с статистикой произведённых выгрузок.

Проекты СОРМ	Статистика					
ID проекта СОРМ	Дата выгрузки	Кол-во выгруженных договоров	ID первого выгруженного договора	ID последнего выгруженного договора		Статус выгрузки на FTP
5	28.08.2020 18:07:56	1924	0	762988		Не выгружено
5	28.08.2020 18:10:30	0	0	0		Не выгружено
5	31.08.2020 14:27:59	0	0	0		Не выгружено
5	31.08.2020 15:09:52	1924	0	762988		Не выгружено
5	31.08.2020 15:25:07	1924	0	762988		Не выгружено
5	31.08.2020 15:43:16	1924	0	762988		Не выгружено
5	31.08.2020 15:51:57	1924	0	762988		Не выгружено
5	15.09.2020 11:38:08	1924	0	762988		Не выгружено
5	22.09.2020 09:11:18	0	0	0		Не выгружено
5	22.09.2020 09:12:31	0	0	0		Не выгружено
5	22.09.2020 09:14:00	1924	0	762988		Не выгружено
5	22.09.2020 09:56:08	1924	0	762988		Не выгружено
5	22.09.2020 10:00:18	1924	0	762988		Не выгружено
5	22.09.2020 10:22:18	1924	0	762988		Не выгружено
5	22.09.2020 10:22:51	1924	0	762988		Не выгружено
5	22.09.2020 15:31:57	1	0	762913		Не выгружено

Таблица с статистикой содержит информацию:

- ID проекта СОРМ, выгрузка по конфигурации которого была произведена
- Дата и время начала выгрузки
- Общее кол-во договоров, которое было выгружено
- ID первого и ID последнего выгруженного договора, и всех сопутствующих этим договорам данных.
- Статус выгрузки на FTP. Выгружено или Не выгружено. В случае, если будет проблема с соединением к FTP серверу и выгрузка не будет произведена, то эта информация отразится в статусе

Переопределение выгружаемых файлов

Имеется возможность применить свою реализацию для формирования каких-либо выгружаемых файлов. При старте выгрузки будет сформировано синхронное событие `ru.bitel.bgbilling.modules.sorm.server.event.SormUploadEvent`.

В динамическом коде можно заменить один или несколько файлов, можно добавить один или несколько файлов к выгрузке.

Пример

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import ru.bitel.bgbilling.kernel.script.server.dev.EventScriptBase;
import ru.bitel.bgbilling.modules.sorm.common.bean.SormProject;
import ru.bitel.bgbilling.modules.sorm.server.event.SormUploadEvent;
import ru.bitel.bgbilling.modules.sorm.server.upload.Upload;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.model.Pair;
import ru.bitel.common.sql.ConnectionSet;

import java.nio.file.Path;
import java.util.Collections;
import java.util.List;

public class CustomUpload
    extends EventScriptBase<SormUploadEvent>
{
    @Override
    public void onEvent( SormUploadEvent event, Setup setup, ConnectionSet set )
        throws Exception
    {
        SormProject sormProject = event.getSormProject();

        CustomUploadSorm customCustomUploadSorm = new CustomUploadSorm( sormProject, true );

        event.setUploadVersion( customCustomUploadSorm );
    }

    /**
     * - .
     */
    static class CustomUploadSorm
        extends Upload
    {
        SormProject sormProject;

        public CustomUploadSorm( SormProject sormProject )
        {
            //true - , false - . . . , .
            super( sormProject, true );
            this.sormProject = sormProject;
        }

        @Override
        public Path abonents()
        {
            return null;
        }

        @Override
        public Path abonentsAddresses()
        {
            return null;
        }

        @Override
        public Path abonentLogins()
        {
            return null;
        }
    }
}
```

```
@Override
public Path services()
{
    return null;
}

@Override
public Path abonentServices()
{
    return null;
}

@Override
public Path ipNumbering()
{
    return null;
}

@Override
public Path abonentsIpNumbering()
{
    return null;
}

@Override
public Path switches()
{
    return null;
}

@Override
public Path gateways()
{
    return null;
}

@Override
public Path bunches()
{
    return null;
}

@Override
public Path phoneNumbering()
{
    return null;
}

@Override
public Path phoneNumberingAbonents()
{
    return null;
}

@Override
public Path documentTypes()
{
    return null;
}

@Override
public Path paymentTypes()
{
    return null;
}

@Override
public Path abonentsPayments()
{
    return null;
}
```

```

    }

    @Override
    public Path operatorInfo()
    {
        return null;
    }

    /**
     * , ,
     * , :
     * key - ,
     * value -
     *     null,
     *
     * @return
     */
    @Override
    public List<Pair<String, Path>> getOtherFiles()
    {
        return Collections.emptyList();
    }
}

```

Метод `getOtherFiles()` может быть использован для формирования и выгрузки любых файлов, которые должны быть отправлены на ftp при процессе выгрузки.

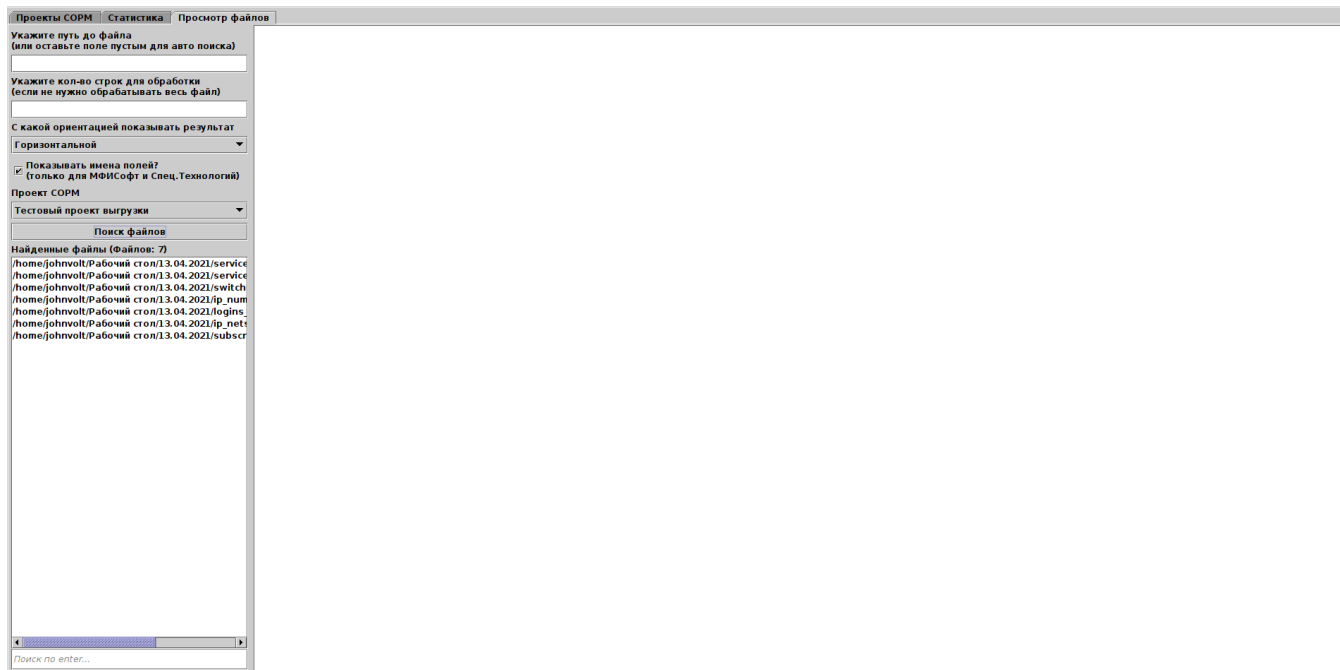
Просмотр сформированных файлов

На вкладке "Просмотр файлов" можно быстро и удобно просмотреть сформированные файлы. После запуска выгрузки, можно перейти к просмотру файлов.

Есть возможность указать точный абсолютный путь до файла на сервере или найти автоматически все файлы.

Примечание: для того, чтобы работал автоматический поиск, в настройке проекта выгрузке не должно быть включено автоматическое удаление локальных файлов.

В списке слева будет показано список всех доступных для просмотра файлов.



Для того, чтобы открыть файл для просмотра нужно дважды кликнуть на выбранном файле.

Так же, для удобства, есть возможность указать какое кол-во строк нужно показывать.

Выбор ориентации отображения отображает записи из файла либо в горизонтальном виде (одна запись - одна строка), либо в вертикально виде (каждое значение на новой строке).

Если поставщик МФИСофт или Спец.Технологии, то так же есть возможность отображать название каждого поля, тогда будет более удобно соотносить поле с его содержанием.

Как пример, на скриншоте открыт файл с данными абонентов сформированный для ВасЭкспертс в горизонтальном отображении, с выводом первых 5-ти строк в файле

Укажите путь до файла
(или оставьте поле пустым для авто поиска)

0) *4* 1) *455* 2) *455* 3) *00000* 4) *0* 5) *06.02.2005* 6) ** 7) *1* 8) ** 9) ** 10) ** 11) ** 12) ** 13) ** 14) ** 15) ** 16) ** 17) ** 18) ** 19) ** 20) ** 21) ** 22) ** 23) ** 24) ** 25) ** 26) ** 27) ** 28) ** 29) *0* 30) ** 31) ** 32) ** 33) ** 34) ** 35) ** 36) **
0) *4* 1) *32343* 2) *32343* 3) *001482* 4) *0* 5) *18.12.2006* 6) ** 7) *0* 8) *1* 9) ** 10) ** 11) ** 12) ** 13) ** 14) *1* 15) ** 16) ** 17) ** 18) ** 19) ** 20) ** 21) ** 22) ** 23) ** 24) ** 25) ** 26) ** 27) ** 28) ** 29) *0* 30) ** 31) ** 32) ** 33) ** 34) ** 35)
0) *4* 1) *32593* 2) *32593* 3) *0000001* 4) *0* 5) *28.01.2008* 6) ** 7) *0* 8) *1* 9) ** 10) ** 11) ** 12) ** 13) ** 14) *1* 15) ** 16) ** 17) ** 18) ** 19) ** 20) ** 21) ** 22) ** 23) ** 24) ** 25) ** 26) ** 27) ** 28) ** 29) *0* 30) ** 31) ** 32) ** 33) ** 34) ** 35)
0) *4* 1) *32603* 2) *32603* 3) *kml* 4) *0* 5) *29.01.2008* 6) ** 7) *0* 8) *1* 9) ** 10) ** 11) ** 12) ** 13) *16.03.2012* 14) *1* 15) ** 16) ** 17) ** 18) ** 19) ** 20) ** 21) ** 22) ** 23) ** 24) ** 25) ** 26) ** 27) ** 28) ** 29) *0* 30) ** 31) ** 32) ** 33) ** 34)
0) *4* 1) *352715* 2) *352715* 3) *dmon* 4) *0* 5) *04.02.2009* 6) ** 7) *0* 8) *1* 9) ** 10) ** 11) ** 12) ** 13) ** 14) *1* 15) ** 16) ** 17) ** 18) ** 19) ** 20) *Бюджет* 21) ** 22) ** 23) ** 24) ** 25) ** 26) ** 27) ** 28) ** 29) *0* 30) ** 31) ** 32) ** 33) **

Укажите кол-во строк для обработки
(если не нужно обрабатывать весь файл)

5

С какой ориентацией показывать результат

Горизонтальной

Показывать имена полей?
(только для МФИСофт и Спец.Технологий)

Проект СОРМ

Тестовый проект выгрузки

Поиск файлов

Найденные файлы (файлов: 7)

- ючий стол/13.04.2021/services_13042021103218
- ючий стол/13.04.2021/service_list_v1_13042021103219
- ючий стол/13.04.2021/switches_13042021103219
- ючий стол/13.04.2021/ip_numbering_13042021103219
- ючий стол/13.04.2021/rogues_1304202110321114
- ючий стол/13.04.2021/ip_nets_v1_1304202110321
- ючий стол/13.04.2021/subscribers_v2_130420211

Формирование радиус-пакетов для СОРМ

Реализовать данную возможность можно используя класс из штатной поставки дин.кода **ru.bitel.bgbilling.modules.inet.dyn.device.misc.RadiusFanoutServiceActivator**.

Данный класс необходимо установить обработчиком активации сервисов на **Типе устройства** для устройства

В шапке класса приведены примеры макросов для формирования пакетов по структуре необходимой для СОРМ.

Свои макросы для старт и стоп пакетов нужно прописать в конфигурации типа устройства. Пример:

```
sa.radius.fanout.accountingStart.attributes.macros=User-Name=$contractTitle();Acct-Session-Id=$connectionId()-$acctSessionId();Acct-Status-Type=1;Framed-IP-Address=$ip();Calling-Station-Id=$callingStationId()
sa.radius.fanout.accountingStop.attributes.macros=User-Name=$contractTitle();Acct-Session-Id=$acctSessionId();Acct-Status-Type=2;Framed-IP-Address=$ip();Calling-Station-Id=$callingStationId();
```

А так же хост и порт для отправки сформированных пакетов

```
sa.radius.fanout.host=XXX.XXX.X.XX
sa.radius.fanout.port=1813
```

Затем, в **Типе сервиса** для тех договоров по которым нужно будет формировать пакеты прописать настройку: `sa.connection.auto.accountingEvent=1`

Из документации к модулю Инет: "sa.connection.auto.accountingEvent - нужно ли генерировать start-stop accounting-пакеты для сессий по трафику. Это может понадобиться для генерации фейковых radius- пакетов для СОРМа при помощи сервис-активатора RadiusFanoutServiceActivator. Возможные значения 0 (false) - отключено, 1 (true) - включено. По умолчанию 0."

Обработчик RadiusFanoutServiceActivator отправляет стоп пакеты при разрыве соединения и т.к. сессии по трафику логически приостанавливаются по таймауту или на границе суток, то возможно, что пакеты формироваться не будут, в таком случае нужно добавить в планировщик скрипт на приостановку активных соединений. Простейший пример такого скрипта представлен ниже.

```
package ru.sorm3;

import org.apache.log4j.Logger;
import ru.bitel.bgbilling.common.BGException;
import ru.bitel.bgbilling.kernel.container.managed.ServerContext;
import ru.bitel.bgbilling.kernel.script.server.dev.GlobalScriptBase;
import ru.bitel.bgbilling.modules.inet.api.common.bean.InetSessionLog;
import ru.bitel.bgbilling.modules.inet.api.common.service.InetSessionService;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.sql.ConnectionSet;

import java.util.Collections;
import java.util.List;

public class InetSessionStopScript
    extends GlobalScriptBase
{
    private final static Logger logger = Logger.getLogger( InetSessionStopScript.class );

    private final int TEST_CID = -1;

    private final int inetModuleId = 12;

    @Override
    public void execute( Setup setup, ConnectionSet connectionSet )
        throws Exception
    {
        ServerContext serverContext = ServerContext.get();
        InetSessionService sessionService = serverContext.getService( InetSessionService.class, inetModuleId );

        List<InetSessionLog> aliveSessions = getAliveSessions( sessionService );
        if( aliveSessions != null && aliveSessions.size() > 0 )
        {
            logger.info( "    = " + aliveSessions.size() );
        }
        else
        {
            logger.info( "    " );
        }
    }
}
```

```

        for( InetSessionLog session : aliveSessions )
        {
            try
            {
                sessionService.connectionFinish( session.getContractId(), session.getConnectionId() );
            }
            catch( BGException e )
            {
                logger.error( "    =" + session.getConnectionId() +
                    "    id=" + session.getContractId() +
                    " : " + e.getMessage() );
            }
        }
    }

private List<InetSessionLog> getAliveSessions( InetSessionService sessionService )
{
    try
    {
        return sessionService.inetSessionAliveList( null, TEST_CID > 0 ?
            Collections.singleton(TEST_CID) : null ,null,null,
null,null,null,null,null ).getList();
    }
    catch( BGException e )
    {
        logger.error( e.getMessage() );
    }

    return Collections.emptyList();
}
}

```

Модуль BotManager

Модуль предназначен для автоматизации работы с клиентами в мессенджерах.

Работа с клиентами происходит по средствам создания и настройки ботов, а так же реализации алгоритмов работы с пользователями через этих ботов (**на данный момент существует реализация для Телеграмм и ВКонтакте**).

Сущности модуля:

- **диалог** - настроенный экран с элементами диалога, который будет отображён в чате с пользователем. Может содержать любое кол-во элементов диалога.
- **элемент диалога** - какой-либо визуальный элемент (текст, кнопка, ссылка, вложение(любой файл до 20мб))
- **запись статистики** - отображается в таблице с статистикой в самом модуле или на панели модуля в договоре. Отображает информацию о том когда было отправлено последнее сообщение пользователю, какое сообщение было отправлено, его id в том месседжере, через который было отправлено сообщение и id договора в БГБиллинге, если пользователь авторизовался.
- **группа диалогов** - пользовательская группа, которое содержит в себе диалоги для удобства скрытия/отображения диалогов данной группы в интерфейсе модуля. Группа для диалога устанавливается в редакторе диалога.

Краткое описание логики модуля:

в клиенте БГБиллинга создаются и настраиваются боты, 1 мессенджер = 1 бот.

Для каждого бота обязательно должен быть установлен "стартовый диалог" - диалог, который будет отображён при первом обращении к боту.

"Стартовый диалог" - это входная точка с кнопками для дальнейшей навигации.

Для каждого диалога есть возможность:

- указать команду с которой будет ассоциироваться именно это диалог
- указать класс-обработчик, в котором есть 2 метода для реализации логики. 1 - вызывается до отображения диалога, 2 - вызывается после и позволяет обработать полученную от пользователя информацию
- установить настройку, чтобы диалог отображался только для авторизованных пользователей

Авторизованный пользователь - тот, чей договор был найден. Вся логика авторизации пользователя должна быть реализована в дин.коде. (Несколько простейших примеров реализации идут в дин.коде в поставке с модулем). Если договор был найден, то в обработчике диалога в дин.коде в JSON объект "chatData" нужно положить id договора под ключом "contractId"

Создание ботов

Создание бота для Вконтакте

1) Создаём группу в вконтакте и даём название

2) Заходим в группу -> управление, копируем id созданной группы, затем переходим в интерфейс модуля BotManager и вставляем в конфигурацию модуля id группы под ключом **vkbot.{ID}.groupId** (где {ID} - id бота)

Затем в настройках созданной вконтакте группы:

3) Переходим в "сообщения" и включаем их

4) Переходим в настройки -> работа с API -> Long Poll API, включаем и выбираем последнюю доступную версию.

Далее здесь же переходим в "Типы событий" и выбираем входящие и исходящие сообщения

5) Переходим в "Ключи доступа" и создаём свой ключ, затем указываем в конфигурации модуля под ключом **vkbot.{ID}.accessToken**

Создание бота для Телеграмм

1) В телеграмме в поиске находим @BotFather

2) Пишем команду /newbot, после чего бот попросит указать имя бота.

3) Далее телеграмм просит указать имя пользователя для работы с ботом, снова пишем точно такое же название с приставкой _bot

4) В ответ телеграмм присылает сообщение в котором указан токен, данный токен нужно положить в конфигурацию под ключом **telegrambot.{ID}.token**

5) Чтобы для бота добавить первое сообщение, которое будет отображено после стандартного "Что умеет этот бот", нужно написать BotFather команду /setdescription и указать содержание сообщения.

Так же в данном боте можно указывать команды, с которыми затем напрямую ассоциировать какой-либо диалог в модуле. Для указания команды в боте BotFather нужно ввести /setcommands, выбрать свой бот и перечислить команды и описание действия к команде, затем в модуле в редакторе диалога можно указать данную команду в отдельном поле

Настройки в конфигурации модуля

Дополнительные настройки в конфигурации модуля

#указывается id стартового диалога для конкретного бота

bot.{ID}.start.dialog.id=4

#Для построения клавиатуры можно установить данную настройку с указанием кол-ва кнопок в одной линии. По умолчанию кол-во = 2

bot.{ID}.keyboard.buttoninline.count=3

#серверный к filestorage модуля для хранения файлов-вложений

file.storage.root.path=/home/user/Рабочий стол/

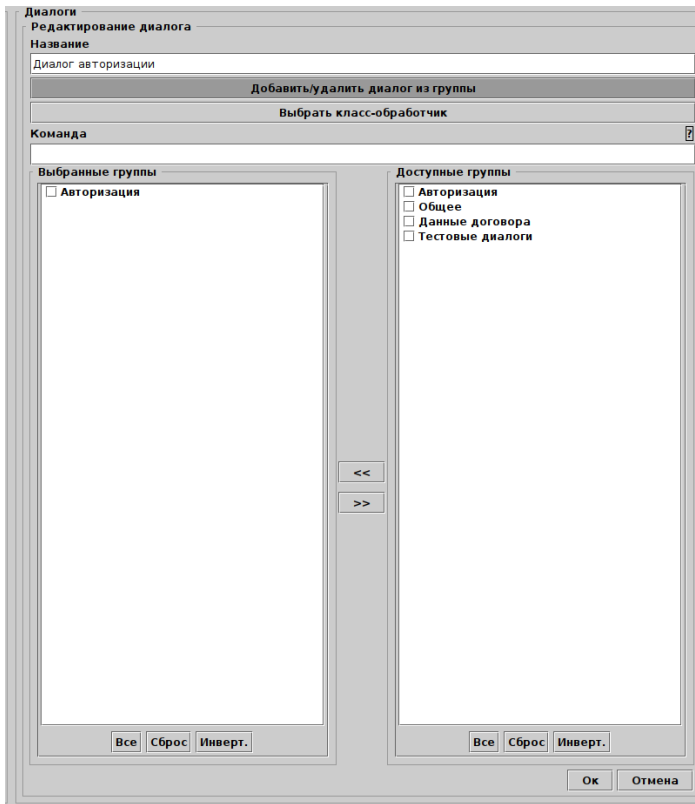
#Можно включать и отключать ведение статистики для каждого бота следующей настройкой

bot.{id}.statistic.enable=true

Создание диалогов

В редакторе диалога необходимо указать название диалога и добавить как минимум 1 элемент диалога (перетащить мышкой элемент из списка элементов на панели справа).

При необходимости данный диалог можно добавить в одну или несколько групп диалогов.



Группы диалогов создаются на панели "Группы диалогов".

ID	Название	Комментарий
3	Авторизация	Набор диалогов для авторизации абонента в боте
4	Общее	Общие диалоги, которые могут быть использованы в любой момент
5	Данные договора	Общие справочные данные договора после авторизации
6	Тестовые диалоги	

В простом варианте диалог - это набор текста и кнопок, которые могут отображать другие диалоги, но в более сложном алгоритме, где нужны, например, проверки введённых пользователем данных, нужно указывать реализацию в дин.коде в виде класса-реализации интерфейса **ru.bitel.bgbilling.modules.botmanager.common.bean.DialogProcessing**

DialogProcessing

```
/**
 *
 */
public interface DialogProcessing
{
    /**
     * id ,
     */
    String CONTRACT_ID = "contractId";
    /**
     *
     */
    String ERROR_MESSAGE = "responseErrorMessage";
    /**
     *
     */
    String REMOVE_CONTRACT = "idContractRemove";
    /** ID ,
     */
    String NEXT_DIALOG_ID = "nextDialogId";
    /**
```

```

*
*     JSONObject,
*     - ,
*/
String MACROS = "macros";

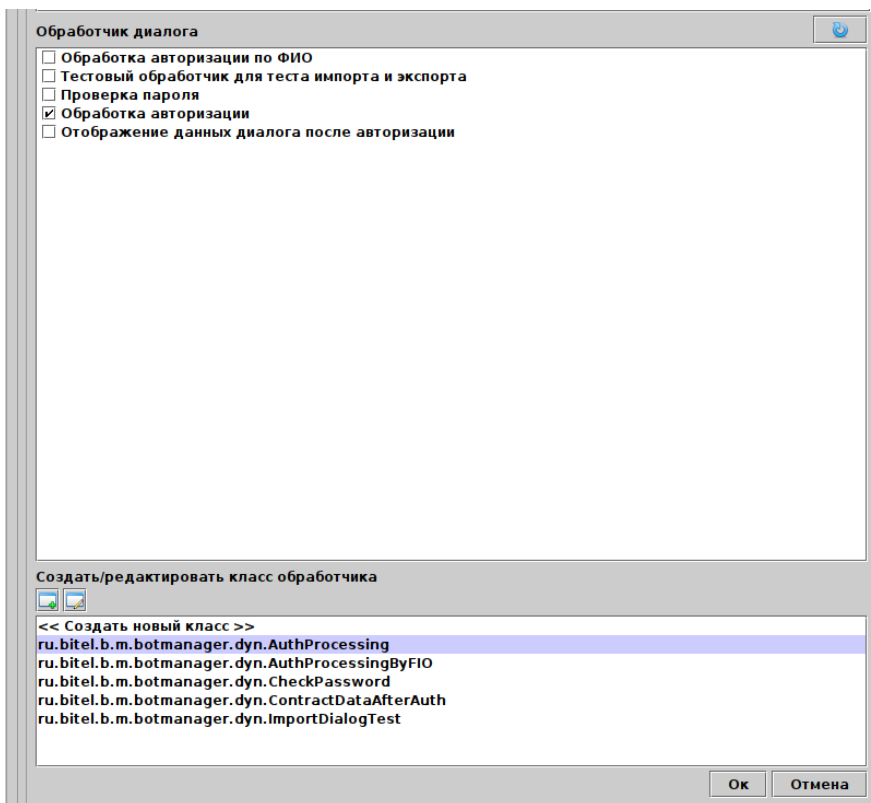
/**
*
* @param botData
*/
void preDialog( ServerContext serverContext, JSONObject botData )
    throws BGEException;

/**
*
* @param botData
* @param inputTextOfUser
*/
void processDialog( ServerContext serverContext, JSONObject botData, String inputTextOfUser )
    throws BGEException;

/**
* @return
*/
String title();
}

```

Свою реализацию указать для диалога можно в реакторе диалога, где после нажатия на "Выбрать класс-обработчик" будет отображена панель с списком классов из дин.кода, которые реализуют интерфейс DialogProcessing



Как пример реализации интерфейса **DialogProcessing** ниже приведён пример класса-реализации логики авторизации пользователя по названию договора. Данный класс обрабатывает после запроса названия договора от пользователя, полученный текст передаётся в метод **processDialog()**, в котором происходит поиск договора и в случае, если договор был найден, указывается, что следующим диалогом должен быть показан заранее созданный диалог с запросом пароля от личного кабинета.

AuthProcessing


```

package ru.bitel.bgbilling.modules.botmanager.dyn;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.json.JSONObject;
import ru.bitel.bgbilling.kernel.container.managed.ServerContext;
import ru.bitel.bgbilling.modules.botmanager.common.bean.DialogProcessing;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class AuthProcessing
    implements DialogProcessing
{
    private static final Logger logger = LogManager.getLogger();

    /**
     * ID
     */
    private final int REQUEST_PASS_DIALOG = 9;

    /**
     * ID
     */
    private final int ERROR_DIALOG = 8;

    @Override
    public void preDialog( ServerContext serverContext, JSONObject botData )
    {
    }

    @Override
    public void processDialog( ServerContext serverContext, JSONObject botData, String inputTextOfUser )
    {
        int cid = findContract( serverContext.getConnection(), inputTextOfUser );
        if( cid > 0 )
        {
            botData.put( CONTRACT_ID, cid );
            // , ,
            botData.put( NEXT_DIALOG_ID, REQUEST_PASS_DIALOG );
            if( logger.isDebugEnabled() )
            {
                logger.debug( "Substring for search contract=" + inputTextOfUser + " Contract was find" );
            }
        }
        else
        {
            botData.put( NEXT_DIALOG_ID, ERROR_DIALOG );
            if( logger.isDebugEnabled() )
            {
                logger.debug( "Cannot to find contract for substring: " + inputTextOfUser );
            }
        }
    }

    //
    private int findContract( Connection connection, String contractTitle )
    {
        int result = -1;
        String query = "select id from contract where title like '%" + contractTitle + "%'";
        try(Statement statement = connection.createStatement();
            ResultSet rs = statement.executeQuery( query ))
        {
            if( rs.next() )
            {
                result = rs.getInt( "id" );
            }
        }
        catch( SQLException ex )
    }
}

```

```
        {
            logger.error( ex );
        }

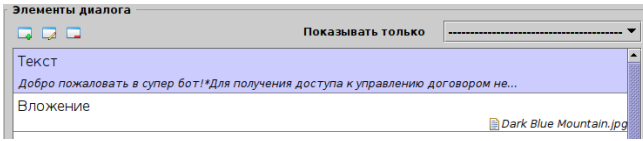
        return result;
    }

    @Override
    public String title()
    {
        return " ";
    }
}
```

Создание элементов диалога

Элементы диалога могут быть 4-х типов: текст, кнопка, ссылка, вложение.

Для создания или редактирования элементов воспользуйтесь кнопками на тулбаре в панели со списком элементов (редактирование элемента так же можно открыть по двойному клику на элементе)



Для удобства навигации можно отображать элементы какого-либо одного типа. Для этого нужно выбрать типа элемента в выпадающем списке справа от тулбара.

Текст

В элементе типа текст можно указывать макросы в текст `${#firstName}`, `${#lastName}`, `${#status}` и т.д.

Затем в обработчике диалога, перед отображением диалога, в методе `preDialog(ServerContext serverContext, JSONObject chatData)` заменить макросы на необходимые значения.

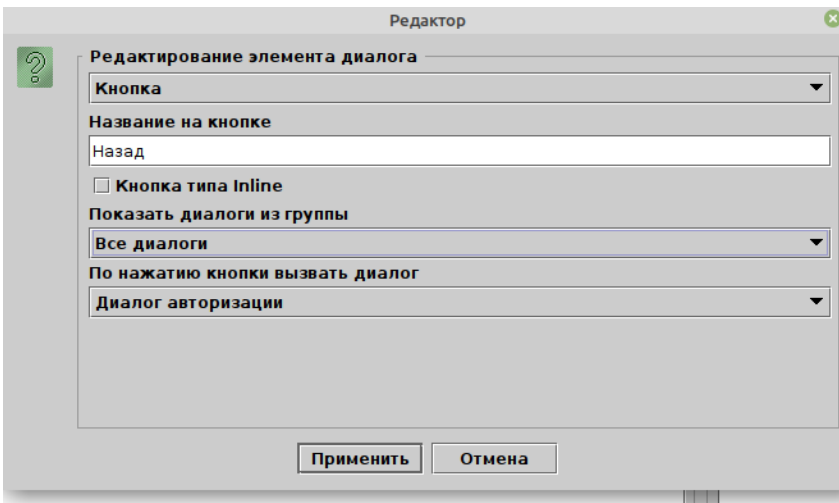
Для этого необходимо создать `JSONObject`, в котором ключи будут самими макросами, а value - значение, которое необходимо установить вместо макроса.

Пример

```
JSONObject macros = new JSONObject();
macros.put( "firstName", Utils.maskBlank( cpm.getStringParam( cid, FIRST_NAME_PID ), "" ) );
macros.put( "lastName", Utils.maskBlank( cpm.getStringParam( cid, LAST_NAME_PID ), "" ) );
macros.put( "status", contract.getStatus() == 0 ? "" : "" );
macros.put( "balance", balanceUtils.getBalance( new Date(), cid ).toString() );
// MACROS("macros") - DialogProcessing
chatData.put( MACROS, macros );
```

Кнопка

При создании элемента с типом кнопка необходимо настроить несколько значений



Название - текст, который будет отображён на самой кнопке. Если стоит флаг "inline" - кнопка будет частью сообщения, а не создаваться отдельно в качестве клавиатуры.

По нажатию на кнопку может быть отображён какой-либо другой диалог, но это не обязательное условие.

Вложение

Вложением может быть любой файл размером не более 20мб. Вложение будет отправляться после всех остальных элементов в диалоге. Т.е сначала отправляется текст, ссылки, а затем вложения для данного диалога.

Отправка собственных сообщений в чаты

Для отправки собственных сообщений в чаты ботов (например из дин.кода) есть реализация класса `ru.bitel.bgbilling.modules.botmanager.server.bots.BotSender` с одним единственным методом `send(BotSender.BotMessage msg)`, который принимает сообщение для отправки.

Пример построения и отправки сообщения

```
new BotSender().send( BotSender.BotMessage.createMessage()  
                                .setText( "- " )  
                                .setContract( 31432 )  
                                .setContract( 22133 )  
                                .setBotId( 2 ) );
```

Класс BotMessage и описание полей

```
/**  
 *  
 */  
public static final class BotMessage  
{  
    /**  
     * ID ,  
     */  
    private Set<Integer> contracts;  
  
    /**  
     * ID .  
     */  
    private List<Long> userIds;  
  
    /**  
     * . , .  
     * , contracts  
     */  
    private Set<Integer> contractLabelIds;  
  
    /**  
     * , ,  
     *  
     */  
    private int botId;  
  
    /**  
     * .  
     * Dialog,  
     *  
     */  
    private String text;  
  
    /**  
     * , BotManager.  
     * ,  
     */  
    private BotDialog dialog;  
  
    private BotMessage() {}  
  
    public static BotMessage createMessage()  
    {  
        return new BotMessage();  
    }  
  
    public BotMessage setContracts(Set<Integer> contracts)  
    {  
        this.contracts = contracts;  
    }  
}
```

```

        return BotMessage.this;
    }

    /**
     *
     *
     */
    public BotMessage setContract(int contractId)
    {
        if( this.contracts == null )
            this.contracts = new HashSet<>();

        this.contracts.add( contractId );

        return BotMessage.this;
    }

    public BotMessage setContractLabelIds( List<Integer> labelIds )
    {
        this.contractLabelIds = new HashSet<>(labelIds);
        return BotMessage.this;
    }

    public BotMessage setUserIds(List<Long> userIds)
    {
        this.userIds = userIds;
        return BotMessage.this;
    }

    public BotMessage addUserId(long userId)
    {
        if( userIds == null )
            userIds = new ArrayList<>();

        userIds.add( userId );
        return BotMessage.this;
    }

    public BotMessage setBotId( int botid )
    {
        this.botId = botid;
        return BotMessage.this;
    }

    public BotMessage setText( String text )
    {
        this.text = text;
        return BotMessage.this;
    }

    public BotMessage setDialog( BotDialog dialog )
    {
        this.dialog = dialog;
        return BotMessage.this;
    }
}

```

Если передаётся список id договоров, то их id в мессенджерах будут браться из статистики в ботах, если ведение статистики не было включено в конфигурации модуля, то для какого-то договора сообщение может не быть отправлено, даже если клиент авторизовался через бота.

Модуль RCMts

Модуль предназначен для интеграции с платформой от МТС - "Красный Конвергент".

В интерфейсе BGBillingClient в конфигурации модуля осуществляется настройка для интеграции. Пример и описание конфигурации приведён ниже.

Параметры фио. Если ФИО абонента хранится в одном параметре, то указать только его

```
fiо.pid=

firstname.pid=
lastname.pid=
middlename.pid=

#id
address.string.pid=

# id параметра типа адрес
address.pid=

#данное значение определено мтс и должно быть указано как в текущем примере

payment.method.default=2

#id статуса договора при котором абонент считается активным
active.status=0

#id статуса договора при котором абонент считается заблокированным
block.status=2

#автоматическое добавление модуля на договор, если на нём нет модуля
add.module.auto=true

#префикс названия договора, который выдаёт МТС для каждого оператора
mts.account.prefix=51911

#список id модулей инета
inet.modules.ids=1

#список id модулей ТВ
tv.modules.ids=26

# id тарифных опций через запятую, которые относятся к мтс конвергент
mts.tariff.options.ids=17,18,19

#запись об услугах модуля прау
#service.{id услуги}=id услуги в мтс(gaar)
service.75=80044555
service.76=80044333

#запись о типах расходов
#charge.type.{id типа расхода}=id услуги в мтс(gaar)
charge.type.62=80044777
charge.type.58=80044999
charge.type.63=80044888

default.payment.type=64
```

В динамическом коде поставляется класс **ru.bitel.bgbilling.modules.rcmts.dyn.RCMtsResponseBuilderDefault**, который является реализацией логики взаимодействия с МТС. Данный класс реализует интерфейс **ru.bitel.bgbilling.modules.rcmts.server.bean.RCMtsResponseBuilder**.

При необходимости можно создать свою реализацию (или наследовать RCMtsResponseBuilderDefault и изменить поведение) данного интерфейса и в таком случае будет работать пользовательская реализация.

Так же по схеме взаимодействия с МТС, если у оператора есть доп.услуги или разовые начисления, то необходимо в планировщике установить задачу модуля - "Задача на выгрузку расходов в таблицу cdr для МТС". Данная задача должна быть настроена на запуск 1 раз в сутки **после задач модуля Прау**.

Для обеспечения возможности для мтс работы с абонентами оператора необходимо:

-завести тарифную опцию и настроить режим активации, например, как указано на скриншоте ниже

Редактор

Название МТС

Период с 02.08.2022 по

Активация на 0 минут с текущего момента ровно

Деактивация моментально Реактивация запрещена

Удаление разрешено для оператора Удаление расхода удалять

Снять расход 0.00 Опция

Ok Отмена

Сохранить опцию Отмена

-добавить созданную тарифную опцию в тариф

-добавить модуль на договор абонента

-добавить тариф с тарифной опцией абоненту

-добавить тарифную опцию абоненту

Плагины

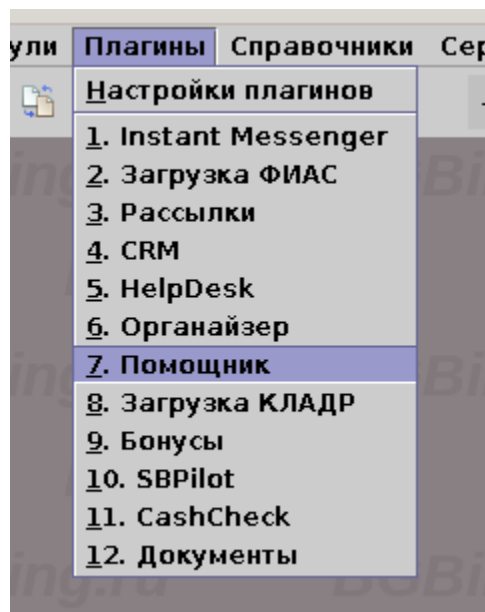
Плагин Assistant

Назначение плагина

Плагин позволяет настраивать ряд мастеров для выполнения часто повторяемых действий с возможностью их контроля. Действия можно фильтровать по пользователям биллинга.

Настройка плагина

Для настройки мастеров используется соответствующий редактор вызвать который можно из меню **Плагины - Помощник**.



Добавить новое действие можно с помощью кнопки  на тулбаре, в результате откроется следующий редактор

Редактор

Action ID (#): debtorsTable

Дочерний элемент: (если будет использоваться как не основное окно Мастера)

Название: Таблица должников

Поля:

Ключ	Тип	Название	Значение	Комментарий
debtorsTable	table			

Комментарий

Класс обработчика

Группы пользователей

Параметры:

Параметр	Значение
Размер окна	фиксированное (ШХВ)
Ширина окна	1200
Высота окна	900
Показать кнопку КНОПКА_1	<input type="checkbox"/>
Текст кнопки КНОПКА_1	
Показать кнопку КНОПКА_2	<input type="checkbox"/>
Текст кнопки КНОПКА_2	
Показать кнопку КНОПКА_3	<input type="checkbox"/>
Текст кнопки КНОПКА_3	
Показать кнопку КНОПКА_4	<input type="checkbox"/>
Текст кнопки КНОПКА_4	
Показать кнопку ОТМЕНА	<input type="checkbox"/>
Текст кнопки ОТМЕНА	
nextActionId	

Ok Отмена

- в нем в поле **Action ID** необходимо указать текстовую строку на которую в дальнейшем нужно ссылаться при переходах между формами.
- если данный мастер будет использоваться, как не главное окно, то можно установить признак **Дочерний элемент**, в дальнейшем это позволит скрывать дочерние элементы в главном окне плагина для более удобной навигации
- в поле **Название** вводится название формы, оно будет отображаться в списке доступных действия и в заголовке формы
- в поле **комментарий** вводится текст подсказки, который будет выводиться в форме (поясняющий в общей форме что и для чего нужно заполнить), если поле оставить пустым подсказка выводится не будет.
- в таблицу **Поля** добавляются поля которые будут отображаться в форме
- в таблице **Параметры** можно задать параметры отображения формы
- на закладке **Группы пользователей** задаются для пользователей каких групп отображать данное действие
- на закладке **Класс обработчика** можно выбрать класс обработчика формы или создать новый

Комментарий

Класс обработчика

Группы пользователей

Выберите класс обработчика:

ru.bitel.bgbilling.assistant.contract.CreateContractStep1

Создать новый класс обработчика:

Создать...

Комментарий

Класс

WiFi

Гости

Группа 1

Группа 2

Контроллеры

Могучие Администр

может всё, или поч

Редактор полей выглядит следующим образом

Ключ:

Тип: не установлен ▾

Название:

Значение:

Комментарий:

Применить Отмена

где

- **ключ** - некая уникальная в пределах формы строка которую можно использовать в коде для доступа к значению поля
- **тип** - тип поля, например строка, дата, флаг, период и т.д.
- **название** - название поля выводимое в форме
- **значение** - значение поля по умолчанию
- **комментарий** - подсказка выводимая при получении фокуса данным полем

Поля:

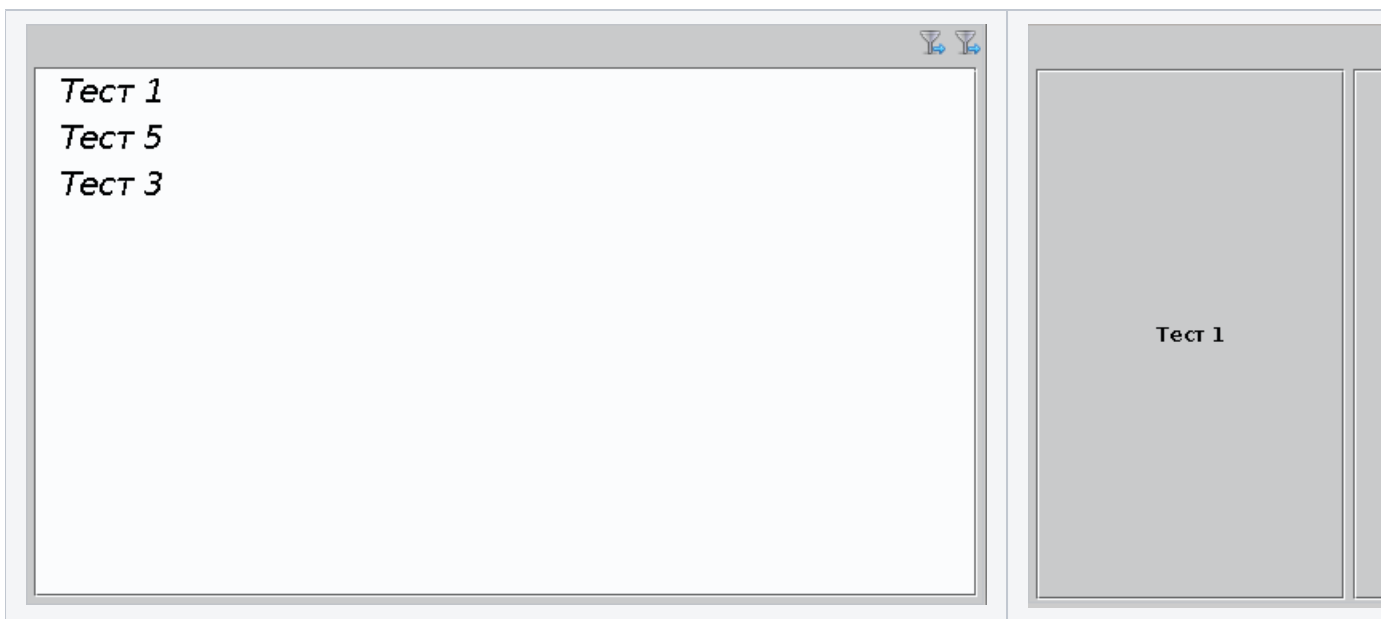
Ключ	Тип	Название	Значение	Комментарий
fio	string	ФИО		Введите Фамилию и инициалы
bdate	date	Дата рождения		Не обязательно для заполнения

последовательность полей на форме можно менять при помощи кнопок со стрелками

Работа с плагином

Помощник

Для вызова доступных действий необходимо кликнуть по кнопке **Помощник** на тулбаре, в результате будет открыто окно с доступными для пользователя действиями, окно имеет два режима отображения (список и кнопки)



выбрав необходимое действие в списке или нажав на кнопку, откроется соответствующий мастер

ФИО	<input type="text"/>
Дата рождения	<input type="text"/>
<input type="button" value="OK"/>	


после заполнения полей и нажатия на кнопку данные будут переданы в заданный обработчик где они могут быть проверены и обработаны. Если при проверке в данных обнаружены ошибки, может быть возвращена ошибка с предложением повторно ввести данные. После обработки формы, можно выполнить какие действия или перейти к следующей форме для ввода дополнительных данных.

Заполнение элементов формы

Выпадающий список

Список

value1
value2

Поля: 

Ключ	Тип	Название	Значение	Комментарий	...
combobox	combobox	Выпадающий список			
list	list	Список			

```
@Override
public AssistantResponse showAction( AssistantRequest assistantRequest )
    throws BGException, BGMessageException
{
    this.assistantRequest = assistantRequest;
    assistantResponse.addFieldData( "combobox", new AssistantKeyValue[] { new AssistantKeyValue(
"key1", "value1" ), new AssistantKeyValue( "key2", "value2" ) } );
    assistantResponse.addFieldData( "list", new AssistantKeyValue[] { new AssistantKeyValue( "key1",
"value1" ), new AssistantKeyValue( "key2", "value2" ) } );
    assistantResponse.addFieldData( "date", new AssistantKeyValue[] { new AssistantKeyValue(
"date", "01.01.2020" ) } );
    assistantResponse.addFieldData( "period", new AssistantKeyValue[] { new AssistantKeyValue(
"dateFrom", "01.01.2020" ), new AssistantKeyValue( "dateTo", "31.12.2020" ) } );
    assistantResponse.addFieldData( "hidden", "" );
    return assistantResponse;
}
```

Элемент - список с возможностью выбора.

В данном примере **listSelector** - ключ, который был задан при конфигурации мастера в графическом интерфейсе плагина

Код

```
JSONArray items = new JSONArray();
for( int i = 1; i < 6; i++ )
{
    JSONObject o = new JSONObject();
    o.put( "id", String.valueOf( i ) );
    o.put( "title", " - " + i );
    o.put( "selected", i%2 == 0 );
    items.put( o );
}
assistantResponse.addFieldData( "listSelector", new AssistantKeyValue[]{ new AssistantKeyValue
("listSelector", items.toString()) } );
```

Получение значений из списка

Выберите что то

- Название - 1 [#1]
- Название - 2 [#2]
- Название - 3 [#3]
- Название - 4 [#4]
- Название - 5 [#5]

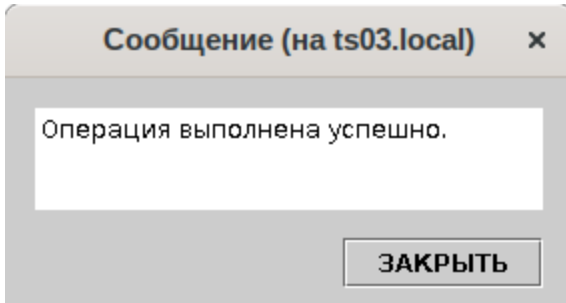
Получение элементов из списка

```
JSONArray items = new JSONArray( assistantRequest.getFields().optString( "listSelector" ) );
for( int i = 0; i < items.length(); i++ )
{
    JSONObject o = items.getJSONObject( i );
    int id = Utils.parseInt(o.optString( "id" ));
    String title = o.optString( "title" );
    boolean isSelected = o.optBoolean( "selected", false );
}
```

Примеры Действий

Окно вывода сообщения

Простое окно может быть использовано как окно сообщения о успешном выполнении действия или в окно может быть выведен результат выполненного действия.



Ниже показана настройка данной формы

Редактор

Action ID (#):

Название:

Поля:

Ключ	Тип	Название	Значение	Комментарий...
------	-----	----------	----------	----------------

Группы пользователей

Комментарий | **Класс обработчика**

Операция выполнена успешно.

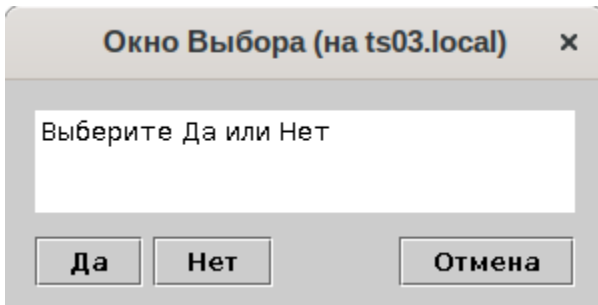
Параметры:

Параметер	Значение
Размер окна	
Ширина окна	300
Высота окна	150
Показывать кнопку ОК	<input type="checkbox"/>
Текст кнопки ОК	
Показывать кнопку НАЗАД	<input type="checkbox"/>
Текст кнопки НАЗАД	
Показывать кнопку ДАЛЕЕ	<input type="checkbox"/>
Текст кнопки ДАЛЕЕ	
Показывать кнопку ОТМЕНА	<input checked="" type="checkbox"/>
Текст кнопки ОТМЕНА	ЗАКРЫТЬ
nextActionId	

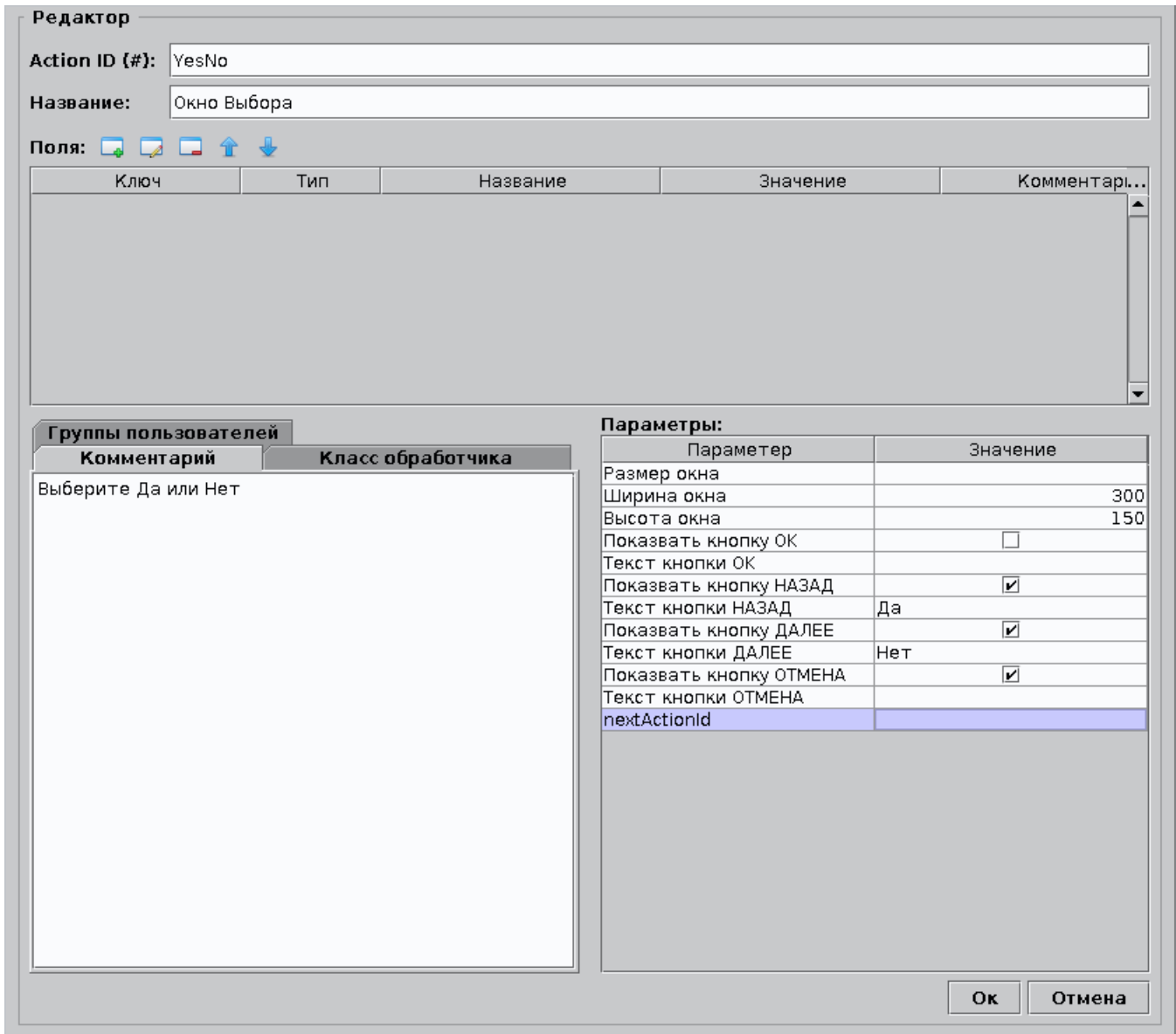
Ок Отмена

в настройке указаны размеры окна, переименована кнопка отмены и задан текст сообщения.

Окно выбора



настройка окна



Класс Обработчика

```
package assistant;  
  
import ru.bitel.bgbilling.common.BGException;
```

```
import ru.bitel.bgbilling.common.BGMessageException;
import ru.bitel.bgbilling.plugins.assistant.server.bean.AssistantActionBase;
import ru.bitel.bgbilling.plugins.assistant.server.bean.AssistantRequest;
import ru.bitel.bgbilling.plugins.assistant.server.bean.AssistantResponse;

public class YesOrNot
    extends AssistantActionBase
{
    @Override
    public void doButtonPrev()
    {
        assistantResponse.setNextActionId( "yes" );
    }

    @Override
    public void doButtonNext()
    {
        assistantResponse.setNextActionId( "no" );
    }
}
```

Создание таблицы в мастере

Если таблица содержит колонку с названием "contractId" Результаты действий мастера можно вывести в таблице. Для этого нужно добавить элемент "таблица" в редакторе мастера. В данном примере будет показан весь путь создания небольшого "мастера".

Допустим, что нужен мастер, который отображает в таблице список абонентов с отрицательным балансом, и пополнение счёта от которых не было более 2 месяцев

Редактор

Action ID (#): findDebtorsForm

Дочерний элемент: (если будет использоваться как не основное окно Мастера)

Название: Поиск должников

Поля:

Ключ	Тип	Название	Значение	Комментарий
correctSum	integer	Остаток на счёте		Введите остаток, меньше которого искать
datePayment	date	Дата последней оплаты		Введите дату последней оплаты

Комментарий | Класс обработчика | Группы пользователей

Выберите класс обработчика:

- assistant.InitComboBox
- ru.bitel.bgbilling.assistant.contract.CreateContractStep1
- ru.bitel.bgbilling.assistant.contract.DebtorsTableResult
- ru.bitel.bgbilling.assistant.contract.FindContractResultTable
- ru.bitel.bgbilling.assistant.contract.FindDebtorsForm
- ru.bitel.bgbilling.assistant.contract.TestAssist

Создать новый класс обработчика:

ru.bitel.bgbilling.assistant.contract.FindDebtorsForm

Создать...

Параметры:

Параметр	Значение
Размер окна	авто
Ширина окна	
Высота окна	
Показать кнопку КНОПКА_1	<input checked="" type="checkbox"/>
Текст кнопки КНОПКА_1	
Показать кнопку КНОПКА_2	<input type="checkbox"/>
Текст кнопки КНОПКА_2	Искать
Показать кнопку КНОПКА_3	<input type="checkbox"/>
Текст кнопки КНОПКА_3	Отмена
Показать кнопку КНОПКА_4	<input type="checkbox"/>
Текст кнопки КНОПКА_4	
Показать кнопку ОТМЕНА	<input checked="" type="checkbox"/>
Текст кнопки ОТМЕНА	Отмена
nextActionId	

Пример обработчика окна с вводом параметров

```
public class FindDebtorsForm
    extends AssistantActionBase
{
    @Override
    public AssistantResponse doAction( AssistantRequest assistantRequest )
        throws BGException, BGMessageException
    {
        assistantResponse.setNextActionId( "debtorsTable" ); // Action ID ,
        return super.doAction( assistantRequest );
    }

    @Override
    public AssistantResponse showAction( AssistantRequest assistantRequest )
        throws BGException, BGMessageException
    {
        this.assistantRequest = assistantRequest;
        return assistantResponse;
    }

    @Override
    public void doButton1()
    {
        super.doButton1();
    }

    @Override
    public void doButton2()
    {
        try
        {
            doAction( assistantRequest );
        }
    }
}
```

```

    }
    catch( BGEException e )
    {
        e.printStackTrace();
    }
}

```

Создаём окно с таблицей, указывая **Action ID: debtorsTable**

Добавим элемент "таблица" с ключем: debtorsTable (ключ может быть любой. Если таблиц несколько, то у каждой должен быть свой уникальный ключ)

Редактор

Action ID (#): debtorsTable

Дочерний элемент: (если будет использоваться как не основное окно Мастера)

Название: Таблица должников

Поля:

Ключ	Тип	Название	Значение	Комментарий
debtorsTable	table			

Комментарий | Класс обработчика | Группы пользователей

Выберите класс обработчика:

- assistant.InitComboBox
- ru.bitel.bgbilling.assistant.contract.CreateContractStep1
- ru.bitel.bgbilling.assistant.contract.DebtorsTableResult**
- ru.bitel.bgbilling.assistant.contract.FindContractResultTable
- ru.bitel.bgbilling.assistant.contract.FindDebtorsForm
- ru.bitel.bgbilling.assistant.contract.TestAssist

Создать новый класс обработчика:

ru.bitel.bgbilling.assistant.contract.DebtorsTableResult

Параметры:

Параметр	Значение
Размер окна	авто
Ширина окна	1200
Высота окна	900
Показать кнопку КНОПКА_1	<input type="checkbox"/>
Текст кнопки КНОПКА_1	
Показать кнопку КНОПКА_2	<input type="checkbox"/>
Текст кнопки КНОПКА_2	
Показать кнопку КНОПКА_3	<input type="checkbox"/>
Текст кнопки КНОПКА_3	
Показать кнопку КНОПКА_4	<input type="checkbox"/>
Текст кнопки КНОПКА_4	
Показать кнопку ОТМЕНА	<input type="checkbox"/>
Текст кнопки ОТМЕНА	
nextActionId	

При запуске мастера "Поиск должников", появится такое окно для задания условий.

Поиск должников
✕

Остаток на счёт

-30

Дата последней оплаты

28.10.2019

Искать

Отмена

В обработчике мы можем получить введённые в мастере данные и произвести необходимые операции с ними. В данном примере показано, как можно получить данные и пример некоторых действий.

Для того, чтобы передать список собственных классов в механизм создания таблицы, нужно передать полный путь до созданного класса, также нужно преобразовать список в формат JSON,

воспользовавшись методом `mapperWrite(List<? extends FilterTableModel> data)` из класса `ru.bitel.bgbilling.plugins.assistant.server.AssistantUtils`

Пример обработчика окна с таблицей.

```

public class DebtorsTableResult
    extends AssistantActionBase
{
    @Override
    public AssistantResponse showAction( AssistantRequest assistantRequest1 )
        throws BGException, BGMessageException
    {
        this.assistantRequest = assistantRequest1;

        //
        JSONObject data = assistantRequest.getData().getJSONObject( "fields" );
        int balanceLimit = Integer.parseInt( data.getString( "correctSum" ) );
        LocalDate searchDate = LocalDate.parse( data.getString( "datePayment" ) );

        //      ,      JSON
        String result = getDataOfTable( balanceLimit, searchDate );
        assistantResponse.addFieldData( "debtorsTable", new AssistantKeyValue[]{ new AssistantKeyValue( "ru.
bitel.bgbilling.assistant.contract.DebtorTableFilter", result ) } );

        return super.showAction( assistantRequest );
    }

    private String getDataOfTable( int balanceLimit, LocalDate startSearchDate )
    {
        List<DebtorTableFilter> debtors = new ArrayList<>();
        Connection con = ServerContext.get( ServerContext.class ).getConnection();
        String query = "SELECT c.id, c.title, cp.dt FROM contract AS c "
            + "LEFT JOIN contract_payment AS cp ON cp.cid=c.id WHERE cp.dt > ? GROUP BY cid ORDER BY
cp.id DESC";
    }
}

```

```

try( PreparedStatement ps = con.prepareStatement( query ) )
{
    ps.setDate( 1, Date.valueOf( startSearchDate ) );
    ResultSet rs = ps.executeQuery();

    while( rs.next() )
    {
        DebtorTableFilter debtor = new DebtorTableFilter();
        debtor.setContractId( rs.getInt( "c.id" ) );
        debtor.setContractTitle( rs.getString( "c.title" ) );
        LocalDate localDate = rs.getDate( "cp.dt" ).toLocalDate();
        debtor.setLastPaymentDate( localDate.toString() );

        debtors.add( debtor );
    }
}
catch( SQLException e )
{
    e.printStackTrace();
}

// ,
BalanceUtils balanceUtils = new BalanceUtils( con );
Iterator<DebtorTableFilter> iterator = debtors.iterator();
while( iterator.hasNext() )
{
    DebtorTableFilter deb = iterator.next();
    BigDecimal balance = balanceUtils.getBalance( new java.util.Date( ), deb.getContractId() );
    if( balance.compareTo( new BigDecimal( balanceLimit ) ) >= 0 )
    {
        iterator.remove();
    }
    else
    {
        deb.setBalance( balance );
    }
}

//
for( DebtorTableFilter deb : debtors )
{
    String query1 = "SELECT mail.email, phone.value FROM contract AS c LEFT JOIN
contract_parameter_type_3 AS mail ON mail.cid=c.id LEFT JOIN contract_parameter_type_phone AS phone ON phone.
cid=c.id WHERE c.id=?";
    try( PreparedStatement ps = con.prepareStatement(query1) )
    {
        ps.setInt(1, deb.getContractId() );

        ResultSet rs = ps.executeQuery();

        while( rs.next() )
        {
            deb.setEmail( rs.getString( "mail.email" ) );
            deb.setTelephone( rs.getString( "phone.value" ) );
        }
    }
    catch( SQLException ex )
    {
        e.printStackTrace();
    }
}

// List JSON-
return AssistantUtils.mapperWrite( debtors );
}
}

```

Для того, чтобы данные корректно отображались в таблице, необходимо в дин.коде создать собственный класс, который должен быть наследником абстрактного класса **ru.bitel.bgbilling.plugins.assistant.common.bean.FilterTableModel**.

Пример пользовательского класса, который будет отображён в таблице

```
@JsonAutoDetect
public class DebtorTableFilter
    extends FilterTableModel
{
    private int contractId;
    private String contractTitle;
    private String telephone;
    private String email;
    private String lastPaymentDate;
    private BigDecimal balance;

    @Override
    public String getCorrectTitleColumn( int columnId )
    {
        switch( columnId )
        {
            case 0: return "ID ";
            case 1: return "";
            case 2: return "";
            case 3: return "";
            case 4: return " ";
            case 5: return " ";
            default: return "< >";
        }
    }

    @Override
    public String getCorrectData( int columnId )
    {
        return null;
    }

    public int getContractId()
    {
        return contractId;
    }

    public void setContractId( int contractId )
    {
        this.contractId = contractId;
    }

    public String getContractTitle()
    {
        return contractTitle;
    }

    public void setContractTitle( String contractTitle )
    {
        this.contractTitle = contractTitle;
    }

    public String getTelephone()
    {
        return telephone;
    }

    public void setTelephone( String telephone )
    {
        this.telephone = telephone;
    }

    public String getEmail()
    {
        return email;
    }
}
```

```

public void setEmail( String email )
{
    this.email = email;
}

public String getLastPaymentDate()
{
    return lastPaymentDate;
}

public void setLastPaymentDate( String lastPaymentDate )
{
    this.lastPaymentDate = lastPaymentDate;
}

public BigDecimal getBalance()
{
    return balance;
}

public void setBalance( BigDecimal balance )
{
    this.balance = balance;
}
}

```

При необходимости, можно реализовать абстрактный метод **getCorrectData()** и передать в таблицу данные для определённого столбца. Например, есть необходимость показать в таблицы текущий статус для договоров, с помощью этого метода, можно подменить число статуса на более понятный "Активен". Если нет необходимости в "подмене" данных, метод должен возвращать null.

Второй метод **getCorrectTitleColumn()** из родительского FilterTableModel должен возвращать корректное название для колонок таблицы.

Итоговый результат работы мастера показан ниже

ID	Название	Телефон	Почта	Последняя оплата	Текущий баланс	...
762913	valera_test	8 (917) 7777777[изменен]	adsasdad <vtoroikak@yandex.r...	2019-12-09	-467.01	▲

Если таблица содержит колонку с названием "contractId", "cid", "idContract" или "ID Договора", то при клике на строке, будет выполняться поиск договора с таким id в БД, если договор будет найден, он будет открыт.

Есть возможность передать выбранную строку в обработчик следующего Мастера. Для этого нужно выбрать в таблице строку, при клике правой кнопкой мыши появится меню с пунктом "Сохранить строку для обработчика следующего Мастера", при клике данные будут сохранены и будет возможность получить их в следующем обработчике. Пример получения показан ниже

Пример получения выбранных в таблице данных


```

public class TestViewSelectedRowData
    extends AssistantActionBase
{
    @Override
    public AssistantResponse showAction( AssistantRequest assistantRequest1 )
        throws BGException, BGMessageException
    {
        this.assistantRequest = assistantRequest1;

        //      data
        JSONObject data = assistantRequest.getData().getJSONObject( "fields" );
        if( data.has( "debtorsTable" ) )
        {
            //      Json
            JSONObject selected = data.getJSONObject( "debtorsTable" );

            //      -      .
            String contractTitle = (String)selected.get( "contractTitle" );
            BigDecimal balance = new BigDecimal( selected.get( "balance" ).toString() );
            String email = (String)selected.get( "email" );
        }

        return super.showAction( assistantRequest );
    }
}

```

Выбранная в таблице строка будет сохранена под ключом самой таблицы, в данном примере ключом является строка "debtorsTable".
 Конкретные данные для выбранной в таблице строки сохраняются под ключом поля объекта, который был представлен в таблице.
 В данном случае в объекте DebtorTableFilter есть поля: contractTitle, balance и email. Используя названия этих полей как ключи,
 можно получить значения.

Мобильное приложение для плагина

Для устройств с ОС Android доступно мобильное приложение плагина Assistant <https://play.google.com/store/apps/details?id=ru.bitel.bgassistantmobile>

Для начала работы необходимо настроить подключение к серверу.

Для этого нужно перейти в настройки, указать URL к серверу, логин и пароль.



После чего на главном экране будет отображён список всех доступных для данного авторизованного пользователя "мастеров" плагина.



Assistant Mobile

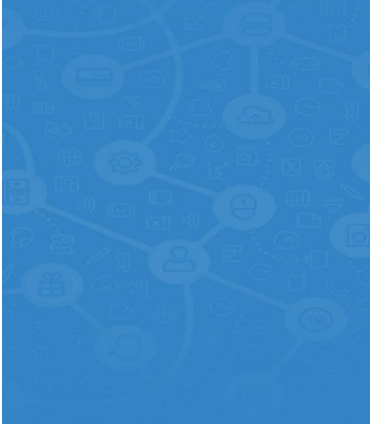
Поиск договора

Создание нового договора

Тестирование скрытых полей

Поиск должников

TestLink



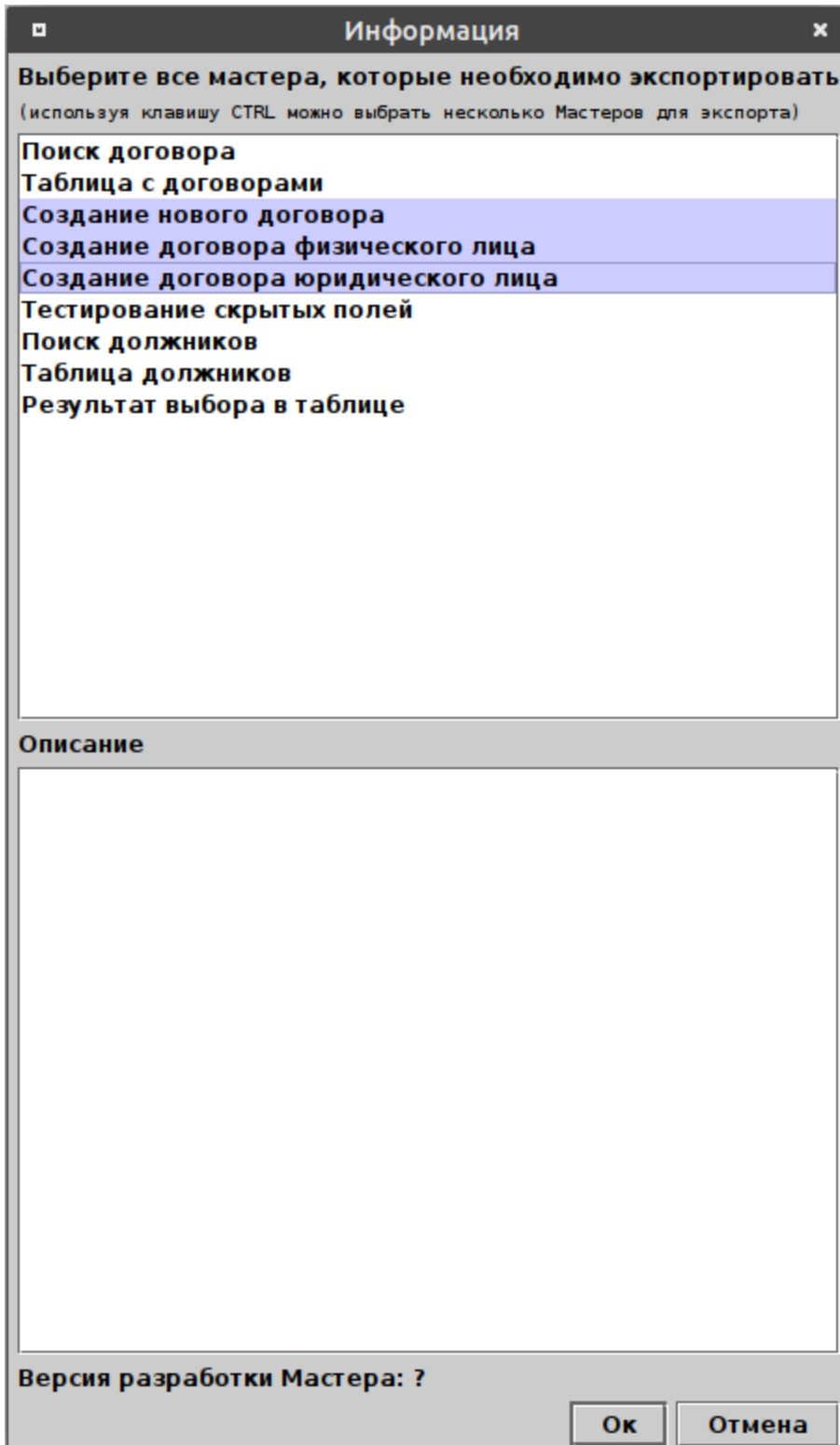
Экспорт/Импорт мастеров плагина

Экспорт

Для того, чтобы экспортировать готовые мастера для плагина Ассистент, нужно на панели со списком всех мастеров кликнуть правой кнопкой мыши и выбрать пункт меню "Экспортировать".

После чего будет открыто окно для выбора мастеров для экспорта. Поле "описание" опционально и необходимо только для показа оставленного описания при последующем импорте в плагин.

После нажатия "Ок", будет предложено выбрать директорию для сохранения мастеров в формате zip-архива, в котором будут присутствовать: файлы описания самих мастеров, файл с оставленным описанием, классы-обработчики из динамического кода.

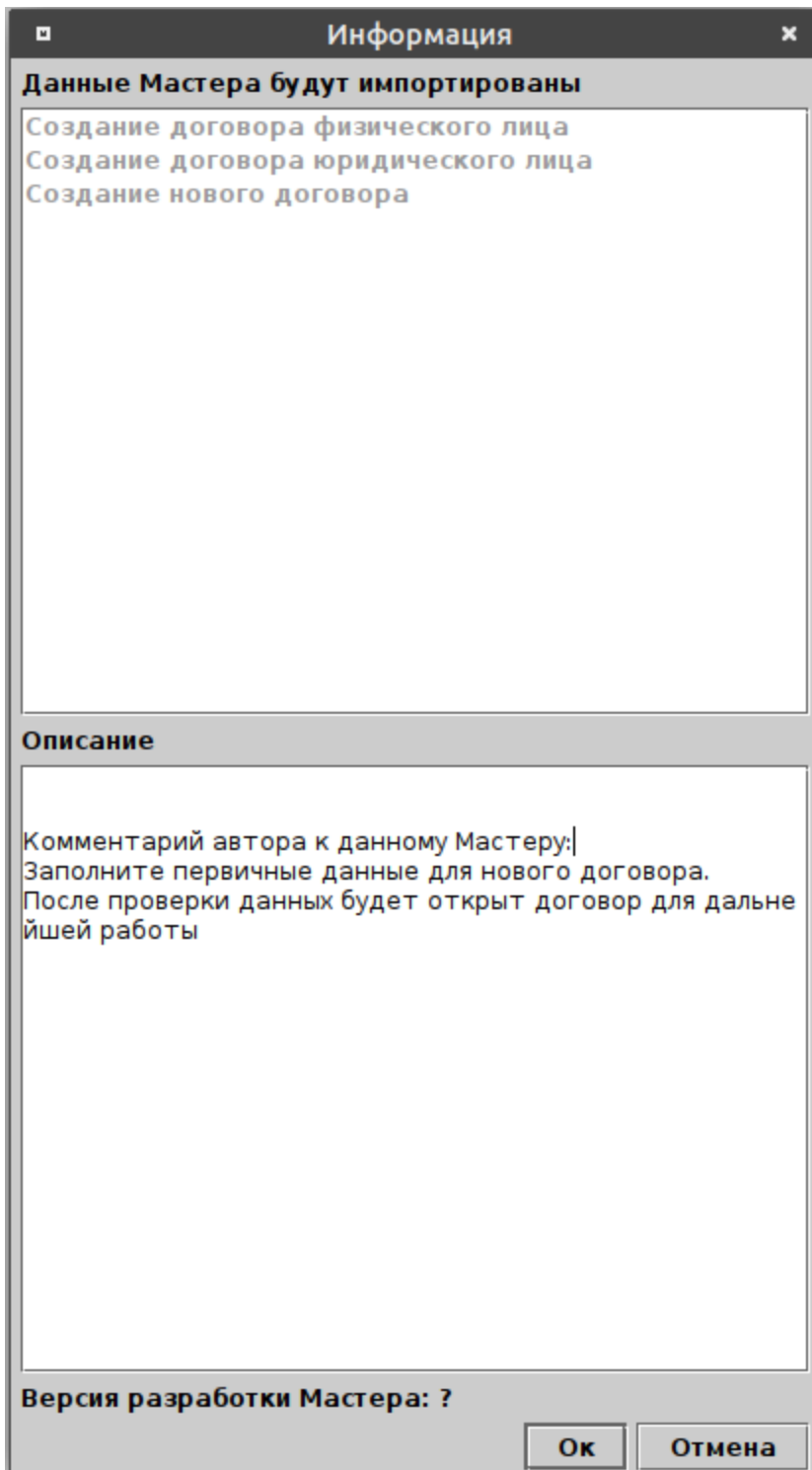


Импорт

Для того, чтобы импортировать готовые мастера в плагин Ассистент, нужно на панели со списком всех мастеров кликнуть правой кнопкой мыши и выбрать пункт меню "Импортировать".

После чего будет открыто окно для выбора zip-файла с расширением **.action**. После выбора файла будет показано окно информацией о импортируемом мастере.

После нажатия кнопки "Ок" мастера будут импортированы в плагин, а классы-обработчики будут сохранены в динамическом коде. (Классы-обработчики будут помещены в директории в соответствии с их полным путём при экспорте)



Плагин Bonus

Назначение плагина

Плагин позволяет начислять и расходовать бонусные баллы для договора. В зависимости от бонусной программы баллы могут начисляться как процент от дохода, расхода, наработки или зависеть от определенных событий.

Настройка плагина

Плагин устанавливается стандартным образом. В настройках плагина (**Сервис->Настройка Плагинов->Bonus**) задайте конфигурацию:

```
# , rate = -
rate=10
# , ,
charge=44,54,39,55
#
charge.percent.44=50
charge.percent.54=35
charge.percent.39=0
charge.percent.55=100
# -
charge.percentDefault=20
# web-e
tariffOptionsPaymentOnlyWithBonuses=10,3,5
#
dynamicBonusPrograms=ru.bitel.bgbilling.bonus.myPrograms.FirstProgram,ru.bitel.bgbilling.bonus.myPrograms.
SecondProgram
```

Например, пусть $rate = 2$ и создается расход на 100 рублей. Тогда максимальное количество бонусных баллов, которыми можно оплатить, равно 50.

Если код расхода не указан, то берется значение по умолчанию, если значение по умолчанию отсутствует, то будет возвращен 0 (то есть оплата бонусами будет запрещена).

Также необходимо настроить местоположение вкладки "Бонус" в дереве договора с ключом **bonus** в [настройках сервера](#), если вы заводи данный параметр(иначе вкладка не будет отображаться в договоре) и после редактирования перезапустите клиент.

```
client.gui.contract.tree.order=parameters objects hierarchy status limit mode face balance tariff modules
groups web tariffGroup bonus script addAction memo
```

Бонусные приходы

Бонусные приходы можно просмотреть во вкладке "Бонус" в дереве вкладок договора. При выборе прихода в нижней таблице будут отображены все расходы, которые списали у данного прихода, если таковые имеются.

The screenshot shows a software interface with a left sidebar containing a tree view of account categories. The main area displays account balances and two tables of transactions.

Текущий период: с 08.2012 по 08.2012

Выбор периода: Договор, Месяц, Квартал, Год, 3 Мес., 6 Мес., 12 Мес.

Баланс (август 2012):

- Входящий остаток: -1075394.96
- Приход: 0.00
- Наработка: 0.00
- Расход: 94.00
- Исходящий остаток: -1075488.96

Таблица Приход:

ID	Тип платежа	Дата	Сумма	От даты	До даты
10	первый тип	17.08.2012	35	17.08.2012	14.09.2012

Таблица Расход:

ID	Тип расхода	ID расхода договора	Дата	Сумма
263	Подписка на Антивирус...	869	17.08.2012	17

Слева: Лимит: 0.00, Режим: дебет, Лицо: Физическое, Бонус: 49/+0

Для добавления бонусного платежа нажмите в режиме просмотра платежей кнопку **Новый элемент** на стандартной панели инструментов клиента биллинга. Необходимо выбрать сумму, тип платежа и промежуток времени, на который данный платеж будет активен.

The screenshot shows the "Редактор приходов" dialog box. It contains a dropdown menu for "Тип платежа" (set to "первый тип"), a text field for "Сумма" (567), and date pickers for "с" (21.08.2012) and "по" (30.08.2012). There are "Ок" and "Отмена" buttons at the bottom right.

Типы платежей предварительно добавляются в справочнике, доступном через пункт меню **Справочники->Другие->Bonus-приходы**

The screenshot shows the "Справочники" dialog box. The left sidebar lists various categories, with "Bonus - приходы" selected. The main area displays a table of payment types.

ID	Название	Используется
2	первый тип	<input checked="" type="checkbox"/>
3	второй т	<input type="checkbox"/>

Редактор: Название: new Bonus payment type, Используется. Buttons: "Ок", "Отмена".

Бонусные расходы

Бонусные расходы можно просмотреть во вкладке "Бонус" в дереве вкладок договора при нажатии на кнопку "Расход". Расходы будут отображаться только за выбранный период, который устанавливается выше. При выборе расхода в нижней таблице будут отображены все приходы, у которых данный расход списал бонусы.

The screenshot shows a software interface with a sidebar on the left containing a tree view of account categories. The main area displays a table of transactions. The top navigation bar includes a date range selector (08.2012) and a period selector (Договор, Месяц, Квартал, Год, 3 Мес., 6 Мес., 12 Мес.). Below the navigation bar are three tabs: Приход, Расход, and Баланс. The main table has columns for ID, Тип расхода, ID расхода договора, Дата, and Сумма. A second table below it shows details for a specific transaction, including ID, Тип платежа, Дата, Сумма, От даты, and До даты.

ID	Тип расхода	ID расхода договора	Дата	Сумма
14	Подписка на Антивирус...	869	17.08.2012	31

ID	Тип платежа	Дата	Сумма	От даты	До даты
3	первый тип	24.07.2012	45	24.07.2012	29.08.2012
10	первый тип	17.08.2012	35	17.08.2012	14.09.2012

Бонусные расходы создаются только при создании расходов для договора. В клиенте для этого создается расход на некоторую сумму и выбирается пункт "Оплатить бонусами (макс %)" (в скобках указывается максимальный процент от суммы расхода, который можно оплатить бонусами, который, в свою очередь, устанавливается в настройках плагина). После чего вводится сумма, которая будет оплачена бонусами. Данный пункт будет только при условии, что плагин включен у договора.

Создание расхода договора с бонусами, возможно только на текущий день. И изменение, при редактировании, изначальной суммы или даты у расхода будет невозможно.

Редактор расходов

Сумма: 100 Дата: 22.08.2012

Тип: Комментарий

Название	Код
Типы	
11	38
mb	42
Возврат денег в модуле BVCom	57
Перенос средств	51
Подписка на Антивирус Касперского	58
Покупка валюты	55
Разовые	
Ключи OEM	53
расходы за подключение	
45345	
1111	50
Ticket	56
материалы	47
работы	48

Оплатить бонусами(макс. 25%)

Бонусный баланс: 49 Сумму с бонусов: 25 Сумму с договора: 75

Восстановить Ок Отмена

Также можно расходовать бонусы для активации тарифных опций в личном кабинете Web-статистики (подробности в разделе "Web-интерфейс клиента") и в клиенте биллинга.

Глобальные тарифы | Персональные тарифы | Тарифные опции | Группа тарифов

Текущие | История

Название	Время активации	Время окончания	Стоимость активации	...
Скидка 10%	04.07.2012 13:20:18		0.00	▲
Скидка 20%	10.09.2012 16:12:12		0.00	
test	01.10.2012 00:00:00	01.11.2012 00:00:00	111.00	
Мега турб	31.10.2012 18:00:58	01.11.2012 04:00:58	16.00	

Редактор

Опция: test | Режим: 11111

Оплатить Бонусами(22 руб. бонусами из 111 р.)

Ок | Отмена

Для просмотра подробностей бонусного расхода у расхода договора необходимо нажать правой кнопкой мыши по расходу и в выпадающем контекстном меню выбрать пункт "**Бонусы**" (при условии, что у него был бонусный расход). Строка "Итого" равна сумме реального расхода договора, строка "Скидка [бонусы]" равна сумме скидки в рублях (в скобках указывается сумма бонусного расхода), а строка "Сумма" - это начальная сумма, которая равна сумме скидки и расхода у договора. При этом поле скидка равна сумме бонусного расхода, деленному на **текущий курс** (поэтому важно этот курс не изменять, так как у расходов он не хранится).

ID	Дата	Сумма	Тип расхода	Комментарий	Время изменения
869	17.08.2012	94,00	Подписка на Антивирус Касперского		17.08.2012 08:36:39
870	22.08.2012	25,00	Перенос средств		8.2012 17:46:32

Бонусы

Сумма	: 97.10
Скидка [бонусы]	: 3.10 [31.00]
Итого	: 94.00

Бонусный баланс

Бонусный баланс можно просмотреть во вкладке "Бонус" в дереве вкладок договора при нажатии на кнопку "Баланс".

The screenshot displays a software interface for viewing a bonus balance. On the left is a sidebar with a tree view containing categories like 'Режим', 'Лицо', 'Баланс (август 2012)', 'Тарифные планы', 'Модули', and 'Web'. The main area is titled 'Дата' and shows the date '22.08.2012' with navigation buttons for 'Сегодня', 'Вчера', 'Неделя', 'Месяц', and 'Год'. Below this are three tabs: 'Приход', 'Расход', and 'Баланс'. The 'Баланс' tab is active, showing a table with columns: ID, Тип операции, Дата, Сумма, Кредит, Дебит, and Остаток. The table contains four rows of transactions. Below the main table, there is a section for 'Платежи не активные на 22.08.2012 16:43:27' with a table showing columns: ID, Тип платежа, Дата, Сумма, От даты, and До даты. At the bottom right, there are two summary boxes: 'Текущий баланс: 549' and 'Ожидаемые баллы: 123'.

ID	Тип операции	Дата	Сумма	Кредит	Дебит	Остаток
3	Приход	24.07.2012	45	45	----	45
10	Приход	17.08.2012	35	----	----	80
14	Расход	17.08.2012	31	----	31	49
12	Приход	22.08.2012	500	----	----	549

ID	Тип платежа	Дата	Сумма	От даты	До даты
11	первый тип	21.08.2012	123	21.10.2012	17.11.2012

В верхней таблице отображены все расходы и приходы (активные на выбранный момент времени), из которых образовался текущий баланс. Изменив дату, можно просмотреть баланс на конкретное число. В нижней таблице отображены приходы, которые еще не активны (то есть они будут доступны для оплаты в будущем) и чуть ниже поле "Ожидаемые баллы", равное сумме этих приходов. При этом данная таблица не зависит от выбранной выше даты и отображает только текущее состояние неактивных платежей.

Бонусные программы

Бонусные программы позволяют начислять бонусные баллы. Изначально создается бонусная программа в плагине с заполненными параметрами (**Плагины->Bonus**), затем она может быть добавлена к конкретному договору.

Бонусные программы обладают, как минимум, следующими параметрами:

Название - символьное обозначение программы;

Тип программы - один из возможных типов бонусных программ;

Период - период, в течение которого будет учитываться данная программа и можно будет добавить программу к договору. Как минимум должна быть дата начала, а дата закрытия может быть открытой;

Тип бонусного прихода - предварительно созданный бонусный тип прихода(**Справочники->Другие->Bonus-приходы**);

Момент активации - устанавливает начало действия бонусного прихода; имеет следующие возможные значения:

1. **день** - конкретный календарный день; если на момент начисления бонусов дата будет прошедшей, то установится датой начисления;
2. **кол-во дней** - кол-во дней, через которое данными бонусными баллами можно будет воспользоваться для оплаты(с момента начисления бонусов); если оставить пустым, то будет считаться равным 0 (будут активны сразу после начисления);
3. **начало недели** - с начала следующей недели;
4. **начало месяца** - с начала следующего месяца;
5. **начало года** - с начала следующего года.

Завершение периода - устанавливает момент завершения действия бонусного прихода. Для всех значений, кроме первого, считается относительно момента активации. Имеет следующие возможные значения:

1. **день** - конкретный календарный день; если на момент начисления бонусов дата будет прошедшей, то установится датой начисления (что приводит к тому, что срок действия бонусов = 0);
2. **кол-во дней** - кол-во дней, в течение которого можно будет воспользоваться данными бонусами; если оставить пустым, то будет считаться равным 0 (будут активны только в день начисления);
3. **конец недели** - до конца недели;
4. **конец месяца** - до конца месяца;
5. **конец года** - до конца года.


При добавлении программы к договору необходимо также выставить период действия для договора. Он может как дублировать период действия самой программы так и сужать его, либо не ограничивать дату завершения (в этом случае ограничением будет служить дата завершения программы, которую проще редактировать, чем у всех договоров с данной программой).

Операционная программа

Данная бонусная программа позволяет начислять бонусные баллы за приходы, расходы и наработку.

Для работы программы необходимо добавить задачу планировщика (**Сервис->Администрирование->Планировщик заданий**) "**Bonus => начисление бонусов**" и настроить ее запуск на каждый день в удобное вам время. Если параметр "Период начисления" будет выбран "моментальный", то наличие задачи не обязательно.

в конфиге задаче можно задать параметр, список запускаемых задач программ

 program.ids=коды_бонусных_программ

Операционная программа имеет следующие параметры:

Тип операции - приход, расход или наработка;

Период начисления - имеет следующие возможные значения: ежедневный - начисление бонусов за каждый день; месячный - за каждый календарный месяц; периодический - за произвольное количество дней; моментальный - начисление бонусов в момент прихода или расхода. Параметр для приходов и расходов;

Дни - параметр заполняется, если "Период начисления" имеет значение месячный или периодический. Для месячного - это день месяца, в который будет происходить начисление бонусов за предыдущий календарный месяц. Для периодического - это количество дней, за которое будут начислены бонусы;

Минимальная сумма операции - если сумма конкретной операции будет меньше выставленной в данном параметре суммы, то данная операция не будет учитываться;

Минимальная сумма за период - если сумма всех учетных операций будет меньше выставленной в данном параметре суммы, то начисление бонусов не произойдет. Также этот параметр имеет значение шага в следующем параметре;

Тип начисления бонусов - имеет следующие возможные значения: процентный - начисляемая сумма равна проценту от учетной суммы, умноженную на бонусный курс; абсолютный - сумма бонусных баллов, которое будет зачислено; пошаговый - сумма бонусов равна значению от деления без остатка суммы всех учетных операций на минимальную сумму за период и умноженную на значение данного параметра (**Пример: минимальная сумма за период = 300, сумма за период вышла = 1499, значение параметра = 500. В результате будет начислено 2000 бонусов**); произвольный - сумма зачисления зависит от того в какой интервал попала сумма операции (**Пример. значение параметра = " 300-1200, 400-1250,450-1800 ", сумма за период вышла = 434. В результате будет начислено 1250**). Первым значением идет сумма учетных операций, второй - сумма баллов.

ID операций - коды приходов, расходов или коды услуг для наработки через запятую, которые будут учитываться. Можно оставить пустым, тогда будут учитываться все операции.

Динамические программы

Динамические программы - это бонусные программы логика реализации которых основана на динамических Java классах. То есть, если вы хотите свой тип бонусной программы, то самым простым способом будет - это реализовать ее через динамические программы.

Для наглядности сейчас будет описан пример создания ДБП. Допустим нам нужна ДБП которая будет на каждый приход договора начислять бонусы, причем начисление зависит от некоторого коэффициента, который будет зависеть от кол-ва лет с момента создания договора, и начисление бонусов будем совершать только на первые два прихода сумма которых более 100р.

1) Создадим динамический класс и обязательно отнаследуем его от **BonusProgramDynamicBase**. Реализуем методы getTitle и accrualOfBonusImpl, первый соответственно возвращает название программы(оно будет отображаться в клиенте), а второй метод занимается начислением или/и расчетом некоторых значений(все зависит от ваших фантазий) для договоров. В нашем случае нам нужно реагировать на событие прихода платежа, потому мы еще переопределим метод onEvent.

Данный код можете использовать в качестве заготовки для вашего дин. кода(само собой изменив перед этим название класса и пакет).

Код

```
package ru.bitel.bgbilling.bonus.myPrograms;

import java.sql.Connection;
import java.sql.SQLException;

import ru.bitel.bgbilling.common.BGException;
import ru.bitel.bgbilling.kernel.event.Event;
import ru.bitel.bgbilling.plugins.bonus.common.bean.*;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.sql.ConnectionSet;

public class FirstProgram extends BonusProgramDynamicBase
{
    @Override
    public String getTitle()
    {
        return " ";
    }

    @Override
    public void accrualOfBonusImpl( Connection con, BonusProgram program ) throws SQLException, BGException
    {
    }

    @Override
    public void onEvent( Event event, Setup setup, ConnectionSet set ) throws Exception
    {
    }

    @Override
    public List<String> getWebBonusStrings(Connection con, BonusContractProgram program )
    {
        return null;
    }
}
```

2) Далее заполним все методы, в результате у нас готовый дин. класс реализующий всю нашу логику. BonusProgramDynamicBase содержит основные методы для работы с бонусами, не забывайте ими пользоваться.

Код

```
package ru.bitel.bgbilling.bonus.myPrograms;

import java.math.BigDecimal;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Calendar;
```

```

import java.util.Date;
import java.util.List;

import ru.bitel.bgbilling.common.BGException;
import ru.bitel.bgbilling.kernel.contract.api.common.bean.Contract;
import ru.bitel.bgbilling.kernel.contract.api.server.bean.ContractDao;
import ru.bitel.bgbilling.kernel.contract.balance.server.event.PaymentEvent;
import ru.bitel.bgbilling.kernel.event.Event;
import ru.bitel.bgbilling.kernel.module.common.bean.User;
import ru.bitel.bgbilling.plugins.bonus.common.bean.BonusContractProgram;
import ru.bitel.bgbilling.plugins.bonus.common.bean.BonusProgram;
import ru.bitel.bgbilling.plugins.bonus.common.bean.BonusProgramDynamicBase;
import ru.bitel.bgbilling.plugins.bonus.server.dao.BonusDao;
import ru.bitel.bgbilling.server.util.Setup;
import ru.bitel.common.Utills;
import ru.bitel.common.sql.ConnectionSet;
import bitel.billing.common.TimeUtills;

public class FirstProgram extends BonusProgramDynamicBase
{
    @Override
    public String getTitle()
    {
        return " ";
    }

    @Override
    public void accrualOfBonusImpl( Connection con, BonusProgram program ) throws SQLException, BGException
    {
        Calendar now = Calendar.getInstance();
        //
        if( now.get( Calendar.DAY_OF_MONTH ) != 1 )
            return;

        ContractDao contractDao = new ContractDao( con, User.USER_SERVER );
        for( int contractId : bonusDao.getContractsOfThisProgram( program.getId(), new Date() ) )//
        {
            if( !bonusDao.pluginInclude( contractId ) )// , .
            {
                continue;
            }
            map = bonusDao.getDataProgramOfContract( program.getId(), contractId );// .
            Contract contract = contractDao.get( contractId );
            // - ( , , )
            int year = TimeUtills.hourDelta( TimeUtills.convertDateToCalendar( contract.getDateFrom() ), Calendar.
getInstance() ) / 24 / 365;
            int percent = 0;
            switch( year )// . = 5%, > 3 = 10%, > 5 = 15%
            {
                case 1:
                    percent = 5;
                    break;
                case 3:
                    percent = 10;
                    break;
                case 5:
                    percent = 15;
                    break;
            }
            if( percent > 0 )
            {
                //
                map.put( "percent", Integer.toString( percent ) );
            }
            map.put( "count", "0" );//
            updateProgramData( program.getId(), contractId, map );// .
        }
    }

    @Override
    public void onEvent( Event event, Setup setup, ConnectionSet set ) throws Exception
    {

```



```

//      ,      .
super.onEvent( event, setup, set );
//      .
if( event instanceof PaymentEvent && bonusDao.pluginInclude( event.getContractId() ) )
{
//
List<BonusContractProgram> list = super.getContractPrograms( this.getClass(), new Date() );
for( BonusContractProgram contractProgram : list )
{
    BonusProgram program = contractProgram.getProgram();
    BigDecimal perc = Utils.parseBigDecimal( getParam( "percent", program.getId() ), BigDecimal.ZERO );
    int count = Utils.parseInt( getParam( "count", program.getId() ) );
    if( count < 3 && perc.compareTo( BigDecimal.ZERO ) == 1 )//      -      3 .
    {
//      . !      ->.
        BigDecimal sum = ((PaymentEvent)event).getPayment().getSum().multiply( perc.divide( new BigDecimal( 100 ) ) );
    }
    updatePayment( sum, program );//      .
    map.put( "count", Integer.toString( count + 1 ) );
    updateProgramData( program.getId() );//      -      .
    }
}
}

@Override
public List<String> getWebBonusStrings( Connection con, BonusContractProgram program )
{
    ArrayList<String> list = new ArrayList<String>();
    contractId = program.getContractId();
    bonusDao = new BonusDao( con );
//      ,      .
    return list;
}
}

```

3) Далее необходимо прописать параметр **dynamicBonusPrograms**(если не был прописан до этого) в конфигураторе плагина(Плагины->Настройки), а в качестве значения прописать полные имена ваших дин. классов используемых для дин. программ, через запятую. В нашем случае еще понадобится наш класс повесить на событие прихода платежа(Сервис->Автоматизация->Функции скриптов поведения).

В нашем случае оно будет иметь следующий вид: `dynamicBonusPrograms=ru.bitel.bgbilling.bonus.myPrograms.FirstProgram`. Допустим если вы написали там же еще вторую дин. программу с названием `SecondProgram`, то параметр будет иметь следующий вид: `dynamicBonusPrograms=ru.bitel.bgbilling.bonus.myPrograms.FirstProgram,ru.bitel.bgbilling.bonus.myPrograms.SecondProgram`

4) Далее стандартным способом создаете бонусную программу, просто вместо операционного типа выберите ваш тип. Далее, так же стандартным способом, добавляете вашу программу на договора.

Редактор

Название

Тип программы Первая программа

Период с по

Тип бонусного прихода превый тип

Момент активации День

Завершение периода День

5) Начисление бонусных баллов происходит с помощью задачи планировщика **Bonus => начисление бонусов**

Web-интерфейс плагина Bonus

На сайте статистики клиент выбирает пункт меню "Бонусы", где будет отображен его баланс и ожидаемые баллы (если токовые есть), а чуть ниже таблица, описывающая каким образом этот баланс был получен.

Меню

- » Новости
- » Уведомления
- » Просмотр баланса
- » Смена пароля на доступ к статистике
- » Смена тарифных планов
- » Тарифные опции
- » Карточки
- » Управление лимитом
- » Управление статусом
- » Дополнительные действия
- » Примечания
- » Оплата через BVCom
- » Оплата через кошелек Qiwi
- » Счета
- » Счета-фактуры
- » Управление лицензиями
- » Бонусы
- » Подписка на рассылки
- » Документы
- » Выход

Баланс

Бонусный баланс: **549.00**
Ожидаемые баллы: **123.00**

Номер	Тип операции	Дата	Сумма	Кредит	Дебит	Остаток
3	Приход	24.07.2012 13:10	45.00	45.00	---	45.00
10	Приход	17.08.2012 01:50	35.00	35.00	---	80.00
14	Расход	17.08.2012 08:36	31.00	---	31.00	49.00
12	Приход	22.08.2012 16:43	500.00	500.00	---	549.00

Для активации тарифных опций с использованием бонусов необходимо в тарифных опциях нажать на кнопку **Активировать** и, если тарифная опция не является бесплатной и бонусный баланс больше 0, то произойдет переход на страницу, где пользователю будет предложено оплатить часть расхода бонусами. Причем пользователю будет предложена максимально возможная сумма для оплаты бонусами, которая зависит от бонусного баланса и максимального процента для данного расхода.

В настройках плагина есть возможность указать тарифные опции которые можно оплатить только бонусами, для таких тар. опц. переход на страницу где будет предложено оплатить бонусами будет произведен, если у клиента достаточно для этого бонусов(причем не важно какой макс. процент у данного расхода, которым можно оплатить бонусами), иначе будет выведено сообщение об ошибке.

Меню

- » Новости
- » Уведомления
- » Просмотр баланса
- » Смена пароля на доступ к статистике
- » Смена тарифных планов
- » Тарифные опции
- » Карточки
- » Управление лимитом
- » Управление статусом
- » Дополнительные действия
- » Примечания
- » Оплата через BVCom
- » Оплата через кошелек Qiwi
- » Счета
- » Счета-фактуры
- » Управление лицензиями
- » Бонусы
- » Подписка на рассылки
- » Документы
- » Выход

Оплата бонусами

Бонусный баланс: **50.00**

Опция	Режим активации	Сумма	Макс. процент бонусами	Описание
option	Платный	1000.00	50%	

Не оплачивать бонусами Оплатить бонусными баллами(5 рубля) Отмена

Шаблоны договоров плагина Bonus

В шаблоне договора можно указать автоматическое добавление бонусной программы и включение плагина при создании договора. Для включения плагина у договоров нужно установить галочку Включен. Для добавления бонусных программ выберите их из списка ниже (предварительно создав их через меню **Плагины->Bonus**); в списке будут присутствовать только действующие на данный момент программы.

Если на момент создания договора бонусная программа будет не действующей (по истечению времени или по причине закрытия), то она не будет добавлена.

Модули Плагины Группы Шаблоны комментариев Параметры Группы тарифов и тарифы Скрипт поведения Объекты

SbPilot
CLADR
Бонусы
CashCheck
HelpDesk
Dispatch
CRM
Документооборот

вкл/выкл плагина, программы

Включен

Бонусные программы

За расход, моментальный.

Групповые операции плагина Bonus

Вкл/выкл плагина, добавление бонус. программ, установка периода программам

Как следует из названия данная групповая операция позволяет включать/выключать плагин, добавлять бонусные программы и устанавливать период действия для них.

Для включения плагина у договоров нужно установить галочку Включен, для добавления бонусных программ первым делом установите период, а уже затем выберите программы из списка ниже (предварительно создав их через меню **Плагины->Bonus**); в списке будут присутствовать только программы, удовлетворяющие периоду (добавление программ, дата закрытия которых уже прошла невозможно).

The screenshot shows a dialog box titled "Бонусы: Вкл/выкл плагина, добавление бонус. программ, установка периода программам." It contains a "Период" section with a date range from "14.11.2012" to an empty field, and a checked "Включен" checkbox. Below is a list of "Бонусные программы" with one item, "За расход, моментальный.", which is checked. At the bottom are buttons for "Все", "Сброс", and "Инверт."

Продление периода действия бонусных платежей

Данной групповой операцией можно расширить период действия бонусных платежей, которые еще не были списаны или списаны не полностью для договоров с включенным плагином.

"Только активные платежи" - если выбрано, то операция будет производиться только с платежами период которых еще не завершился. Рекомендуется оставлять его включенным.

"Платежи за период" - данным параметром выбираются все платежи дата которых входит в заданный период.

"Тип расширения и его период" - задает способ увеличения периода бонусов

- "До даты" - конкретная фиксированная дата до которой будут увеличены все платежи, если у некоторых платежей дата до которой они действуют окажется больше заданной, изменения их не коснутся.
- "На кол-во дней" - относительная величина, увеличивает на заданное кол-во дней каждый платеж. Если заданное кол-во дней не хватает платежу стать активным, то действия его не затронут, так как не имеют смысла.

"Тип бонусных платежей" - операция будет произведена только с выбранными типами. Минимум один тип должен быть выбран.

The screenshot shows a dialog box titled "Бонусы: Продление периода действия бонусных платежей" with the subtitle "Изменяется таблица bonus_contract_payment". It has a checked "Только активные платежи" checkbox. The "Платежи за период" section shows a date range from an empty field to another empty field. The "Тип расширения и его период" section has a dropdown menu set to "До даты" and an empty field. Below is a list of "Типы бонусных платежей" with two items, "превый тип" and "второй т", both unchecked. At the bottom are buttons for "Все", "Сброс", and "Инверт."

Плагин CashCheck

Назначение и структура плагина, архитектура системы

Плагин CashCheck позволяет при внесении платежей в биллинг печатать кассовые чеки на подключённом оборудовании: контрольно-кассовой машине (ККМ, Регистратор) или принтере или онлайн-сервисах. Также возможна печать приходных кассовых ордеров на обычном принтере. Архитектура плагина подразумевает любые подобные действия, связанные с операцией прихода денежных средств. Далее любое оборудование называется принтер.

Список поддерживаемых на данный момент физических принтеров находится в разделе [Настройка сервера печати и оборудования, поддерживаемые устройства](#).

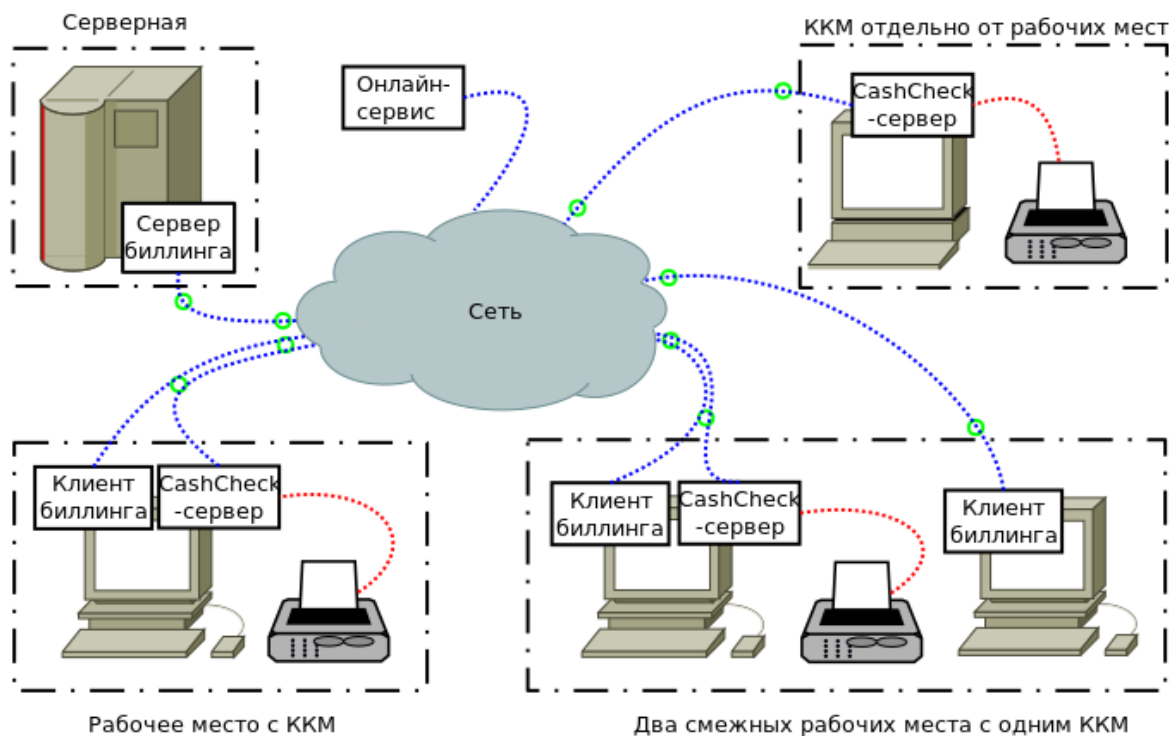
Список поддерживаемых онлайн-сервисов см. в настройках фискализаторов.

Плагин (а именно его серверная часть) для работы с физическими(локальными) ККТ работает в связке с **сервером печати** - отдельным приложением, запущенном на некотором удалённом или локальном компьютере и доступном для сервера биллинга по сетевому протоколу TCP. Сервер печати служит для сетевого доступа к принтеру. Сервер печати придуман для того, чтобы несколько принтеров могли быть установлены в любых местах с возможностью любого клиента биллинга (то есть, в конечном итоге, сервера биллинга) печатать на любой из них.

Один сервер печати даёт возможность сделать привязку только одного конкретного принтера к одному сетевому порту. Для двух подключённых к одной рабочей машине устройств (одинаковых или разных) требуется запустить два сервера печати на разных портах. (см. раздел [запуск двух копий сервера](#))

Итак, помимо установленных плагинов на сервер биллинга и на необходимые клиенты биллинга, устанавливается несколько серверов печати - по числу работающих принтеров. Серверы печати устанавливаются и запускаются на тех компьютерах, к которым физически подключено оборудование. Ниже представлена схема работы системы и принцип взаимодействия клиентов биллинга, сервера биллинга и принтеров.

Сервер печати - один из вариантов **фискализаторов**, другие фискализаторы могут работать с какими-то онлайн-системами, например, атол-онлайн и т.п.



Несмотря на то, что сервер печати нумеруется версиями, сходными с версиями биллинга, эта нумерация условная и носит больше номенклатурный характер. Соответствие версии сервера печати и версии самого биллинга в общем случае необязательно (в отличие от версии самого плагина CashCheck). Но злоупотреблять этим не стоит.

Необходимо предупредить, что очень часты проблемы при использовании схем подключения, отличных от прямого подключения к com-порту. Например, при использовании переходников com-usb, устройств типа prog и прочего, особенно ненадлежащего качества. Если вы используете такие устройства, убедитесь, что они корректно работают в вашей системе, для них правильно установлены и настроены свежие драйверы. При проблемах обмена с принтерами ("ошибка связи", "не отвечает на ENQ", "некорректно ответил в течение 5 попыток" и т. д.) сначала переустановите драйверы этих устройств, а также попробуйте заменить сами устройства. Дешёвые переходники гарантированно будут давать проблемы. Игак, дешёвые китайские — плохие, качественные (в т.ч. на базе ИМС CP2102 и подобных) — хорошие.

Настройка плагина

Настройка плагина в биллинге

После стандартной установки и подключения плагина требуется настроить печать уже в самом биллинге. Это делается в конфигурации плагинов для соответствующего плагина. Конфигурация состоит из нескольких частей:

- настройки регистраторов — указывается адрес и порт сервера печати и заголовок для отображения наименования в биллинге;
- настройки привязки типов платежей к регистраторам отделам — указывается на каких регистраторах разрешена печать;
- некоторые флаги.

Пример конфигурации для локальных ККТ, т.е. сервера печати (отражён не весь набор параметров/флагов):

```
#
# fr.< >.connector=< >:< >
# fr.< >.title=< >
# fr.< >.hidden=< >, 0>

fr.1.connector=127.0.0.1:9876
fr.1.title= ()

fr.2.connector=192.168.0.1:9876
fr.2.title=

fr.100.connector=192.168.0.2:9876
fr.100.title=
fr.100.hidden=1

#
# pt.<id >.fr=< >, >

pt.1.fr=1
pt.25.fr=2
pt.37.fr=1,2

# "" ( false), fop-,
disable.login.hello=0
```

Отдел не привязывается к типам платежей в конфигурации плагина - это делается в скрипте поведения для события "формирование вида чека". Отдел тут подразумевается в контексте отдела продаж в понятиях фискального регистратора, по ним можно будет дополнительно делать отчёт на регистраторе. Подробнее о том, как использовать эту возможность см. ниже.

Тип платежа идентифицируется по его ID - числовому номеру, который можно узнать в справочнике платежей. Не привязанные никуда платежи вообще нельзя будет напечатать на принтере, также они не будут попадать в очередь возможных для печати платежей.

Если при попытке печати чека у вас выдаётся сообщение что "печать невозможна", то это указывает на проблему с настройкой маппинга. Такая ошибка возникает, а) когда платежа, который пытается быть превращён в чек, нет в настройках; б) когда этот платёж уже есть в таблице лога распечатанных платежей. Платёж либо можно сразу распечатать, либо он окажется в "очереди печати". Туда попадают ВСЕ платежи, тип которых назначен на текущий регистратор. После распечатки платежи регистрируются в логе распечатанных платежей.

Если установить параметр hidden для регистратора, то он не рисуется в списке для кассира, но доступен из скриптов, настроек автоплатежей и т. д.

Настройка внешнего вида чеков (динамический код)

Для формирования вида чеков, задания фискальных атрибутов и прочих параметров используется динамический код. В конфигурацию плагина прописывается нужный класс:

```
#
checkbuilder=ru.bitel.bgbilling.cashcheck.SimpleCheck
```

Есть возможность задать отдельный класс для отдельного ККТ или для отдельного маппинга по типу. Это может потребоваться для теста нового шаблона. Или для удобства, если фискализаторы совсем разные (например, локальный и онлайн-служба) - вместо проверки по ККТ внутри шаблона просто используются разные шаблоны.

```
#
fr.1.checkbuilder=ru.bitel.bgbilling.cashcheck.CheckKKT1
#
pt.666.checkbuilder=ru.bitel.bgbilling.cashcheck.CheckTestType666
```

В таком случае порядок выбора класса следующий: ищется для маппинга, потом для ККТ, потом общий. Про случай настройки для маппинга есть оговорка: не работает для возвратов (в данный момент там нет маппинга), при ручном чеке проверяется что все платежи в чеке (если их несколько - например при распределении или просто при печати кучи платежей одного договора в один чек) нашли один и тот же настроенный дин.код.

Несколько примеров реализации класса идёт в комплекте с плагином. Подразумевается, что какой-либо класс обязательно должен быть найден и должен сработать. О работе с динамическим кодом можно прочитать в соответствующем разделе справки. Внутри можно проверить любые условия и сформировать чек любой формы для каждой позиции/платежа, добавляемой в чек. Класс должен расширять абстрактный класс `ru.bitel.bgbilling.cashcheck.CheckMaker`.

Методы динамического класса служат для настройки внешнего вида чеков: чека продажи, чека возврата, чека для автоматической печати платежей.

- Позиция - это несколько строк, представляющие одну позицию в чеке, добавляются в объект **check** с помощью **addPayment/addString**
- один раз должна присутствовать установка платежа **addPayment()**, это будет "фискальная" строка, которая собственно представляет собой продажу, в этой строке может быть ещё одна строка, типа наименование продажи и отдел. Теоретически может не быть фискальной строки, в печать не выведется тогда платёж, но внимание - по всем остальным признакам платёж пометится напечатанным. Замечание касается работы с фискальным оборудованием. При печати на произвольном принтере (термопринтере или обычном) возможно формирование с помощью скриптов любых данных для вывода.
- в остальных строках - по одному параметру **String** которая выведется
- или: прямой текст указан, так и выведется
- или: просто ничего (пробел), это будет пустая строка, типа вертикальная табуляция (например, для красоты при отделении блоков текста друг от друга и т.д.)
- для объекта **check** настраиваются дополнительные параметры: типы платежа, признаки электронного чека итд - см. следующий раздел.

Примеры кода:

- **добавление позиции**

```
public class SimpleCheck
    extends CheckMaker
{
    ...
    public void addPayment( Payment payment, Check check, Printer printer ) throws BGException
    {
        // 1) -, ( , )
        check.addPayment( payment.getSumma(), payment.getContractTitle(), 0 );
        // 2)
        check.addString( payment.getContractComment() );
        // 3)
        check.addString( " " );
    }
    ...
}
```

- **завершение формирования**

```
public class SimpleCheck
    extends CheckMaker
```

```

{
...
public void endCreate( Set<Integer> cids, Check check, Printer printer ) throws BGEException
{
    check.addString( "footer 1" );
    check.addString( "footer 2" );
    check.addString( "footer 3" );
}
...
}

```

Обратите особое внимание, что в каждом скрипте формирования внешнего вида чека (а именно происходит формирование каждой отдельной позиции чека) обязательно должна присутствовать (как правило*) ровно одна команда **addPayment** для всех устройств, являющихся ККМ. Дополнительно может быть любое количество **addString**. Для устройств, представляющих обычный принтер, для FOP-устройств (см. ниже) и т.п. команда **addPayment** не нужна, так как там не происходит добавление продажи во внутреннюю память. Но сумма платежа будет считаться только для позиций, добавленных через **addPayment**.

* Исключение может быть для случаев, если вы специально хотите разбить один платёж на две позиции в чеке (например, с разными названиями или атрибутами), в этом случае нужно самостоятельно разбить сумму на части и добавить несколько **addPayment**, в логе платежей будет одна запись с итоговой суммой.

Далее приведём пример кода "добавление позиции" для формирования FO-документа, для FOP-драйвера. Эти строки соответствуют шаблону **cas hcheck_pko.xml**, находящемуся в стандартной поставке сервера печати.

```

import bitel.billing.common.*;
import bitel.billing.server.admin.bean.*;

public class SimpleCheck
    extends CheckMaker
{
...
public void addPayment( Payment payment, Check check, Printer printer ) throws BGEException
{
    // : "120,00." (    addPayment,    )
    check.addPayment( payment.getSumma(), String.valueOf(payment.getSumma()) + ".", 0 );
    //
    check.addString( payment.getContractTitle() );
    // : " .."
    check.addString( payment.getContractComment() );
    // : 20-04-2009
    check.addString( TimeUtils.format( payment.getDate(), "dd-MM-yyyy" ) );
    //: ""
    check.addString( " \\\"" );
    // ( )
    UserManager um = new UserManager( con );
    User user = um.getUserByID( payment.getUserID() );
    check.addString( user.getName() );
    // : " "
    check.addString( " " );
    // : " 00 "
    check.addString( SummaToString.summaToString( String.valueOf(payment.getSumma()), true ) );
}
...
}

```

В данном случае мы формируем 8 строк с произвольной информацией, которая передаётся в FOP-драйвер сервера печати, который из напрямую передаёт в FO-шаблон обычной линейной xml. Это позволяет формировать документ из любых строк, подготовленных в скрипте и из любого написанного шаблона.

Также в методы класса передаётся объект "printer" - объект "принтер", на который производится печать. Это может понадобиться, например, при наличии двух разных принтеров и желания печатать на них разного вида информацию и на разные принтеры использовать разные скрипты (например, для FOP и ККМ скрипты всегда разные будут). Например, можно получить ид принтера (такой, какой он в конфиге плагина):

```
int printerId = printer.getId();
```

В этом объекте есть метод **switchPrinter**, который позволяет переключить на нужный принтер внутри дин.кода, это может понадобиться в случаях, когда выбор принтера сложнее привязки "по типам". Весь конфиг принтера доступен через метод **getConfig**.

Последовательность вызова дин.кода:

Для чека продажи вызывается **addPayment** для каждого платежа биллинга, которые направлены на печать в один чек и в конце один раз **endCreate**. Оговорка про "для каждого платежа" - потому что из клиента биллинга есть возможность напечатать сразу несколько приходов биллинга в один чек, в виде нескольких продажных позиций. Если печатается просто по приходу платежа галочкой, то будет один вызов **addPayment** дин. кода.

Для чека возврата вызывается **addPaymentRefund** для одного возврата биллинга (возвраты только по отдельности можно печатать) и в конце **endCreateRefund**. В отличие от платёжного чека обычно нельзя печатать строки итд, потому должно быть там просто один addPayment внутри **addPaymentRefund** и установка доп.атрибутов (налоги, адрес покупателя итд).

Для чека, который печатается через механизм автопечати платежей - вызывается один раз **addPaymentAuto** для единственного платежа. Т.е. в отличие от методов штатной функциональности вызывается один раз для каждого кассового чека (т.к. в чек попадает всегда один платёж биллинга), так что подразумевает в себе логику и addPayment и endCreate. (Для возвратов в данный момент аналогичная ситуация, но исторически там осталось два метода). Про автопечатать платежей см. соответствующий раздел.

Аналогично auto-платежам для delay-платежей реализуется метод **addPaymentAutoDelay** с такой же сигнатурой.

Разделение по отделам в ККМ, задание типа платежа, налога, признак электронного чека и т.д.

Этот раздел относится логически к предыдущему про настройку внешнего вида чеков, так как затрагивает использование тех же скриптов.

Имейте ввиду, что для каждого ККМ номера отделов, типов платежа и остального имеют какой-то диапазон или формат. Следует уточнить это в документации по устройству. Также устройство может не поддерживать то или иное.

Внимание: особенно внимательно надо переходить с одного фискализатора (или даже драйвера сервера печати) на другой и не забывать проверить дин.код и все значения. Например, в драйверах atoll3 и atollibptr разные значения кодов типов платежей.

При желании разделять некоторые платежи по **разным отделам** можно использовать такую возможность, предоставляемую большинством ККМ. Для этого при добавлении в скрипте "формирование позиции чека" очередной позиции с помощью команды addPayment надо третьим параметром передать номер отдела. Таким образом отдел привязывается к продажной позиции.

Вычислить номер отдела можно на основании любых данных, например, типа платежа, номера принтера и т.д. и т.п. Пример использования:

```
int paymentDep = 0;
switch( payment.getPaymentTypeID() )
{
case 1:
    paymentDep = 1;
    break;
case 2:
    paymentDep = 2;
    break;
// ....
}
check.addPayment( payment.getSumma(), payment.getContractTitle(), paymentDep );
```

Можно просто отдавать в метод 0 (ноль), если вам не нужна эта возможность.

Тип платежа - это "наличные", "картой" итд. Поддерживается многими ККМ. Он задаётся для всего чека сразу. Вам нужен setType() у объекта Check. Туда передаётся целочисленный тип, назначение идентификаторов уточняйте в документации устройства или у поставщика. Так как тип платежа привязан ко всему чеку, подходящее место для установки - метод "завершение формирования чека" (но можно ставить и в момент формирования каждой позиции, нужно только учесть, что в следующей позиции он может перекрыться).

Номер налоговой группы - задаётся для всего чека, метод setTax() у объекта Check. Назначение номеров настраивается в самом ККТ, уточняйте в документации устройства.

Признак электронного чека - указание для ККТ что чек электронный, если он умеет такое, то печати бумажного чека не будет. Метод setOnlyEiCheck(true) у объекта Check. Обратите внимание, что как правило ККТ игнорируют эту настройку, если не передаётся customerAddress.

Указание customerAddress - для электронных чеков необходимо указание customerAddress (можно указывать и при бумажном). Метод setCustomerEmail("email/phone") у объекта Check. Формат передаваемого телефона/адреса уточняйте в документации ОФД.

Указание произвольного тега для ОФД - можно задать дополнительный любой тег. За это отвечают методы setTag(int tag, String value), setTag(int tag, byte value), setTag(int tag, byte[] value) у объекта Check.

Признак способа расчёта (Check.setPayMethod(int payMethod)) и **Признак предмета расчёта** (Check.setPayObject(int payObject)) - атрибуты ФФД 1.05, вообще это атрибут позиции чека, но здесь оно на весь чек задаётся.

Значение **кассира (тег 1021)** для чеков (Check.setCashierName(String cashierName)) и **инн кассира (тег 1203)** для чеков (Check.setCashierINN(String cashierINN)) - более удобные методы задания этих тегов, либо основной способ авторизации/указания кассиров в драйвере ru.bitel.frk.driver.atollibptr.Driver (там нет авторизации по паролю кассира).

Автопечать платежей

Существует штатный механизм для печати чеков по факту прихода в биллинг произвольного платежа, т.е. не только добавления оператором вручную. В основном это может быть полезно для реализации печати электронных чеков. Или можно перевести на автопечать и печать тех чеков, которые добавляет оператор (или как альтернатива: настроить чтобы он не мог снять галочку в интерфейсе), тогда оператор не должен ставить галочку (можно настроить недоступность её). Есть вариант автоматической отложенной печати (например, ночью) всех накопленных подходящих под настройки маппинга платежей.

Конфигурация целиком:

```
#
# pt.< >.fr=< >
# pt.< >.pass=< >
# pt.< >.print=< : manual( )/auto/delay>
pt.666.fr=100
pt.666.print=auto
pt.666.pass=01

fr.N.title=...
fr.N.connector=...
# ,
fr.N.autoclose=1
# ,
fr.N.autoclose.pass=30
# , , , ( , " "), ( )
fr.N.pass=01

# delay- ( ) auto-
#
delay.payment.back.days=5
# delay- ( auto- )
delay.payment.timeout.ms=10

# ( )
#
check.status.back.days=5
#
check.status.timeout.ms=10
```

Маппинг настраивается аналогично ручным платежам, привязываясь к типу платежа; только указывается не разрешённые регистраторы, а один конкретный. Также указывается пароль для регистратора. с которым надо заходить на ККТ.

Режим печати - одно из значений: **manual** (по умолчанию) - обычный режим печати при добавлении платежа оператором, этот режим рассматривался выше; **auto** - режим автоматической печати по приходу платежа, рассматриваемый в данный момент; **delay** - режим отложенной автоматической печати, при котором печатается не по приходу платежа, а все накопленные нераспечатанные платежи при запуске специальной задачи.

Для режима **auto** при приходе платежа из любого источника проверяется маппинг и если тип платежа настроен - вызывается метод **addPaymentAuto** обычного дин.кода указанного в **checkbuilder=** (параметр "динамический класс для формирования вида чека") похожим способом, как это делается для обычных manual-платежей.

Для режима **delay** всё аналогично, только делается это при запуске задачи "CashCheck => печать отложенных платежей". Вызывается метод **addPaymentAutoDelay**.

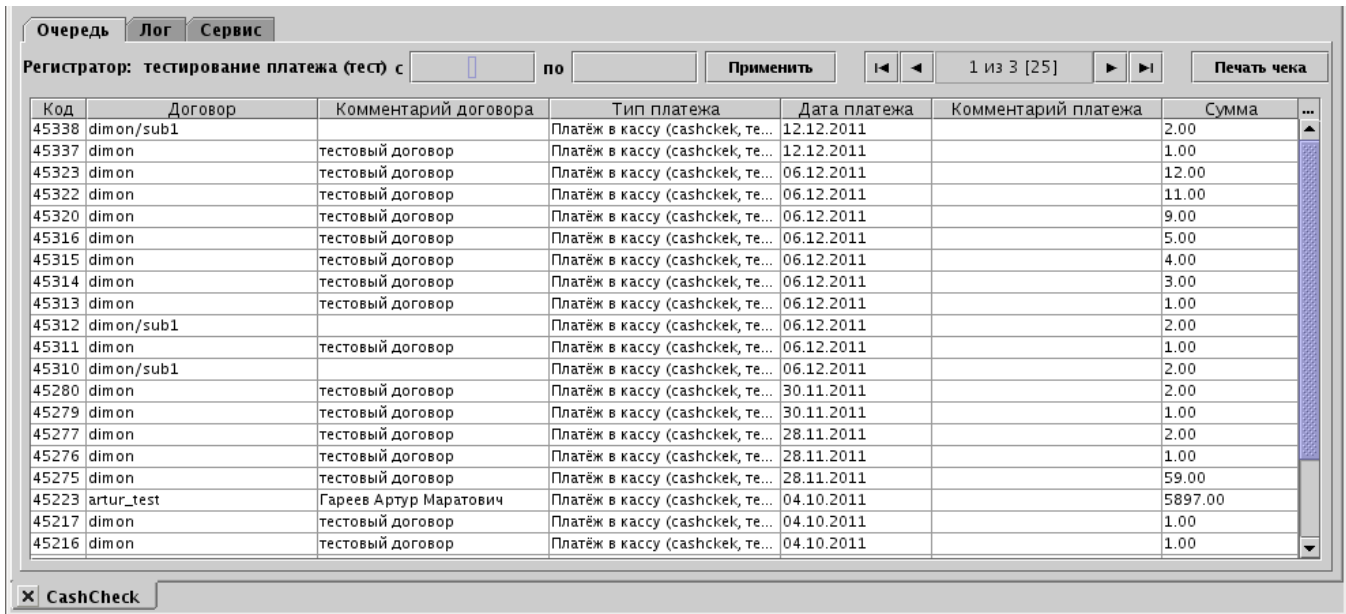
Для автоплатежей актуален вопрос автоматического закрытия дня на ККТ. Для этого существует задача "CashCheck => автоматическое закрытие дня (снятие z-отчёта)". Можно настроить на середину ночи, например. Настройки этой задачи централизованно находятся в конфиге плагина и дополняют настройки каждого регистратора (см. выше). Может быть настроено на автозакрытие любое количество ККТ, все они закроются последовательно одним запуском задачи.

Если будут проблемы с печатью чеков или снятием отчёта - будет отправлено оповещение по стандартной системе алармов, убедитесь, что она корректно настроена.

Использование плагина

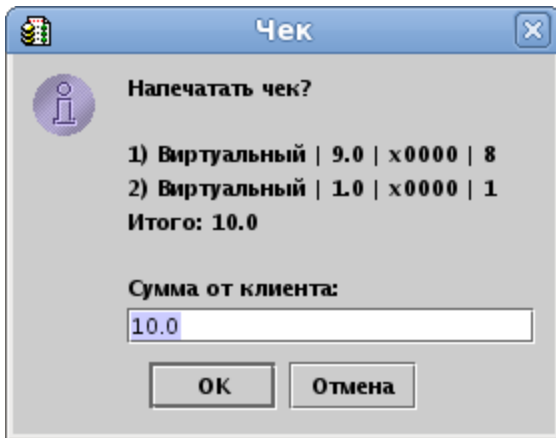
Очередь печати

"Очередь платежей для печати" отображает все платежи, подходящие для печати по ним чека, но по которым чек напечатан ещё не был. Вызывается через **Плагины**->**CashCheck**, вкладка **Очередь**.



Код	Договор	Комментарий договора	Тип платежа	Дата платежа	Комментарий платежа	Сумма
45338	dimon/sub1		Платёж в кассу (cashcek, те...	12.12.2011		2.00
45337	dimon	тестовый договор	Платёж в кассу (cashcek, те...	12.12.2011		1.00
45323	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		12.00
45322	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		11.00
45320	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		9.00
45316	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		5.00
45315	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		4.00
45314	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		3.00
45313	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		1.00
45312	dimon/sub1		Платёж в кассу (cashcek, те...	06.12.2011		2.00
45311	dimon	тестовый договор	Платёж в кассу (cashcek, те...	06.12.2011		1.00
45310	dimon/sub1		Платёж в кассу (cashcek, те...	06.12.2011		2.00
45280	dimon	тестовый договор	Платёж в кассу (cashcek, те...	30.11.2011		2.00
45279	dimon	тестовый договор	Платёж в кассу (cashcek, те...	30.11.2011		1.00
45277	dimon	тестовый договор	Платёж в кассу (cashcek, те...	28.11.2011		2.00
45276	dimon	тестовый договор	Платёж в кассу (cashcek, те...	28.11.2011		1.00
45275	dimon	тестовый договор	Платёж в кассу (cashcek, те...	28.11.2011		59.00
45223	artur_test	Гареев Артур Маратович	Платёж в кассу (cashcek, те...	04.10.2011		5897.00
45217	dimon	тестовый договор	Платёж в кассу (cashcek, те...	04.10.2011		1.00
45216	dimon	тестовый договор	Платёж в кассу (cashcek, те...	04.10.2011		1.00

Очередь отображается для текущего выбранного (данном залогиненным пользователем биллинга) принтера. Для одного или нескольких платежей из очереди можно из этой таблицы напечатать чек.



Чек

Напечатать чек?

1) Виртуальный | 9.0 | x0000 | 8
2) Виртуальный | 1.0 | x0000 | 1
Итого: 10.0

Сумма от клиента:
10.0

OK Отмена

Обратите внимание - запуск (или не запуск) сервера печати с каким-либо драйвером (реальным или виртуальным) никак не влияет на добавление или не добавление платежа в очередь. Сервер биллинга ничего об этом аспекте не знает - он посылает команды, которые не зависят от активированного драйвера. Можно ли платёж напечатать — зависит только от того, прописана ли соответствующая строка с типом платежа в конфиге плагина. Конечно, если указана принудительная печать чека, а сервер печати не запущен или настроен некорректно - будет выведена ошибка печати, но в очередь чек всё равно попадёт, если не был напечатан при добавлении платежа.

Не разрешено печатать платежи разных договоров на одном чеке. Это сделано специально - может возникнуть недоразумение. Исключение - если при печати чека по галочке было "распределение средств", и все эти договоры зависимы друг от друга, для этого же есть возможность передать несколько id договоров в "завершение формирования чека".

Подразумевается, что в этой очереди скапливаются платежи, которые забыли напечатать или не хотели. Рассматривать как справочную информацию и/или удобный поиск заглочивших при распечатке платежей. Двойным кликом можно перейти к договору.

Лог распечатанных платежей

Всё что уже напечаталось находится здесь. Можно перейти к договору. Печати чека нет. Есть фильтр по принтерам, в независимости от текущего выбранного. Рассматривать, как справочную информацию для истории.

Код	Договор	Комментарий до...	Тип платежа	Дата платежа	Комментарий пл...	Сумма	Дата логa	Регистратор	...
446...	x0000	ФИО - Иванов Ива...	Банк	19.05.2010		200.00	19.05.2010 12:56:24	2	
446...	x0000	ФИО - Иванов Ива...	Банк	19.05.2010		100.00	19.05.2010 12:42:59	2	
445...	dimon	тестовый договор	Банк	29.12.2009		1.00	29.12.2009 19:04:38	2	
444...	dimon	тестовый договор	Виртуальный	03.09.2009		200.00	03.09.2009 16:45:31	2	
444...	dimon	тестовый договор	Виртуальный	03.09.2009		100.00	03.09.2009 16:42:55	2	
443...	dimon	тестовый договор	Виртуальный	25.05.2009		1.00	25.05.2009 17:49:46	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 20:14:02	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 20:12:28	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 20:11:20	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 20:06:19	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:54:11	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:26:42	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:23:50	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:18:28	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:15:15	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:10:23	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:05:24	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 19:01:34	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 18:47:30	2	
438...	x0000	ФИО - Иванов Ива...	Банк	23.04.2009		1.00	23.04.2009 18:45:53	2	

× CashCheck

Выбор принтера, отчёты, сервис

Для того, чтобы можно было работать с каким-либо регистратором, его надо предварительно выбрать для пользователя. Это можно сделать во вкладке **Сервис** диалога **Плагины->CashCheck**.

Очередь Лог **Сервис**

Регистратор: **тестирование платежа (тест)** Пароль: **привет** **Отсоединить** **Обновить состояние**

Снятие отчётов: **Отчёт без гашения** **Отчёт с гашением** **Отчёт по секциям**

Возврат продажи: Сумма: Отдел: **Возврат**

Принудительное закрытие чека: Сумма: **Закреть чек**

Повтор чека (копия) **Продолжение печати** **Гудок** **Аннулирование чека**

Регистратор #5

Устройство (драйвер):
driver-emulator

Назначены типы платежей:
#24

Статус (драйвер):
Состояние: ОК

Статус сервера:
CmdTcpServer v build from
on: 0.0.0.0/0.0.0.0:9876
device driver: ru.bitel.frk.driver.emu.Driver@1ee2c2c
device status: good (:
thread counter: 3
started: 12.12.2011 18:25:00, uptime: 00:08:56
memory total: 55 312 384, max: 820 248 576, free: 45 548 240

X CashCheck

Выбирается из списка принтер, вводится пароль пользователя и нажимается **Выбрать**.

Для не выбранного принтера нельзя выполнять никакие операции. В этих случаях при первой попытке использовать принтер появится окошко с предложением залогиниться на принтер.

Вход на ККМ

Регистратор: **----** Пароль:

Ок **Отмена**

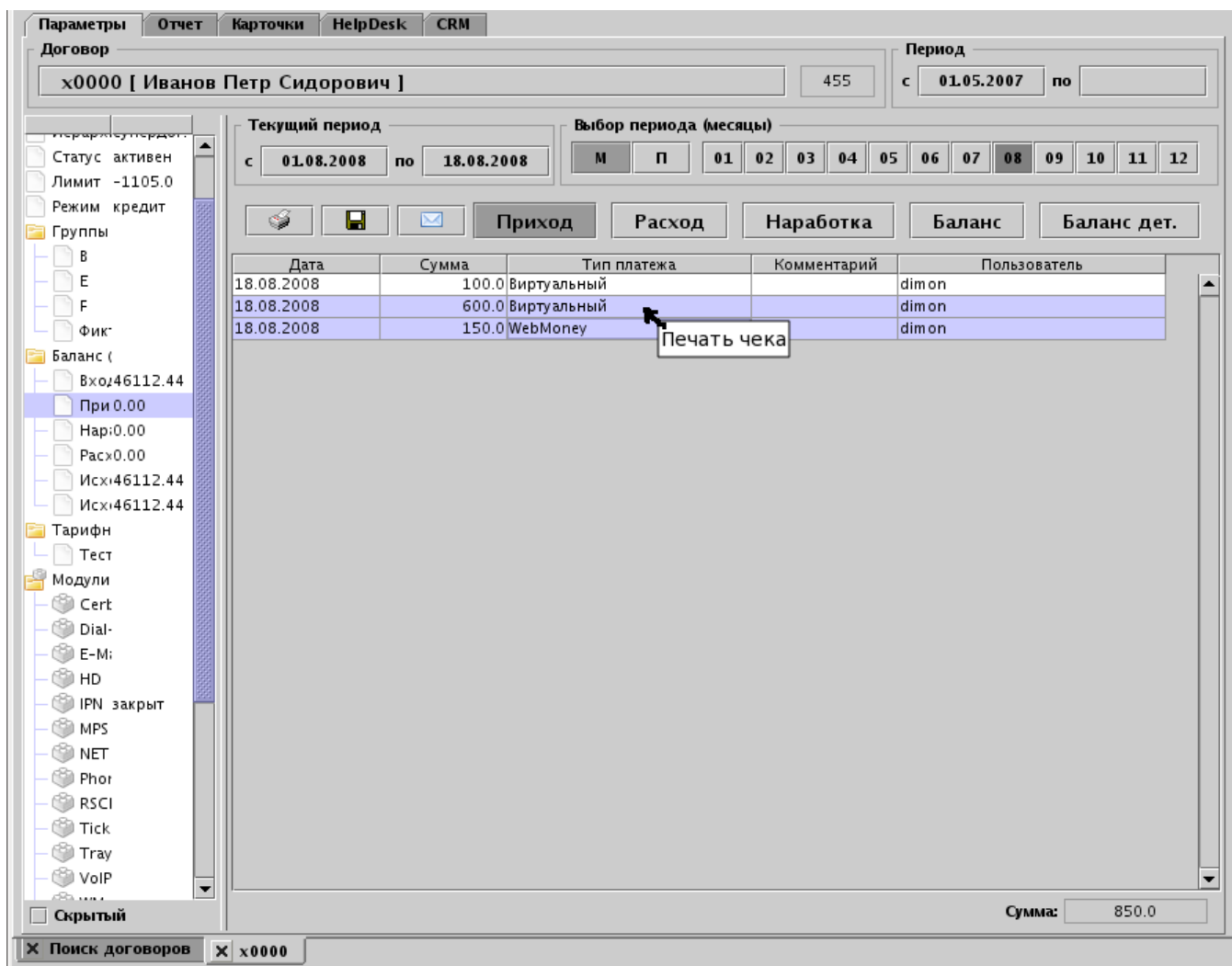
На этой же вкладке **Сервис** можно снимать отчёты, делать возвраты и производить другие действия (см. скриншот).

Также имеется кнопка "**привет**", по которой на выбранном в списке принтере печатается приветствие. Это было добавлено по просьбам пользователей, у которых много принтеров и кассиров - так проще контролировать тот ли принтер находится рядом с текущим рабочим местом.

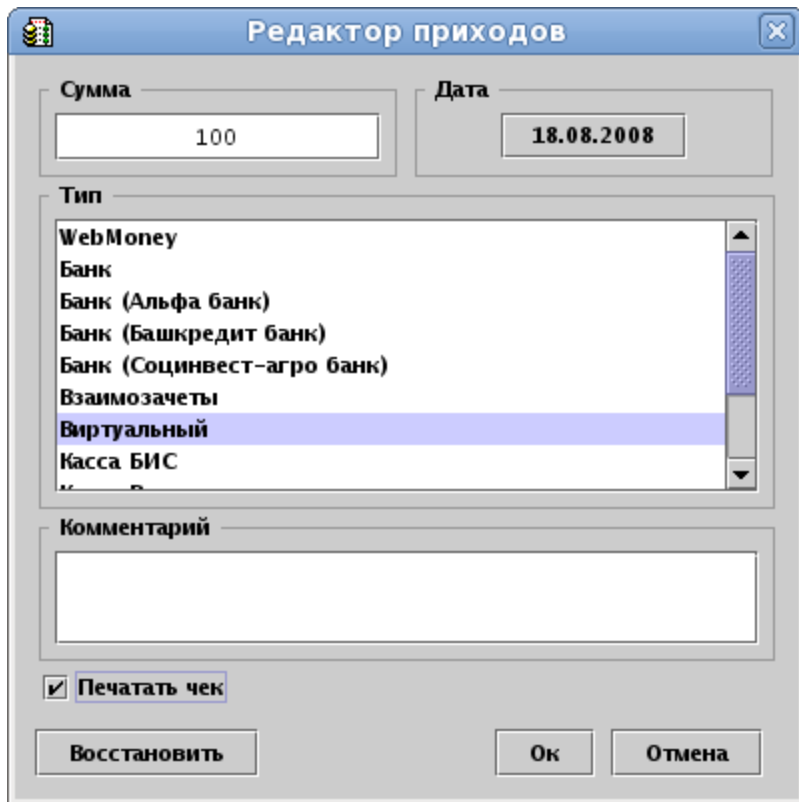
Печать чека

Помимо печати чека из очереди печати (см. выше) имеется возможность печатать чеки прямо из списка платежей в договоре или сразу же при добавлении платежа.

В первом случае надо выбрать один или несколько платежей в таблице платежей (в части, соответствующей платежам-приходам), и в контекстном меню по правой кнопке мыши выбрать пункт **Печать чека**.



Во втором случае можно при добавлении платежа поставить внизу "Редактора приходов" галочку **Печатать чек**. В этом случае в момент добавления платежа инициируется печать чека.



Редактор приходов

Сумма: 100

Дата: 18.08.2008

Тип:

- WebMoney
- Банк
- Банк (Альфа банк)
- Банк (Башкредит банк)
- Банк (Социнвест-агро банк)
- Взаимозачеты
- Виртуальный**
- Касса БИС

Комментарий:

Печатать чек

Восстановить Ок Отмена

Обратите внимание, если вы добавляете платёж с галочкой "напечатать", то платёж не добавится в очередь при успешной печати - предполагается, что он сразу ушёл на принтер. Если же не ушёл, то тогда он будет в очереди печати. Образно можно повторить [сказанное выше](#): очередь печати - это очередь ещё не распечатанных платежей.

Если при добавлении платежа происходит распределение средств, то напечатается чек с несколькими позициями.

Настройка галочки в диалоге прихода платежа

Имеется возможность настроить галочку "печатать чек" в диалоге прихода платежа. По умолчанию она снята и доступна для изменения. Возможно, вы захотите запретить всем пользователям, кроме некоторых менять её состояние. Или наоборот. Для этого предусмотрены следующие настройки **в конфигурации сервера биллинга** (sic!).

```
#  cashcheck,
#  " "
# defaultoff -      , (default)
# defaulton -      ,
# off -      ,
# on -      ,
client.gui.cashcheck.user.dimon.checkbox.mode=defaultoff
#
#      ,
#      "defaultoff"
client.gui.cashcheck.default.checkbox.mode=defaultoff
#      :
client.gui.cashcheck.user.dimon.checkbox.refund.mode=defaultoff
client.gui.cashcheck.default.checkbox.refund.mode=defaultoff
```

Таким образом, можно задать общее поведение чекбокса, а также перекрывающее его поведение для каждого отдельно взятого логина. Например, запретить всем и отдельно навсегда назначить её кассирам.

Фискализаторы

Фискализаторы - это первый слой абстракции плагина. Каждому настроенному **fr.*** в плагине сопоставлен фискализатор соответствующего типа. По умолчанию используется **cashcheckserver** - фискализатор работающий с сервером печати (для остальных фискализаторов сервер печати не нужен!). Для каждого возможен дополнительный конфиг для **fr.***. Тип фискализатора указывается:

```
fr.1.fiscalizator=atolonlinev4
```

Все возможности, описанные в разделе настроек плагина могут совершенно прозрачно работать с фискализаторами любых типов - дин.код, автоплатежи, отложенные платежи, возвраты, автозакрытия и т.д. и т.п. Но не все фискализаторы умеют всё. Большинство онлайн-сервисов не умеет ничего кроме "печати чеков", и настройки закрытия дня для них обычно бессмысленны.

OFD.ru Ferma

Реализует Программный интерфейс приложений (API) для работы с ИС "Ferma" от OFD.ru

Конфиг:

```
fr.101.title=OFD.ru Ferma
fr.101.fiscalizator=fermaofdru
fr.101.addr=https://ferma-test.ofd.ru/api/
fr.101.login=***
fr.101.pass=***
fr.101.timeout=5000

# ,      ( )
fr.101.Inn=6666666666
#
# "Common" "0" - ; "SimpleIn" "1" - ( ); "SimpleInOut" "2" - ( );
# "Unified" "3" - ; "UnifiedAgricultural" "4" - ; "Patent" "5" - .
fr.101.TaxationSystem=Common

# ( ) 1057
# ( AgentType, ..., SupplierPhone , ( " " ))
# BANK_PAYMENT_AGENT - ; BANK_PAYMENT_SUBAGENT - ; PAYMENT_SUBAGENT -
# CONFIDANT - ; COMMISSIONER - ; AGENT -
fr.101.AgentType=BANK_PAYMENT_AGENT
#
# ,
fr.101.TransferAgentPhone=79173412345
#
fr.101.TransferAgentName=
#
fr.101.TransferAgentAddress=
#
fr.101.TransferAgentINN=6666666666
#
fr.101.PaymentAgentOperation=
#
fr.101.PaymentAgentPhone=79173412345
#
fr.101.ReceiverPhone=79173412345
#
fr.101.SupplierPhone=79173412345
```

В дин.коде обязательно нужно задавать **setDocumentId(...)** - "Идентификатор счета, на основании которого генерируется чек", т.е. передавать туда id платежа удобнее всего.

Можно задавать **setDocumentDate(...)** - "Локальная дата и время чека", т.е. дата платежа. Если платежи уходят не сразу, а через delay-например.

Обязательно нужно задавать customerEmail.

Все параметры из конфига заявлены в протоколе как необходимые.

Налог задаётся через **setTax** целым числом и соответствует предусмотренным в системе налогам таким образом:

не задано/-1: «VatNo» – НДС не облагается;
0: «Vat0» – НДС по ставке 0%;
10: «Vat10» – НДС чека по ставке 10%;
18: «Vat18» – НДС чека по ставке 18%;
110 или **10110:** «CalculatedVat10110» – вычисленный НДС 10% от 110% суммы;
118 или **18118:** «CalculatedVat18118» – вычисленный НДС 18% от 118% суммы;
20: «Vat20» – НДС чека по ставке 20%;
120 или **20120:** «CalculatedVat20120» – вычисленный НДС 20% от 120% суммы.

Всё что добавлено через check.addString - в фискализации не используется и молча игнорируется.

Для проверки статуса чека предусмотрена отдельная **Задача запроса статуса чеков**. TODO

АТОЛ-Онлайн v4

Фискализатор **atolonlinev4** поддерживает онлайн-сервис "АТОЛ Онлайн, версия сервиса v4".

Основная конфигурация включает доступ к аккаунту и некоторые общие параметры чеков, которые не задаются в дин.коде формирования чеков.

```
fr.1.title=-
fr.1.fiscalizator=atolonlinev4

fr.1.addr=https://testonline.atol.ru/possystem/v4/
fr.1.groupcode=v4-online-atol-ru_***
fr.1.login=v4-online-atol-ru
fr.1.pass=***
fr.1.timeout=5000

# 1117
fr.1.company.email=chek@romashka.ru
# 1018
fr.1.company.inn=1234567891
# 1187
fr.1.company.payment_address=http://magazin.ru/
```

Особенности фискализатора:

В дин.коде обязательно нужно задавать **setDocumentId(...)** - "идентификатор документа внешней системы", т.е. передавать туда id платежа удобнее всего.

Можно задавать **setDocumentDate(...)** - "дата и время документа внешней системы", т.е. дата платежа. Если платежи уходят не сразу, а через delay-например.

Обязательно нужно задавать customerEmail. Можно задавать **setCashierName(...)**.

Налог задаётся через **setTax** целым числом и соответствует предусмотренным в системе налогам таким образом:

не задано/-1: «none» – без НДС;
0: «vat0» – НДС по ставке 0%;
10: «vat10» – НДС чека по ставке 10%;
18: «vat18» – НДС чека по ставке 18%;
110: «vat110» – НДС чека по расчетной ставке 10/110;
118: «vat118» – НДС чека по расчетной ставке 18/118;
20: «vat20» – НДС чека по ставке 20%;
120: «vat120» – НДС чека по расчетной ставке 20/120.

Всё что добавлено через check.addString - в фискализации не используется и молча игнорируется.

Для проверки статуса чека предусмотрена отдельная **Задача запроса статуса чеков**.

Комтет Касса

Фискализатор **komtet** поддерживает онлайн-сервис "комтет касса" (<https://kassa.komtet.ru/>).

Основная конфигурация включает доступ к аккаунту и некоторые общие параметры чеков, которые не задаются в дин.коде формирования чеков.

```
fr.102.title=komtet
fr.102.fiscalizator=komtet
fr.102.shopId=547654hgh676
fr.102.secret=6765tfhfg67
fr.102.queue=666
fr.102.timeout=10000
#
fr.102.sno=0
#
fr.102.payment_address=bgbilling.ru
# ("supplier_info"),
# [: 1171] .
# , «+», 19 . ( )
fr.102.supplier_info.phones="+79270989548
# [: 1225]
fr.102.supplier_info.name= ''
# [: 1226]
fr.102.supplier_info.inn=366406939791
```

Особенности фискализатора:

В дин.коде обязательно нужно задавать **setDocumentId(...)** - "идентификатор документа внешней системы", т.е. передавать туда id платежа удобнее всего (с префиксами "p" / "с" для платежей/расходов чтобы не пересекались ид в платежах/возвратах).

Обязательно нужно задавать customerEmail. Можно задавать **setCashierName(...)** + **setCashierINN(...)**. Используется признак OnlyElCheck (но неизвестно как применяется в системе).

Всё что добавлено через **check.addString** - в фискализации не используется и молча игнорируется.

Налог задаётся через **setTax** целым числом и соответствует предусмотренным в системе налогам таким образом:

не задано/-1: "no - [Ter: 1105] Без НДС"
0: "0 - [Ter: 1104] НДС 0%"
10: "10 - [Ter: 1103] НДС 10%"
110: "110 - [Ter: 1107] НДС 10/110"
20: "20 - [Ter: 1102] НДС 20%"
120: "120 - [Ter: 1106] НДС 20/120"

[Ter: 1055] Система налогообложения, возможные значения (настраивается в конфиге):

0 - ОСН
1 - УСН доход
2 - УСН доход - расход
4 - ЕСХН
5 - Патент

Вид оплаты Возможные значения (цифры взяты условно по значению тегов) (устанавливается в дин.коде):

1081: "card - [Ter: 1081] оплата безналичными (по умолчанию)"
1031: "cash - [Ter: 1031] оплата наличными"
1215: "prerepayment - [Ter: 1215] сумма предоплатой (зачет аванса и/или предыдущих платежей)"
1216: "credit - [Ter: 1216] сумма постоплатой (кредит)"
1217: "counter_provisioning - [Ter: 1217] сумма встречным предоставлением"

[Ter: 1214] Способ расчета Возможные значения (цифры взяты из значений тега в ффд) (устанавливается в дин.коде):

1: pre_payment_full - полная предварительная оплата до момента передачи предмета расчета «ПРЕДОПЛАТА 100%»
2: pre_payment_part - частичная предварительная оплата до момента передачи предмета расчета - «ПРЕДОПЛАТА»
3: advance - аванс
4: full_payment - полная оплата, в том числе с учетом аванса (предварительной оплаты) в момент передачи предмета расчета - «ПОЛНЫЙ РАСЧЕТ»
5: credit_part - частичная оплата предмета расчета в момент его передачи с последующей оплатой в кредит - «ЧАСТИЧНЫЙ РАСЧЕТ И КРЕДИТ»
6: credit - передача предмета расчета без его оплаты в момент его передачи с последующей оплатой в кредит - «ПЕРЕДАЧА В КРЕДИТ»
7: credit_pay - оплата предмета расчета после его передачи с оплатой в кредит (оплата кредита) - «ОПЛАТА КРЕДИТА»

[Тег: 1212] Предмет расчета Возможные значения (цифры взяты из значений тега в ффд) (устанавливается в дин.коде):

1: product - товар, за исключением подакцизного товара

3: work - работа

4: service - услуга

10: payment - об авансе, задатке, предоплате, кредите, взносе в счет оплаты, пени, штрафе, вознаграждении бонусе и ином аналогичном предмете расчета

11: commission - вознаграждении пользователя, являющегося платежным агентом (субагентом), банковским платежным агентом (субагентом), комиссионером, поверенным или иным агентом

13: other - о предмете расчета, не относящемуся к предметам расчета, которым может быть присвоено значение от «0» до «12» (0-12 -- это вышеперечисленные)

Для проверки статуса чека предусмотрена отдельная **Задача запроса статуса чеков**.

Локальные ККТ (сервер печати)

Фискализатор **cashcheckserver** поддерживает локальные ККТ. Для его работы требуется установить отдельное приложение - сервер печати (для каждого настроенного фискализатора - по одному серверу печати - для каждого отдельного ККТ). Подробно рассмотрено в соответствующем разделе.

Основная конфигурация включает коннектор до сервера печати. **Обратите внимание**, что сервер печати (часто он **на клиентской машине**, куда физически подключен ККТ, если речь не про автоплатежи) должен быть доступен по этому ip:port с машины **сервера биллинга**.

```
fr.1.title=  
fr.1.fiscalizator=cashcheckserver  
fr.1.connector=192.168.6.66:9876  
  
#  
# , " ( 30-60 ) .  
fr.1.timeout.connect=60000
```

Пустой фискализатор

Фискализатор **empty** может потребоваться для тестирования. Или для того, чтобы часть чеков нигде не печатать. Например, чеки одного типа для физ.лиц печатаются на настоящем принтере, а для юр.лиц в дин.коде заворачиваются на пустой фискализатор. Если просто их проигнорировать - они будут везде выглядеть как ненапечатанные, в очереди мешаться и т.д. Если же не печатать, но отметить как напечатанные, то не будут в очереди, но будет неотличимо от настоящих чеков, что не всегда удобно. (Более неудобная альтернатива: настроить фискализатор "сервер печати" и в нём пустой драйвер-эмулятор).

Основная конфигурация пустая.

```
fr.1.title=  
fr.1.fiscalizator=empty
```

Фискализатор на динамическом классе

Можно написать фискализатор в дин.коде.

```
fr.99.title=  
fr.99.fiscalizator=dyncode  
fr.99.dyncode=ru.bitel.bgbilling.cashcheck.TestDyncodeFiscalizator
```

Класс должен наследовать **ru.bitel.bgbilling.plugins.cashcheck.server.fisc.Fiscalizator** и реализовывать по меньшей мере методы: **status**, **name**, **getFiscalizatorStatus**, **isAsynchronous** а также, очевидно, **check** и **checkreturn**. Если фискализатор асинхронный (требует последующего запроса статуса платежа), то должен быть реализован метод **checkPending**.

CashCheck-сервер (сервер печати)

Сервер печати используется в качестве фискализатора для локальных ККТ, т.е. в случае использования в плагине фискализатора **cashcheckserver**. Сервер печати распространяется в виде отдельного пакета BGCashCheckServer, его структура во многом схожа с другими серверами BGBilling. Установка, настройка и запуск осуществляется аналогично. Подготовительные действия смотрите в разделах [установка сервера под Linux](#) и [установка сервера под Windows](#), здесь же будут приведены уточнения. Обновление происходит с сервера биллинга, как и в случае остальных приложений. Не забудьте прописать в application.xml параметры и обновить сразу после первоначальной установки!

Необходимы переменные окружения JAVA_HOME (полный путь до корня используемой JRE) и BGCASHCHECK_SERVER_DIR (полный путь до корня установленного bgcashcheckserver).

Под Windows это должны быть обязательно системные (а не пользовательские) переменные окружения, если требуется запускать как windows-службу. Обратите внимание, т.к. это очень распространённая ошибка.

Для запуска сервера как службы эти условия обязательны. Для проверки просто запуска через bat/sh можно указать вручную переменные в файлах запуска server.bat/sh, testserver.bat/sh и т.д. (см. в начале скриптов установку этих переменных, раскомментировать их установку и скорректировать на нужные пути).

Установка службы под Windows аналогична установке других серверов, запуск производится через JSL, установка службы - server_install.bat.

Установка демона под Linux тоже стандартна (приведён пример, для настройки под ваш дистрибутив обратитесь к документации по системе):

- Копируем скрипт(ы) из linux_service в /etc/rc.d/init.d
- Устанавливаем права

```
chmod 755 /etc/rc.d/init.d/bgcashcheckserver
```

- Добавляем службу

```
chkconfig --add bgcashcheckserver
```

- Включаем службу, например, для уровней 2,3,4 и 5

```
chkconfig --level 2345 bgcashcheckserver on
```

Касается установки под Linux. Обратите внимание на то, что пользователю, от которого запускается приложение, должны быть доступны порты (устройства /dev/ttyS*). Проверьте их группу и владельца, а также права доступа. При необходимости добавьте пользователя в нужную группу (может быть uucp, dialout или другая в вашей системе). Иначе порт будет недоступен, а в testserver выведется пустой список портов. Если сервер стартует как демон, то это, в общем случае, не актуально (стартует от суперпользователя), но утилиты (testserver и пр.) всё равно не будут корректно работать.

Настройка сервера печати и оборудования, поддерживаемые устройства

Конфигурация сервера печати находится в файле **setupfrk.config**. Конфигурация содержит комментарии и все варианты своих параметров, так что настройка не должна вызвать затруднения.

Два параметра - **driver** и **port** - управляют связкой "устройство - сетевой порт", остальные параметры конфигурации специфичны для каждого поддерживаемого драйвера (см. параметры в описании каждого драйвера).

Параметр **port**, как понятно, определяет порт, на котором сервер печати слушает подключения сервера биллинга.

Параметр **driver** указывает на класс драйвера устройства. В данный момент поддерживаются следующие устройства (в каждом подразделе даётся строка для указания в качестве имени драйвера) со следующими параметрами (см. ниже).

На данный момент это всё оборудование, которое поддерживается сервером печати. Работоспособность других устройств не гарантируется (более того - маловероятно). Но реализация этой поддержки, конечно, возможна.

В конфигурации сервера печати прописаны некие параметры "по умолчанию" для каждого устройства на момент написания драйвера. Вполне возможно, что у вас настройки будут иные. Внимательно проверяйте все параметры! Например, несовпадение параметров СОМ-порта в устройстве, настройках драйвера и в самой системе чаще всего приведёт к невозможности работы с этим устройством.

Фискальный регистратор Штрих-ФР-К для использования его в VGBilling

```
driver: ru.bitel.frk.driver.shtrih2.Driver
```

Конфигурация:

- **portName** - имя порта, например, в Windows **com<X>**, в Linux **/dev/ttyS<X>**;
- **baudRate** - скорость обмена (**2400, 4800, 9600, 19200, 38400, 57600, 115200**);
- **flowControlIn, flowControlOut** - описание типа flow control (**None, Xon/Xoff Out, Xon/Xoff In, RTS/CTS In, RTS/CTS Out**);
- **databits** - настройка data bits (**5, 6, 7, 8**);
- **stopbits** - настройка stop bits (**1, 1.5, 2**);
- **parity** - настройка parity (**None, Even, Odd**);
- **oneByteTimeout** - таймаут приёма одного байта, характеристика порта (по умолчанию 100).

Драйвером **поддерживаются** все современные устройства от Штрих по 54-ФЗ.

Проверены не все устройства, но согласно описанию протокола должны работать все. Подробная сверка моделей и наличия в них используемых команд не проводилась (за исключением обычных, распространённых моделей - ФР-К и т.д.), поэтому, возможно, что среди указанных моделей есть специфические устройства, которые будут работать с ограниченным функционалом.

Ограниченно поддерживаются обычные, без ФП принтеры.

Полноценной заявки поддержки нет ввиду нереализованных в них команд типа "продажа" и подобных. Но можно подключить и запустить это устройство, но в скрипте обработки формирования чека (см. ниже) формировать нужные строки, без команды добавления платежа, только командами добавления строк, аналогично тому, как это описано для FOP-драйвера сервера печати. Таким образом, несложно вручную сформировать любой вид чека, аналогично обычным ККМ. Но, конечно, чека в данном случае нефискального.

Имеются дополнительные параметры драйвера (использовать по ситуации):

```
#
waitNoprintTimeout: 1000

# - ( )
waitNoprintNumtry: 20

# ( )
# - 0 ( )
cutterDisabled: 1

# ( )
# - 0 ( , )
#cutterForceManual: 0
```

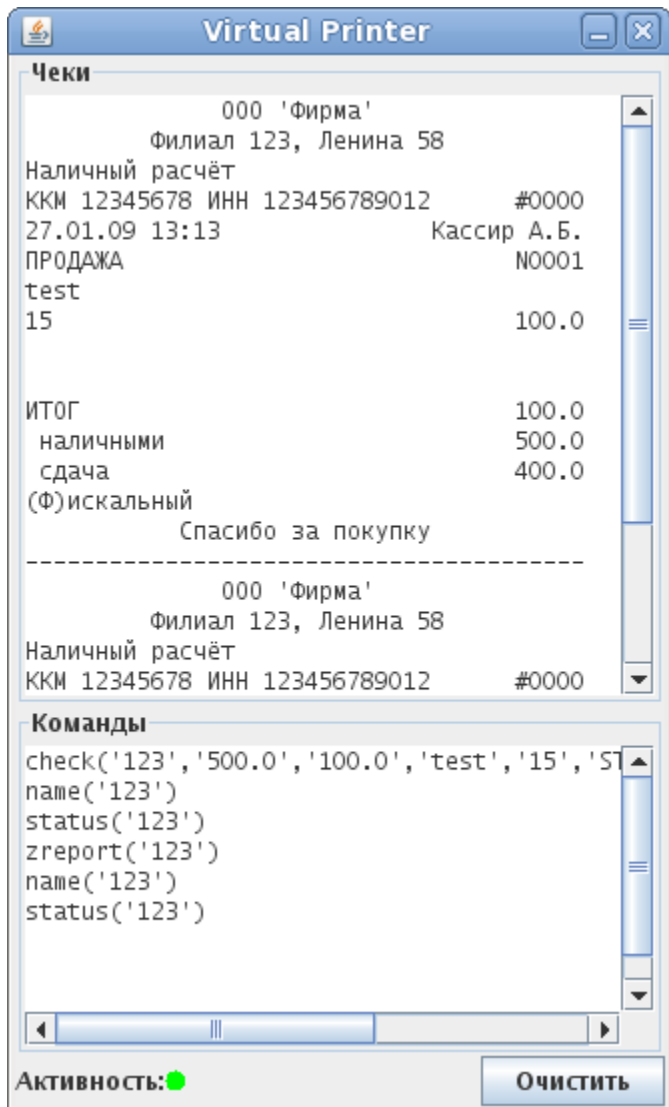
Здесь и далее - флаг **cutterDisabled** действует только на ручную команду отрезки при печати произвольного текста или при приветствии итд итп. А отрезка или неотрезка при чеках/отчётах настраивается в принтере. После печати чека должно резать автоматически, это заложено в программе регистратора и настраивается в нём самом.

Эмулятор принтера, подразумевающегося к использованию в BGBilling

```
driver: ru.bitel.frk.driver.emu.Driver
```

Представляет собой некую заглушку. Если нет реального устройства или оно не поддерживается/не работает, можно запустить и протестировать с этим драйвером-эмулятором. При этом некоторый вывод драйвера (вместо вывода на реальную бумагу) выкидывается в консоль (и, следовательно, попадает в cashserver.out). Дополнительной конфигурации этот драйвер не требует.

К этому драйверу прилагается визуальный "эмулятор принтера", очень грубым образом отображающий возможный вид "напечатанных" чеков. Этот эмулятор интерпретирует консольный вывод драйвера-заглушки и рендерит чеки, отдалённо напоминающие чеки регистраторов Штрих. Может применяться для оценки работы сервера печати и системы в целом.



Приложение виртуального принтера запускается из папки сервера печати: virtualprinter.sh(.bat). Параметры можно увидеть при запуске.

Использование: <virtualprinter>[.bat].sh [--noskipexist] [--encoding=<encoding>] <logfilename>

где --noskipexist - не пропускать уже существующие на момент запуска виртуального принтера строки в выводе драйвера-эмулятора;

--encoding=<encoding> - указание кодировки исходного файла (вывода эмулятора);

logfilename - путь к файлу 'cashserver.out'. Например:

```
./virtualprinter.sh ./log/cashserver.out
```


Любой системный принтер для печати на нём XSL-FO шаблонов

```
driver: ru.bitel.frk.driver.fop.Driver
```

Представляет собой FO-транслятор, который действует так:

1. формирует из приходящих данных (текстовые строки) **xml**;
2. обрабатывает с помощью этой **xml** указанный заранее **xsl:fo** шаблон;
3. выводит его на печать на указанном принтере.

Необходимая конфигурация сервера печати:

```
# ( , , , , )
printer=pdf

# xsl. ( , , xsl
# , , , )
#xsl=http://127.0.0.1:8080/bgbilling/xsl/cashcheck_fop.xsl
#xsl=file:///home/dimon/workspace/bgbilling/modules/cashcheck/server/server_files/xsl/cashcheck_fop.xsl
xsl=file:///usr/local/BGCashcheckServer/xsl/cashcheck_fop.xsl

# ( ), " "
jobName=fop driver printing

# , " "
jobCopies=1
```

К этому драйверу нужны две сущности: скрипт поведения, который формирует нужные строки для драйвера, и xsl-шаблон, который преобразуется с помощью xml, сформированной драйвером, в готовый документ для печати. Пример скрипта поведения см. ниже, в разделе "настройка внешнего вида чеков". Смысл скрипта состоит в заполнении набора строк с помощью конструкций вида

```
check.addString( " " );
```

порядок строк важен для шаблона, там они выбираются по номерам. Список строк драйвер преобразует в xml такого вида:

```
<?xml version="1.0" encoding="UTF-8"?>
<data clientsumma="10.0" summa="10.0">
  <line n="1" text="120,00." />
  <line n="2" text="2" />
  <line n="3" text="3" summ="10.0" dep="15" />
</data>
```

Такую xml и надо иметь ввиду при создании и редактировании xsl-шаблона. Последняя, третья, строка получена с помощью метода **addPayment** в скрипте обработки, а не **addString**. Сам шаблон представляет собой любой валидный FO-документ. Получить в этом шаблоне значения строковых параметров из драйвера можно таким кодом (см. исходную xml выше):

```
<!-- : 120,00. -->
<xsl:variable name="param_summ" select="line[@n='1']/@text" />
```

Таким образом, в xsl-переменной окажется значение атрибута "text" из первой строки ("n"=1).

При правке или замене XSL-FO шаблона при необходимости видеть результат внесённых изменений необходимо перезагрузить CashCheck-сервер, т.к. шаблон загружается и распознаётся при инициализации драйвера во время старта сервера. Для оптимизации по скорости.

Устройства с протоколом от компании АТОЛ

Данный драйвер сервера печати работает с ККТ только по транспортному протоколу версии 2.4 (не 3.x), нужно вручную переключить в сервисном меню устройства (т.е. через кнопку протяжки ленты, а не в утилите и не где-то ещё). К сожалению, родная утилита "тест драйвера" от Атола до версии 9 при этом перестаёт работать, до переключения обратно, в новых версиях утилиты/драйвера есть возможность выбора режима подключения.

В данный момент драйвер не рекомендуется и устарел в пользу atollibfptr.

```
driver: ru.bitel.frk.driver.atol3.Driver
```

Драйвер реализует поддержку протоколов, основанных на разработках компании АТОЛ. В данный момент это все современные устройства АТОЛ, поддерживаемые по 54-ФЗ.

Параметры для этого драйвера такие:

portName - имя порта, например, в Windows COM<X>, в Linux /dev/ttyS<X>;

baudRate - скорость порта. По протоколу заявлены следующие возможные значения: 1200, 2400, 4800, 9600, 14400, 38400, 57600, 115200.

Остальные параметры порта по умолчанию заявлены в протоколе следующими: 1 стартовый бит, 8 битов данных, 1 стоповый бит, без проверки на четность, 3 линии (TXD, RXD, GND);

type - так как драйвер поддерживает несколько принтеров, разновидностей и их платформ, то необходимо указать тип принтера заранее. В конфигурации приведены все возможные значения;

accessPassword - пароль доступа к ККМ (не пароль кассира/админа/сисадмина!). 4цифры, по умолчанию "0000", согласно протоколу.

Имеются дополнительные параметры драйвера (использовать по ситуации):

```
#      (      )
# - 0 ( )
cutterDisabled: 1

#      (      )
# - 0 ( , , )
#cutterForceManual: 0
```

Также есть несколько специфических параметров, см. комментарии в конфиге сервера печати.

ККТ Старрус РП-Система 1ФА

```
driver: ru.bitel.frk.driver.starrus.Driver
```

Поддерживаются устройства Старрус ККТ РП-Система 1ФА, работающие по сетевому протоколу обмена.

Параметры для этого драйвера такие:

address, port - адрес и порт TCP;

cutterDisabled - запрет отрезчика.

Устройства АТОЛ работающие с родным драйвером ККТ

```
driver: ru.bitel.frk.driver.atollibfptr.Driver
```

Драйвер реализует поддержку родного драйвера от АТОЛ для ККТ. На данный момент реализована поддержка десятой версии драйвера (через вращатель libfptr10.jar).

Параметры для этого драйвера см. в **setupfrk.config**. Поддерживаются (передаются в драйвер без изменений) любые настройки допустимые для драйвера, они перечислены в конфиге, более подробные описания: http://integration.atol.ru/?java#driver_settings. В том числе любые методы коннекта, которыми может оперировать драйвер АТОЛ ККТ. Данная схема включает нативную часть, различную под разные платформы. Для установленного через инсталлятор драйвера должно всё найтись само. Для ручных установок в дистрибутиве сервера печати лежат бинарные библиотеки под разные платформы. TODO ручная установка нативных библиотек.

Особенность: драйвер не авторизует кассиров по двузначному паролю, для всех команд этот пароль по сути игнорируется (но требуется фейковый логин с любым паролем). Вместо этого используются параметры "имя кассира" и "инн кассира". Для чека их можно (нужно) передавать из скрипта методами `check.setCashierName` и `check.setCashierINN`, для операций открытия и закрытия смены используются значения из конфига драйвера сервера печати (`setupfrk.config`). Они же используются и как ФФД-теги.

Запуск сервера печати

Подготовка службы (для Windows-системы) и запуск сервера печати аналогична подобной процедуре при запуске сервера биллинга. Скрипты с аналогичными названиями присутствуют.

Некоторыми пользователями утверждается, что бывают проблемы при маппинге последовательных портов на тонких клиентах в определённых средах. См. предостережение в главе 1.

Запуск двух копий сервера

Запуск двух копий сервера может понадобиться, например, при подключении двух разных или одинаковых устройств к одному компьютеру. Часто такое бывает при запуске клиентов кассиров через терминальную архитектуру или в каких-то других случаях.

Чтобы запустить два сервера на одной машине, вам придётся сделать две копии папки программы и две разных переменных окружения, например, **%BGCASHCHECK_SERVER_DIR_1%** и **%BGCASHCHECK_SERVER_DIR_2%**. Соответственно, все используемые файлы запуска (bat,sh, server.ini) тоже надо соответствующим образом подправить на эти разные переменные. Не забудьте указать разные порты в конфигурации серверов.

В итоге, файлы запуска будут выглядеть примерно так (для Windows):

```
set CLASSPATH=%BGCASHCHECK_SERVER_DIR_1%;%BGCASHCHECK_SERVER_DIR_1%\lib\*
java -cp %CLASSPATH% ru.bitel.frk.server.CmdTcpServer %1 %2 %3 > %BGCASHCHECK_SERVER_DIR_1%\log\cashserver.out
2>&1
```

Для запуска двух серверов, как Windows- служб надо соответственно поправить файлы **server_install.bat**, **server_uninstall.bat**, **server.ini**. Необходимо заменить в них переменные окружения **%BGCASHCHECK_SERVER_DIR%** на разные, а также изменить названия службы (параметры appname, servicename, displayname в конфиге **server.ini**). Не забывайте про корректное указание двух соответствующих переменных окружения, которые должны прописываться как системные переменные окружения (см. установка и запуск BGBilling-сервера).

Тестирование

В сборке сервера имеется скрипт **testserver.bat(.sh)** для запуска программы самотестирования принтера и сервера печати. Производится проверка:

1. физического доступа к serial-портам;
2. прямого доступа к железной части устройства принтера;
3. сетевого доступа к принтеру, эмулируется работа клиента (в данном случае - сервера биллинга);
4. доступа к серверу для получения статуса;
5. всех драйверов, которые физически нашлись в сборке сервера, для каждого проводится процедура самотестирования (процедура driver touch).

Результаты работы скрипта выводятся в stdout и stderr, при необходимости можно самостоятельно перенаправить вывод в лог файл (например, **/log/testserver.out**) для анализа. Пример строки запуска можно увидеть ниже. Нормально отработал или нет - видно по отсутствию ошибок в логе и надписи **TEST COMPLETED** в конце вывода скрипта. Эта строка не означает, что всё закончилось успешно, а просто что всё закончилось!

Если сервер запустился и скрипт **testserver** выдаёт приличные результаты, то серверная сторона печати чеков готова принимать команды и работать с принтером.

Программа testserver также использует log4j, настройки находятся в том же конфиге, но под меткой "testserver".

В программу testserver опционально передаётся пароль для кассира/администратора. То есть тот, что передаётся в драйвер, при логине кассира. Пароли доступа, если они нужны - будут браться из конфига.

При запуске приложения необходимо указать некоторые параметры, при этом делается либо тест, либо touch, либо всё вместе. Параметры запуска:

```
testserver[.bat|.sh] [--dotest] [--dotouch] [--passwordXXX] [--driversDRIVER1,DRIVER2,...]
```

где --dotest делать тесты;

--dotouch делать тач драйверов;

--password(пароль) задать пароль оператора для некоторых тестов (без пробелов и скобок), иначе не задан (равен нулю);

--drivers(драйвер1),(драйвер2)... задать драйвер(ы) для теста (без пробелов и скобок), иначе выбраны все.

Также, как сказано выше, при необходимости надо перенаправить вывод в лог, к примеру, так:

```
./testserver.sh --dotest --dotouch > test.log 2>&1
```

Так, к примеру, выглядит тест драйвера штриха с созданием файла testserver.log для анализа (под Linux и Windows соответственно):

```
./testserver.sh --dotouch --driversru.bitel.frk.driver.shtrih.Driver,ru.bitel.frk.driver.shtrih2.Driver >
testserver.log 2>&1
./testserver.bat --dotouch "--driversru.bitel.frk.driver.shtrih.Driver,ru.bitel.frk.driver.shtrih2.Driver" >
testserver.log 2>&1
```

Анализ ошибок и логирование

Сервер печати CashCheck логирует большинство своих действий. Для вывода используется библиотека log4j. Также в некоторых местах используется прямой вывод в консоль. В основном это специальный консольный вывод и незалогированные ошибки сервера. По умолчанию основной лог пишется в файл `./log/cashserver.log`, а консольный вывод - в `./log/cashserver.out`.

В разных режимах работы логгера log4j может выводиться разная информация. Примерное распределение информации по уровням логгирования:

- **error**: ошибки сервера, потоков команд, работы драйверов;
- **warn**: ошибки протокола (неверные пропущенные заголовки итд), ошибки, возвращенные драйвером и переданные вызывающему;
- **info**: старт, стоп, получение команд;
- **debug**: получение сервером запросов, старт/стоп каждого потока команд;
- **trace**: содержимое команд и ответов на них, низкоуровневый вывод драйверов и устройств.

Полный вывод может понадобиться для отладки конкретного устройства. Например, когда возникает ошибка соединения, с помощью подробной трассировки можно понять на каком этапе была авария. Часто подобная информация нужна разработчикам при дописывании программы для поддержки какого-либо устройства. Чтобы сменить режим, поменяйте значение параметра **priority** в конфигурации **log4j-cashcheck.xml** (может потребоваться перезагрузка сервера печати):

```
<root>
  <priority value="TRACE" />
  <appender-ref ref="cashserverfilelog" />
</root>
```

Для более полного понимания логики работы логгера читайте документацию по log4j.

Плагин CRM

Введение

Данный плагин предоставляет лишь простейший функционал CRM-системы. В случае необходимости использования полноценной CRM /ERP системы рекомендуем использовать систему [BGERP](#), имеющую функционал интеграции с [BGBilling](#).

Плагин CRM позволяет учитывать звонки клиентов, вести учёт и разбор проблем, а также имеет гибкую подсистему выдачи, отслеживания, выполнения и обработки выполненных задач.

Журналы задач и звонков доступны как в глобальном контексте в меню **Плагины=>CRM**, так и в контексте одного договора на вкладке **CRM**. Журнал проблем только глобальный.

Данные вкладки доступны только после установки и активации плагина CRM.

Настройка плагина

В конфигурации плагина вы можете поменять темы писем с отчётами по звонкам, проблемам и задачам. Другие опции описаны ниже.

```
register.task.call.subject=
register.problem.mail.subject=
register.task.mail.subject=
#
#close.task.future=1

# . ,
#
#
#contract.fio.param.id=10, 20

# . , .
#register.param.phone=10

# . , .
#, 49 50 phone, 24 text.
#contract.phone.param.id=49,text:24,50

#
#
#problem.active.record.limit=50
#
#problem.active.period.days=60
```

Справочники

Справочники Журнала проблем расположены в меню **Справочники=>Другие**, их названия начинаются с префикса "CRM -". Для начала работы необходимо указать:

- **типы звонков** клиентов и группу по умолчанию, на которую переводится проблема по каждому типу звонка;
- **группы решения** проблем и задач и e-mail адреса, на которые слать информацию по проблемам/задачам/звонкам при нажатии кнопки ОК+Е-Mail (несколько адресов вводятся через запятую);
- **исполнители**, каждый из которых может входить в несколько групп;
- **категория** - характеристика, которая назначается проблеме после её разрешения (конечная причина), используется для отчётности;
- **тип задачи** - характеристика, которая назначается задаче.

Журнал звонков

Звонки обладают только одной настраиваемой характеристикой: **тип** и могут быть привязаны к какой-либо не закрытой на данный момент проблеме. При этом проблема может быть создана автоматически и привязана к данному звонку.

При привязке к звонку проблемы и нажатии **OK+Mail** письмо отсылается на группу решения проблемы. Шаблон письма - файл **register_call.xml**

Не допускайте скопления большого количества не закрытых проблем в журнале проблем! Это может привести к очень медленному открыванию редактора в журнале звонков.

Если звонок добавлен в договоре на вкладке **CRM**, то он привязывается к данному клиенту, иначе звонок считается не клиентским.

Начиная с 4.3 версии звонок может быть привязан к объекту договора.

Код	Договор	Тип	Время	Принят	Проблема
130...	27NK9271-08	Перевод на сисадм...	09.12.2008 00:50:03	Подалко Антон Вл...	28035;закрота (Админы - настройка ПО);27NK
130...	27NK7599-08	Перевод на сисадм...	09.12.2008 01:50:21	Подалко Антон Вл...	28036;закрота (Админы - Настройка оборудо
130...	27NK7610-08	Перевод на сисадм...	09.12.2008 07:20:07	Подалко Антон Вл...	28037;закрота (Админы - настройка ПО);27NK
130...		Есть ли в нашем до...	09.12.2008 09:14:06	Гудовских Лидия ...	
130...	27КФ11638-08	Качество вещания (...)	09.12.2008 09:19:35	Гвбанова Юлия С...	28038;закрота (Иная проблема);пер.Музыкаль

Редактирование звонка разбито на 3 вкладки .

Вкладка "Тип Звонка" : Тут можно выбрать тип звонка.

Редактор

Тип звонка | Группа | Проблема

- Местонахождение офиса
- Пакет каналов
- Перевод в офис
- Перевод на отдел подключений (Интернет)
- Перевод на отдел подключений (КТВ)
- Перевод на отдел продаж**
- Перевод на расчётную службу
- Перевод на сисадминов
- Провести обследование КТВ
- Пункты приёма платежей
- Ремонт интернет
- Состояние счёта (Интернет)
- Состояние счёта (КТВ)

Договор: 27NK9271-08

OK OK + EMail Отмена

Вкладка "Группа": Здесь выбирается группа звонка.

Журнал проблем

Проблема глобальна, но к ней могут быть привязаны звонки клиентов. Проблема обладает следующими характеристиками:

- текущая группа решения;
- текущий список исполнителей;
- список привязанных к проблеме звонков;
- комментарий - исходное описание проблемы;
- участвовавшие группы - список групп, между которыми переходила проблема (по ходу разбора);
- ход решения - текстовое описание процесса решения проблемы;
- категория - характеристика, назначаемая проблеме после её разрешения (конечная причина), используется для отчётности.

В ходе решения проблема проходит следующие состояния: открыта - принята - закрыта. При этом принятие проблемы обычно означает назначение ей конкретной группы решения и исполнителей. Кроме этого можно добавлять свои статусы. Для настройки статусов нужно в добавить в конфигурацию плагина:

```
# , "0;1;2:"
register.problem.status.list=0;1;2;3:
#
register.problem.status.default=0
#
register.problem.status.close=2
```

Для большего удобства проблема имеет характеристику "срочность", в зависимости от которой меняется цветовая индикация в таблице проблем.

Скриншот интерфейса "Журнал проблем". Вверху видны вкладки: Звонки, Проблемы, Задачи, Работы. Активна вкладка "Проблемы".

Фильтр:

- Статусы: открыта, принята
- Группы: [пусто]
- Исполнители: [пусто]
- Кнопки: Все, Сброс, Инверт.

Сортировка:

- время откр. обр. [выпадающий список]
- срочность обр. [выпадающий список]

Всего проблем: 117

Код	Срочн...	Группа	Комментарий	Статус	Статус(время)	Статус(пользов.)	Длительность	Договора
175	0	Диспетчера	зззззззззз...	закрыта	20.03.09 17:25	Сергеев Кирилл	0 d 00:09:34	
174	0	Диспетчера	Не работае...	принята	24.03.09 11:45	Сергеев Кирилл	6 d 02:27:52	king
173	-2	Диспетчера	zxc	отложена	20.03.09 18:48	Сергеев Кирилл	9 d 19:24:18	king, king
172	0	Диспетчера	ggggggggg...	отложена	20.03.09 18:49	Сергеев Кирилл	9 d 19:23:54	
171	-2	Диспетчера	888888888...	открыта	12.03.09 15:47	Denis Pimenov	17 d 22:26:04	king
170	-2	Диспетчера	999999999...	открыта	12.03.09 15:43	Denis Pimenov	17 d 22:29:21	king
169	0	Диспетчера	[Блокировк...	открыта	11.03.09 18:48	Сергеев Кирилл	18 d 19:24:16	x00
168	0	Диспетчера	[Консульта...	открыта	11.03.09 18:48	Сергеев Кирилл	18 d 19:25:07	x00
167	0	Диспетчера	[Информац...	открыта	11.03.09 18:47	Сергеев Кирилл	18 d 19:25:41	x00
166	0	Диспетчера	[Блокировк...	открыта	11.03.09 18:30	Сергеев Кирилл	18 d 19:42:50	x00

В ходе решения проблема может переходить между различными группами; все участвовавшие группы запоминаются. В поле **Комментарий** вносится изначальное описание проблемы, **Ход решения** - описание процесса разбора. В столбце "Договора" показаны договоры, звонки которых привязаны к данной проблеме. Их можно также увидеть во всплывающем меню по нажатию правой кнопкой на конкретной проблеме, и, если выбрать в этом меню конкретный договор, то он откроется.

Статус: **открыта** Срочность: **0**

Группа: **Диспетчера**

Исполнители

Вася
 Петя

Комментарий
 fdgdfgd1

Категория: ---

Все **Сброс** **Инверт.**

OK **OK + E-Mail** **OK + Печать наряда** **Отмена**

Также как и звонок проблема может быть отправлена на почту текущей группе решения, шаблон письма - файл register_problem.xml Также возможно создание отчёта по списку проблем с отправкой его в файл/на печать, либо на почту. Шаблон отчёта находится в файле register_problems.xml. С помощью кнопки "OK+печать наряда" можно отправить конкретную проблему на печать (шаблон register_print_problem_order.xml).

Доступна история изменения проблемы. Для просмотра истории щелкните правой кнопкой мышки по проблеме в таблице и выберите пункт "История изменений".

Звонки **Проблемы** **Задачи** **Работы**

Фильтр + сортировка **Всего проблем: 1** 1 из 1 [50]

Код	Срочн...	Группа	Комментарий	Статус
161	0	Диспетчера	[ru.bitel.b...n.model.RegisterSubject@49f...]	принята

x00
История изменений

История

Код	Дата изменения	Кто изменил	Срочн...	Группа	Статус
161	17.09.2009 22:01:03	Сергеев Кирилл	0	Диспетчера	[ru.bitel.bgbilling.p]

Очистить историю **Закреть**

CRM

История

Детализация записи

Статус: принята Срочность: 0

Группа: Диспетчера

Исполнители

- Вася
- Петя

Все Сброс Инверт.

Комментарий Ход решения Участвовавшие группы Звонки

[ru.bitel.bgbilling.plugins.crm.common.model.RegisterSubject@49f8d077]
#XXXfdgdfgd

Категория: ---

Очистить историю << Пред Сергеев Кирилл, 17.09.2009 22:01:03 След >> Назад к списку Закреть

CRM

Журнал задач

Задача - конкретное мероприятие, которое нужно провести с конкретным клиентом. В зависимости от проблемы для задачи заранее определены группа решения и характер работы. Примером задач могут быть подключения и отключения клиентов.

В ходе решения задача проходит следующие состояния: открыта - принята - закрыта.

Как уже упоминалось, задача привязана к конкретному договору и может быть добавлена только в контексте договора на вкладке **CRM**. Кроме того, для удобства задача привязывается к адресу клиента и в ней выводятся ФИО и Телефон клиента. Для этого необходимо добавить в конфигурации плагина:

```
#
# , ( )
contract.fio.param.id=< 1>[, < 2>]
#() , () ( )
contract.phone.param.id=<text: ___>[, <phone: ___>]
```

Задача проходит статусы: открыта - принята - закрыта.

The screenshot shows a web interface for task management. On the left is a 'Фильтр' (Filter) panel with sections for 'Статус, период, обработанные' (Status, period, processed), 'Сортировка' (Sorting), and 'Дополнительно' (Additional). The 'Статус' section has checkboxes for 'открытые' (checked), 'принятые' (checked), and 'закрытые'. The 'Обработаны' section has radio buttons for 'да', 'нет', and 'все' (selected). The 'Сортировка' section has four dropdown menus, all set to '- нет -'. The 'Дополнительно' section has a dropdown for 'Открыть выбранные' and an 'Ok' button. On the right is a list of tasks with columns for 'Группы' (Groups), 'Исполнители' (Executors), and 'Типы' (Types). The 'Группы' column has checkboxes for 'Диспетчера' and 'Оптики'. Below the list are buttons for 'Все', 'Сброс', and 'Инверт.'. At the bottom of the interface is a toolbar with icons for print, save, email, and check, followed by a status bar showing 'Фильтр, сортировка, адрес, дополнительно', 'Всего задач: 8', and '1 из 1 [25]'. Below the toolbar is a table with columns: ФИО, Город, Квартал, Улица, Дом, По..., Кварт..., Телефон, Дата, Открыта, Сроки.

ФИО	Город	Квартал	Улица	Дом	По...	Кварт...	Телефон	Дата	Открыта	Сроки
	с. Ново...		ул. Радужная	14			+8-902-343-72-89; +...	28.09.2005	??? (7)	02.12.2005
								14.08.2008	Сергеев Кирилл	24.09.2008
								14.08.2008	Сергеев Кирилл	30.01.2009
							+8-902-343-72-89; +...	30.01.2009	Сергеев Кирилл	30.01.2009
							fghfh По пиду: Рабочий ...	20.02.2009	Сергеев Кирилл	
					0 (yy...		12345456 По пиду: +7-...	26.02.2009	Denis Pimenov	
							12345456 По пиду: +7-...	02.03.2009	Denis Pimenov	

В отличие от проблем задачи поддерживают блочную обработку - возможно принятие, открытие и закрытие сразу нескольких выбранных задач. Таким образом можно быстро раздавать наряды группам решения.

Также возможна процедура обработки задачи - при этом всем договорам будет прислано сообщение о завершении обработки задачи, которое может быть обработано скриптом BGBS. Например, возможно автоматическое включение абонплаты после подключения клиента.

Для того, чтобы печатать отчёты или отправлять их на почту возможно добавление одной или нескольких форм отчёта. Для этого добавьте в конфигурацию плагина строку, аналогичную приведённой ниже:

```
register.task.report.format=register_tasks.xml: ;register_tasks_1.xml: ;register_tasks_2.xml:
```

Здесь перечислены файлы с форматом отчёта и названия отчётов, выводимые в выпадающем списке в левом нижнем углу фильтра. Файл **register_tasks.xml** идёт с дистрибутивом и может быть использован в качестве примера.

Для получения краткой сводки о задаче необходимо нажать на строку правой кнопкой и выбрать пункт меню **Скопировать в буфер**. Шаблон краткой сводки - **register_task_short.xml**.

3	Юрия Гагарина	40	120	30-60-45	02.11.2005		открыта КФ4...	Рябь на всех канала:
7	Баязита Бикбая	27	103	34-55-81, 8-904-25-23654	10.11.2005		открыта КФ4...	Не показывает ТВ
7	Братьев Кадом...	7	35	77-58-78,8-927-33-40896	12.11.2005		открыта КФ4...	Рябь на всех канала:
	Карла Маркса	15/2	27	50-96-74, 8-903-10-20889	15.11.2005		открыта КФ1...	Срочное подключени
3	Максима Рыль...	19	114	8-905-35-02308	18.11.2005		открыта КФ4...	Иное
	8 Марта	32	51	8-904-25-97646	18.11.2005		открыта КФ1...	Замена кабеля
	Кирова	93	87	Скопировать в буфер	8.11.2005		открыта КФ1...	Иное
5	Кольцевая	22	36	42-47-67, 8-917-41-31407	28.11.2005		принята КФ1...	Замена кабеля
	Гафури	19/1	45	50-60-15	29.11.2005	22.01.2006	открыта КФ7...	Я не подключен не р
	Карла Маркса	54	53	29-11-22	29.11.2005	29.12.2004	открыта КФ1...	Я не подключен не р
	Муксинова	7	8	55-00-36	29.11.2005	24.02.2005	принята КФ1...	Я не подключен не р
	Достоевского	49	37	8-901-475-40-03	29.11.2005	22.01.2006	открыта КФ1...	Я не подключен не р
	Менделеева	122/1	44	56-76-29	29.11.2005	28.01.2006	открыта КФ2...	Я не подключен не р
	Достоевского	64	34	8-917-40-65-500	29.11.2005	09.06.2005	открыта КФ2...	Я не подключен не р
3	бульвар Тюльк...	7/3	72	33-78-46; 8-927-3199-626	29.11.2005	30.05.2005	открыта КФ2...	Я не подключен не р
2	Клавдии Абрам...	2	247	32-06-51, 8-927-30-55052	29.11.2005	19.07.2005	открыта КФ3...	Я не подключен не р
	Кирова	91	294	28-73-62	29.11.2005	13.12.2005	принята КФ3...	Истек срок подключе
	Кирова	97	294	28-73-62	29.11.2005	13.12.2005	принята КФ3...	Истек срок подключе

При печати наряда по конкретной задаче можно настроить отдельные шаблоны для них. Для этого следует добавить в конфигурацию плагина строки, аналогичные приведенным ниже:

```
# XML
order.print.add.contract.params=<true|false>
# xsl-
register.print.task.order.<id>.xslt=connecting.xsl
```

Где <id> - это id задачи в справочнике "CRM - типы задач".

Редактирование задачи выглядит следующим образом:

С помощью кнопки "OK+печать наряда" можно отправить конкретную задачу на печать (шаблон **register_print_task_order.xsl**).

Есть возможность выделить задачи определенным цветом в зависимости от количества часов, оставшихся до поставленных сроков выполнения. Для этого следует добавить в конфигурацию плагина строки, аналогичные приведенной ниже:

```
task.hour=24,48,96
```

Где цифры 24,48,96 - это часы оставшиеся до выполнения задач. По умолчанию они будут закрашиваться градациями серого цвета. Чем ближе задача к наименьшему интервалу, тем темнее она будет выделена.

Но есть так же возможность вместо градации серого задавать свои цвета. Для этого помимо добавления предыдущей строки следует добавить в конфигурацию плагина строки, аналогичные приведенным ниже:

```
task.color.24=00FFFF
task.color.48=80CC80
task.color.96=00FF00
```

Где 00FFFF - цвет, которым будут выделены задачи, у которых оставшееся часы до выполнения менее 24, 80CC80 - оставшееся время лежит в пределах 24-48 часов и 00FF00, если 48-96.

Фильтр, сортировка, адрес, дополнительно						
Телефон	Дата	Открыта	Срок	Статус	Догов...	Ком
17-477-34-98 [моб...	03.12.2010	kostya	22.12.2010 00:00	открыта	test_k...	ФИО
05-180-59-39 [Aira...	09.12.2010	shamil1	21.11.2011 00:00	открыта	boian2	ООО
	25.01.2011	Denis Pimenov	23.11.2011 00:00	открыта	ilshat_...	ком
	25.01.2011	Denis Pimenov	30.11.2011 00:00	открыта	ilshat_...	ком
17-477-34-98 [моб...	09.03.2011	kostya	18.11.2011 00:00	открыта	test_k...	ФИО
17-477-34-98 [моб...	09.03.2011	kostya	17.11.2011 00:00	открыта	test_k...	ФИО
17-477-34-98 [моб...	09.03.2011	kostya	16.11.2011 00:00	открыта	test_k...	ФИО
17-346-99-25; +2 (...)	10.11.2011	Ildar	15.11.2011 22:00	открыта	Ildar	

Журнал работ

Функционал работ в текущий момент не доведен до конца и поэтому тут не описан . Пока можете их не использовать.

Плагин Dispatch

Назначение плагина Dispatch

Плагин рассылок предназначен для управления пользовательскими рассылками. Данным плагином предоставляется единый интерфейс управления как новостными рассылками (т.е. рассылками, отправка сообщений которых происходит по мере поступления новой информации), так и периодическими рассылками о балансе, трафике, наработке и т.п. Существует возможность отправки сообщений различным транспортом: email, sms, а также можно реализовать собственный механизм отправки, используя встроенное API.

Настройка плагина Dispatch

Перед началом настройки плагина необходимо его установить при помощи утилиты `bg_installer` аналогично другим модулям и плагинам.

После установки необходимо настроить задачу **Задача отправки рассылок** на выполнение **каждую минуту!** Это важно, поскольку в противном случае возможны пропуски отправки по расписанию определённых рассылок.

В конфигурации плагина указываются следующие опции:

```
#           , none - ( )
#web.menuItem1= | none
#
#message.title.prefix=PROVIDERNAME -

#,
# email. email .
# : true - email- url
# , false -
dispatch.email.unsubscribe=
# ,
# . http(s)://<server>[:<port>]/<context>/dispatchexecutor
billing.url=
#URL, , .
dispatch.after.unsubscribe.url=
# , /message
file.storage.root.path=
```

Другие опции описаны в соответствующих разделах документации по плагину и относятся к конкретным вариантам использования плагина.

Общий обзор

Основным понятием плагина является **Рассылка**. Рассылка представляет из себя, по сути, контейнер для **шаблонов сообщений** (или просто: **сообщений**) с определённым набором характеристик: названием, расписанием отправки (или его отсутствия), способом отправки, типом контактов, которые могут осуществлять подписку на рассылку, набором условий отправки рассылки.

На данный момент системой поддерживаются следующие **типы отправки** сообщения:

- по E-mail;
- по SMS (через шлюзы smsc.ru, enterix.ru и smsaero);
- через скрипт.

Сообщение рассылки представляет из себя шаблон, который перед отправкой конечному подписчику преобразуется в конкретное сообщение путём подстановки данных вместо различных макросов, например, номера договора, фамилии клиента, баланса на счёте и т.п. К сообщению также можно прикреплять файлы (однако их отправка может не поддерживаться какими-либо отправителями). Также имеется возможность задать собственное тело сообщения с помощью соответствующих макросов.

Рассылки можно функционально разделить на новостные и периодические рассылки:

- **Новостные рассылки** - это рассылки, отправка сообщений которых происходит на каждой итерации срабатывания задачи отправки рассылки. При этом для отправки выбираются ещё неотправленные сообщения, время отправки которых уже пришло. После отработки задачи они помечаются как "**отправленные**".
- **Периодические рассылки** - это рассылки, у которых есть фиксированное расписание отправки, задаваемое аналогично расписанию выполнения задач планировщика (об этом ниже). Отправка сообщений таких рассылки происходит по указанному расписанию, причем при каждой отправке конечному пользователю отправляются **все** сообщения данной рассылки! В периодических рассылках, как правило, заводится одно сообщение.

Рассылки могут быть глобальными или персональными:

- **Глобальные рассылки** - по умолчанию для новых рассылки. Сообщения таких рассылки приходят всем подписчикам в одном и том же виде, в одно и то же время. Время отправки устанавливается для всей рассылки в целом.
- **Персональные рассылки** - это рассылки, для которых подписчики могут указать дни недели, в которые они хотят получать сообщения. Также подписчики персональных рассылки могут указывать дополнительные условия, при которых нужно присылать сообщения, например, диапазон баланса для рассылки баланса или логин модуля, для получения детализации по нему.

Каждая рассылка характеризуется также **типом контактов**, которыми возможно подписаться на эту рассылку. Типы контактов заводятся администратором биллинга вручную, при этом указывается их описание и регулярное выражение для проверки корректности вводимых данных.

Каждый пользователь может заводить неограниченное количество **контактов** определённого типа. Каждый пользователь может создавать неограниченное количество **подписок** на рассылки. Подписка на рассылки характеризуется самой рассылкой, на которую была осуществлена подписка, списком контактов, а также (опционально) набором индивидуальных условий отправки сообщений рассылки.

Основное окно плагина расположено в меню **Плагины => Рассылки**.

Название	Метод отправки	Расписание отправки	Активна	Персональная	Дата создания	Дата последней модификации	Пользователь
albert_personal	По e-mail	каждую минуту	<input type="checkbox"/>	<input checked="" type="checkbox"/>		18.05.2015 17:07:43	Гареев Артур
kostya_test	В консоль	в 0:0	<input type="checkbox"/>	<input checked="" type="checkbox"/>		06.10.2016 12:33:51	Борис
Борис_тест	По e-mail	в 0:0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	24.08.2016 15:12:42	12.12.2016 16:18:28	Гареев Артур
Рассылка Dialup	По e-mail	в 0:0	<input type="checkbox"/>	<input checked="" type="checkbox"/>		27.09.2017 17:32:23	admin
Рассылка баланса	По e-mail	каждую минуту	<input type="checkbox"/>	<input checked="" type="checkbox"/>			<Пользователь неизвестен>
Тест email с вложен...	По e-mail	каждую минуту	<input type="checkbox"/>	<input checked="" type="checkbox"/>		27.09.2017 17:32:17	admin
artur_test_sms	По SMS через сервис Enterix.ru	каждую минуту	<input type="checkbox"/>	<input type="checkbox"/>		20.02.2016 14:19:55	admin
inet	По e-mail		<input type="checkbox"/>	<input type="checkbox"/>		27.09.2017 17:32:14	admin
Рассылка для группо...	По e-mail		<input type="checkbox"/>	<input type="checkbox"/>		27.09.2017 17:32:11	admin
Рассылка по занесе...	По e-mail		<input type="checkbox"/>	<input type="checkbox"/>		27.09.2017 17:32:09	admin
Рассылка по приходу...	По e-mail		<input type="checkbox"/>	<input type="checkbox"/>		27.09.2017 17:32:06	admin
Рассылка по салдо ...	По e-mail	каждую минуту	<input type="checkbox"/>	<input type="checkbox"/>	19.05.2015 12:14:18	19.05.2015 15:46:55	Гареев Артур
Рассылка с условием...	По e-mail	каждую минуту	<input type="checkbox"/>	<input type="checkbox"/>		24.11.2016 12:13:55	Гареев Артур
Тест периодич. расс...	По e-mail		<input type="checkbox"/>	<input type="checkbox"/>		29.07.2015 17:53:06	Гареев Артур
Тестирование услов...	По e-mail	каждую минуту	<input type="checkbox"/>	<input type="checkbox"/>	29.05.2015 15:44:02	20.02.2016 11:33:38	admin

На нем расположены три вкладки: **Менеджер рассылок**, **Менеджер контактов** и **Методы отправки**. Подробнее смотрите в следующих разделах документации.

Типы контактов

Настройка типов контактов происходит в **Менеджере контактов**.

Код	Название
3	E-mail
3080036	email-2
3080034	Для уведомлений об авариях (email)
3080035	Для уведомлений по балансу (email)
1	Любой
5	Телефон

Редактор

Тип контакта:

Шаблон:

Описание (подсказка):

Параметр по умолчанию:

OK Отмена

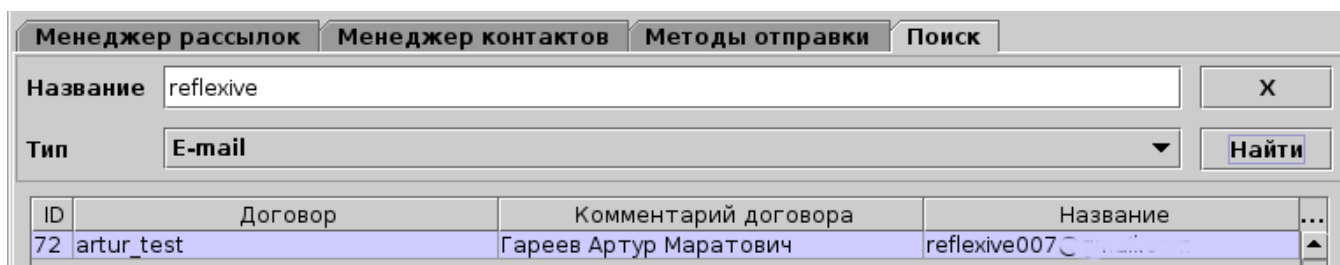
Для создания нового типа контакта необходимо указать его название, описание (которое будет видно пользователям) и регулярное выражение (шаблон), которым будет проверяться валидность введённых данных. Существует ряд предустановленных шаблонов типов контактов, это:

- E-mail, которому соответствует шаблон `^[A-z0-9._%+ -]+@[A-z0-9.-]+\.[A-z]{2,4}$`
- Пользовательский шаблон, необходимо указать его вручную, пользовательское регулярное выражение должно начинаться с `^` и оканчиваться `$`.

Для параметра можно задать параметра договора, из которого будет браться значение если контакт в договоре не будет заполнен.

Поиск по контактам

Поиск можно производить по названию и типу контакта.



ID	Договор	Комментарий договора	Название	...
72	artur_test	Гареев Артур Маратович	reflexive007	▲

При двойному клику на строке в таблице результатов открывается договор с найденным контактом.

Методы отправки

Метод отправки - это серверная логика реальной фактической отправки конечного сообщения подписчику. Настройка методов отправки происходит во вкладке **Методы отправки**. На данный момент существуют следующие встроенные механизмы отправки сообщения и возможные опции, которые задаются в настройках плагина:

- Email - отправка сообщения осуществляется на почтовый ящик подписчика, при этом отправляются также все приложенные файлы. Существует возможность отписываться от рассылки.

Настройка отписки от рассылки

```
#,      email-
dispatch.email.unsubscribe=true
# ,      .
#      nginx      .
billing.url=http://localhost:8080/bgbilling/
# ,      ,
dispatch.unsubscribe.text=
#url,
dispatch.after.unsubscribe.url=http://www.bgbilling.ru
```

- SMS (smsc.ru) - отправка сообщения осуществляется на мобильный телефон подписчика, посредством смс сообщения.

```
sender.sms.smsc.login=
sender.sms.smsc.sender=
sender.sms.smsc.password=
#      .      :
# 0 -      .
# 1 ( ) -      "translit".
# 2 -      "mpaHc/Ium"
sender.sms.smsc.translit=
```

- SMS (enterix.ru) - отправка сообщения осуществляется на мобильный телефон подписчика, посредством смс сообщения.

```
sender.sms.enterix.login=
sender.sms.enterix.sign=
sender.sms.enterix.password=
```

- SMS (smaero.ru) - отправка сообщения осуществляется на мобильный телефон подписчика, посредством смс сообщения.

```
sender.sms.smaero.login=
sender.sms.smaero.password=D06071FC526EA03696 - md5.( -> Digest)
sender.sms.smaero.sign=SiGnAtUrE - ( / smaero.ru ).
```




Помимо встроенных способов отправки сообщения существует возможность реализации пользовательских способов отправки. Для реализации этой возможности необходимо воспользоваться механизмом [динамически загружаемых Java-классов](#). Необходимо создать динамический класс, реализующий интерфейс `ru.bitel.bgbilling.plugins.dispatch.server.sender.Sender`, а затем создать метод отправки нажатием на кнопку **Добавить** в панели инструментов, указав этот класс в выпадающем списке.

Менеджер рассылок Менеджер контактов **Методы отправки** Поиск

Код	Название
-1	По e-mail
-2	По SMS через сервис Enterix.ru
-3	По SMS через сервис Smsc.ru
-4	По SMS через сервис SmsAero.ru
1	В консоль

Редактор

Имя метода отправки:

Динамический класс:   

OK Отмена

✕ Рассылки ✕ Управление динамическим кодом

Создание и редактирование рассылки

Создание новой рассылки осуществляется через кнопку **Добавить** в панели инструментов во вкладке **Менеджер рассылок**. При этом откроется редактор рассылки.

Менеджер рассылок Менеджер контактов Менеджер...

Редактор

Название:
Новая рассылка

Тип рассылки:
По e-mail

Тип контактов:
-- Выберите тип контактов --

без контакта

только один контакт

активна

персональная

не помечать сообщения отправленными

Отправка по расписанию

Минуты *
Часы *
Дни *
Дни недели *
Месяцы *

Показать сообщения =>

Показать условия отправки =>

В редакторе рассылки необходимо указать следующие параметры:

- **Название** - название рассылки, его будут видеть и пользователи, и операторы биллинга;
- **Тип рассылки** - выпадающий список выбора способа отправки сообщений;
- **Тип контактов** для подписки на рассылку;
- **Без контакта** - для рассылок, которые используют другие каналы связи с абонентом и не требуют указания какого-либо контакта абонента. Например, в самописном методе отправки можно реализовать вызов определенного url для уведомления абонента, либо вызов какой-либо команды в терминале.
- **Активна** - если не отмечено, рассылка отправляться не будет;
- **Персональная** - позволяет пользователям через личный кабинет самим устанавливать дни недели, в которые они хотят получать рассылку. Время отправки сообщений (часы и минуты), устанавливаются для самой рассылки. При создании подписки на персональные рассылки пользователи также могут указывать условия, при которых им нужно отправлять сообщения рассылки (напр., условия по балансу, для модулей Inet и VoiceIP - выбирать логин).

- **Не помечать сообщения отправленными** - позволяет не отмечать сообщения отправленными для глобальных неперiodических рассылок. Это можно использовать, к примеру, когда нужно отправлять одно и то же сообщение, в теле которого присутствуют макросы, а само сообщение должно отправляться по событию (например, сообщение о пополнении баланса договора или о занесении расхода). Данный параметр автоматически устанавливается для персональных и периодических рассылок.
- **Только один контакт** - если отмечено, будет позволено выбрать только один контакт;
- **Отправка по расписанию** - это флаг, обозначающий наличие или отсутствие расписания отправки у рассылки;
- **Минуты, Часы, Дни, Дни недели и Месяцы**, когда происходит отправка рассылки, данные в этих полях заполняются аналогично расписанию задач в планировщике заданий. Для пользовательских рассылок доступны для изменения только поля Часы и Минуты. Дни недели для отправки в этом случае определяются настройками подписки для конкретного договора.

По кнопке **Показать условия отправки** откроется окно условий отправки, которые накладываются на данную рассылку глобально для всех подписок. Подробнее об условиях отправки см. [здесь](#).

По кнопке **Показать сообщения** откроется окно редактирования сообщений, подробнее о редакторе сообщений см. [далее](#). При создании новой рассылки сообщения не будут созданы.

После заполнения необходимых полей нажмите на кнопку **ОК**.

Подписка на рассылки

Настройка подписок договора осуществляется на вкладке **Рассылки** в договоре.

Контакты

Для заведения контактов определённого типа необходимо перейти на вкладку **Контакты**. Далее нажать на кнопку **Добавить** в панели инструментов. После этого необходимо выбрать тип заводимого контакта и указать его значение.

Тип контакта	Значение
Мыло	pechenie@bitel.ru
Мыло	preved@kag.dila
Любой	Woot

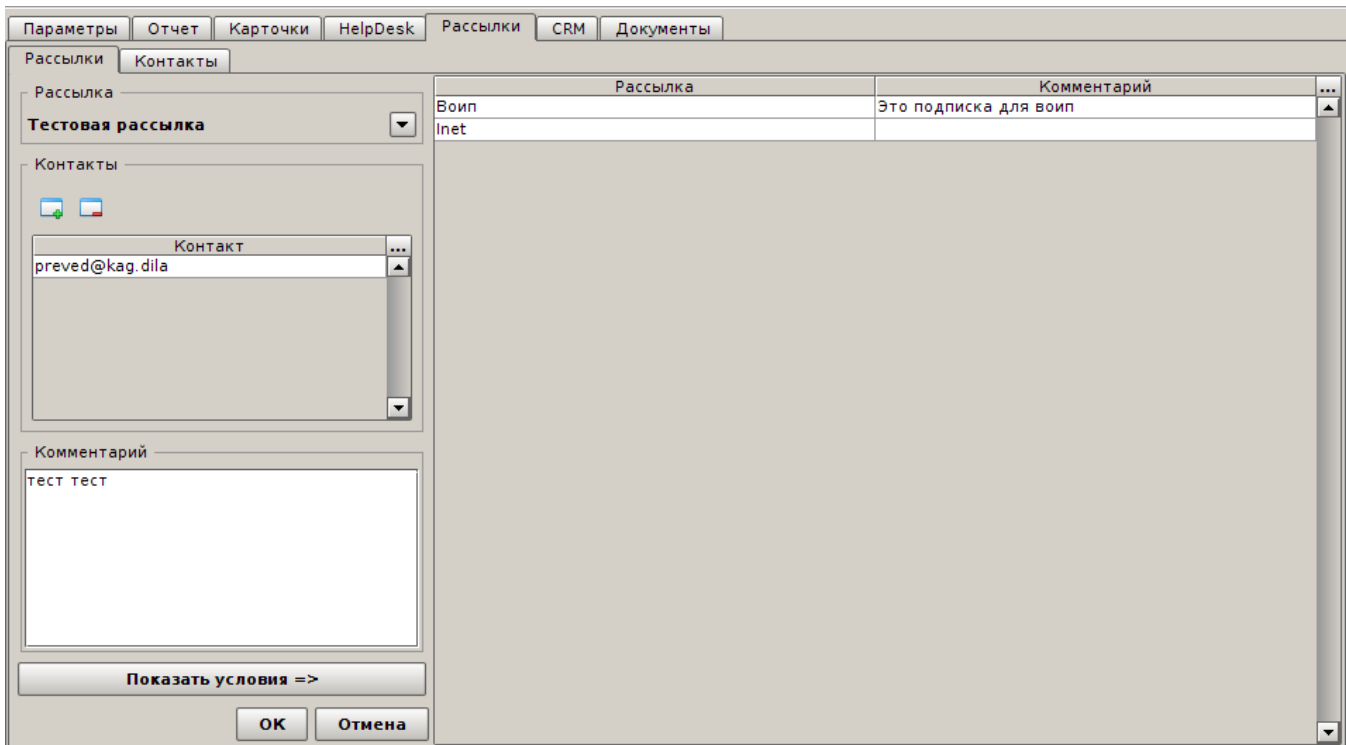
Тип контакта: **Мыло**

Значение:

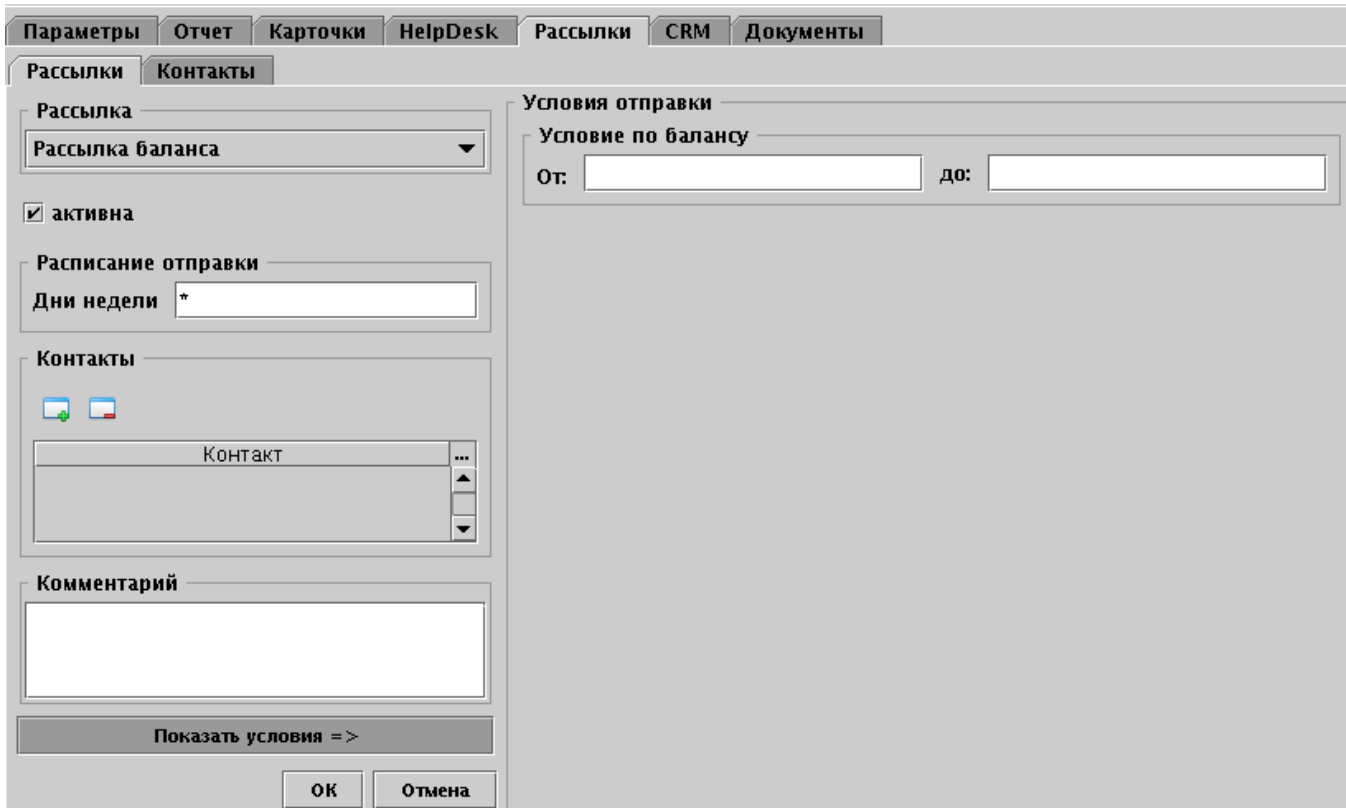
OK Отмена

Рассылки

Для подписки на рассылки необходимо перейти на вкладку **Рассылки**. Здесь можно увидеть таблицу осуществлённых подписок. Для создания новой подписки необходимо нажать на кнопку **Добавить** в панели инструментов.

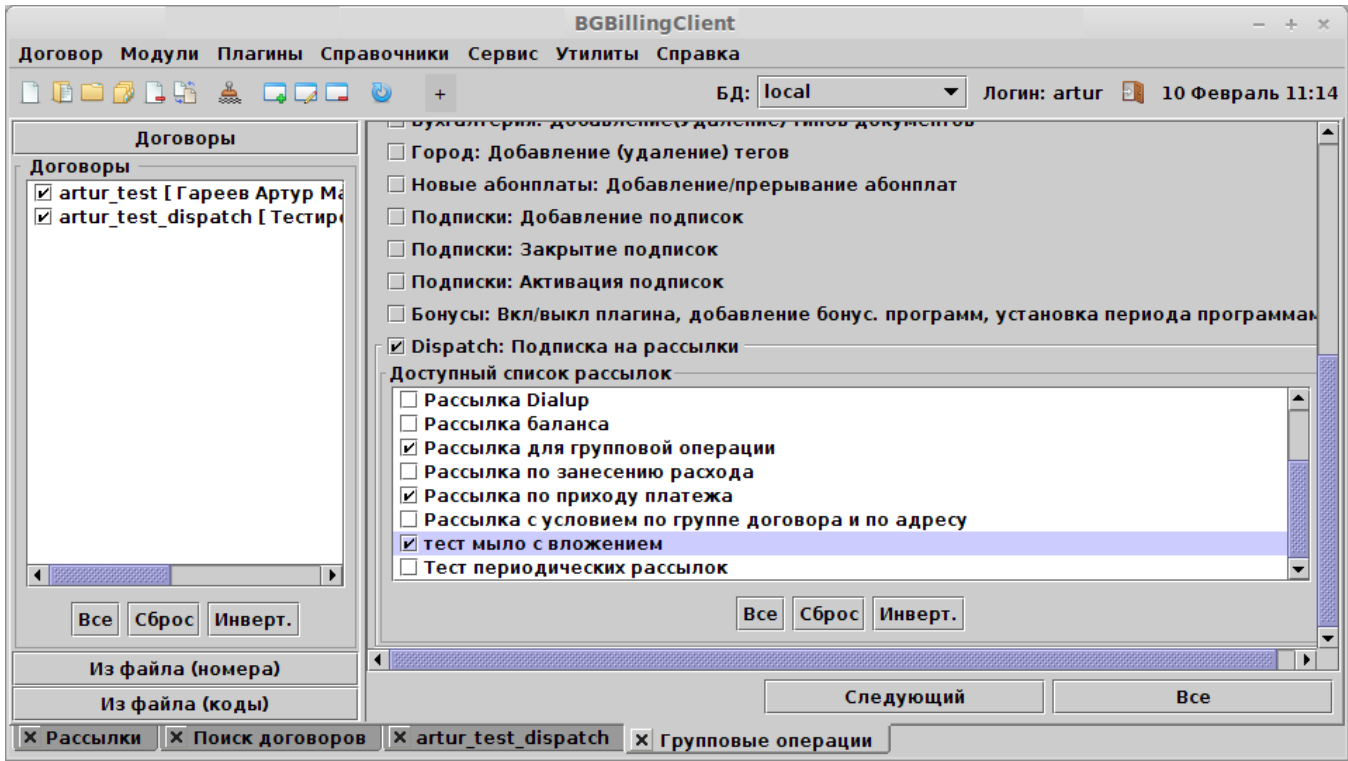


В открывшейся слева панели необходимо выбрать вид рассылки в выпадающем списке сверху, затем указать необходимые контакты из списка контактов и заполнить комментарий к рассылке. Для персональных рассылок необходимо проверить наличие условий отправки и, по необходимости, заполнить их данными. Для персональных рассылок можно также изменить дни недели, в которые будет приходить рассылка.



Автоматизация подписки на рассылки с помощью групповой операции

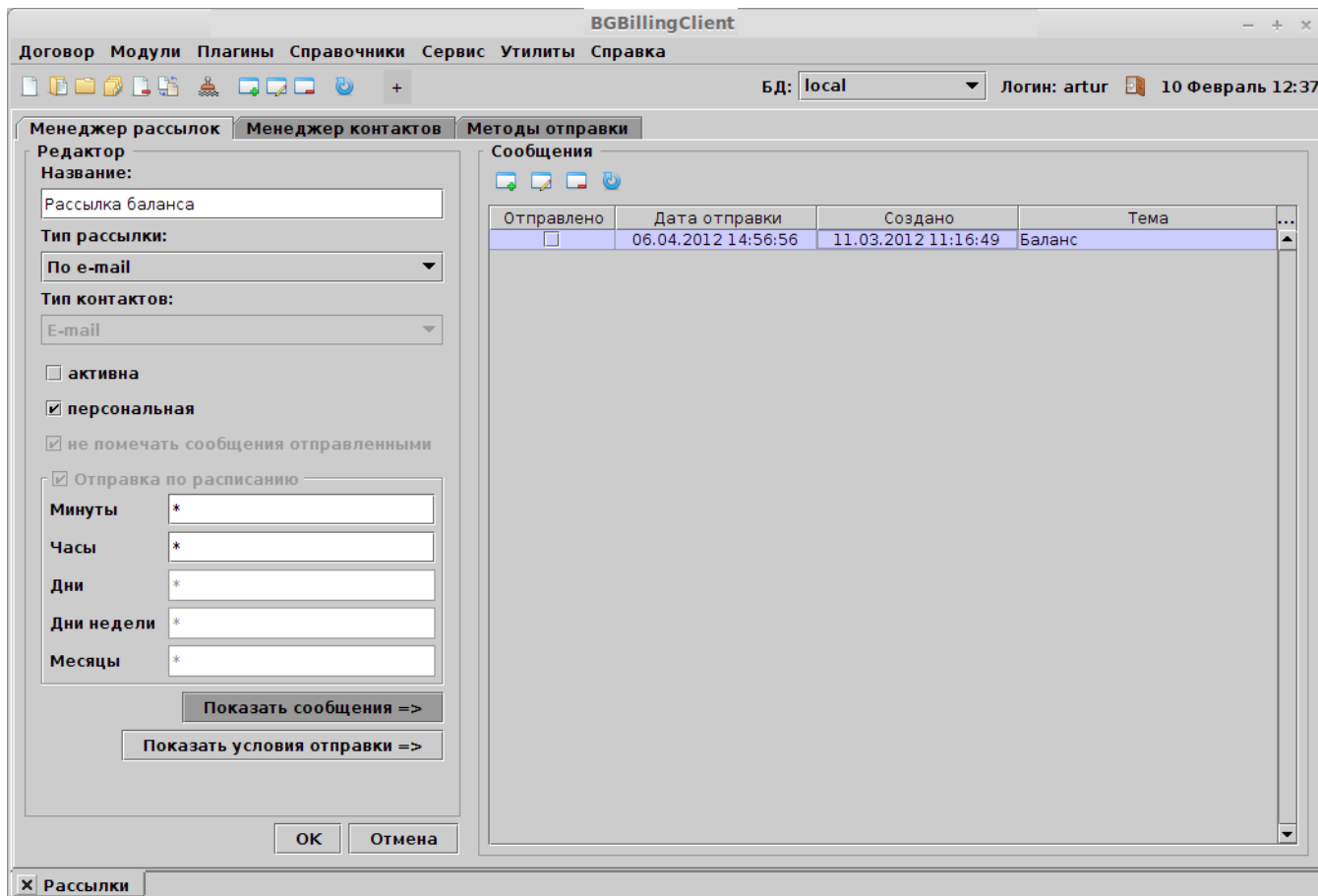
Для автоматизации создания подписок на одни и те же рассылки для большого числа договоров, отфильтрованных по определенным условиям, есть возможность использовать групповую операцию. Для этого необходимо предварительно отфильтровать список договоров с помощью поиска, для которых необходимо добавить подписку, после этого перейти в меню **Сервис->Администрирование->Групповые операции**, активировать групповую операцию плагина **Dispatch: Подписка на рассылки**, нажать кнопку **Договоры** в левой верхней части окна и выбрать из появившегося списка необходимые договоры.



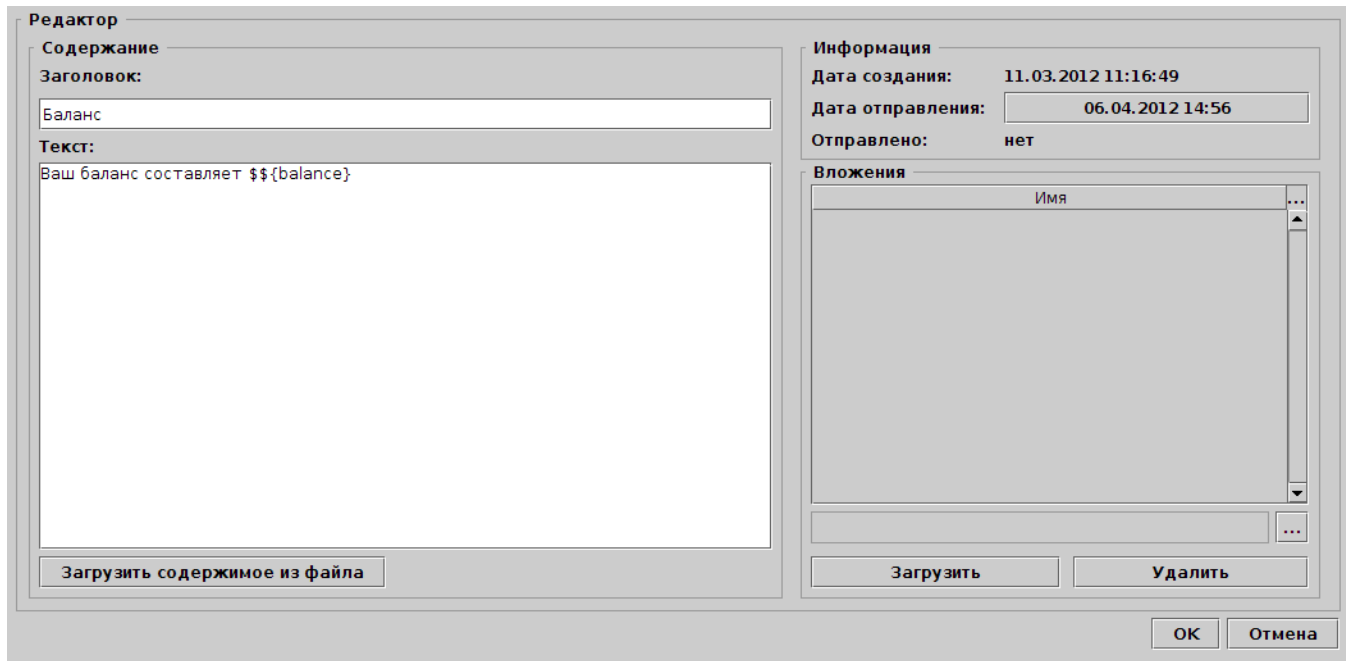
Для приведения групповой операции в действие необходимо нажать кнопку **Все** или **Следующий** для выполнения всех выбранных групповых операций в автоматическом или пошаговом режиме соответственно.

Создание и редактирование сообщений

Для просмотра сообщений конкретной рассылки откройте вкладку **Плагины => Рассылки => Менеджер рассылок**, нажмите кнопку **Редактировать** в панели инструментов и далее нажмите кнопку **Показать сообщения**.



На панели справа можно увидеть список сообщений. Для редактирования/добавления/удаления сообщений необходимо воспользоваться соответствующими кнопками в панели инструментов редактора сообщений. При редактировании или создании сообщения откроется редактор сообщений.



Для создания сообщения необходимо указать заголовок сообщения и тело сообщения. Для загрузки в тело сообщения данных из файла, необходимо нажать кнопку **Загрузить содержимое из файла** и выбрать файл.

Кодировка содержимого в файле должна быть **UTF-8**.

Также есть возможность прикрепления файлов к сообщению. Для этого в окне справа нажмите кнопку ... и выберите файл. После этого нажмите кнопку **Загрузить**.

Для сообщения также есть возможность указания времени отправки сообщения для отложенной отправки сообщения. Будут отправлены только те сообщения, время которых уже пришло.

В теле сообщения могут присутствовать макросы, вместо которых при отправке этого сообщения конечному подписчику будут подставлены соответствующие данные этого подписчика. На данный момент системой поддерживаются следующие макроподстановки:

- **\$\$ {title}** - номер договора;
- **\$\$ {comment}** - комментарий договора;
- **\$\$ {parameter:<PID>}** - параметр договора, здесь **<PID>** - это код параметра договора, например **\$\$ {parameter:5}**;
- **\$\$ {balance}** - баланс договора на момент отправки;
- **\$\$ {saldo}** - сальдо договора на момент отправки;
- **\$\$ {module:<MID>:<FUNCTION>}** - вставка модульных данных, здесь **<MID>** - код модуля, **<FUNCTION>** - название функции запрашиваемых данных, например, **\$\$ {module:10:detail}**;
- **\$\$ {class:<FULLY.QUALIFIED.CLASS.NAME>}** - глобальная макроподстановка, позволяющая полностью изменять тело сообщения с помощью пользовательских динамических классов рассылок, здесь **<FULLY.QUALIFIED.CLASS.NAME>** - полное имя пользовательского класса. Пользовательский класс должен наследовать абстрактный класс `ru.bitel.bgbilling.plugins.dispatch.server.bean.message.CustomDispatchMessage` и переопределять метод `String getMessageBody()`. Возвращаемое значение и будет телом сообщения. В классе доступен объект `java.sql.Connection` для соединения с БД, а также код договора, для которого предназначено данное сообщение.

Также есть возможность применения макросов в заголовке сообщения.

Рассылки модулей

Inet

В плагине рассылок есть возможность вставки Отчёта по сессиям пользователя модуля **Inet** в рассылку. Для этого используется имя функции **session**. При этом в сообщении будет прикреплена детализация в формате HTML, а на месте макроса будет фраза "Детализация по сессиям во вложениях".

VoiceIp

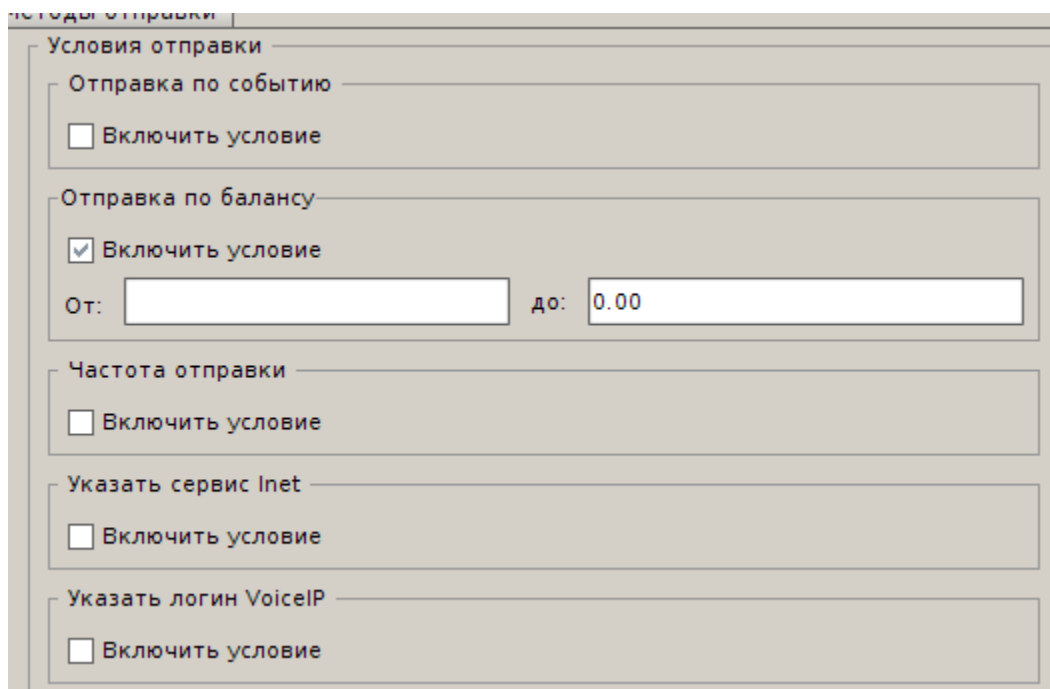
В плагине рассылок есть возможность вставки Отчёта по направлениям модуля **VoiceIp** в рассылку. Для этого используется имя функции **direct**. При этом в сообщении будет прикреплена детализация в формате HTML, а на месте макроса будет фраза "Детализация по направлениям во вложениях".

Phone

В плагине рассылок есть возможность вставки Отчёта по сессиям модуля **DialUp** в рассылку. Для этого используется имя функции **session**. При этом в сообщении будет прикреплена детализация в формате HTML, а на месте макроса будет фраза "Детализация по сессиям во вложениях".

Условия отправки

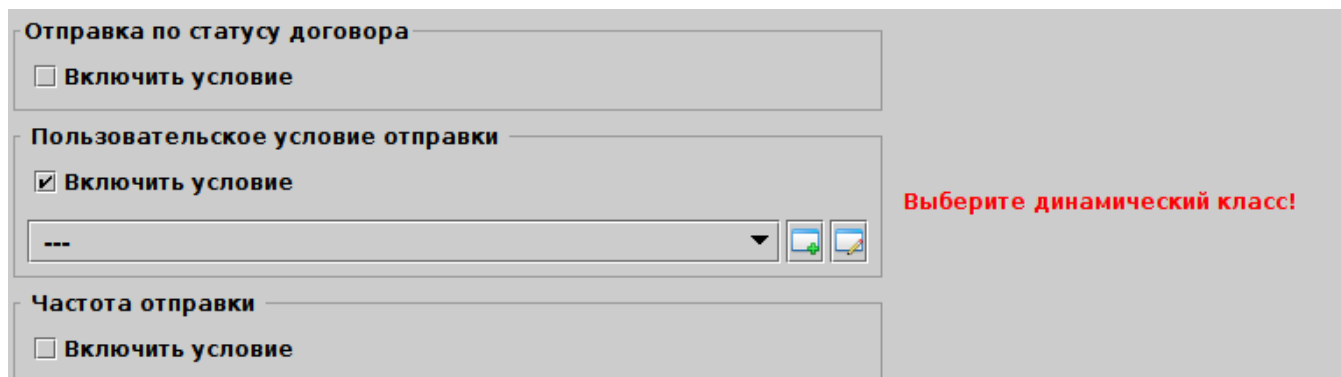
Условия отправки представляют из себя набор ограничений, условий или настроек, которыми обладает рассылка.



Только выполнение требований этих условий в каждый момент отправки конечному подписчику позволяет выполнить отpravку. Каждое условие отправки состоит из трёх составных частей:

- интерфейса настройки условия отправки для всей рассылки глобально (в простейшем случае может иметь только флаг включено\отключено);
- интерфейса настройки условия отправки для конкретной подписки на рассылку (в простейшем случае может отсутствовать);
- логики проверки выполнения условия на сервере.

При сохранении рассылки происходит валидации условий отправки, чтобы не было противоречий. В случае некорректности условия плагин выдаст соответствующее предупреждение. Например, если в пользовательском условии отправки не указать динамический класс, то предупреждение будет выглядеть следующим образом.



Ниже перечислены все условия отправки, поддерживаемые системой.

Отправка по событию

Данное условие представляет собой универсальный способ отправки рассылки. Настройка данного условия в меню настройки рассылки заключается только во включении\отключении условия. Настройка данного условия в меню настройки подписки отсутствует.

Для успешной проверки на выполнение данного условия перед отправкой конечному подписчику в настройках конкретной подписки на рассылку должен присутствовать флаг "событие произошло". Предполагается, что данный флаг устанавливается из какого-либо скрипта поведения, среагировавшего на то или иное событие. Например, если у клиента изменился статус на "Отключён", то соответствующий скрипт, обработавший данное событие, устанавливает соответствующий флаг в настройках его (клиента) подписки на рассылку, информирующую об отключении абонента за долги. Тогда задача отправки рассылок на очередной итерации отправки при проверке условия данной подписки установит, что данное условие выполнено и отправка уведомления об отключении произойдет.

Пример части скрипта, обрабатывающего событие **Статус изменён**, который устанавливает флаг в настройках подписки:

```
// " "
int dispatchId = X;
// ,
int contractId = Y;
//
List<Subscription> subscriptions = service.getSubscriptions( dispatchId, contractId );
//      1
for( Subscription subscription : subscriptions )
{
    subscription.getPreferences().set( DispatchEventCondition.EVENT_OCCURED_FLAG, "1" );
    service.updateSubscriptions( subscription );
}
```

с версии 6.2

```
// " "
int dispatchId = X;

ServerContext context = ServerContext.get();
DispatchService service = context.getService( DispatchService.class, 0 );
service.setEventForSubscription( dispatchId, event.getContractId() );
```

Отправка по балансу

Данное условие выполняется только тогда, когда баланс договора на момент отправки находится в указанных в условии пределах. Настройка данного условия в меню настройки рассылки заключается в установке диапазона баланса, для которого происходит отправка. Настройка данного условия в меню настройки подписки отсутствует.

Отправка по балансу

Включить условие

От: до:

Ограничение частоты отправки

Данное условие выполняется только тогда, когда последняя отправка той же рассылки произошла не ранее, чем через указанное количество дней в настройках. Настройка данного условия в меню настройки рассылки заключается в установке количества дней между отправкой рассылки. Настройка данного условия в меню настройки подписки отсутствует.

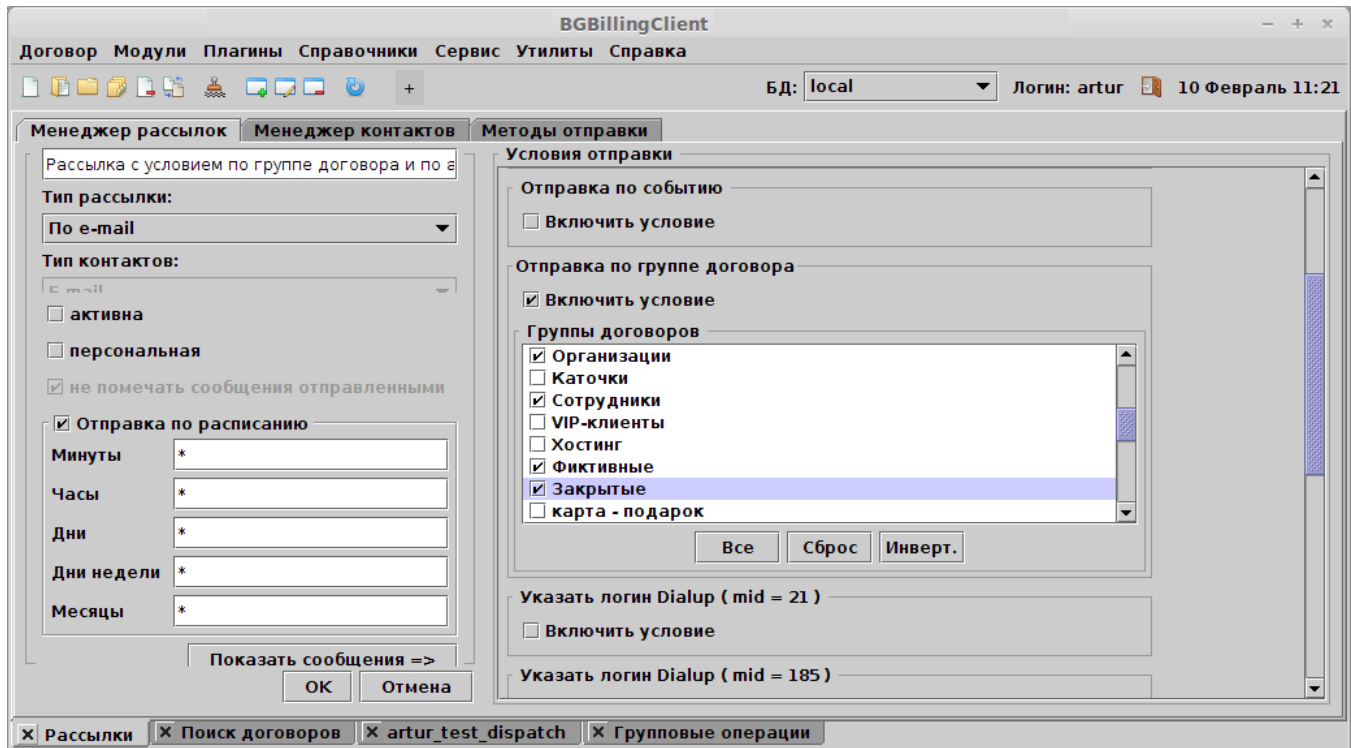
Частота отправки

Включить условие

Отправлять не чаще, чем раз в дней

Ограничение по группе договора

Условие по группе договора выполняется только тогда, когда подписчик состоит хотя бы в одной из выбранных групп договоров. Настройка условия в настройках подписки отсутствует.



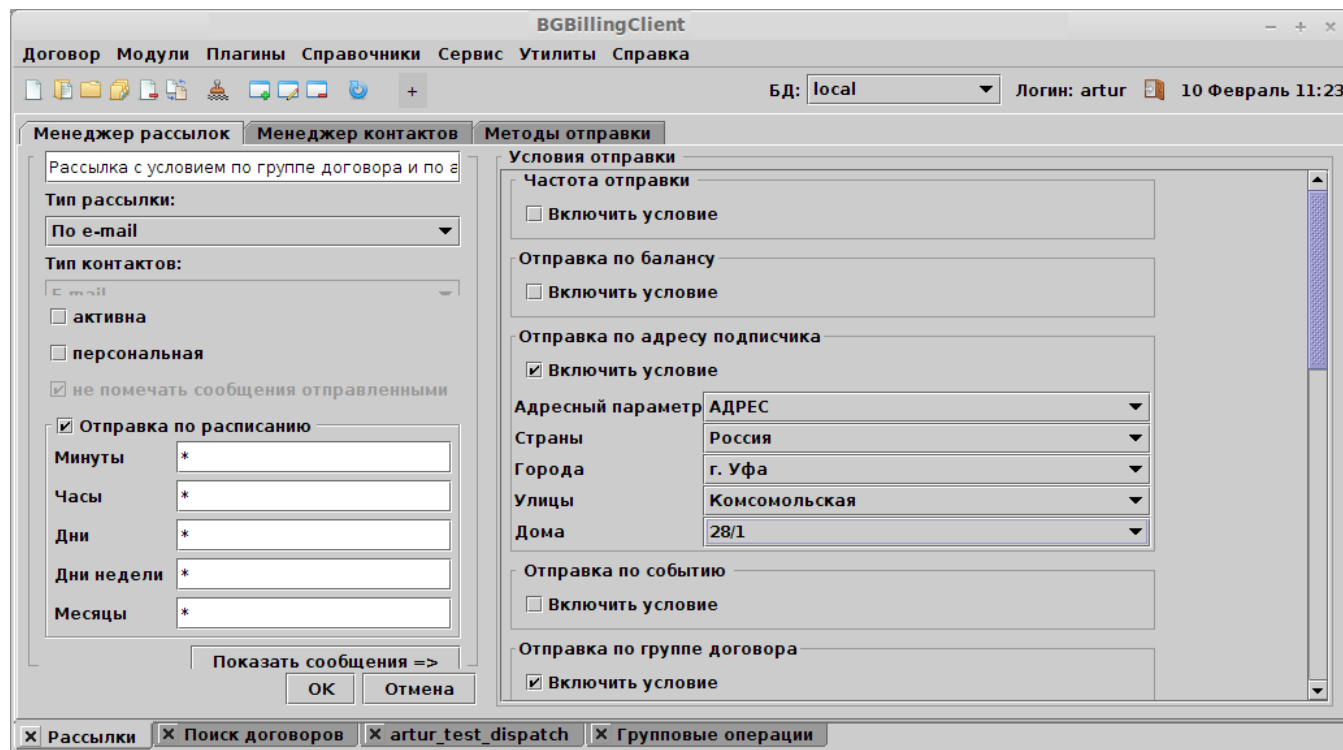
Ограничение по адресу подписчика

Данное условие позволяет ограничить рассылку с учетом адреса подписчика. Корректная работоспособность данного условия гарантируется только в том случае, если адреса абонентов представляют собой параметр договора с типом Адрес. В настройках условия сперва необходимо выбрать тип адресного параметра, по которому будет включено ограничение. Далее из справочника адресов биллинга подгружается список стран. Выбрав страну, подгружается список городов. Для выбранного города загружается список улиц, а для выбранной улицы подгружается список домов на этой улице.

Принцип работы ограничения:

1. Если вы выбираете только город, то рассылка будет отправляться всем абонентам из этого города.
2. Если вы выбираете город и улицу, то рассылка будет отправляться всем абонентам, проживающим на выбранной улице выбранного города.
3. Если вы выбираете город, улицу и дом, то рассылка будет отправляться всем абонентам по указанному адресу.

Настройка условия в настройках подписки отсутствует.



Указание сервиса модуля Inet

Данное условие актуально только при наличии установленного модуля Inet. **Активация данного условия обязательна для рассылок детализаций модуля Inet.**

Данное условие выполняется только тогда, когда в настройках подписки будет указан какой-либо сервис модуля Inet. Настройка данного условия в меню настройки рассылки заключается во включении\отключении данного условия. Настройка данного условия в настройках подписки заключается в установке необходимого сервиса модуля Inet из списка.

Указание логина модуля VoiceIP

Данное условие актуально только при наличии установленного модуля VoiceIP. **Активация данного условия обязательна для рассылок детализаций модуля VoiceIP.**

Данное условие выполняется только тогда, когда в настройках подписки будет указан какой-либо логин модуля VoiceIP. Настройка данного условия в меню настройки рассылки заключается во включении\отключении данного условия. Настройка данного условия в настройках подписки заключается в установке необходимого логина модуля VoiceIP из списка.

Указание логина модуля DialUp

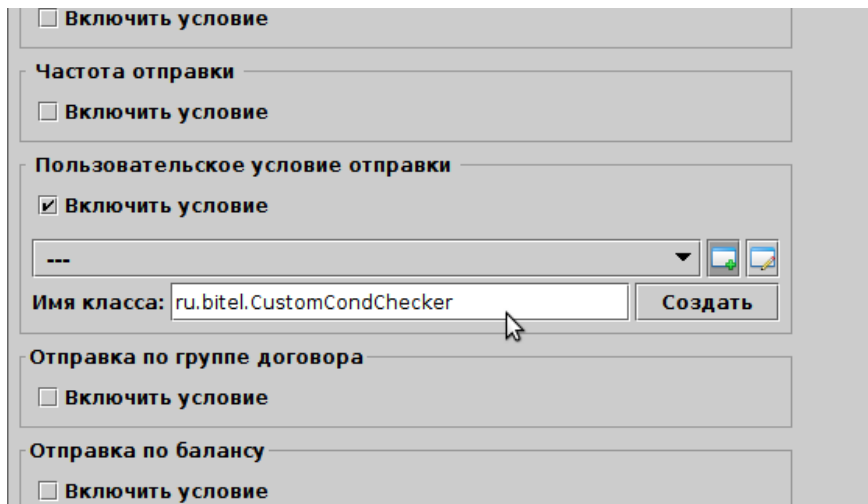
Данное условие актуально только при наличии установленного модуля DialUp. **Активация данного условия обязательна для рассылок детализаций модуля DialUp.**

Данное условие выполняется только тогда, когда в настройках подписки будет указан какой-либо логин модуля DialUp. Настройка данного условия в меню настройки рассылки заключается во включении\отключении данного условия. Настройка данного условия в настройках подписки заключается в установке необходимого логина модуля DialUp из списка.

Пользовательское условие отправки

В случае, если стандартных условий отправки недостаточно и требуется большая гибкость при настройке ограничений, то существует возможность реализовать собственные проверки путем написания динамического скрипта.

В настройках рассылки условие выглядит следующим образом:



Включить условие

Частота отправки

Включить условие

Пользовательское условие отправки

Включить условие

Имя класса:

Отправка по группе договора

Включить условие

Отправка по балансу

Включить условие

В динамическом коде необходимо создать класс, который реализует интерфейс `ru.bitel.bgbilling.plugins.dispatch.server.bean.condition.CustomDyncodeChecker`. Проверку осуществляет метод `check`, который имеет следующую сигнатуру:

Сигнатура метода `check` пользовательского условия отправки

```
/**
 *
 * @param con
 *
 * @param dispatch
 *
 * @param subscription
 *
 * @return
 *         true - , false - .
 * @throws BGException
 */
boolean check( Connection con, Dispatch dispatch, Subscription subscription )
    throws BGException;
```

В результате работы метод должен вернуть либо **true** - можно отправлять, либо **false** - отправка запрещена по данному условию.

Web-интерфейс плагина Dispatch

Осуществлять подписку на рассылки может как оператор биллинга, так и непосредственно сам пользователь. Плагин добавляет в меню пользователя пункт **Подписка на рассылки**. Данное меню содержит две "вкладки": **Мои контакты** и **Мои подписки**. По умолчанию открывается вкладка с контактами.

Мои контакты

На вкладке с контактами отображается таблица текущих контактов.

Мои контакты

Мои контакты Мои рассылки

Тип	Значение	X
Мыло	pechenie@bitel.ru	Удалить
Мыло	preved@kag.dila	Удалить
Любой	Woot	Удалить

Новый

Для удаления ненужных контактов используется кнопка **Удалить**. Для добавления нового контакта необходимо нажать кнопку **Новый**.

Новый контакт

Выберите тип:

Введите e-mail, например, example@example.org

Введите значение:

Создать Отмена

Далее необходимо указать тип добавляемого контакта и ввести его значение.

Мои рассылки

Здесь отображается таблица с подписками на рассылки. Для отказа от рассылки необходимо нажать на кнопку **Удалить**.

Мои рассылки

Мои контакты Мои рассылки

Подписка	Контакты	Расписание	Комментарий	V	X
Воип	pechenie@bitel.ru		Это подписка для воип	Редактировать	Удалить
Inet	pechenie@bitel.ru	каждую минуту		Редактировать	Удалить

Inet

Для редактирования рассылки необходимо нажать кнопку **Редактировать**. Для добавления новой рассылки необходимо выбрать её из выпадающего списка снизу и нажать кнопку **Новая подписка**. В обоих случаях откроется редактор подписок.

Подписка на рассылку

Мои контакты

Мои рассылки

Редактирование подписки "Воип"

Выберите контакты:

- pechenie@bitel.ru
- preved@kag.dila

Укажите логин VoiceIP:

118[adasdsadasdasdasdasdads] ▾

Ваш комментарий:

Это подписка для воип

Подписаться

Для осуществления подписки необходимо выбрать контакты и указать условия отправки, если таковые имеются. Также при подписке можно добавить к ней комментарий.

Конвертирование рассылок с версии 5.1

На данный момент реализована возможность переноса данных о рассылках баланса с версии 5.1. Для этого нужно запустить следующую команду в директории сервера:

```
 ${JAVA_HOME}/bin/java -Xmx256m -cp ./lib/*:lib/jms/*:classes:./lib/ext/bgcommon-boot.jar ru.bitel.common.bootstrap.Boot ru.bitel.bgbilling.plugins.dispatch.server.utils.Converter arg1 arg2
```

Конвертер принимает на входе 2 параметра:

- **arg1** - имя базы данных, в которой находятся таблицы с данными по старым рассылкам. Если база данных текущая, нужно указать пустую строку (в кавычках);
- **arg2** - id контакта типа e-mail. Тип контакта можно добавить вручную в меню **Плагины=>Рассылки=>Менеджер контактов** и указать id этого типа контакта при запуске скрипта. Другой вариант - при запуске указать "-1" (без кавычек), при этом этот тип контакта будет создан автоматически.

После выполнения конвертации будет создана рассылка "Рассылка баланса" со стандартным сообщением рассылки, которое можно будет в дальнейшем редактировать. Для договоров, которые были подписаны ранее на рассылку баланса будут созданы соответствующие подписки.

Плагин Documents

- [Назначение плагина Documents](#)
- [Установка и настройка плагина Documents](#)
- [Использование плагина Documents](#)
- [Шаблоны документов плагина Documents](#)
- [Массовое добавление файлов в договоры](#)
- [Web-интерфейс плагина Documents](#)
- [Отправка ссылки на файлы документа](#)

Назначение плагина Documents

Плагин предназначен для хранения документов, привязанных к договору клиента, и предоставления клиенту возможности загрузки документов через страницу Web-статистики. Также с помощью плагина есть возможность генерации документов в формате docx/xlsx/odt по заранее сформированному шаблону.

Установка и настройка плагина Documents

Плагин устанавливается с помощью утилиты **bg_installer**. После установки плагин должен быть активирован в меню **Плагины=>Настройки плагинов** и у него должна быть установлена конфигурация.

В конфигурации должно быть определено место на диске, где сервер биллинга будет хранить документы и шаблоны документов. Каталог должен быть доступен для записи пользователю, под которым запускается сервер биллинга. Также имеется возможность открытия файлов напрямую с расшаренной папки без скачивания, в случае необходимости.

```
#####
#####

# , (
file.storage.root.path=/var/billing_doc
# ,
file.storage.pattern.path=/var/billing_pattern
# ,
# ! !
file.net.share.root.path=/var/billing_files
# , .
# ! !
file.pattern.net.share.root.path

#####
#####

#
default.status=< >
#
# odt,
# LibreOffice
libre.path=

#####
#####

# email-
#
documents.mail.subject=
# Web-
web.menuItem1=
# ,
# /
# .
# , .
contract.type.<type_id>.prefix=<prefix>
contract.type.<type_id>.dir=<dir>
contract.type.<type_id>.extract.regexp=<regexp1>
contract.type.<type_id>.compare.regexp=<regexp2>
# ,
contract.type.<type_id>.dir.auto.create=<true|false>
# .
# :
```

```
# <extA1> | <extA2> | ... | <extAN> - <descriptionA>;<extB1> | <extB2> | ... | <extBN> - <descriptionB>; ...
# <ext..> , , .
client.documents.contract.file.upload.extensions=
#
client.documents.contract.file.max.size=
```

Где:

- **<type_id>** - уникальный числовой идентификатор типа договора в пределах конфигурации;
- **<prefix>** - префикс номера договора;
- **<dir>** - доступный с клиентской машины общий каталог с подкаталогами договоров;
- **<regex1>** - регулярное выражение для извлечения из номера договора ключевой последовательности;
- **<regex2>** - регулярное выражение для определения подкаталога, с ключевым словом <EXTRACT>, заменяемым на извлечённую ключевую последовательность.

Например, файлы договоров x<номер> размещаются в каталогах /tmp/TT<номер>.

```
contract.type.1.prefix=x
contract.type.1.dir=/tmp
contract.type.1.extract.regex=x(\d+)
contract.type.1.compare.regex=TT<EXTRACT>
```

Использование плагина Documents

После установки и активации плагина в каждом договоре появляется вкладка **Документы**. В редакторе справочников, доступном через меню **Плагин=>Документы**, должны быть определены перечни статусов документов, типы документов и журналы документов.

Справочники Шаблоны документов

Типы Журналы Статусы



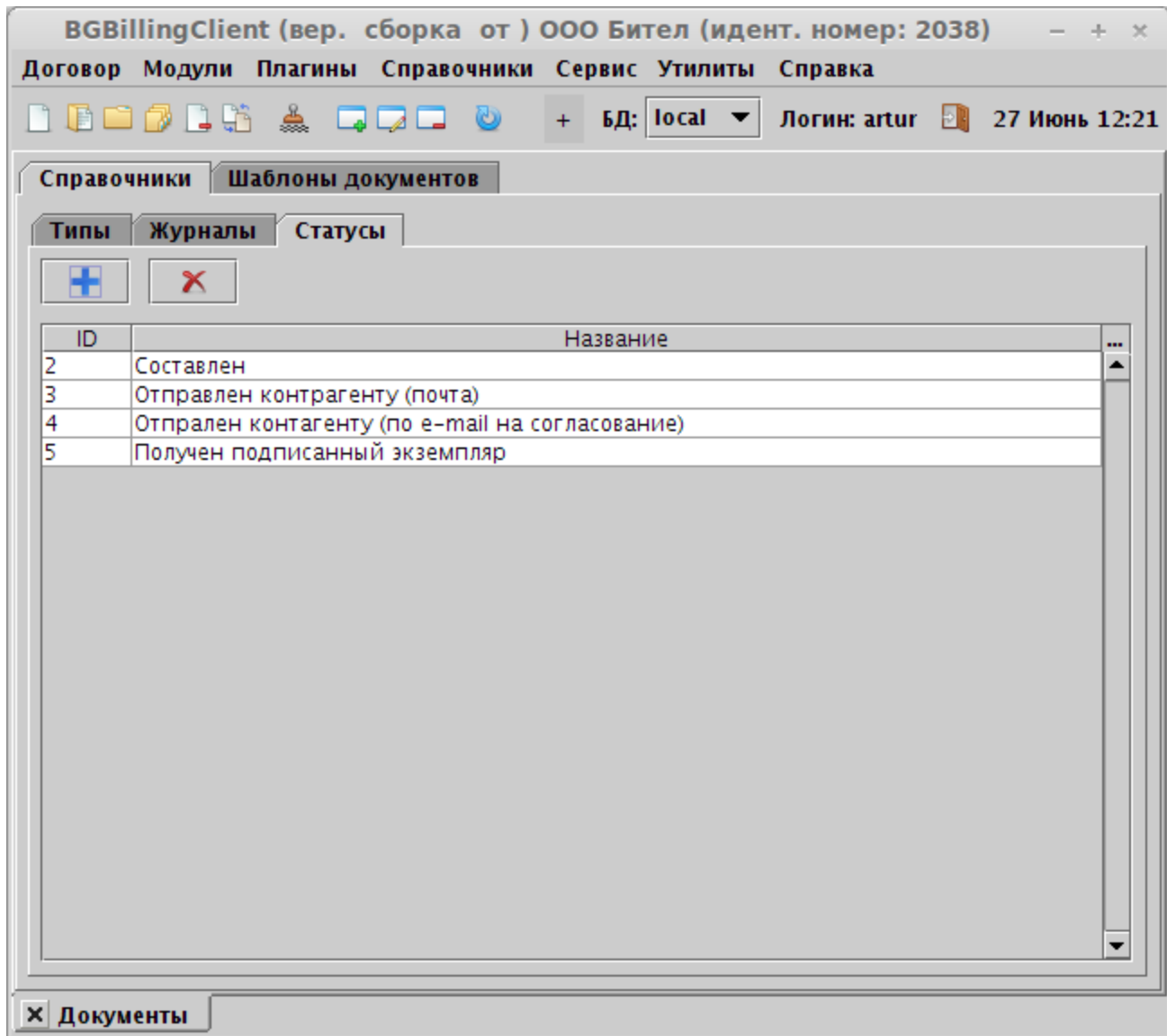
ID	Название	...
1	Договор	▲
2	Договор (приложение)	
3	Счет	
4	Акт выполненных работ	
5	Другой	

Справочники Шаблоны документов

Типы Журналы Статусы



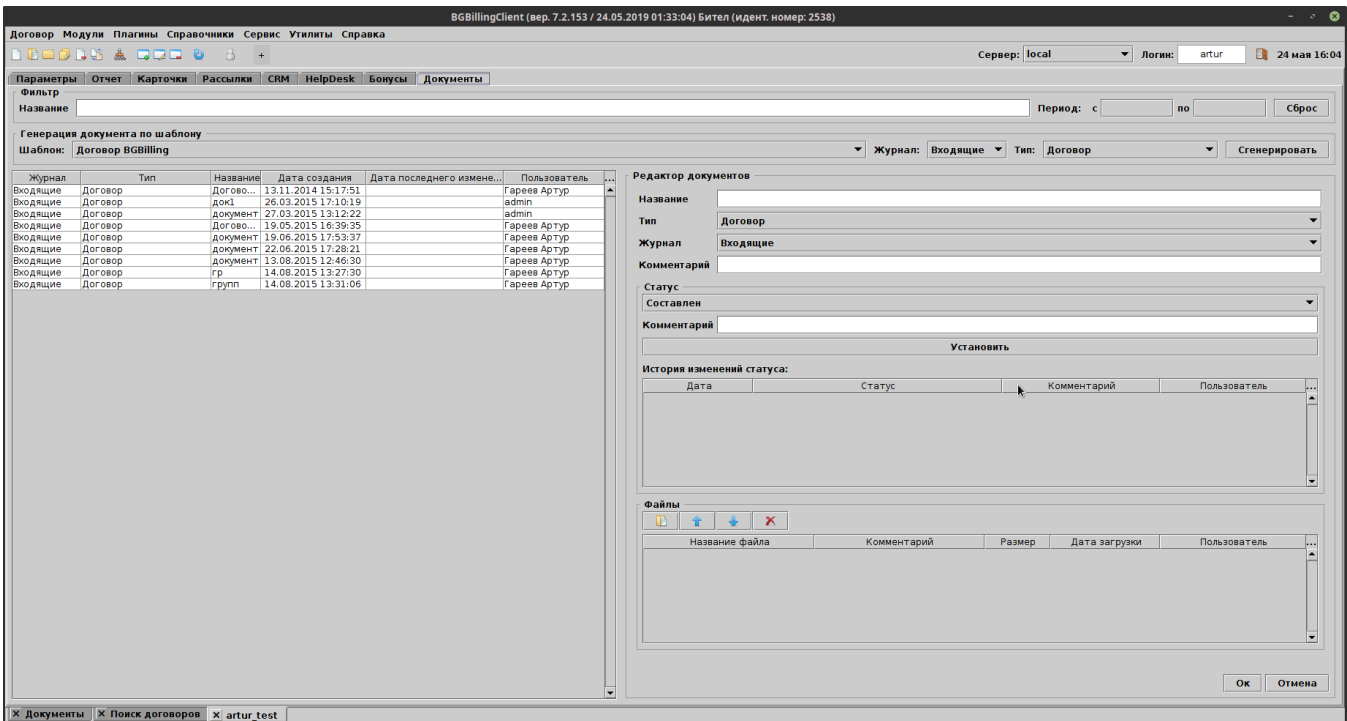
ID	Название	...
1	Входящие	▲
2	Исходящие	
3	Разное	




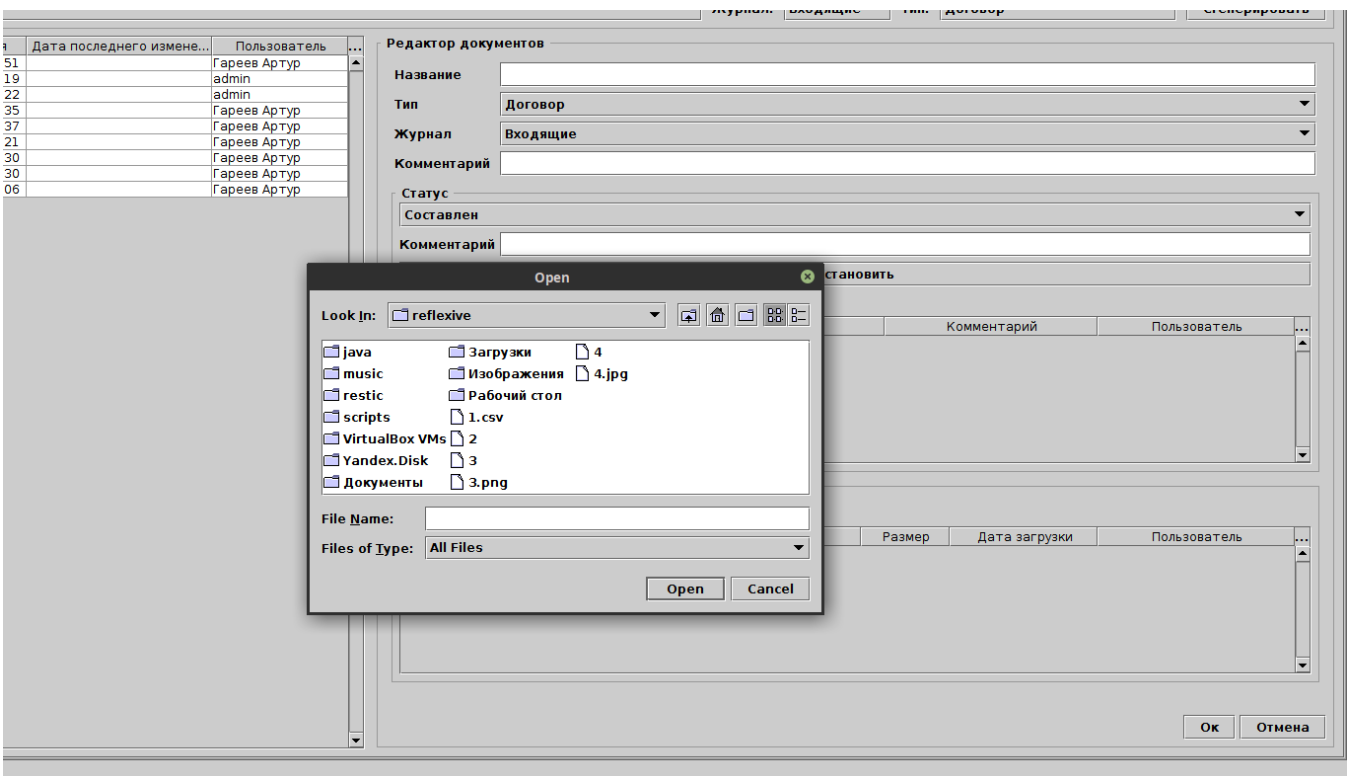
В договоре на вкладке **Документы** отображаются документы договора. Для добавления нового документа выберите **Новый элемент** на стандартной панели инструментов.


Далее введите название документа, выберите его тип, журнал и текущий статус. Для статуса можно ввести комментарий. История изменений статуса документа отображается в соответствующей таблице справа.

Нажмите **ОК** для сохранения документа.




На вкладке **Файлы** выберите файл, соответствующий текущему статусу документа, и загрузите его на сервер, нажав кнопку .



В дальнейшем вы можете загружать новые версии документа и выгрузить любое его состояние, выбрав строку в таблице и нажав на кнопку .

Для удаления документа нажмите кнопку .

Нажав кнопку  можно открыть документ в приложении операционной системы, с которым ассоциирован данный тип файлов.

Для отправки документов по почте необходимо выбрать **Отправить файл по почте** в контекстном меню, которое можно вызвать кликом правой клавишей мыши по желаемому файлу. В появившемся окне можно выбрать почтовые ящики или ввести их вручную. Далее необходимо нажать кнопку **Ок**.

Выбор e-mail

sfdgsd <adfsd@aa.ro>

mkj <sfgsdf@fd.rp>

bb <fghf@dsfsa.rp>

Все **Сброс** **Инверт.**

Другой e-mail:

*Для дополнительных e-mail
(по одному на строку).*

Ок **Отмена**

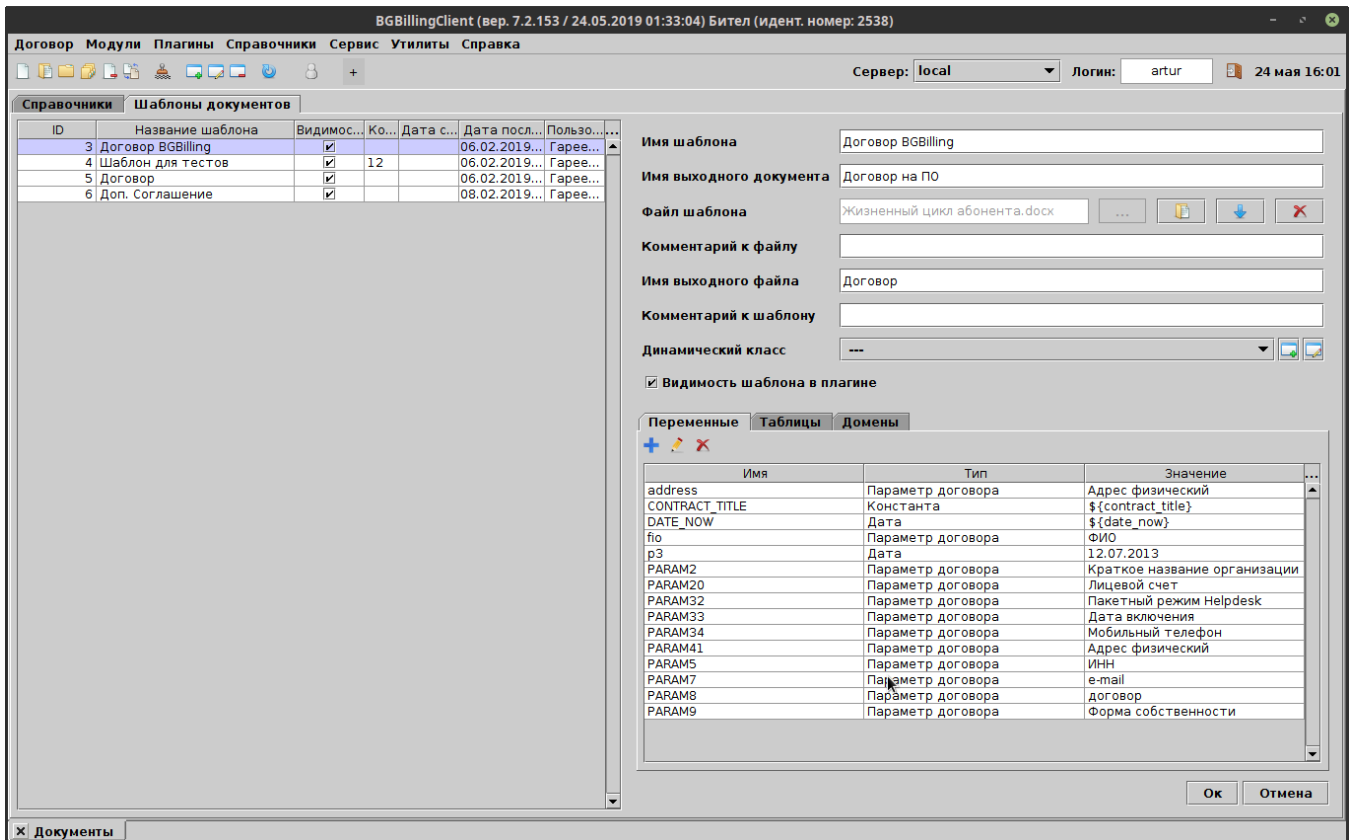
Шаблоны документов плагина Documents

Для автоматизации заполнения типовых форм документов предназначена функция генерации документов по шаблону. Шаблон документа заводится в редакторе шаблонов, доступном через меню **Плагины=>Документы=>Шаблоны документов**.

ID	Название шаблона	Видимость в плагине	Коммента...	Дата создания	Дата последней модификации	Пользователь	...
3	Договор VGBilling	<input checked="" type="checkbox"/>			28.01.2016 20:10:05	Marsel	▲
4	Шаблон для тестов	<input checked="" type="checkbox"/>	12		03.04.2017 15:03:37	Гареев Артур	
5	Договор	<input checked="" type="checkbox"/>			24.03.2016 13:58:46	Гареев Артур	
6	Доп. Соглашение	<input checked="" type="checkbox"/>			27.08.2015 18:17:33	Сергеев Кирилл	
7	Доп. Соглашение (копия 27.09.2017 18:48:55)	<input checked="" type="checkbox"/>		27.09.2017 18:48:55	27.09.2017 18:48:55	admin	▼

На данной вкладке отображается список существующих шаблонов документов. В контекстном меню каждого шаблона доступен пункт **Создать копию шаблона**, которая создает полную копию указанного шаблона, изменяя название.

Для создания нового шаблона документа щелкните на кнопку **Новый элемент** на стандартной панели инструментов. В появившемся редакторе заполните необходимые поля, выберите файл шаблона - документ в формате **docx** (предпочтительный формат), **odt** (на сервере должен быть установлен пакет LibreOffice), **xlsx**.



На представленном выше рисунке отображены следующие поля:

- **Имя шаблона** - имя шаблона документа;
- **Имя выходного документа** - название документа, который будет сгенерирован и привязан к договору;
- **Файл шаблона** - загружаемый шаблон документа в формате docx, odt, xlsx;
- **Комментарий к файлу** - комментарий к загружаемому файлу шаблона;
- **Имя выходного файла** - имя сгенерированного файла;
- **Комментарий к шаблону** - комментарий шаблона;
- **Динамический класс** - Java-класс, в котором возможно определить логику получения данных для шаблона;
- **Видимость шаблона в плагине** - флаг того, что шаблон будет отображен на вкладке Документы в договоре;
- **Переменные** - список переменных, содержащихся в файле шаблона документа;
- **Таблицы** - список таблиц, которые требуется сгенерировать в файле шаблона документа.
- **Домены** - настройка доступности шаблона документа на вкладке Документы карточки договора в зависимости от доменов.

Файл шаблона представляет собой текстовый документ, содержащий в своем тексте специальные последовательности, на место которых будут подставлены данные. Формат таких последовательностей имеет вид:

```
{_} {_( )}
```

где **имя переменной** - название этой последовательности; в тексте шаблона данное название должно совпадать с именем переменной в таблице Переменные;

значение по умолчанию - то значение, которое будет подставлено вместо последовательности в случае, если по имени переменной возвратится пустое значение (н-р, если в договоре отсутствует какой-либо заполненный параметр, то на его место можно поставить прочерк или какую-нибудь фразу).

Новая переменная заводится с помощью редактора переменных, который вызывается нажатием на кнопку . Кнопками и можно отредактировать и удалить переменную соответственно.

В диалоговом окне редактора переменной задается имя переменной, ее тип и значение. Тип переменной выбирается в соответствующем выпадающем списке. Доступны следующие варианты:

- **Константа** - значение переменной данного типа подставится в тело шаблона документа как есть. Данный тип переменной поддерживает макроподстановки, о которых речь пойдет ниже.
- **Параметр договора** - как следует из названия, значение переменной данного типа берется из параметра договора. Для этого необходимо выбрать тип параметра договора и непосредственно сам параметр.
- **Дата** - значение переменной данного типа можно выбрать из календаря (статическая дата, например, дата принятия устава), либо выбрать одно из предустановленных значений получения даты (динамическая дата. Возможные варианты: текущая дата, дата создания договора). Для переменной типа дата возможно ввести формат даты, который отформатирует дату требуемым образом. Подробнее о форматах дат см. [SimpleDateFormat](#).
- **URL** - предполагается, что значение переменной данного типа расположено на каком-то удаленном узле;
- **SQL** - получение значения переменной данного типа осуществляется путем выборки из БД с помощью SQL-запроса.

В переменных типа **Константа** возможны следующие макроподстановки:

- **#{contract_title}** - название договора, для которого генерируется документ;
- **#{contract_comment}** - комментарий договора;
- **#{current_user}** - имя пользователя, который генерирует документ;
- **#{user_<userId>}** - имя пользователя с кодом **<userId>**. Код пользователя можно посмотреть в редакторе пользователей биллинга (**Сервис =>Администрирование=>Пользователи и права**), выбрав интересующего пользователя в таблице и нажав комбинацию клавиш **Ctrl+i**.

В переменных типа **SQL** возможны следующие макроподстановки:




- **#{cid}** - код договора, для которого генерируется документ.

Если в шаблоне документа задан динамический Java-класс, то значения, возвращаемые данным классом, имеют БОЛЬШИЙ приоритет и перетирают значения одноименных переменных в списке переменных.

Динамический Java-класс реализует интерфейс **ru.bitel.bgbilling.plugins.documents.server.bean.pattern.PatternDataExtractor**, содержащий метод **extractData(Connection con, int contractId, DocumentPattern pattern, DocumentType type, DocumentJournal journal, Map<String, String> apiMap)**, который должен возвращать объект типа **Map<String, String>** - список переменных в формате "**ключ=>значение переменной**". Здесь в качестве ключа выступает название переменной, соответствующей переменной в файле шаблона документа.

Простой пример реализации метода **extractData(Connection con)**:

```
public Map extractData( Connection con, int contractId, DocumentPattern pattern, DocumentType type,
DocumentJournal journal, Map<String, String> apiMap ) throws BGException
{
    Map<String, String> result = new HashMap<String, String>();
    result.put( "param1", "value1" );
    result.put( "param2", "value2" );
    return result;
}
```

Если в файле шаблона документа есть таблицы, которые необходимо сгенерировать, то в редакторе шаблонов необходимо добавить соответствие между таблицей в файле шаблона и таблицей в шаблоне, создаваемом в клиенте биллинга. Это осуществляется на вкладке **Таблицы** редактора шаблонов путем нажатия кнопок **Добавить** , **Редактировать**  и **Удалить** .

Редактор таблицы x

Имя таблицы

Идентификатор в файле шаблона

Метод получения данных SQL ▼

Значение

В представленном редакторе задается символьное имя таблицы, ее уникальный идентификатор в файле документа, метод получения данных и, соответствующее методу получения данных, значение для таблицы. На данный момент поддерживается 2 метода получения данных:

- **SQL** - полученная SQL-запросом таблица как есть попадет в тело сгенерированного документа. В SQL-запросе возможно использование макроподстановки `#{cid}` - код договора, для которого производится генерация документа;
- **Динамический класс Java** - класс на Java, который возвращает данные для генерируемой таблицы.

Для генерации содержимого таблицы необходимо в файле шаблона создать заголовок таблицы и первую строку-шаблон, содержащую образец форматирования текста и ячеек. В этой строке-шаблоне также необходимо прописать названия столбцов, которые будут выступать в качестве ключей при получении данных (только для метода получения данных - Динамический класс). Если названия столбцов не заданы, то используются названия по умолчанию (`col1...colN`). В методе получения данных SQL по умолчанию используется нотация `col1...colN`, следовательно, задавать названия столбцов не обязательно.

Следует отметить, что для поиска таблицы в тексте документ необходимо задать уникальный идентификатор, который должен соответствовать идентификатору, заданному в редакторе таблицы клиента биллинга. Этот идентификатор можно поместить в любую ячейку таблицы, но рекомендуется поместить ее в самую первую ячейку заголовка - потребуется меньше времени для поиска нужной таблицы. Формат идентификатора соответствует формату переменных в шаблоне - **{уникальный_идентификатор}**. В редакторе таблиц этот уникальный идентификатор должен быть без фигурных скобок.

Динамический класс должен реализовывать интерфейс `ru.bitel.bgbilling.plugins.documents.server.bean.pattern.TableDataExtractor` и метод `extractData(Connection con, int contractId, List<Map<String, String>> apiRowList)`, который возвращает `List<Map<String, String>>`, т.е. список строк и содержимое столбцов в виде "ключ->значение". В качестве ключа выступает название столбца в строке-шаблоне.

Закладка **Домены** предназначена для настройки доступности шаблона документа оператору биллинга в зависимости от домена. Под доступностью понимается отображение создаваемого шаблона в выпадающем списке шаблонов документов для генерации на вкладке Документы карточки договора. Действуют следующие правила:

1. Если в договоре не установлен домен и в настройках шаблона не проставлен ни один домен, то отображаются все доступные шаблоны
2. Если в шаблоне проставлено несколько доменов и хотя бы один из них совпадает с установленным доменом договора, то такой шаблон отображается

При необходимости отображаемый список шаблонов документов можно отфильтровать самостоятельно с помощью динамического кода. Для этого необходимо написать обработчик события "Запрос списка шаблонов документов для договора" (класс события `ru.bitel.bgbilling.plugins.documents.server.bean.event.DocumentGetPatternListEvent`). Добавить обработчик события можно через меню **Сервис => Автоматизация => Функции скриптов поведения => вкладка Классы Java**. Список шаблонов, сформированный в обработчике события, имеет БОЛЬШОЙ приоритет независимо от настроек доменов договора и шаблона. В событии необходимо заполнить список шаблонов, используя следующие методы события:

Доступные методы события "Запрос списка шаблонов документов для договора"

```
/**
 *
 *
 *
 *
 *
 * @param patternList
 *
 */
public void setPatternList( List<DocumentPattern> patternList )
```

```

/**
 *
 *
 *
 *
 *
 * @param pattern
 *
 */
public void addPattern(DocumentPattern pattern)

```

Замечания:

1. Рекомендуется использовать в качестве шаблона документа файл Microsoft Office с расширением **docx**.
2. При использовании в качестве шаблона документа файл с расширением **odt** необходимо, чтобы на сервере был установлен пакет LibreOffice. Также необходимо учитывать, что при использовании данного формата файла возможно искажение форматирования документа.
3. При использовании в качестве шаблона документа файл с расширением **xlsx** будет производиться подстановка ТОЛЬКО переменных в таблице на всех листах книги.

Для генерации документа необходимо открыть договор, для которого требуется сгенерировать документ по шаблону, выбрать вкладку **Документ**, выбрать **Шаблон** документа из выпадающего списка, **Тип** генерируемого документа и **Журнал** и нажать кнопку **Сгенерировать**. Сгенерированный документ появится в списке документов договора.

Массовое добавление файлов в договоры

Для массового добавления файла (ов) к договору (ам) предназначена соответствующая групповая операция. Выберите пункт меню **Сервис->Администрирование->Групповые операции**, активируйте пункт **Документооборот: Добавление файлов к договорам**.

В результате работы групповой операции в выбранных договорах будет создан новый документ (его параметры задаются в групповой операции), к которому привяжутся выбранные файлы.

The screenshot shows the BGSBillingClient application window. The title bar indicates the version (7.1.62) and the date (21.09.2017). The main menu includes 'Договор', 'Модули', 'Плагины', 'Справочники', 'Сервис', 'Утилиты', and 'Справка'. The current view is the 'Документы' section for a selected contract, showing a list of documents and a 'Документооборот' (Document Management) section. The 'Документооборот' section is set to 'Добавление файлов к договорам' (Add files to contracts). Below this, there is a 'Список файлов' (File List) table with columns for 'Название файла' (File Name) and 'Размер файла, байт' (File Size, bytes). The table contains the following files:

Название файла	Размер файла, байт
DHCP-авторизация.pdf	254913
imgpsh_fullsize.png	400896
java-usb-src.tar.gz	10407
lic.properties	6596

Below the file list, there is a 'Параметры документа' (Document Parameters) form with the following fields:

- Название: Новый документ
- Комментарий: Документ добавлен с помощью групповой операции
- Статус: Составлен
- Тип: Договор
- Журнал: Входящие

At the bottom of the window, there are buttons for 'Выбрать файл(ы)', 'Удалить выбранные файл(ы)', 'Следующий', and 'Все'. The taskbar at the bottom shows several open windows, including 'Групповые операции'.

Web-интерфейс плагина Documents

В Web-интерфейсе клиент может посмотреть все связанные с ним документы в меню **Документы** и загрузить их.

Список документов

Название	Тип	Дата	Команды	
документ 1	Договор	07.09.2007 13:06:45	Статусы	Файлы
документ 2	Акт выполненных работ	11.09.2007 19:03:50	Статусы	Файлы
документ 3	Счет	11.09.2007 19:19:48	Статусы	Файлы
Новый документ	Акт выполненных работ	03.10.2007 11:46:31	Статусы	Файлы
Договор	Договор	24.10.2007 13:49:34	Статусы	Файлы

Список документов

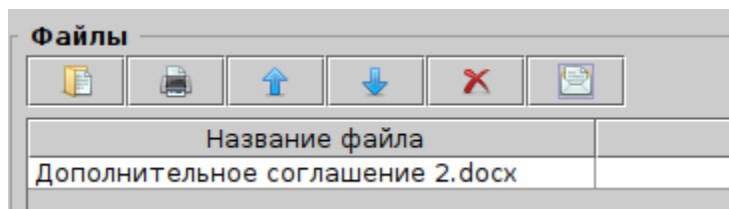
Название	Размер, кб	Дата	Пользователь	Комментарий	
test.pdf	59591	03.10.2007 11:48:26	Шамиль		загрузить

[Вернуться к списку документов](#)

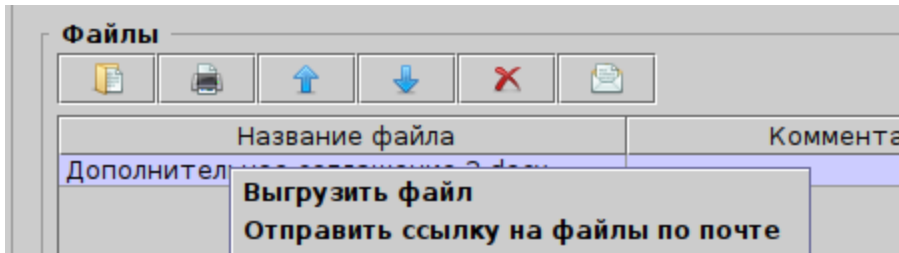
Отправка ссылки на файлы документа

С версии 9.2204+ доступна возможность отправить ссылку на файл(ы) документа по E-mail с последующим скачиванием файла без авторизации в ЛК.

Отправка ссылки возможна из редактора документа по клику на соответствующую иконку или из контекстного меню **"Отправить ссылку на файлы по почте"**



Если нужно отправить только некоторые из файлов документа их необходимо предварительно выбрать, если файлы не выбранные по ссылке будет доступен архив со всеми файлами документа.



При отправке через контекстное меню файл(ы) должны быть выбраны.

Если в документе содержится только один файл или был выбран только один файл, по ссылке будет доступен этот файл в том виде в каком он хранится в документе, если выбрано больше одного файла по ссылке будет доступен zip архив с файлами.

()

```
#
documents.file.url=http[s]://host[:8080]/bgbilling/
#
#documents.file.email.subject= "{document.title}"
```

Возможно задание времени жизни ссылки в секундах

Формат ссылки:

[http\[s\]://host\[:8080\]/bgbilling/docs/3162:4872-1649776743-0-68ec7f02f52048a0c460e2e776b3f57d4e63a97668fa446b6eb93a772e00e52a](http[s]://host[:8080]/bgbilling/docs/3162:4872-1649776743-0-68ec7f02f52048a0c460e2e776b3f57d4e63a97668fa446b6eb93a772e00e52a)

где

3162 - ID документа

4872 - ID файла(ов)

1649776743 - время создания ссылки

0 - время жизни ссылки (0 - бесконечно)

68ec7f02f52048a0c460e2e776b3f57d4e63a97668fa446b6eb93a772e00e52a - подпись

Плагин HelpDesk

Назначение плагина

Плагин предназначен для работы с клиентами через Web. Клиент в случае возникновения проблемы пишет на сервере статистики (которая может быть доступна ему даже, если у него нет счета). По запросу клиента создается отдельное сообщение и в дальнейшем вся переписка с клиентом по этой проблеме идёт в рамках этого сообщения. Клиент может вести обсуждение с провайдером, прикреплять файлы к обсуждению.

Алгоритм работы

Пользователь заходит на страницу статистики и может создавать новую тему, может оставить сообщение в одну из созданных им тем. Пользователь видит только те темы, которые созданы от его имени. Провайдер ведёт обсуждение с пользователем.

Активные темы - темы на которые не было ответов со стороны провайдера.

Закрытые темы - темы, по которым обсуждение уже завершено.

Новые темы - темы, которые не помечены как прочитанные.

Менеджеры могут просматривать сообщения от пользователей и отвечать на них. Менеджер может назначить себя ответственным за конкретную тему.

Информация о новых темах и сообщениях рассылается на почту провайдера. Оповещения о новых сообщениях от менеджеров приходят на почту клиента.

Темы могут автоматически закрываться по прошествии некоторого периода.

Опционально можно включить поддержку платных обращений. Она реализуется в пакетном режиме. Пакет - это некоторый набор обращений на определённый срок. Указывается стоимость пакета. Клиент через Web-интерфейс может активировать определённый пакет, если у него на счёте достаточно средств для оплаты этого пакета. При этом ему начисляется расход, равный стоимости пакета.

Установка и настройка плагина

Плагин устанавливается стандартным образом. В настройках плагина (**Плагины->HelpDesk->Конфигурация**) задайте конфигурацию:

Конфигурация

```
#
file.storage.root.path=/home/bill/BGBillingServer/filestorage/helpdesk
#
mode.list=off;on;package:
#
mode.default.id=on
# ( , ): 0 - , 1 -
state.mode=0
# : 0 - ; 1 - ; 2 - e-mail
default.comm.mode=0
# (true/false)
default.comm.phone.enable=false
# e-mail (true/false)
default.comm.email.enable=true
#
default.comm.email.subject=HelpDesk => : [{id}] {title}
# " " (true - , false - )
topic.avto.closed.default=false
# Web- ( )
web.menuItem1=HelpDesk
# ,
topic.charge.type.id=51
#
topic.default.comment= {topicId}
#
topic.default.cost=2000.0
```

Для рассылки оповещений нужно настроить задачу планировщика "**Рассылка уведомлений о приходе новых сообщений**" для менеджеров. В параметрах запуска задачи надо указать:

```
#
mail.to=billing@zzz.ru
# , , ,
#contract.title.regex=
# , , ,
#contract.group.mask=
#     helpdesk
mail.subject=HelpDesk =>      - {countAll}/{countDay}/{countHour} (// )
#     helpdesk
mail.body={countAll} - \n{countDay} - \n{countHour} -
```

Для автоматического закрытия старых тем, на которые нет ответов менеджеров, нужно настроить задачу "**Автозакрытие тем через заданный интервал**". Параметры запуска это задачи :

```
#
topic.avto.closed.period=5
#
mail.to=billing@zzz.ru
```

Для рассылки персональных оповещений надо настроить задачу "**Рассылка персональных уведомлений о новых сообщениях**", она позволяет настроить для каждого менеджера уведомления на e-mail и/или на jabber о появлении новых ответов в его теме, а также в темах, которые ещё не закреплены за менеджером. Пример конфигурации:

```
#     ():
mail.body= :\n{youNews}\n:\n{unmanagedNews}
mail.subject=HelpDesk => {youCountAll} ( {unmanagedCountAll} )
mail.bodypart={title} (ID {id}) - {count} \n

#     - .
user.dimon.mail=dimon@mail.ru
user.dimon.jid=dimon@jabber.ru
user.test1.mail=test1@bitel.ru
user.test2.jid=test2@jabber.ru
```

Для того, чтобы работали оповещения на jabber, в конфигурации СЕРВЕРА должны быть прописаны данные jabber-аккаунта (бота), например:

```
# jabber
# @/
jabber.jid=bgbilling@jabber.***.ru/helpdesk
#
jabber.password=***
```

Также есть "**Рассылка персональных уведомлений о новых сообщениях**" и у клиентов. При выборе уведомления по e-mail в конфигурации плагина следует добавить url сервера биллинга, чтобы в письмо добавлялась ссылка на топик. Пример конфигурации:

```
url.reference.topic=http://localhost:9580/bgbilling/webexecuter?
```

Можно, чтобы при создании темы пользователь выбрал дополнительную информацию которая подставиться к названию темы в квадратные скобки(тогда название темы будет иметь следующий вид - [выбор пользователя] Название темы)

```
#
topic.additionalTextToSubject.title= :
#
topic.additionalTextToSubject.options=5.1,5.2,6.0,6.1
```

Настройка групп и категорий

1) Создаем группы

The screenshot shows the 'Группы' (Groups) tab selected. The 'Фильтр' (Filter) field is empty. Below it is the 'Таблица Группы' (Groups Table) with the following data:

ID	
1	Биллинг
3	Домены
2	Хостинг

2) создаем категории

The screenshot shows the 'Категории' (Categories) tab selected. The 'Фильтр' (Filter) field is empty. Below it is the 'Таблица Категории' (Categories Table) with the following data:

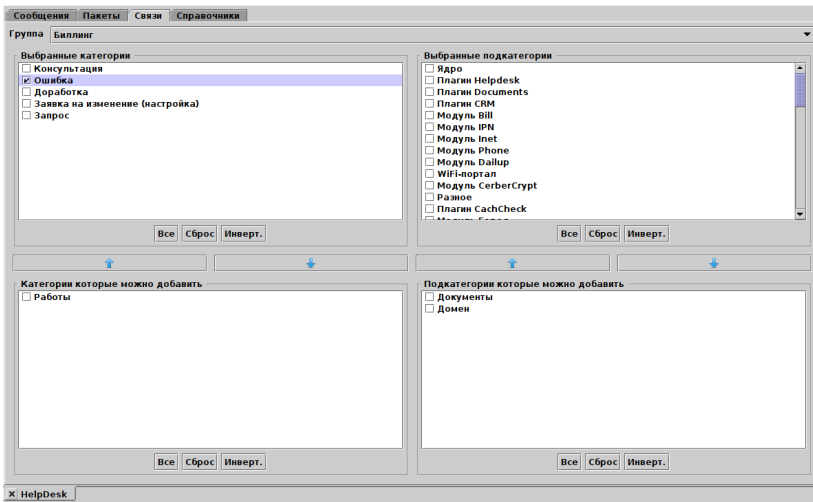
ID	
5	Доработка
8	Запрос
6	Заявка на изменение (настройка)
4	Консультация
3	Ошибка
7	Работы

3) Создаем подкатегории

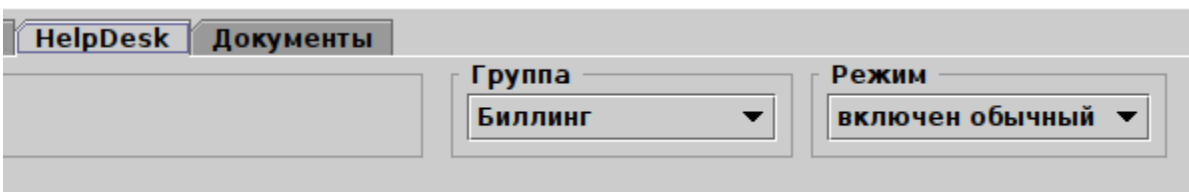
The screenshot shows the 'Подкатегории' (Subcategories) tab selected. The 'Фильтр' (Filter) field is empty. Below it is the 'Таблица Подкатегории' (Subcategories Table) with the following data:

ID	
10	WiFi-портал
49	Документы
22	Домен
26	Лицензии
54	Личный кабинет
48	Модуль Alfabank
23	Модуль Assist
5	Модуль Bill
17	Модуль BVCom

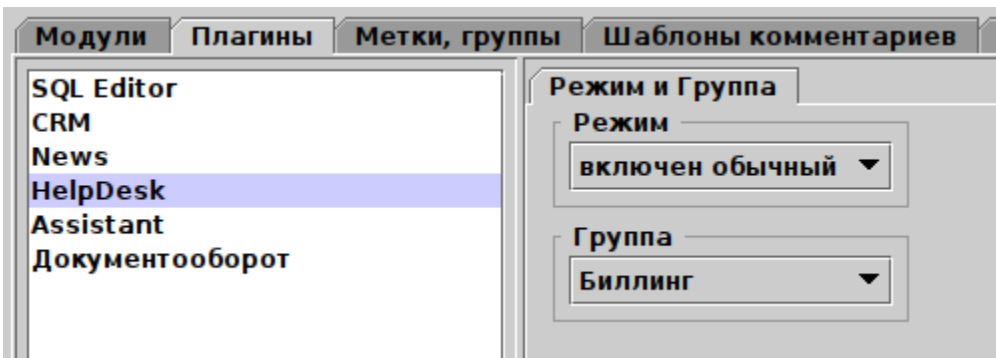
4) Для каждой группы созданной на первом шаге создаем связи категорий с подкатегориями



5) на договорах указываем группу и режим



6) для новых договоров в шаблоне указываем режим и группу по умолчанию



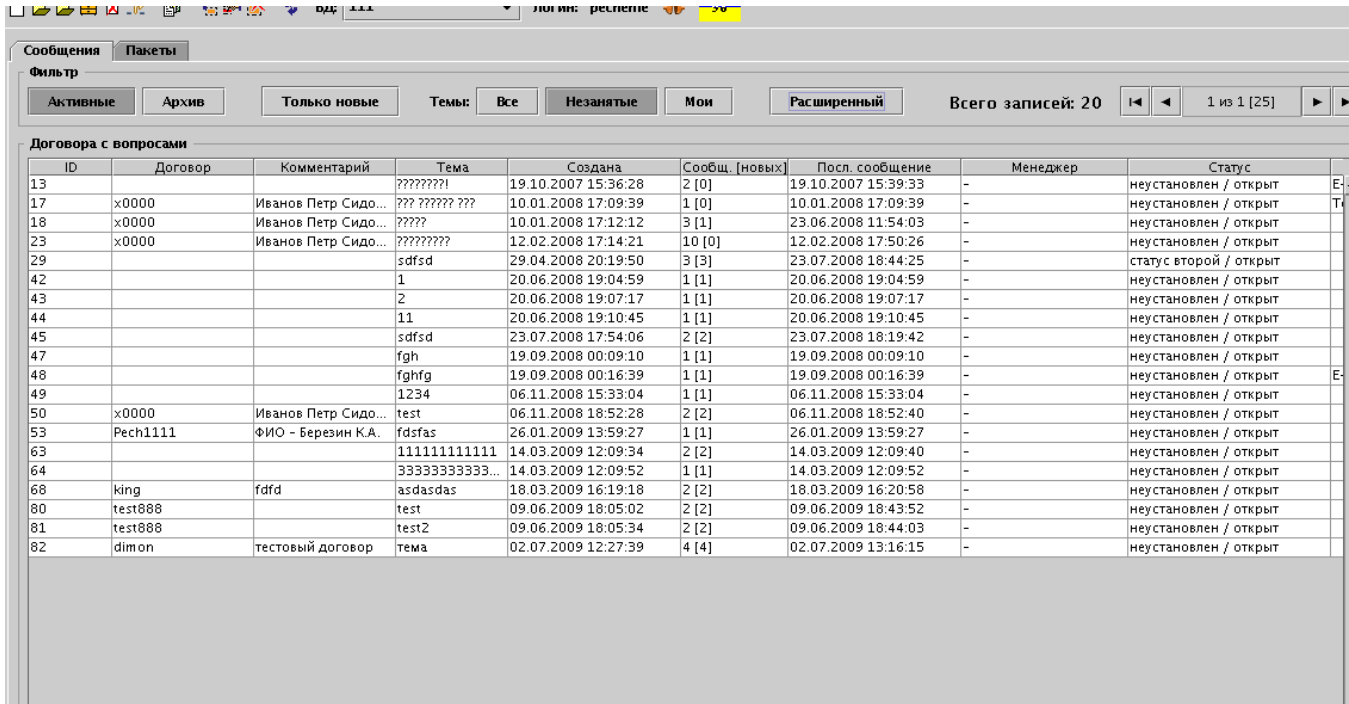
Статусы тем

В справочнике необходимо настроить статусы тем Меню - Справочники - Другие

Справочники	Helpdesk - Статус темы
Договоры - параметры	Доработка
Договоры - группы параметров	Запрос
Договоры - группы	Консультация
Договоры - метки	Ошибка
Договоры - шаблоны комментариев	Предложение
Договоры - именованные номера шаблонов	Работы
Договоры - значения списков	я_TODO
Договоры - скрипты поведения	я_Долгий ящик
Договоры - обслуживание	я_К закрытию
Домены	я_Надо бы сделать
Типы платежей	я_Неправильная тема
Типы расходов	
Типы возвратов	
Типы резервов	
Типы времени	
CRM - типы звонков	
CRM - группы решения	
CRM - исполнители	
CRM - категории проблем	
CRM - типы задач	
Helpdesk - Статус тем	

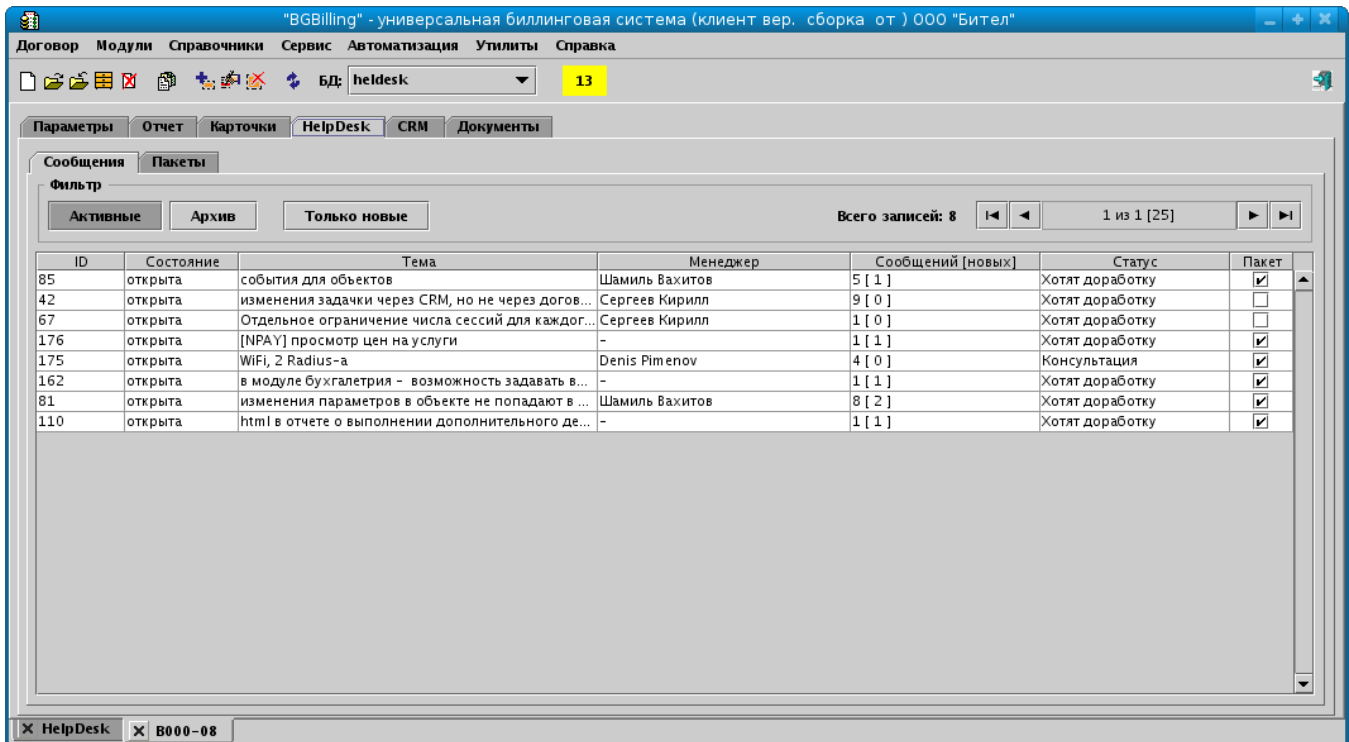
Работа с сообщениями

Для работы с сообщениями клиентов надо открыть вкладку **Плагины->Helpdesk->Сообщения**:

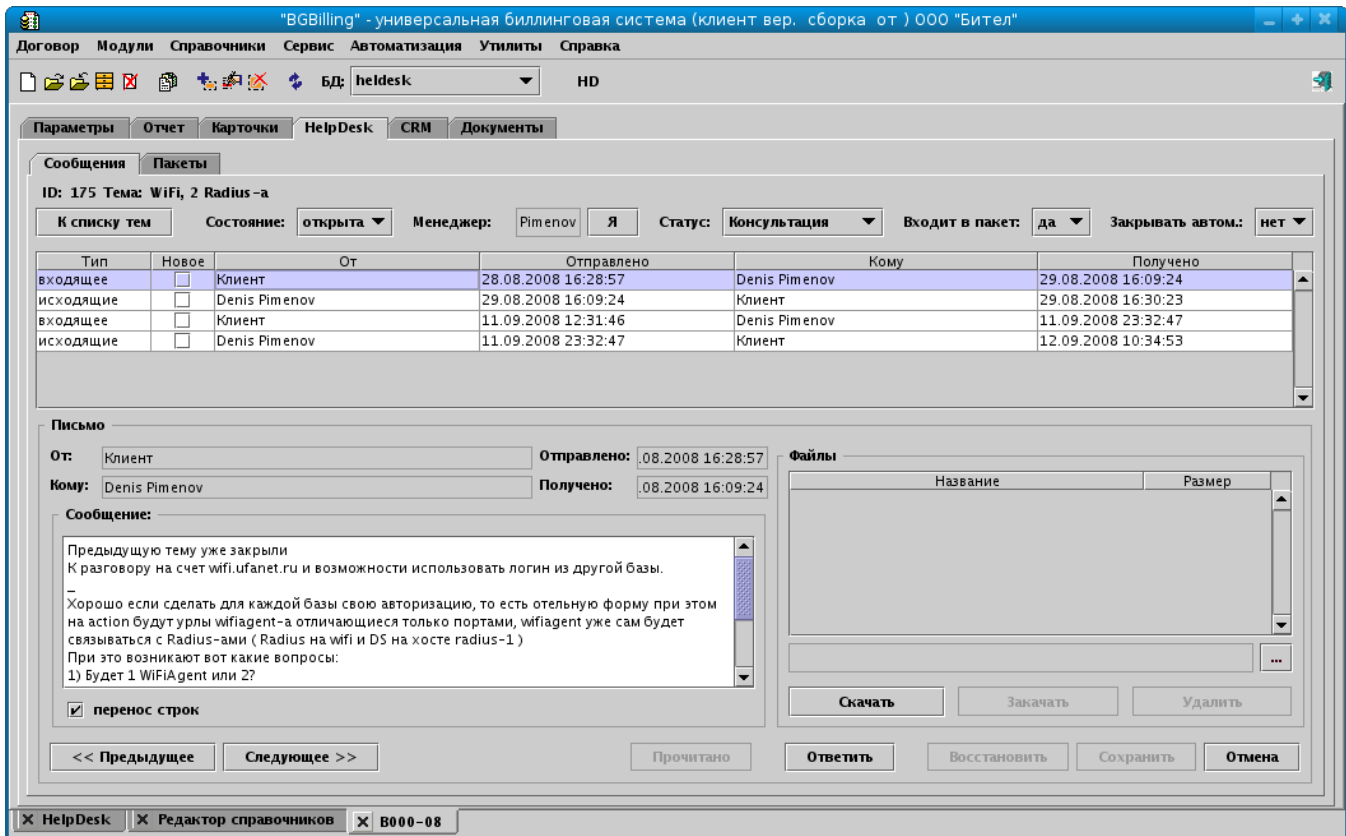


Здесь вы можете просмотреть все темы, созданные пользователями. Кнопка **Активные** - фильтрация активных тем. **Архив** - закрытые темы. **Только новые** - новые сообщения. **Незаятые** - выводит темы, которые на связаны с менеджером. **Мои** - сообщения текущего менеджера. Кнопка **Расширенный** открывает дополнительный фильтр тем по ID, названию, содержанию сообщений, дате и статусу.

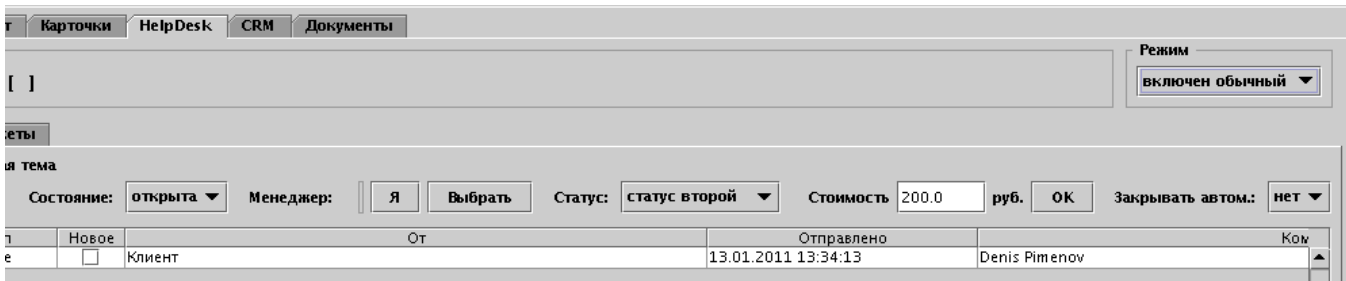
При двойном клике на теме открывается договор, там уже показаны только темы данного договора:



Двойным кликом на теме открывается редактор этой темы. Там отображается уже список сообщений внутри этой темы. Двойным кликом на конкретном сообщении можно открыть его для просмотра:



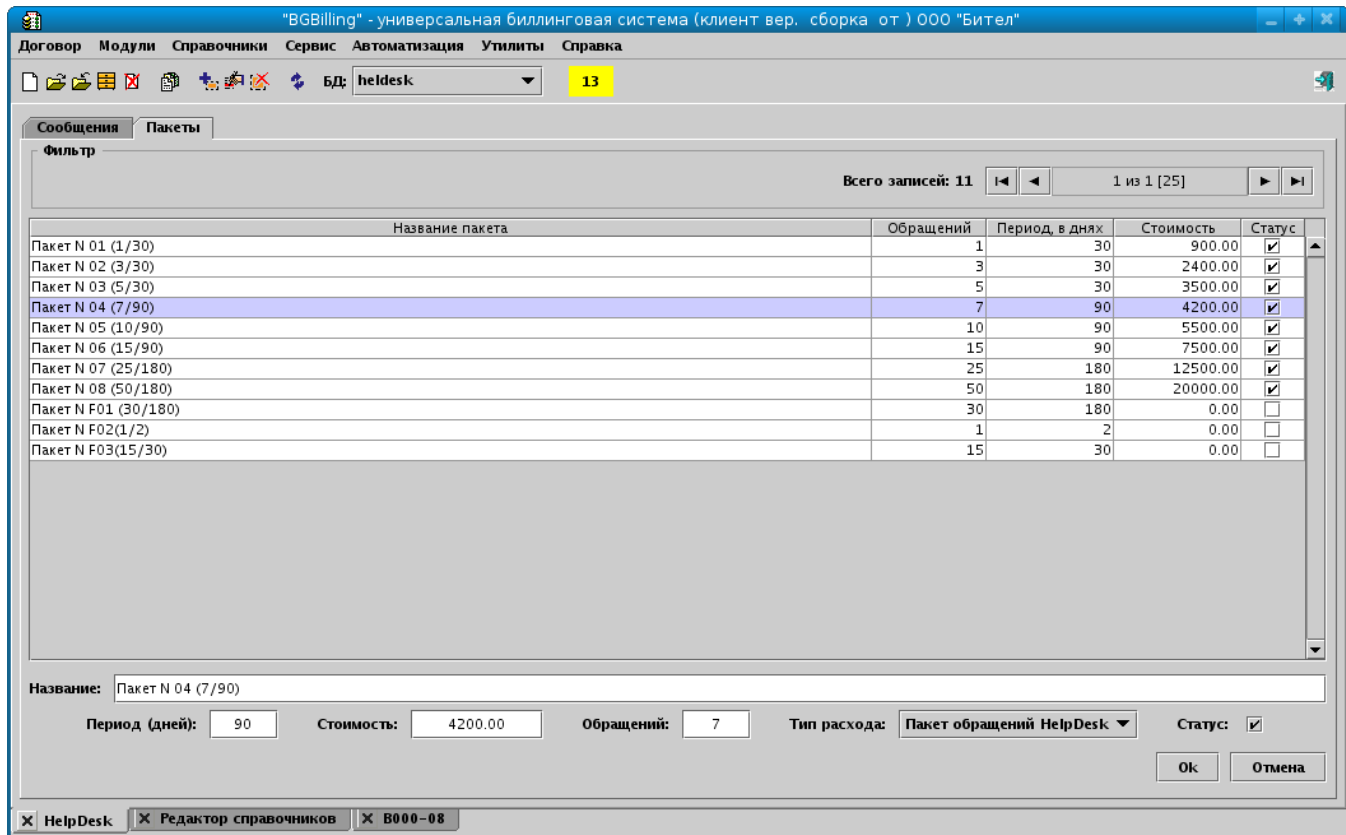
Здесь можно менять состояние темы - открыта/закрыта. Можно установить себя менеджером данной темы. Можно установить дополнительный статус темы. Статусы выбираются из справочника "Helpdesk - Статус тем". При использовании пакетного режима обращений можно указать входит ли данная тема в пакет. Так же можно указать закрывается ли данная тема автоматически. При просмотре конкретного сообщения можно на него ответить, есть возможность прикреплять файлы к своему сообщению и скачивать файлы.



Если для договора установлен непакетный режим (то есть режим выключен или включен обычный), то можно указать стоимость обращения вручную в соответствующем поле. Значение стоимости непакетного обращения по умолчанию задается в конфигурации плагина. При закрытии такого обращения договору начислится расход с указанной суммой. Тип расхода и комментарий к нему задаются также в конфигурации плагина. При открытии закрытого обращения соответствующий расход удаляется.

Работа с пакетами обращений

Для просмотра пакетов надо открыть вкладку **Плагины->Helpdesk->Пакеты**:



Здесь создаётся пакет, для которого задаётся период его действия, его стоимость, количество обращений, которое входит в данный пакет, тип расхода (тут можно выбирать из нередатируемых типов расходов, заданных в справочнике). Статус - может ли клиент использовать данный пакет.

Web-интерфейс клиента

На сайте статистики клиент выбирает helpdesk. Там по умолчанию ему показываются активные сообщения :

The screenshot shows the 'HelpDesk - Активные темы' section. It includes navigation buttons for 'Новая тема', 'Активные', 'Архив', 'Параметры', and 'Пакеты'. There are search filters for 'ID Темы', 'Название', and 'В периоде с' followed by 'по'. A search bar labeled 'Строка поиска в сообщениях' is present with 'Фильтр' and 'Очистить' buttons. Below is a table of active topics.

ID	Заголовок	Оповещение	Дата создания	Состояние	Дата закрытия	Статус	Входит в пакет	Выгрузка
2575	Новая тема	-	27.01.2011 11:05:02	открыта			да	сохранить

С помощью кнопки **"Новая тема"** клиент может добавить новую тему . Кнопка **"Активные"** - отображает активные темы, **"Архив"** - закрытые темы.

Возможна фильтрация тем по ID, названию, периоду и тексту сообщений. Также есть возможность сохранять историю переписки по темам в файл.

Клиент может выбрать одну из уже созданных тем и увидеть переписку по ней :

The screenshot shows the 'HelpDesk - Переписка' interface. It includes navigation buttons for 'Новая тема', 'Активные', 'Архив', 'Параметры', and 'Пакеты'. The message thread for topic ID #85 is displayed, with the subject 'Тема: события для объектов'. The thread contains three messages with their respective IDs, senders, and timestamps.

Меню:

- Новости
- Просмотр баланса
- Смена пароля на доступ к статистике
- Подписка на рассылки
- Смена тарифных планов
- Дополнительные действия
- Примечания
- Счета
- Акты
- HelpDesk
- Документы
- Выход

HelpDesk - Переписка

Новая тема Активные Архив Параметры Пакеты

ID: # 85; Статус: Активна; Дата открытия: 02.07.2008 17:19:08; Оповещение: нет; Менеджер: Шамиль Вахитов

Тема: события для объектов

Сообщение ID: #331; Отправлено: Клиент в 23.06.2008 16:05:14; Прочитано: Шамиль Вахитов в 23.06.2008 16:58:14

А будут ли в биллинге какие-нибудь события, связанные с действиями, выполняемыми над объектами? Например, создание объекта, изменение параметра объекта и т.п.?

Сообщение ID: #337; Отправлено: Шамиль Вахитов в 23.06.2008 16:58:14; Прочитано: Клиент в 23.06.2008 16:58:59

Клиент писал (23.06.2008 16:05:14):

>>А будут ли в биллинге какие-нибудь события, связанные с действиями, выполняемыми над объектами? Например, создание объекта, изменение параметра объекта и т.п.?

Попросите - будут. Какие нужны?

Сообщение ID: #367; Отправлено: Клиент в 30.06.2008 13:03:57; Прочитано: Шамиль Вахитов в 02.07.2008 15:22:54

Для начала, думаю, хватит вот таких:

Готово

С этой же страницы он может посылать новые сообщения в текущую тему .


При выборе **"Параметры"** клиент может задать то, как он будет получать уведомления о новых сообщениях :

HelpDesk - Параметры | BGBilling - Mozilla Firefox

Файл Правка Вид Журнал Закладки Инструменты Справка

http://billing.office.bitel.ru/bgbilling/webexecuter?action=HelpDeskParam&m

Самые популяр... LINUX.ORG.RU - Пу... BG price download BG тема BG Doc JOBS dbinfo BG home BG forum



Договор № В000-08

Меню

- »» Новости
- »» Просмотр баланса
- »» Смена пароля на доступ к статистике
- »» Подписка на рассылки
- »» Смена тарифных планов
- »» Дополнительные действия
- »» Примечания
- »» Счета
- »» Акты
- »» HelpDesk
- »» Документы
- »» Выход

HelpDesk - Параметры

Новая тема Активные Архив Параметры

Выберите режим оповещения об ответах и значения по умолчанию:

- не оповещать
- по телефону
- по email

Сохранить

© 2007-2008 BiTel, Ltd.

Готово

"Пакеты" :

HelpDesk | BGBilling - Mozilla Firefox

Файл Правка Вид Журнал Закладки Инструменты Справка

http://billing.office.bitel.ru/bgbilling/webexecuter?action=HelpDeskPackage&mo

Самые популяр... LINUX.ORG.RU - Ру... BG price download BG тема BG Doc JOBS dbinfo BG home BG forum

Меню

- » Новости
- » Просмотр баланса
- » Смена пароля на доступ к статистике
- » Подписка на рассылки
- » Смена тарифных планов
- » Дополнительные действия
- » Примечания
- » Счета
- » Акты
- » HelpDesk
- » Документы
- » Выход

HelpDesk

Новая тема Активные Архив Параметры Пакеты

Активированные пакеты:

ID	Период	Стоимость	Обращений [использовано]	Активный	Активировал
2	10.06.2008-06.12.2008	20000.00	50 [50]	нет	Администратор
3	24.06.2008-21.12.2008	0.00	30 [6]	да	Администратор

Доступные пакеты:

V	Название пакета	Период [дней]	Обращений	Стоимость
<input type="radio"/>	Пакет N 01 (1/30)	01.10.2008-31.10.2008 [30]	1	900.00
<input type="radio"/>	Пакет N 02 (3/30)	01.10.2008-31.10.2008 [30]	3	2400.00
<input type="radio"/>	Пакет N 03 (5/30)	01.10.2008-31.10.2008 [30]	5	3500.00
<input type="radio"/>	Пакет N 04 (7/90)	01.10.2008-30.12.2008 [90]	7	4200.00
<input type="radio"/>	Пакет N 05 (10/90)	01.10.2008-30.12.2008 [90]	10	5500.00
<input type="radio"/>	Пакет N 06 (15/90)	01.10.2008-30.12.2008 [90]	15	7500.00
<input type="radio"/>	Пакет N 07 (25/180)	01.10.2008-30.03.2009 [180]	25	12500.00
<input type="radio"/>	Пакет N 08 (50/180)	01.10.2008-30.03.2009 [180]	50	20000.00

Активировать (выбранный пакет будет активирован только при наличие на счете суммы большей или равной стоимости пакета)

© 2007-2008 BiTel, Ltd.

Готово

Здесь можно увидеть активированные пакеты и пакеты, которые клиент сам может себе активировать.

Оповещение (для пользователей биллинга)

Поддерживается несколько режимов оповещения о поступлении новых сообщений от клиентов

- 1) На E-mail (в конфиге пользователя биллинга необходимо прописать параметр `helpdesk.mail=user@domain.ru`)
- 2) В Jabber
- 3) В Telegram (в конфиге пользователя биллинга необходимо прописать параметр `telegram.user.id=311204768`)

Настройка оповещения E-mail

настроить запуск задачи планировщика (HelpDesk => рассылка персональных уведомлений о новых сообщениях) с конфигом

```
mail.body= ( {youCountAll} ):\n{youNews}\n ( {unmanagedCountAll} ):\n{unmanagedNews}
mail.subject=HelpDesk => {youCountAll} ( {unmanagedCountAll} )
mail.bodypart={title} (ID {id}) - {count} \n
#
user.<__>.mail=nickname@mail.ru
```

Настройка оповещения Jabber

В конфигурации сервера прописать

```
# , ,
# @/
jabber.jid=billing@jabber.local/helpdesk
jabber.server=jabber.local
#
jabber.password=123456
```

настроить запуск задачи планировщика (HelpDesk => рассылка персональных уведомлений о новых сообщениях) с конфигом

```
#
user.<__>.jid=nickname@jabber.local
```

Настройка оповещения в Telegram

- 1) В Telegram с помощью бота [@BotFather](#) создаете бота и получаете токен типа `983377823:dsahfuiwehfufqywdg26dqf6Vzi8klN8`

Информация о боте



BotFather

бот



BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

Информация

@BotFather

Имя пользователя

2) в конфиге плагина прописываем

```
telegram.bot.token=983377823:dsahfaiwehfufqywdg26dqf6Vzi8klN8  
telegram.webhook.url=https://[ ]/bgbilling/telegramHelpdeskWebhook
```

Для ограничения доступа к адресу **webhook** рекомендуем маскировать его с помощью проксирования **nginx**

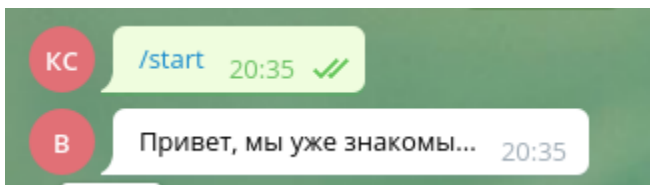
т.е. на **nginx** настроить какойнибудь **location** типа **/bgbilling/telegramHJgweHJUgheby** с пробросом на **/bgbilling/telegramHelpdeskWebhook**

а прямой доступ к **/bgbilling/telegramHelpdeskWebhook** запретить

соответственно в этом случае в параметр нужно будет прописать

telegram.webhook.url=https://[адрес биллинга]/bgbilling/telegramHJgweHJUgheby

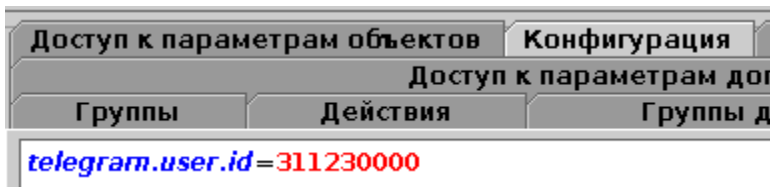
3) Далее подключаемся к созданному ранее боту и вводим **/start**



4) вводим команду **/showMyId**



5) полученный выше ID прописываем в конфигурации пользователя биллинга

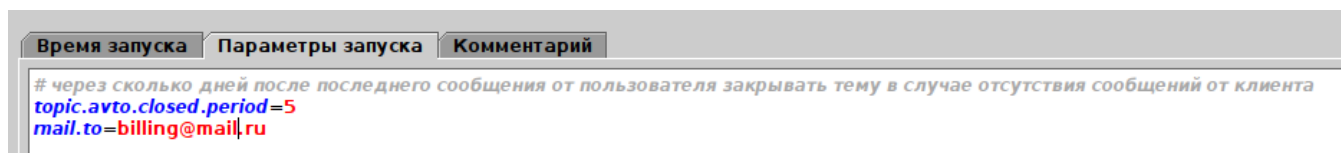


6) После указанных настроек в Telegram должны приходить сообщения Менеджером в которых указан данный пользователь, а также все сообщения у которых не установлен Менеджер

Автозакрытие тем

В плагине доступна задача планировщика "**Плагин HelpDesk. Задача автозакрытия тем.**" с помощью которой можно закрыть темы по которым заданное время не было новых сообщений.

Настройка задачи производится в параметрах запуска задачи, рекомендуемая частота запуска задачи раз в сутки.



Для того что бы задача могла закрыть тему необходимо соблюдение следующих условий:

- у темы должен стоять режим **Закрывать автоматически Да**
- все сообщения в теме прочитаны
- последнее сообщение было от менеджера и оно прочитано клиентом
- с даты прочтения клиентом прошло заданное в параметре **topic.avto.closed.period** дней

Плагин Organizer

Назначение плагина

Часто возникают ситуации планирования задач непосредственно внутри биллинга. Особенно затруднительно помнить о необходимости выполнения каких-либо действий через достаточно большой промежуток времени (месяц, квартал, год). Например, действие договора было приостановлено по просьбе клиента, и, в случае, если клиент самостоятельно не сообщит о необходимости возобновления договора в течение двух месяцев, необходимо закрыть договор. Для планирования таких (или подобных) задач внутри клиента биллинга можно воспользоваться плагином **Organizer**.

Настройка плагина

В конфигурации плагина можно указать временной диапазон (в годах), который будет отображаться в календарном дереве (см. далее) при поиске заданий и при просмотре журнала заданий.

```
calendar.year.from=< >  
calendar.year.to=< >
```

Здесь **<год начала>** и **<год конца>** - это годы (например, 2009 и 2011 соответственно).

По умолчанию, если в конфигурации ничего не указано, то в качестве диапазона берётся текущий год и следующий за ним.

Общий обзор плагина Organizer

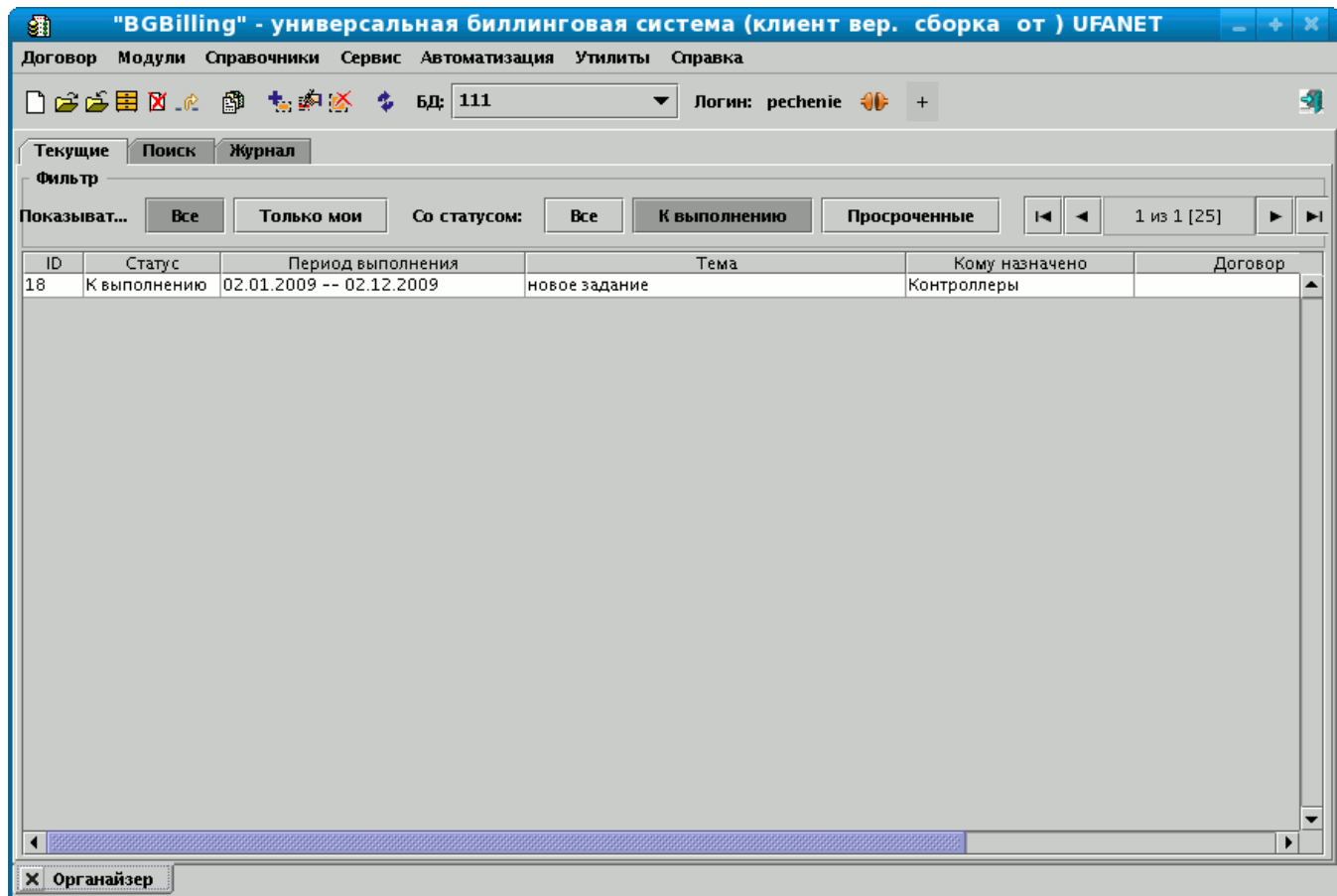
Основная сущность, с которой работает плагин, - это **задание**. **Задание** - это некоторое предписание конкретному пользователю биллинга или целой группе пользователей со ссылкой на некоторый договор или без неё, которое необходимо выполнить в установленные сроки. Задания могут, фактически, иметь четыре статуса:

- **не наступили сроки выполнения задания** - это означает, что задание уже создано в системе, но период его выполнения ещё не наступил;
- **к выполнению** - это означает, что задание создано, и наступил срок выполнения этого задания;
- **просрочено** - это означает, что время выполнения задания уже истекло;
- **выполнено**.

Ответственность за корректное выполнение заданий целиком лежит на самих пользователях. Только пользователь, выполнивший (по его мнению) задание, решает - пометать ли его как выполненное. К слову, выполнять задания могут и пользователи, которым изначально они не были предписаны. Выполненные же задания можно вновь пометать как невыполненные. При каждом снятии или установке пометки о выполнении задания в журнал заданий заносится соответствующая запись. Это позволяет отслеживать историю задания, а также попытки "переоформить" выполнение задания на себя.

Как уже было сказано выше, основными характеристиками задания являются непосредственно текст задания, назначение некоторому пользователю или группе пользователей, сроки выполнения, а также договор, с которым связано задание. Например, если нужно будет выполнить какие-либо действия с некоторым договором, то сразу после прочтения текста задания можно перейти во вкладку этого договора.

Основное окно плагина находится во вкладке **Плагины->Организер**.



Здесь расположены три вкладки: **Текущие**, **Поиск** и **Журнал**. Вкладка **Текущие** позволяет просмотреть задания, время выполнения которых уже наступило (в том числе и просроченные задания). Вкладка **Поиск** позволяет искать задания (как выполненные, так и невыполненные) по различным критериям. Вкладка **Журнал** отображает журнал изменения заданий. Данный функционал более подробно будет рассмотрен ниже.

В статусной панели клиента биллинга при включении плагина появляются кнопка "+" (добавить новое задание) и кнопка, уведомляющая пользователя о текущем состоянии назначенных ему заданий (в случае, если на данный момент отсутствуют задания, необходимые к выполнению или просроченные, то кнопка скрыта).



Заданий [просрочено]: 1 [0] +

Добавление задания

Создать новое задание можно двумя способами:

- нажатием на кнопку +;
- нажатием на кнопку "Добавить", находясь во вкладках **Текущие** или **Поиск**.

Полезное примечание: если нажать на кнопку "+" находясь непосредственно на вкладке некоторого договора, то он автоматически добавиться в поле связанного с заданием договора.

В обоих случаях откроется **Редактор задания**.

Редактор

Тема

Комментарий

Договор

Кому назначено:

Пользователю:

или группе:

Период

 по

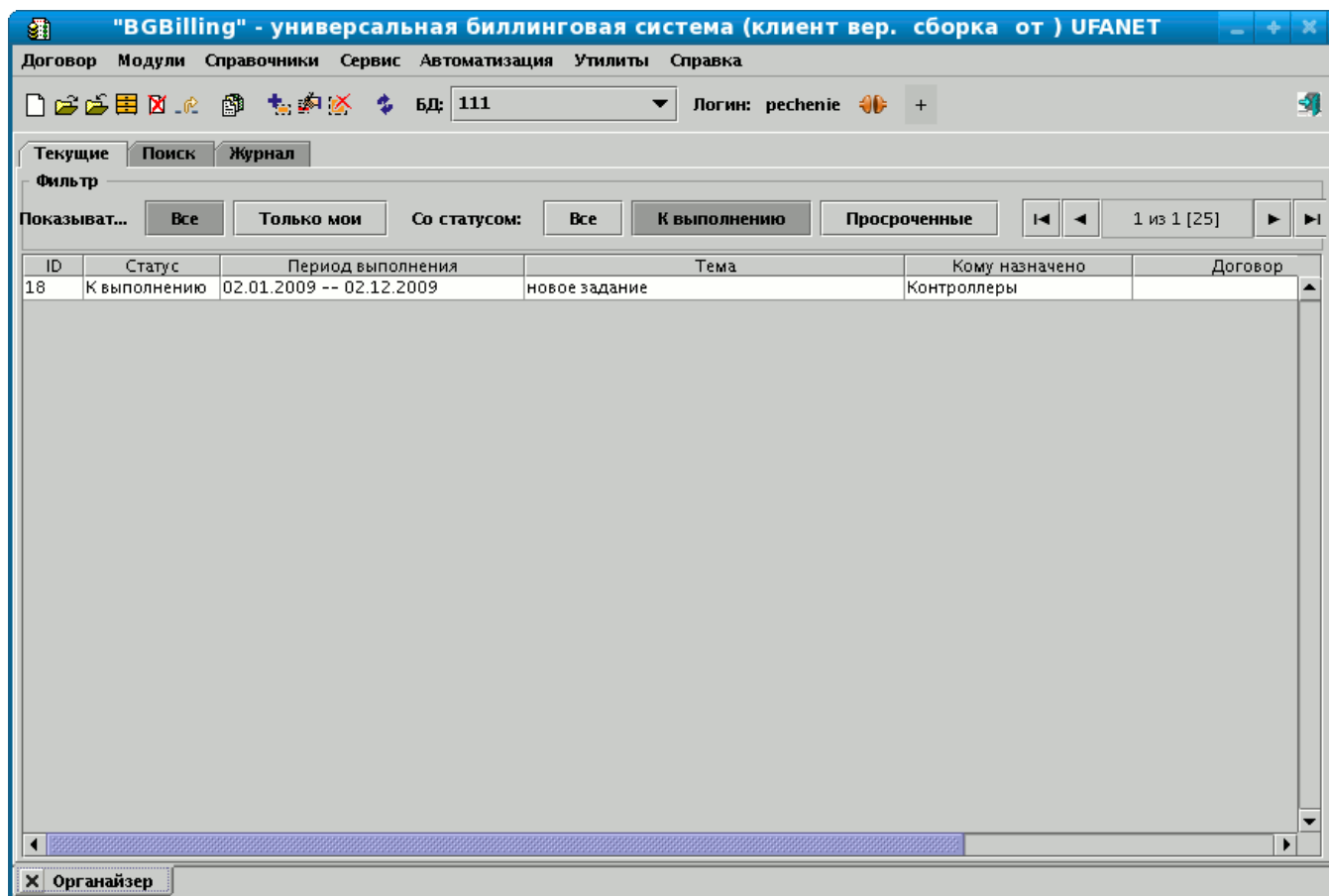
Статус выполнения

Здесь **Тема** - краткое описание задания, **Комментарий** - полное описание задания, **Договор** - связанный с заданием договор (для добавление договора в это поле откройте вкладку этого договора, а затем нажмите на кнопку >>>), **Кому назначено** - пользователь или группа, которой (которому) назначено задание, **Период** - соответственно, период выполнения, и **Статус** - это метка о статусе договора.

После заполнения необходимых полей нажмите на кнопку **OK**. Создать сразу выполненное задание нельзя.

Просмотр текущих заданий

Для просмотра текущих заданий необходимо зайти в меню **Плагины->Организер**, либо, в случае имеющихся к выполнению заданий, нажать на кнопку текущих заданий в статусной панели клиента.



Выбирая соответствующие условия показа заданий, можно просматривать как свои, так и чужие задания со статусом к выполнению, просроченные, либо и те, и другие.

Переход к договору, с которым связано задание, осуществляется выбором соответствующей опции в меню, открываемом щелчком правой клавишей мыши по заданию.

Поиск заданий

Для поиска заданий по различным критериям необходимо перейти во вкладку **Поиск**. Поиск можно осуществлять по периоду, назначенному для выполнения задания, по времени непосредственного выполнения задания, по ID, по тексту внутри темы или развернутого сообщения, по пользователям (группам), которым предписано задание, а также по признаку выполненности и по пользователям, выполнившим задание.

The screenshot shows the "VBilling" software interface. The main window title is "VBilling" - универсальная биллинговая система (клиент вер. сборка от) UFANET. The interface is divided into several sections:

- Menu Bar:** Договор, Модули, Справочники, Сервис, Автоматизация, Утилиты, Справка.
- Toolbar:** Includes icons for home, search, and other functions. It also shows "БД: 111" and "Логин: reshenie".
- Navigation:** Tabs for "Текущие", "Поиск" (selected), and "Журнал".
- Calendar:** A tree view on the left showing months from April to December, with a selection for the year 2010.
- Search Form:** Contains fields for "ID:" (with a red '1' pointing to it), "Содержит:" (with a red '2' pointing to it), "Кому назначено:" (with a red '3' pointing to it), "Пользователю:" (with a red '4' pointing to it), "или группе:" (with a red '3' pointing to it), "Статус:" (with a red '4' pointing to it), "Период:" (with a red '6' pointing to it), and "с:" (with a red '7' pointing to it).
- Buttons:** "Фильтр", "Поиск", "Сброс".
- Results:** A table showing search results with columns: ID, Статус, Период выполнения, Выполнил, Когда, Тема, and Кому назн. The table contains three rows of data.

1 - ID задания, 2 - текст внутри темы или комментария к заданию, 3 - кому задание было назначено, 4 - признак выполненности, 5 - пользователь, выполнивший задание, 6 - период, когда задание было выполнено, 7 - период, на который было назначено выполнение задания. 8 - календарное дерево, выбор месяца, года или всех годов автоматически заполняет поле 7.

Выполнение заданий

Ответственность за фактическое исполнение заданий лежит на самих пользователях, система только лишь распределяет задания, но не контролирует реальные его цель и статус. Для того, чтобы пометить задание как выполненное, необходимо дважды щёлкнуть на нем в списке заданий (либо выбрать его в таблице и нажать на кнопку **Редактировать**).

Редактор

Тема

Комментарий

Договор

>>>

Кому назначено:

Пользователю:

или группе:

Период

с по

Статус выполнения

В открывшемся редакторе необходимо отметить задание как **Выполненное**, нажав соответствующую кнопку, и подтвердить изменения, нажав кнопку **ОК**.

Также можно присвоить выполнение задания себе (например, если задание выдано не конкретному человеку, а группе пользователей, то можно взять его выполнение на себя). Для этого необходимо нажать кнопку **Назначить себе** и подтвердить изменения, нажав кнопку **ОК**.

Журнал

Для просмотра журнала истории заданий необходимо открыть вкладку Журнал. Аналогично поиску заданий, здесь можно воспользоваться календарным деревом для поиска записей в журнале по периоду. Также можно искать записи в журнале для конкретного пользователя.

The screenshot shows the "BGBilling" software interface. The title bar reads "BGBilling" - универсальная биллинговая система (клиент вер. сборка от) UFANET. The menu bar includes "Договор", "Модули", "Справочники", "Сервис", "Автоматизация", "Утилиты", and "Справка". The toolbar contains various icons and a dropdown menu for "БД:" with the value "111", and a "Логин:" field with the value "pechenie".

The "Журнал" tab is active. On the left, there is a "Календарь" (Calendar) tree view showing a hierarchy of "Годы" (Years) with sub-items for "2009" and "2010", each containing months from "Январь" (January) to "Декабрь" (December). The "2010" folder is selected.

The main area is titled "Поиск" (Search) and contains a search form with the following fields and buttons:

- "Пользователь:" (User): A dropdown menu with a dashed line indicating no selection.
- "с" (from): A date field containing "01.01.2010".
- "по" (to): A date field containing "31.12.2010".
- "Поиск" (Search) button.
- "Сброс" (Reset) button.

Below the search form, it displays "Всего записей: 0" (Total records: 0) and "1 из 1 [25]" (1 of 1 [25]).

The main table has the following columns: "ID", "ID записи", "Кто", "Действие", and "Дата". The table is currently empty.

At the bottom left, there is a tab labeled "Организатор" (Organizer).

Плагин SBPilot

Назначение плагина

Плагин предназначен для интеграции с терминалами Сбербанка через систему/утилиты sb_pilot для приёма оплаты через банковские карты на рабочих местах кассиров.

Структура и настройка плагина

Есть два режима работы терминала в плагине - на клиентской стороне и через сервер печати cashcheck.

Клиентский режим

В этом варианте плагин по большей части работает на клиентской стороне. Взаимодействие с терминалами производится через утилиты и настройки Сбербанка на компьютере, где установлен клиент биллинга (рабочее место кассира). Серверная часть плагина используется только для ведения истории платежей. Работа осуществляется через обращение к утилите через командную строку. Путь до неё на текущем локальном компьютере и до файлов, которые она генерирует, прописываются в настройках. Основная конфигурация производится в файле настройки клиента биллинга (файл `client_ru_RU.properties`):

```
# sb_pilot ( linux windows),
# /
#sbpilot.path.bin=/home/bill/sb-pilot/sb_pilot
#sbpilot.path.bin=C:\sb-pilot\SB_PILOT.EXE
sbpilot.path.bin=/home/bill/sb-pilot/dowindow.sh
# e p, (. sb_pilot)
sbpilot.path.e=/home/bill/sb-pilot/e
sbpilot.path.p=/home/bill/sb-pilot/cheque.txt
# sbpilot.path.e , e - sbkernel.log
# sbkernel.log
sbpilot.path.e2=/home/bill/sb-pilot/sbkernel%YY%MM.log
# ,
sbpilot.cashcheckserver.connector=127.0.0.1:9876
#
sbpilot.cashcheckserver.oppass=30
#
sbpilot.cashcheckserver.endcut=1
# ( hex) (, PrinterEnd=01 PrinterEnd=010D0A)
# .. , sbpilot.cashcheckserver.endcut
sbpilot.cashcheckserver.charcut=0D0A01
# , " "
sbpilot.abort.timeout=60
# , (!) ,
#sbpilot.success.close.timeout=60
```

Для распечатки чека используется сервер печати cashcheck. Необходимо установить и настроить его согласно инструкции, приведённой в соответствующем разделе.

Необходим лишь сервер печати — приложение BGCashcheckServer. Сам плагин cashcheck устанавливать не нужно, если вы не собираетесь его использовать.

Режим сервера печати

В этом варианте пинпад настраивается не в клиенте, а в сервере печати cashcheck со специальным драйвером `ru.bitel.frk.driver.sbpilot.Driver`, все настройки в точности такие же, как для локального варианта (см. в конфиге сервера печати). Работа с пинпадом ведётся через сервер биллинга, который через сервер печати включает терминал, показывает диалог и т.д. Для доступа к серверу печати с драйвером sbpilot указывается коннектор в конфиге плагина:

```
# sbpilot ( )

cashserver.1.title=
cashserver.1.connector=127.0.0.1:9876

cashserver.2.title=
cashserver.2.connector=127.0.0.1:9876
```

Обратите внимание, что в конфиге сервера печати (тот что с драйвером **sbpilot**) указывается коннектор на другой сервер печати (на другой машине или на другом порту), который печатает чеки (как и в клиентском варианте).

Поддерживается несколько терминалов, с выбором налету нужного в интерфейсе, выбор запоминается для каждого юзера отдельно (в серверной сессии).

Настройка утилиты `sb_pilot`

Настройка утилиты производится сотрудниками Сбербанка и в данном руководстве не рассматривается. Помимо настройки связи с банком необходимо уточнить в какие места и под каким именем сохраняются выходные файлы (см. настройку в клиенте). Также нужно попросить настроить ширину генерируемого чека в соответствии с шириной ленты в используемом вами принтере чека. Для Linux имена файлов обычно **e** и **cheque.txt**, для windows — **e** и **p**. Также есть версия утилиты, которая вместо файла **e** пишет лог операций в файл **sbkernel.log**.

Донастройка утилиты для биллинга заключается в том, чтобы вместить в один скрипт вызов утилиты с каким-то настройками окружения под каждую систему, и передать в утилиту переданные этому скрипту параметры (не менее четырёх). Скрипт должен запустить утилиту с параметрами, отобразить на экране в приличном виде, **ожидать завершения утилиты, а после её завершения - закрыться (передать управление)**.

Это распространённая ошибка: утилита корректно не запускается через этот скрипт, либо запускается и сразу закрывается, либо запускается но пишет файлы не туда, либо что-то ещё подобное. Надо отдельно проверить запуск какой-либо командой с параметрами в виде вызова именно этого скрипта и убедиться, что она ведёт себя как надо.

Ниже приведены примеры (не стоит рассматривать их как конечные решения).

Донастройка в Linux

Используется консольная linux-версия утилиты. Для справки: в каталоге этой версии программы находятся файлы `sb_pilot`, `config`, `upnixmn.out` и прочие.

Необходимость донастройки заключается в том, что утилита работает в архаичной однобайтовой кодировке `koI8-r`. Потому для корректного отображения окна с приглашением ввода карты, меню и т.д. потребуется обернуть вызов утилиты в скрипт, который установит текущую локаль, шрифты, а также, при желании, размер окна, расположение, заголовок и т.п.. Предпочтительнее использовать именно скрипт вместо прописывания длинной строки запуска в параметре `sbpilot.path.bin`.

Скрипт можно написать совершенно любой под любой эмулятор терминала. Суть скрипта заключается в том, чтобы корректно отобразить на экране утилиту `sb_pilot`. Окно должно ожидать завершения каких-либо действий, а после завершения работы утилиты — закрываться. Также скрипт должен передавать не менее четырёх параметров утилите `sb_pilot`, т.е. необходимы переменные `$1 $2 $3 $4` в командной строке утилиты.

В линуксе при использовании `gnome-terminal` скрипт `dowindow.sh` может иметь, например, такой вид:

```
#!/bin/sh

export LANG=ru_RU.KOI8-R; gnome-terminal --disable-factory -e "/home/bill/sb-pilot/sb_pilot $1 $2 $3 $4" --hide-menubar -t "SB-Pilot" --working-directory=. --profile=sbpilot
```

В этом примере в профиле с названием `sbpilot` (см. настройки терминала) можно указать любые размеры и цвета окна терминала. Новый `gnome-terminal` настраивается по-другому, есть проблемы с ожиданием запущенного в нём процесса.

При использовании эмулятора терминала `xterm`:

```
#!/bin/sh

export LANG=ru_RU.KOI8-R; export LC_ALL=ru_RU.KOI8-R; xterm -T SbPilot -fn -misc-*-*-*-*-*13-*-*-*-*-*-koI8-r -e "/home/bill/sb-pilot/sb_pilot $1 $2 $3 $4"
```

В данном случае для отображения кириллических символов, псевдографики и т. д. в `xterm` в системе должны быть установлены соответствующие шрифты `xorg-x11-fonts-misc`, например, или/и `xorg-x11-fonts-cyrillic` или, возможно, пакеты с другим названием для вашей системы. Установка и настройка отображения кириллических `koI8-r` шрифтов в вашей системе выходит за рамки этого руководства.

Под эмулятор терминала `kde` не получилось написать корректный скрипт, выполняющий вышеобозначенные требования (при вызове из стороннего приложения окно закрывается сразу).

Донастройка в Windows

Используется консольная win32-версия (иногда она называется у сотрудников сбербанка почему-то DOS). Работа происходит через командную строку, аналогичную командной строке linux-версии. Для справки: в каталоге этой версии программы находятся файлы SB_PILOT.EXE, pinpad.ini, updoscf.exe и прочие. Некоторая настройка также необходима для указания рабочих каталогов. Файл **dowindow.bat** может выглядеть так:

```
start /d c:\dos /wait c:\dos\sb_pilot.exe %1 %2 %3 %4
```

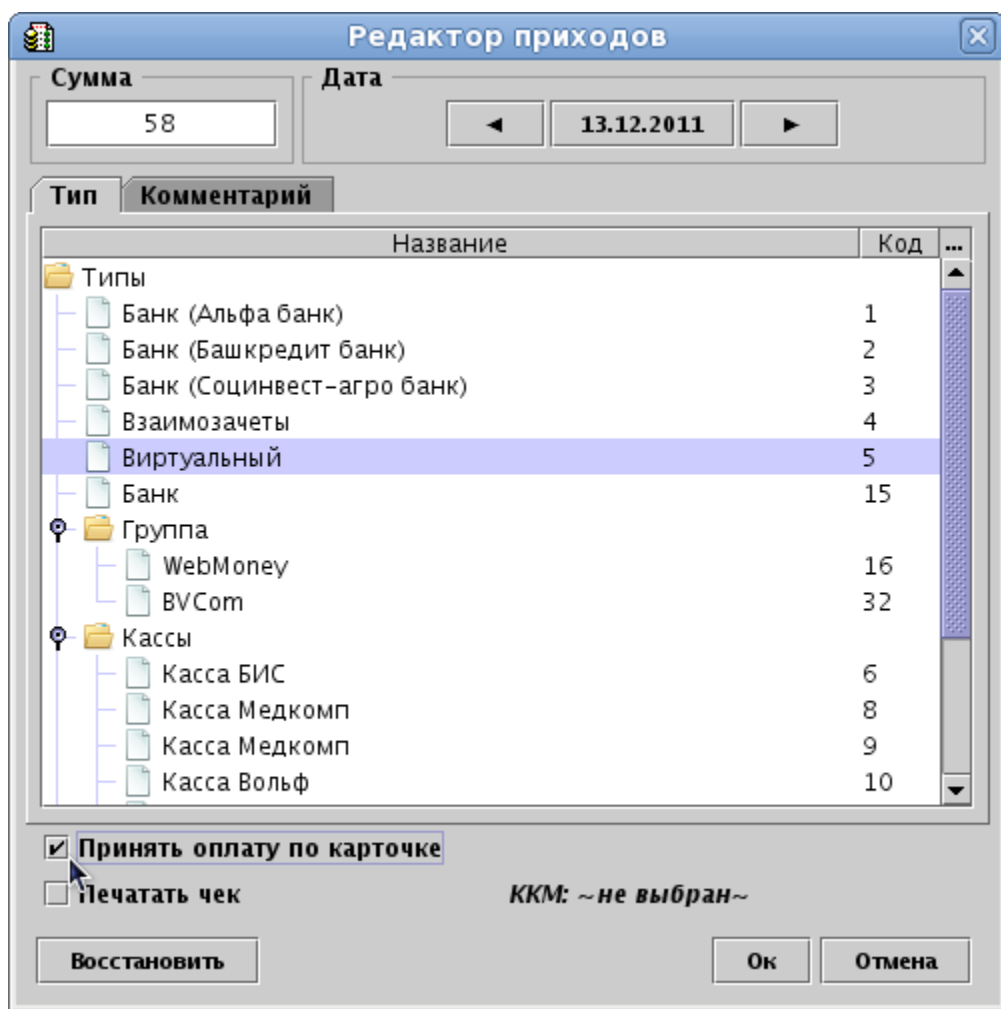
, либо каким-то другим образом. При этом обратите внимание на корректное указание рабочего каталога утилиты, без этого она не будет корректно работать (ищет свои файлы около себя) и файлы выходные тоже потеряются (пишет тоже в рабочий каталог). Кодировка выходных файлов — cp866.

После этого не забудьте прописать в файле **client.properties** полные пути до файлов **dowindow.bat**, **e**, **p**.

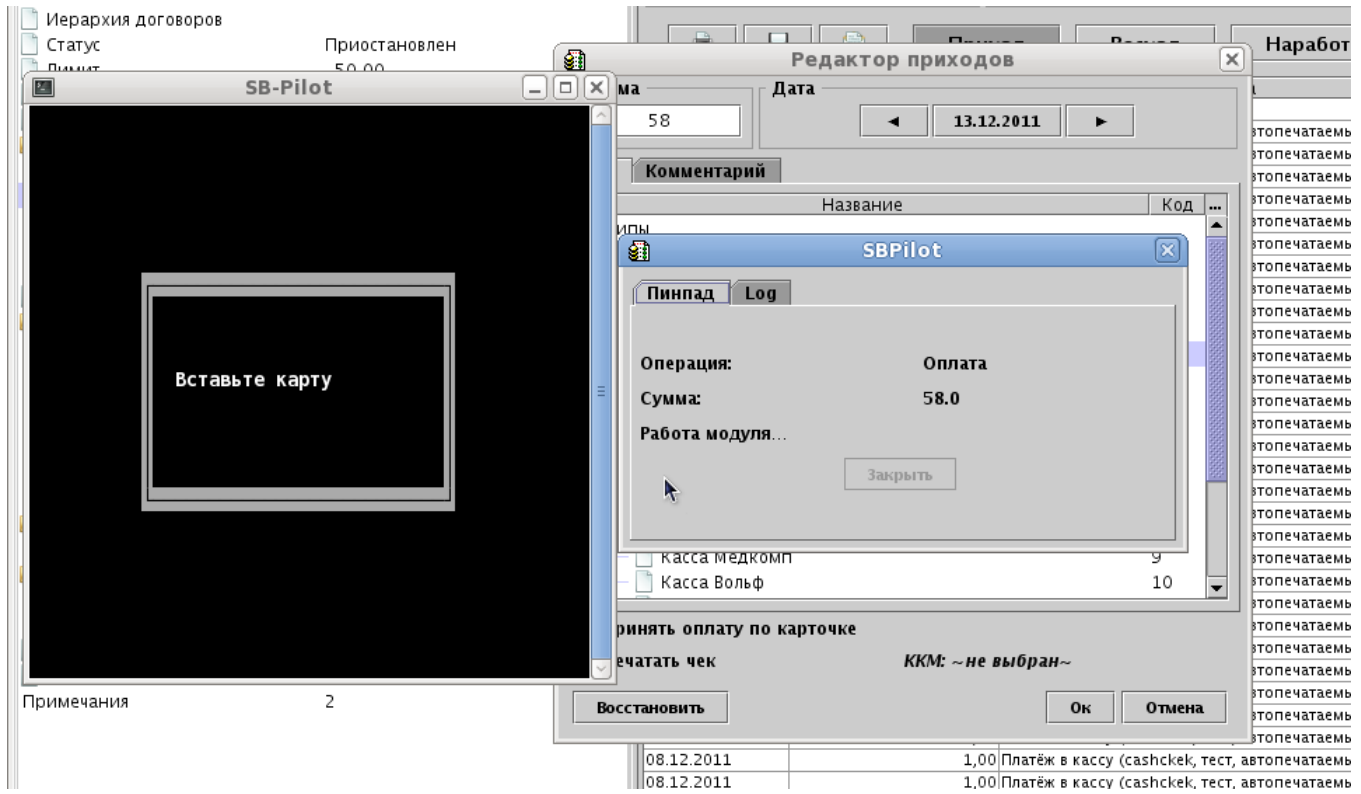
Имеются данные, что конкретно эта утилита не работает корректно в 64-битной версии windows.

Использование плагина

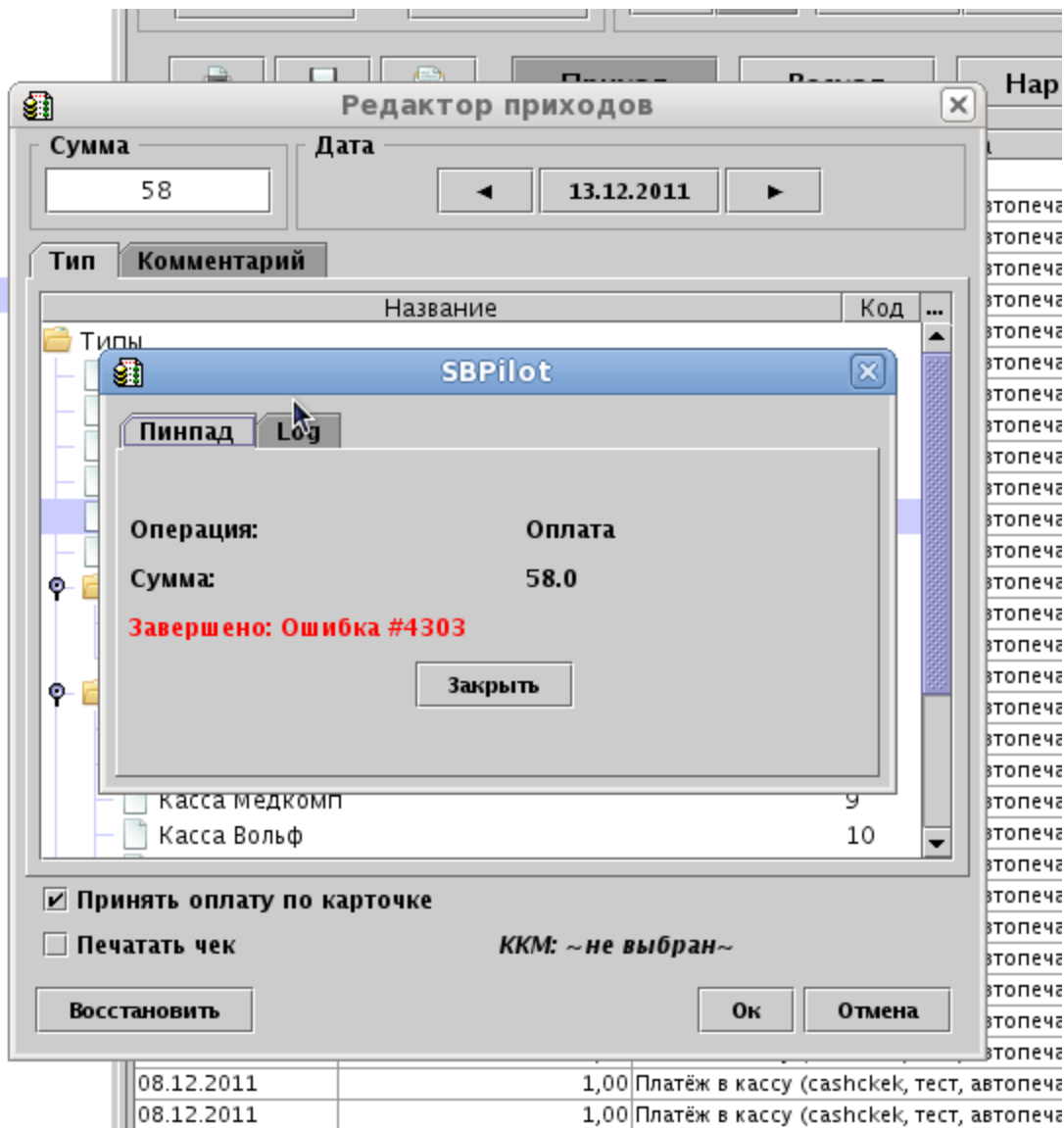
При установке и активации плагина в диалоге добавления платежа появляется галка "принять оплату по карте".



При попытке добавления платежа с установленной галочкой активируется диалог работы с утилитой и сама утилита.



После завершения работы утилиты успешно или же с ошибкой диалог можно закрыть.



На второй вкладке "Log" можно увидеть более подробную информацию о взаимодействии с утилитой. Если работа была завершена с ошибкой, то платёж не совершается и мы по-прежнему имеем дело с диалогом добавления платежа. Если оплата проведена успешно, то платёж совершается, диалог закрывается, в историю платежей заносится запись. В истории платежей (**Плагины->SBPilot**) также имеется возможность совершения дополнительных действий - отмены, повторы, некоторые отчёты итп.

Код	Договор	Комментарий договора	Тип платежа	Время	Комментарий...	Сумма	Номер...	Дата к...	Код а...	Внутр...	Тип к...
450...	dim on	тестовый договор	Платёж в кассу (cas...	25.04.2011 18:1...	платёж по карте	1.00					
450...	dim on	тестовый договор	Виртуальный	10.05.2011 00:0...		1.00					

Также обратите внимание на одну особенность: если пользоваться распределением средств на зависимых договорах (т.е. одним платежом инициировать занесение в разные договоры), то в логе припишется только к одному из платежей, а оплата и чек пройдёт на всю сумму изначального платежа.

Плагин КЛАДР

Назначение плагина

Плагин позволяет синхронизировать базу улиц с Классификатором адресов России (КЛАДР), предоставляемым Федеральной Налоговой Службой. https://www.gnivc.ru/technical_support/classifiers_reference/kladr/

Установка и настройка

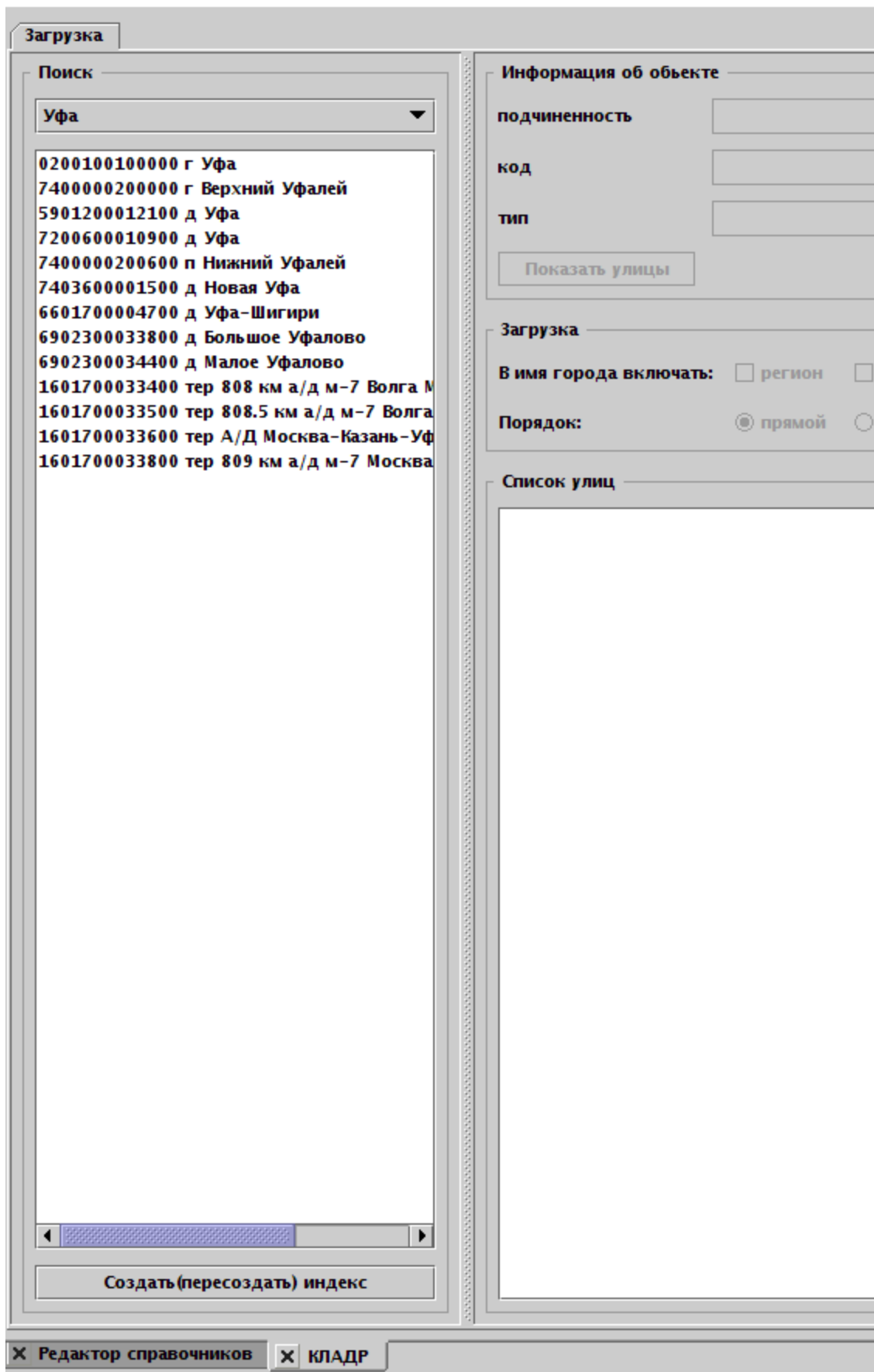
Плагин устанавливается с помощью утилиты **bg_installer**. После установки плагин должен быть активирован в меню **Плагины=>Настройки плагинов** и у него должна быть установлена конфигурация.

В конфигурации должно быть определено место на диске, где у клиента храниться база данных КЛАДР

```
driver.class.name=com.hxtt.sql.dbf.DBFDriver
db.url=jdbc:DBF:///home/billing/KLADR/base
harset=Cp866
```

Использование плагина

После установки плагина в клиенте появиться меню **Плагины=>Загрузка КЛАДР**



Кнопка **Создать (пересоздать) индекс** используется для создания индексов базы КЛАДР при первом обращении к базе, либо при обновлении базы КЛАДР.

Для загрузки нужного Вам населённого пункта используется поиск по названию. Перед загрузкой необходимо создать населённый пункт в справочнике городов. После завершения поиска выводятся населённые пункты, совпадающие по названию с выбранным городом. При выборе населённого пункта в списке слева в правой верхней части отображаются его данные. Кнопка **Показать улицы** позволяет увидеть список улиц.

При загрузке настраивается какие части включать в название города. В частности, можно включить регион и район в прямом или обратном порядке. По умолчанию сокращение "ул." в названии улицы игнорируется, но можно отменить данное ограничение. При нажатии кнопки **Обновить базу** начнётся обновление базы.

Информация об объекте

подчиненность: Респ Башкортостан->р-н Уфимский->г Уфа

код: 0200100100000

тип: г название: Уфа

Загрузка

В имя города включать: регион район Игнорировать сокращение в улицах:

Порядок: прямой обратный

Выбор названия

Старое название: Демская

02001001000024900 пл Демская 450018
02001001000025000 ул Демская 450018

При обновлении базы происходит синхронизация названий улиц с базой КЛАДР. Для улиц, не имеющих синхронизации по коду, предлагается сделать выбор вручную. Есть возможность выбрать из списка с какой из похожих улиц ассоциировать улицу из КЛАДР, также есть возможность пропустить улицу (оставить в базе как есть) и удалить улицу из базы.

Плагин ФИАС

Инструкция по обновлению

Инструкция по обновлению с версии 7.1 на версию 7.2

1. Создайте резервную копию БД, и всех приложений биллинга.
2. Остановите сервер биллинга, планировщик и загрузчик логов, переименуйте папку со старым BGBillingServer (например в BGBillingServer_old).
3. Загрузите последнюю версию BGBillingServer 7.2 с сайта. Произведите настройку согласно документации:
[Особенности установки под различные платформы](#)
!!! Пропустите момент с заливкой дампа базы !!!
[Установка сервера биллинга \(для версий до 8.0\(вкл.\)\)](#)

Настройте data/data.properties согласно старому файлу.

Правьте только IP адреса, логины, пароли, значения переменных в соответствии со старым файлом. Не забудьте проверить context.path!
Перетрите тестовую лицензию в файле lic.properties вашей коммерческой лицензией.
Перенесите, если есть files, archive, .keystore.
Если в старой версии использовались динамический код, необходимо скопировать содержимое каталога BGBillingServer/dyn из старой версии в новую.

В файле setenv.sh пропишите JAVA_HOME.

4. Загрузите ActiveMQ 5.14.4 с официального сайта <http://activemq.apache.org/activemq-5144-release.html> или установите из репозитория, если данная версия (>=5.14) в нем есть. Установите и настройте согласно [инструкции](#).
5. Загрузите **update_7.2.zip** с <https://bgbilling.ru/download/7.27.2>, сохраните его в папку BGBillingServer и установите инсталлятором.

```
./bg_installer.sh update_7.2.zip!
```

6. Установите обновления всех используемых вами модулей.

Для этого запустите
для Linux : ./bg_installer.sh update
для Windows: bg_installer.bat update

Как работает автоматическая система обновлений описано в документации.

7. Загрузите новую версию клиента биллинга и настройте ее(пропишите JAVA_HOME - jdk 1.8 и сделайте файл bgbilling.sh запускаемым (для Linux)).
8. Запустите сервер биллинга(проверьте по логам все ли стартануло нормально), подключитесь к нему клиентом.
В окне авторизации установите опцию "загружать обновления с этого сервера".
9. Установите обновления всех приложений (BGInetAccess, BGInetAccounting, BGDaiusDialup и т.п). Обновлять нужно через штатную систему update.sh
[Установка обновлений биллинга](#)
Еще один момент - во всех приложениях нужно вручную почистить папки со сторонними библиотеками (lib/ext) и скопировать туда библиотеки из соответствующего архива на сайте.
10. Для приложений BGDHCP_IPN, которое не поддерживают автоматическую систему обновлений, скачайте и распакуйте новую версию, произведя аналогичные старым настройки .properties файлов и .sh-файлов .

Инструкция по обновлению с версий 3.5-7.1 до версии 7.2

Обязательно раскрыть и выполнить те пункты, которые удовлетворяют по условию той версии, с которой вы обновляетесь.

1. Создайте резервную копию БД, и всех приложений биллинга.
2. BGBilling версии с 6.2 должен быть запущен под JAVA JDK 1.8. Это обязательное условие, он скомпилирован под jdk 1.8, а под jdk 1.6-1.7 работать вообще не будет (даже не запустится). Установите JDK 1.8.

```
1) , .
1) MQ .
   http://bgbilling.ru/v5.2/doc/ch01s05s03.html
2) "SQL " mysql :
   ALTER TABLE `installed_modules` ADD COLUMN `uninstall` LONGTEXT;
3) BGBilling c 5.1 JDK JRE! JDK

1) mysql root
   mysql -uroot -p bgbilling
:
   GRANT ALL PRIVILEGES ON bgbilling.* TO 'bill'@'localhost' IDENTIFIED BY 'bgbilling';
   GRANT ALL PRIVILEGES ON bgbilling.* TO 'bill'@'%' IDENTIFIED BY 'bgbilling';
   ALTER, .

2) IPN -
   manad.

1) "=>SQL " mysql :
   ALTER TABLE installed_modules ADD COLUMN `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY;

2) MPS 4.3 "SQL " mysql :
   ALTER TABLE mps_payment_$mid_$yyyy$mm ADD `type` INT NOT NULL DEFAULT 0
   ALTER TABLE mps_payment_$mid_$yyyy$mm ADD rawNumber VARCHAR(20) NOT NULL DEFAULT ''
   $mid - MPS, $yyyy - $mm - , .
3) "=>SQL " mysql :
   SHOW COLUMNS FROM contract_status;
```

Field	Type	Null	Key	Default	Extra
cid	int(10) unsigned	NO		0	
mid	int(11)	NO		0	
status	tinyint(3) unsigned	NO		0	

```
, :
   DROP TABLE contract_status;
```

```
4) "=>SQL " mysql :
   describe contract_status_log;
```

Field	Type	Null	Key	Default	Extra
dt	datetime	NO		0000-00-00 00:00:00	
cid	int(10) unsigned	NO		0	
mid	int(11)	NO		0	
action	tinyint(4)	NO		0	
uid	int(11)	NO		0	
comment	char(200)	NO			

```
, :
   DROP TABLE contract_status_log;
```

```
1) "=>SQL " mysql :
   ALTER TABLE `installed_modules` ADD COLUMN `uninstall` LONGTEXT;
```

```
1) , contract.status.list, contract.status.no.manual.set: http://bgbilling.ru/v5.2/doc/ch01s18s09.html
   , 5.1 : http://bgbilling.ru/v5.2/doc/ch01s10s01.html
```

```
2) : http://bgbilling.ru/v5.2/doc/ch01s18s09.html#d0e3685
   credit.contract.active.status credit.contract.open.by.payment.status.
```

```
3) :
   RentSoft: http://bgbilling.ru/v5.2/doc/ch06s03.html
   VoiceIp: http://bgbilling.ru/v5.2/doc/ch31s03.html
   DialUp: http://bgbilling.ru/v5.2/doc/ch15s03.html
   IPN: http://bgbilling.ru/v5.2/doc/ch19s02.html
```


E-Mail: <http://bgbilling.ru/v5.2/doc/ch17s02s02.html>
CerberCrypt: <http://bgbilling.ru/v5.2/doc/ch13s02.html>
Phone: <http://bgbilling.ru/v5.2/doc/ch24s02.html>

```
contract.status.active.codes contract.status.suspend.codes ( ).  
  
dbinfo.allow.to.change.status ,  
contract.status.no.manual.set ( - ).
```

4) NPay contract.status.suspend.codes: <http://bgbilling.ru/v5.2/doc/ch22s02.html>

```
service.no.suspend, service.no.suspend.in.suspend, service.no.suspend.in.close, wrap.service  
( , ).
```

```
,  
:  
http://bgbilling.ru/v5.2/doc/ch21s07.html
```

5) >=5.2 InnoDB. " " .
MyISam, :

http://wiki.bgbilling.ru/index.php/%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D0%B8%D0%B8_%D0%BF%D0%BE_%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B5_MySQL

```
SQL SHOW TABLE STATUS, Engine. ,  
... ..
```

6) " " " " , .

7) activemq ftp://ftp.bgbilling.ru/pub/bgbilling/activemq/linux/apache-activemq-5.4.2/conf/activemq.xml

1) 7.0 lic.properties. 7.0.

3. Выполните инструкцию по обновлению с версии 7.1 начиная с пункта 2 - до конца:

[Инструкция по обновлению с версии 7.1 на версию 7.2](#)

1) 5.2 FO . ()

```
, , - ,  
FO-XSL. :  
)  
) fo-xsl ;  
) fo-, ( ! ) ,  
fo-xsl .  
: http://forum.bgbilling.ru/viewtopic.php?t=4942  
: ftp://ftp.bgbilling.ru/pub/bgbilling/util/fo\_validator.zip
```

2) cachcheck: . , .

```
- ' ' ' ' .  
connector.  
: http://forum.bgbilling.ru/viewtopic.php?t=6194
```

3) " " " " , " "
(5.2 " "), " "
() .

4) WiFi- Dilaup , , .

```
:  
) dialup_wifi.properties,  
) billing.server.dialup.mid billing.server.moduleId, ,  
) activemq  
)  
) update.sh,  
) setenv.sh JAVA_HOME,  
) .
```

5) CRM. 5.2. , register_task_log
, sql , register_task_log

```
insert into register_task_log select id, open_dt, open_uid, 0 from register_task;  
insert into register_task_log select id, accept_dt, accept_uid, 1 from register_task;  
insert into register_task_log select id, close_dt, close_uid, 2 from register_task;
```

6) Slave , URL .
<http://bgbilling.ru/v5.2/doc/ch01s24s02.html>

1) (Windows - cmd Far) BGBillingServer :

```

    LINUX: < Java>/bin/java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* bitel.billing.server.contract.ServiceNormalizer
    Windows: < Java>\bin\java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* bitel.billing.server.contract.ServiceNormalizer
    FreeBSD: ( ):java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp "../lib/app/*:
/lib/ext/*" bitel.billing.server.contract.ServiceNormalizer

```

NPay

```

2) ( Windows - cmd Far) BGBillingServer :
    LINUX: < Java>/bin/java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* bitel.billing.server.contract.ServicePatternNormalizer
    Windows: < Java>\bin\java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* bitel.billing.server.contract.ServicePatternNormalizer
    FreeBSD( ):java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp "../lib/app/*:
/lib/ext/*" bitel.billing.server.contract.ServicePatternNormalizer

```

```

1)
#
findmode.0.title= User-Name=LOGIN
findmode.0.value=User-Name=LOGIN
findmode.1.title= User-Name=ALIAS
findmode.1.value=User-Name=ALIAS
findmode.2.title= Calling-Station-Id=ALIAS
findmode.2.value=Calling-Station-Id=ALIAS
#
find.order=0,1,2

```

User-Name=LOGIN

```

2) !!!
NetFlow IPN

```

, NetFlow

dataloader NetFlow ,

() FTP

NetFlow

```

1) voiceip -

```

```

1) , -10000000.

```

```

2) "=>SQL " mysql :
    UPDATE contract SET closesumma=-1000000 WHERE mode=0

```

```

3) dialup "=>SQL " mysql :
    ALTER TABLE session_detail_<mid>_<yyyyMM> ALTER COLUMN summa SET DEFAULT 0;
<mid> -
<yyyyMM> - , 200801
    Unknown column 'summa' in 'session_detail_...'

```

```

1) IPN.
    4.5 Manad, Cisco, Mikrotik, BGRadiusIPN .

```

```

1.1.
    .. manand_128, manand_256, cisco..

```

```

1.2. Manad FreeBSD <LOOP></LOOP>. . :
    pipe {P0} config bw 512000
    pipe {P1} config bw 512000
    add {N1} pipe {P0} all from any to {A}
    add {N1} pipe {P1} all from {A} to any

```

```

:
<LOOP>
    pipe {P0} config bw 512000
    pipe {P1} config bw 512000
    add {N1} pipe {P0} all from any to {A}
    add {N1} pipe {P1} all from {A} to any
</LOOP>

1.3.   Manad, Cisco, Mikrotik . 4.5( ).
Mikrotik , <OPEN></OPEN> <CLOSE></CLOSE> [OPEN][/OPEN] [CLOSE][/CLOSE] .
[CLOSE][/CLOSE] [DELETE][/DELETE].

1.4.   Manad, Cisco, Mikrotik, BGRadiusIPN .
" ".

1.5.   Manad, isco, Mikrotik. . , Manad :
256:
    <LOOP>
        pipe {P0} config bw 256000
        pipe {P1} config bw 256000
        add {N2} pipe {P0} all from any to {A}
        add {N2} pipe {P1} all from {A} to any
    </LOOP>
512:
    <LOOP>
        pipe {P0} config bw 512000
        pipe {P1} config bw 512000
        add {N1} pipe {P0} all from any to {A}
        add {N1} pipe {P1} all from {A} to any
    </LOOP>
:
[DEFAULT]
    <LOOP>
        pipe {P0} config bw ${speed}
        pipe {P1} config bw ${speed}
        add {N2} pipe {P0} all from any to {A}
        add {N2} pipe {P1} all from {A} to any
    </LOOP>
[/DEFAULT]

. 256: speed=256000; 512: speed=512000.

2)   Web ,
:
http://bgbilling.ru/v4.5/doc/ch01s21s08.html

3)   , , .
:
"=>SQL " mysql :
    UPDATE address_street SET cityid=<id >
    UPDATE address_area SET cityid=<id >
    UPDATE address_quarter SET cityid=<id >

<id > Ctrl+i.

1)   ( ) 9.2 . .

2)   3 .

3)   DHCP_IPN. DHCP (Dlink Cisco2).
4.6 dhcp.82.key.position dhcp.82.key.option.code.
Dlink:
dhcp.82.key.option.code=1
dhcp.82.key.position=5
Cisco2:
dhcp.82.key.option.code=1
dhcp.82.key.position=2

4)   IPN. Zyxel, <LOOP></LOOP> <LOOP_PORT></LOOP_PORT>.

5)   Card. , ( ).

6)   IPN. "SQL " mysql :
ALTER TABLE ipn_contract_data_${mid}_${yyyy}$mm MODIFY COLUMN cost FLOAT(10,5) NOT NULL DEFAULT 0;
${mid} - IPN, ${yyyy} - $mm - ,
( ).

7)   IPN. Cisco2 Zyxel ,
:
gate_manager.class=bitel.billing.server.ipn.vlan.CiscoVlanGateWorker
:
gate_manager.class=bitel.billing.server.ipn.vlan.CiscoVlanParentGateWorker

```

```

1) IPN.
   <LOOP_NET> <LOOP_NET_MASK>,
      a) LOOP_NET {NET} {IP}/{MASK_BIT}.
         ) LOOP_NET_MASK {NET_MASK} {IP}/{MASK}.

2) IPN.
   port.range,      gate.range.

3) IPN.      vlan      range,
   vlan.range.

4) . (
   *.java reports),
   public void fillReport( con, filter, bitel.billing.server.reports.BGCSVReport.ReportResult result, pageSize,
   pageIndex )

   public void fillReport( con, filter, bitel.billing.server.reports.BGCSVReport.ReportResult result )
       pageSize, pageIndex( ).

5) .
   ( Windows - cmd Far) BGBillingServer :
   LINUX: < Java>/bin/java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* ru.bitel.bgbilling.server.util.ContractParameterPhoneConvertor
   Windows: < Java>\bin\java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* ru.bitel.bgbilling.server.util.ContractParameterPhoneConvertor

6) Dr.Web.
   ( Windows - cmd Far) BGBillingServer :
   LINUX: < Java>/bin/java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* ru.bitel.bgbilling.modules.drweb.server.convert.Convert $mid
   Windows: < Java>\bin\java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib
/app/*:../lib/ext/* ru.bitel.bgbilling.modules.drweb.server.convert.Convert $mid

7) .
   " " " "
   " ( )" " (-)" ,
   :
http://bgbilling.ru/v5.1/doc_op/billing.html#d0e18884.

8) , client.gui.contract.tree.order, "face". :
client.gui.contract.tree.order=parameters objects hierarchy status limit mode face balance tariff modules groups
web tariffGroup script addAction memo

9) . , . 2 :
, , , ( group.mask), ,
: (cmd Windows; xterm, konsole ( ) - Linux),
BGBillingServer :
LINUX: < Java>/bin/java -Xmx256m -cp ../lib/app/*:../lib/ext/* bitel.billing.server.gorod.utils.
Converter_firm2tag <_> <mid>
Windows: < Java>\bin\java -Xmx256m -cp ../lib/app/*:../lib/ext/* bitel.billing.server.gorod.utils.
Converter_firm2tag <_> <mid>
FreeBSD( ):java -Xmx256m -cp " ../lib/app/*:../lib/ext/*" bitel.billing.server.gorod.utils.
Converter_firm2tag <_> <mid>
<_> - ( ), ( group.mask );
<mid> - .

10) ( " "):

http://bgbilling.ru/v5.1/doc/ch01s25s02.html
1) contract.xslt, contract.csv.xslt , ,
contract.balance.xslt=contract_balance_print.xsl
contract.balance.csv.xslt=contract_balance_print_csv.xsl
[!] contract.balance.xslt, contract.balance.csv.xslt, contract.balance.detail.csv.xslt,
contract.balance.detail.xslt - - .

2) : , , .

3) RSCM , ( unix- ).
,
rscm_service_account_{mid}_yyyyMM
rscm_service_account_{mid}.

#!/bin/bash

DB_PARAMS="-ubill -pbgbilling bgbilling"
TT=rscm_service_account_

BB=`mysql $DB_PARAMS -e "show tables like \"\${TT}%\" | grep -P "\${TT}[0-9]+$" | cut -d _ -f 4`
AA=`mysql $DB_PARAMS -e "show tables like \"\${TT}%\" | grep -P "\${TT}[0-9]+_\"`

#echo $BB

```

```

for tableName in $AA;
do
    echo $tableName
    mysql $DB_PARAMS -e "INSERT INTO $TT$BB (cid, sid, date, amount, comment) SELECT cid, sid, date, amount,
comment FROM $tableName"
done

```

```

4)      paymaster,
        paymaster.LMI_MERCHANT_ID      paymaster.merchant.id,
paymaster.LMI_SUCCESS_URL      paymaster.success.url ..
        paymaster.LMI_INVOICE_CONFIRMATION_URL - (      ).

```

```

5)      ( Windows - cmd Far)      BGBillingServer :

```

Для LINUX:

```

< Java>/bin/java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib/app/*:/lib/ext
/* ru.bitel.bgbilling.kernel.task.server.TaskConverter

```

Для Windows:

```

< Java>\bin\java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp ../lib/app/*:/lib/ext
/* ru.bitel.bgbilling.kernel.task.server.TaskConverter

```

Для FreeBSD:

```

java -Dnetworkaddress.cache.ttl=3600 -Xmx256m -Djava.awt.headless=true -cp "../lib/app/*:/lib/ext/*" ru.bitel.
bgbilling.kernel.task.server.TaskConverter

```

```

6)      ( , BGInetAccess, BGInetAccounting, BGDaiusDialup, BGIPNetflowCollector .) url
        characterEncoding=UTF-8      cp1251.      cp1251,
        utf8 collation      utf8_unicode_ci ( utf8_general_ci)      connectionCollation=utf8_unicode_ci ,
collation .
        cp1251,      connectionCollation      ( ).

```

1) Поменяйте настройки всех приложений (BGInetAccess, BGInetAccounting, BGDaiusDialup, BGIPNetflowCollector и т.п), чтобы запускались на jdk 1.8.

2) Модуль MPS. Если не работает приход платежей, то возможно дело в кодировке. Раньше они по умолчанию была cp1251, теперь для каждой системы есть настройка :

<https://docs.bitel.ru/pages/viewpage.action?pageId=43385605>

```

#      ,      utf-8, cp1251
#mps.<mpsId>.request.encoding=

```

Иногда для некоторых протоколов стоит поставить cp1251, особенно если у вас используются русские символы в номере договора.

1. Для модуля Inet: переместите параметр конфигурации accounting.deviceTypeIds из inet-access.xml (<param name="accounting.deviceTypeIds" value="x"/>) в конфигурацию модуля (accounting.deviceTypeIds=x), если еще не сделали этого.

1. Модуль Бухгалтерия(Bill). В связи с тем, что типы реквизитов теперь создаются в справочнике модуля Бухгалтерии, необходимо перенести их из конфигурации модуля. Если типов не много, то можно в ручную, иначе можно воспользоваться конвертером.

1.1. Запуск конвертера осуществляется следующим образом: запустите командную строку (cmd в ОС Windows; xterm, konsole (или любой другой) - в Linux), перейдите в каталог BGBillingServer и запустите

Для LINUX: <путь к Java>/bin/java -Xmx256m -cp ../lib/app/*:/lib/ext/* ru.bitel.bgbilling.modules.bill.server.utils.Converter_attributes <mid>

Для Windows: <путь к Java>\bin\java -Xmx256m -cp ../lib/app/*:/lib/ext/* ru.bitel.bgbilling.modules.bill.server.utils.Converter_attributes <mid>

Для FreeBSD(по рекомендации пользователей):java -Xmx256m -cp "../lib/app/*:/lib/ext/*" ru.bitel.bgbilling.modules.bill.server.utils.

Converter_attributes <mid>

где <mid> - код модуля бухгалтерия(без угловых скобок)

1.2 Убедитесь в том, что данные импортировались(Бухгалтерия->Справочники->Типы Реквизитов).

1.3 Можете удалить поле "bill.attributes" из конфига модуля.

2. Модуль Inet. При обновлении, особенно если оно происходит не в начале месяца, желательно переименовать таблицу

connection_log_entry_<mid>_<yyyyMM> за текущий месяц в, например, connection_log_entry_<mid>_<yyyyMM>_bak. Данная таблица содержит не

важные данные, а вспомогательные, для отображение RADIUS/DHCP-лога сессии. После старта InetAccess/InetAccounting создастся новая таблица.

Или же можете вызвать обновление таблицы вручную (перед запуском InetAccess/InetAccounting), посмотрев длительность выполнения данного запроса сначала на тестовой БД:

```
ALTER TABLE connection_log_entry_<mid>_<yyyyMM> ADD COLUMN `identifier` VARCHAR(50) NULL , DROP INDEX `app-dev-con` , ADD INDEX `app-dev-con` (`deviceId`, `time`, `connectionId`, `acctSessId`, `identifier`)
```

Инструкция по обновлению с версии 7.2 на версию 8.0

1. Создайте резервную копию БД, и всех приложений биллинга.
2. Остановите сервер биллинга, планировщик и загрузчик логов, переименуйте папку со старым BGBillingServer (например в BGBillingServer_old).
3. Загрузите последнюю версию BGBillingServer 8.0 с сайта. Произведите настройку согласно документации:
[Особенности установки под различные платформы](#)
!!! Пропустите момент с заливкой дампа базы !!!
[Установка сервера биллинга \(для версий до 8.0\(вкл.\)\)](#)

Настройте data/data.properties согласно старому файлу (между 7.2 и 8.0 скорее всего ничего менять не потребуется).

Перетрите тестовую лицензию в файле lic.properties вашей коммерческой лицензией для 8.x. Если в конфигурации сервера у вас прописаны параметры **license.contract.title** и **license.contract.password**, то необходимо их поменять на актуальные для версии 8.0, иначе в индикаторе лицензий будут отображаться данные для старой версии биллинга.

Перенесите, если есть files, archive, .keystore.

Если в старой версии использовались динамический код, необходимо скопировать содержимое каталога BGBillingServer/dyn из старой версии в новую.

В файле setenv.sh пропишите JAVA_HOME.

4. Загрузите https://bgbilling.ru/v8.0/download/kernel/update_8.0.zip, сохраните его в папку BGBillingServer и установите инсталлятором:
./bg_installer.sh update 8.0.zip
5. Установите обновления всех используемых вами модулей.
Для этого запустите
для Linux : ./bg_installer.sh update
для Windows: bg_installer.bat update
Как работает автоматическая система обновлений описано в документации.
6. Если не пользуетесь лаунчером клиента: загрузите новую версию клиента биллинга и настройте ее(пропишите JAVA_HOME - jdk 1.8 и сделайте файл bgbilling.sh запускаемым (для Linux)).
7. Запустите сервер биллинга (проверьте по логам все ли стартовало нормально), подключитесь к нему клиентом.
Если не пользуетесь лаунчером клиента: В окне авторизации установите опцию "загружать обновления с этого сервера".
8. Установите обновления всех приложений (BGInetAccess, BGInetAccounting, BGDaiusDialup и т.п). Обновлять нужно через штатную систему update.sh
[Установка обновлений биллинга](#)
Еще один момент - во всех приложениях нужно вручную почистить папки со сторонними библиотеками (lib/ext) и скопировать туда библиотеки из соответствующего архива на сайте (между 7.2 и 8.0 скорее всего не требуется).
9. Для приложений BGDHCP_IPN, которое не поддерживают автоматическую систему обновлений, скачайте и распакуйте новую версию, произведя аналогичные старым настройки .properties файлов и .sh-файлов.

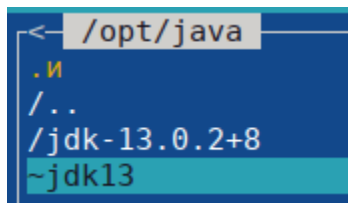
Инструкция по обновлению на версию 8.YYMM

1. Загрузите и установите необходимую версию JDK, (установка [JDK](#)) (для версии 8.2010 - JDK13, 8.2101 - JDK14, 8.2104+ - JDK15, 8.2201+ - JDK17)

Примерный набор команд для загрузки и установки JDK

```
sudo su -
cd /opt
mkdir java
cd java
wget https://github.com/AdoptOpenJDK/openjdk13-binaries/releases/download/jdk-13.0.2%2B8/OpenJDK13U-jdk_x64_linux_hotspot_13.0.2_8.tar.gz
tar -zxvf OpenJDK13U-jdk_x64_linux_hotspot_13.0.2_8.tar.gz
#
#rm -f OpenJDK13U*
ln -s jdk-13.0.2+8 jdk13
```

в итоге должно получиться что то следующее



2. Загрузите и установите сервер биллинга. Для этого перейдите на страницу <https://bgbilling.ru/current/new/download.shtml> скопируйте ссылку на архив сервера, далее загрузите его на сервер и распакуйте.

Примерный набор команд для загрузки и установки

```
sudo su -
cd /opt
# bgbilling , bgbilling.new
mkdir bgbilling
cd bgbilling
wget , https://bgbilling.ru/download/8.2010/sets/2010010305/BGBillingServer_8.2010-2010010305.zip
unzip BGBillingServer_8.2010-2010271940.zip
#
#rm -f BGBillingServer_*.zip
```

результат


```
← /opt/bgbilling/BGBillingServer
.и      Имя
/..
/.run
/archive
/bin
/components
/data
/dyn
/lib
/log
/script
/tmp
.bgclient_keystore
```

3. Если JDK установили в каталог отличный от указанного в п. 1 прописываем путь к ней в файле `/opt/bgbilling/BGBillingServer/bin/setenv.sh`

```
setenv.sh [----] 0 L
#JAVA_HOME=
if [[ -e /opt/java/jdk13 ]]
then
    JAVA_HOME=/opt/java/jdk13
fi
```

4. Копируем из текущей инсталляции файлы:

- файл с лицензией `BGBilling/data/lic.properties`
- файл с настройками сервера `BGBilling/data/data.properties`

Создаем файлы с расширением `.orig` чтобы при обновлении не происходило перетирание файлов с настройками

```
cd /opt/bgbilling/BGBillingServer/data
cp lic.properties lic.properties.orig
cp data.properties data.properties.orig
```

5. Переходив в каталог `/opt/bgbilling/BGBillingServer/bin/` убеждаемся, что у скриптов установлены права на запуск (при необходимости устанавливаем `chmod 0755 *.sh`)

6. Запускаем скрипт `update.sh`, он должен загрузить и установить сервер биллинга (`kernel_8.2010-2010271940.zip`) и конфигурацию сторонних библиотек (`updateLib_8.2010-2010260314.zip`)

после будет предложено установить все модули и плагины указанные в лицензии (если у вас тестовая лицензия со всеми компонентами биллинга, рекомендуем ответить `no` и далее установить только необходимые по отдельности)

после этого будет выведено следующее меню

```
Select:
 1) Install new component      3) Show installed components
 2) Update installed components
 0) Exit
=> 
```

Выбираем пункт 1 (нажимаем клавишу 1 и Enter), выводит список доступных компонентов

```

Select:
 1) agr          22) email       43) onpay      64) simplepay
 2) alfabank    23) enaza      44) organizer  65) smarthouse
 3) assist     24) fias       45) paykeeper  66) softkey
 4) assistant  25) gazprombank 46) paylinks  67) softline
 5) bill       26) gorod     47) paymaster  68) sorm
 6) bonus      27) hd        48) payonline  69) subscription
 7) buyemoney  28) helpdesk  49) payture    70) tinkoff
 8) bvcom      29) im        50) phone      71) trayinfo
 9) card       30) inet      51) psb        72) tv
10) cashcheck 31) ipn       52) pscb       73) twpg
11) cerbercrypt 32) license  53) qiwi       74) uniteller
12) chronopay  33) mangooffice 54) rbkmoney   75) vepay
13) cladr     34) megogo    55) rentsoft  76) vidimax
14) crm       35) mobile    56) reports    77) voice
15) dba       36) mobimoney 57) rfiec      78) voiceip
16) dialup    37) moneta    58) robokassa  79) vseplatezhi
17) dispatch 38) mps       59) rscm       80) wm
18) documents 39) mtsbank   60) runtel    81) yamoney
19) drweb     40) netpay    61) rurupay   82) kernelWeb
20) drwebn    41) npay      62) sberbank
21) edo       42) omega     63) sbpilot
0) Exit

```

устанавливаем необходимые компоненты вводя их номер и нажимая Enter

После установки всех компонент вводим 0 и Enter, при выходе происходит проверка наличия необходимых библиотек и их загрузка.

6. Копируем из текущей инсталляции необходимые файлы и каталоги

7. Пробуем запустить сервер биллинга с помощью скрипта `/opt/bgbilling/BGBillingServer/bin/start.sh`

8. Настройка Автозапуска приложений биллинга при старте сервера.

systemd

```

cd /opt/BGBillingServer/script
cp *.service /usr/lib/systemd/system
systemctl enable bgbilling.service
systemctl enable bgscheduler.service

```

Примечания

С версии 8.2107+, если вы используете модуль NPay обратите внимание на изменение настроек статусов в которых происходит начисление (см. раздел доки про Модуль Абонплат)

Личный кабинет (не поддерживается с версии 9.2204+)

Bitel Личный кабинет

Пожалуйста, введите учетные данные

10000010

Пароль

[Забыли пароль?](#)

ВОЙТИ

BG Билетин 6.2 © Bitel Ltd

Bitel Личный кабинет 81.11 10000010

Новости

09.06.2015
Бета-версия нового личного кабинета
Вы можете посмотреть новую версию личного кабинета, доступную в готовых образах виртуальных машин. Новый личный кабинет будет доступен с версии 6.2.

28.05.2015
Готовые образы виртуальных машин
Готовые образы виртуальных машин доступны на странице загрузки.
[Узнать подробнее.](#)

BG Билетин 6.2 © Bitel Ltd

- НОВОСТИ
- УВЕДОМЛЕНИЯ
- БАЛАНС
- ЛИМИТ
- ТАРИФНЫЕ ОПЦИИ
- ДОГОВОР
- ИНТЕРНЕТ

Баланс договора

Август 2015 ПОКАЗАТЬ

Описание	Сумма
Входящий остаток на начало месяца	92,75 руб.
Приход за месяц (всего)	0,00 руб.
Расход за месяц (всего)	0,00 руб.
Наработка за месяц (всего)	11,64 руб.
Входящий	11,64 руб.
Исходящий остаток на конец месяца	81,11 руб.
Лимит	-110,00 руб.

ПОПОЛНИТЬ СЧЕТ

BG Balance 6.2 © Bitel Ltd

Лимит баланса

Текущий лимит: -110.00 ₪

Лимит будет восстановлен при поступлении суммы равной или большей суммы понижения, а понижение лимита помечено как погашено. Если понижение лимита по истечении срока не погашено, оно помечается как просрочено и лимит восстанавливается к прежнему значению.

ПОНИЗИТЬ ЛИМИТ

Понизить лимит на 3 дня ПОНИЗИТЬ

История понижений лимита

Август 2015 ПОКАЗАТЬ

Сумма	Дата понижения	Дата восстановления	Состояние

Сессии

Активные сессии История сессий

ПОКАЗАТЬ

Для информации по трафикам нажмите на строку сессии.

Время входа	Время выхода	Длительность	Сумма
19.08.2015 00:00:00	22:38:46 [81526]	11,64000 руб.	

BG Balance 6.2 © Bitel Ltd

Лимит баланса

Текущий лимит: -110.00 ₪

Лимит будет восстановлен при поступлении суммы равной или большей суммы понижения, а понижение лимита помечено как погашено. Если понижение лимита по истечении срока не погашено, оно помечается как просрочено и лимит восстанавливается к прежнему значению.

ПОНИЗИТЬ ЛИМИТ

Понизить лимит на

3 дня

ПОНИЗИТЬ

История понижений лимита

Август 2015

ПОКАЗАТЬ

Установка личного кабинета

- [Установка Wildfly](#)
- [Установка ЛК](#)
- [Рекомендация для reverse-proxy \(Nginx\)](#)

Установка Wildfly

При установке далее мы рекомендуем указывать JAVA_HOME вручную в **standalone.conf**, чтобы, если у вас есть установленная JRE/JDK из репозитория, неожиданно для вас не поменялась используемая версия Java. Однако это не касается запуска некоторых команд, требуемых при установке личного кабинета. Для таких команд может использоваться JRE/JDK по умолчанию, установленную из репозитория; или же Вы можете прописать JAVA_HOME глобально, в /etc/environment или в ~/.bashrc.

1. Скачайте сервер приложений Java EE (Java Platform, Enterprise Edition) **Wildfly** с официального сайта <http://wildfly.org/downloads/>. Рекомендуемая версия Wildfly для биллинга 21.0.2.Final. Распакуйте содержимое архива, например, в папку /opt и переименуйте папку wildfly-номерверсии в wildfly. Или же создайте символическую ссылку **/opt/wildfly**, которая будет указывать на /opt/wildfly-номерверсии.
2. Отредактируйте файл **/opt/wildfly/bin/standalone.conf**, укажите правильный путь к JDK в переменной JAVA_HOME:

```
#
# Specify the location of the Java home directory. If set then $JAVA will
# be defined to $JAVA_HOME/bin/java, else $JAVA will be "java".
#
#JAVA_HOME="/opt/java/jdk"
```

Отредактируйте ниже параметры запуска JVM:

```
#
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    #JAVA_OPTS="-Xms64m -Xmx512m -XX:MetaspaceSize=96m -XX:MaxMetaspaceSize=256m -Djava.net.
preferIPv4Stack=true"
    JAVA_OPTS="-Xms64m -Xmx300m -XX:MetaspaceSize=96m -XX:MaxMetaspaceSize=256m -Djava.net.
preferIPv4Stack=true -XX:MaxDirectMemorySize=80m -XX:MinHeapFreeRatio=10 -XX:MaxHeapFreeRatio=20"
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
else
    echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
fi
```

3. Если у вас нет значения JAVA_HOME по умолчанию, то отредактируйте **common.sh** и укажите там путь к **JAVA_HOME** (иначе следующая команда не будет выполнена):

```
#!/bin/sh -x

JAVA_HOME=/opt/java/jdk15
```

4. Выполните команду, чтобы добавить пользователя Wildfly для аутентификации в web-консоли:

```
./add-user.sh
```

5. Отредактируйте файл конфигурации Wildfly **/opt/wildfly/standalone/configuration/standalone.xml** указав используемые HTTP-порты (если необходимо):

```
<!--<socket-binding name="http" port="${ jboss.http.port:8080} " />-->
<socket-binding name="http" port="${ jboss.http.port:8081} " />
```

Возможно вы захотите поменять интерфейс, на котором открыт HTTP-порт. Обратите внимание, что для сокета (порта) интерфейс прописан в атрибуте `socket-binding/@interface`, а также в `socket-binding-group/@default-interface`. Сами интерфейсы прописаны выше:

```
<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:127.0.0.1}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="{jboss.bind.address.unsecure:127.0.0.1}"/>
  </interface>
</interfaces>
```

Не рекомендуем оставлять web-консоль (**management**) открытым наружу (если вы решили поменять интерфейсы).

Или же порты **можно поменять с помощью сдвига (offset)**, используя параметр **port-offset**, например, на 1 (т.е. 8080 станет 8081, 8443 станет 8444), изменив значение параметра **port-offset**:

```
<!--<socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.binding.port-offset:0}">-->
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.binding.port-offset:1}">
```

При изменении port-offset также изменится порт web-консоли Wildfly, значение по умолчанию которого 9990. В этом случае нужно редактировать файл jboss-cli.xml, меняя значение <port>9990</port> на новое.

Данные параметры можно менять в web-консоли Wildfly.

6. Добавьте системного пользователя wildfly и поменяйте владельца у /opt/wildfly:

```
useradd -r wildfly
chown -R wildfly:wildfly /opt/wildfly-XX.Y.Z.Final
chown -R wildfly:wildfly /opt/wildfly
```

7. Создайте файл службы для systemd, (примеры скриптов досту в каталоге /wildfly-XX.Y.Z.Final/docs/contrib/scripts) или используйте в качестве примера код ниже

```
cd /lib/systemd/system
touch wildfly.service
```

и укажите содержимое:

```
[Unit]
Description=WildFly Application Server Daemon
After=network.target
Requires=

[Service]
Type=simple
User=wildfly
Group=wildfly
ExecStart=/opt/wildfly/bin/standalone.sh

[Install]
WantedBy=multi-user.target
```

8. Установите в автозапуск:

```
systemctl enable wildfly
```

9. Запустите

```
systemctl start wildfly
```

и попробуйте открыть web-консоль **на порту 9991, если указали сдвиг портов на 1**, или 9990, если не указывали сдвиг портов. Web-консоль по умолчанию запускается на интерфейсе 127.0.0.1, поэтому может понадобиться проброс портов.

10. Отключить логирование в консоль (в файл /var/log/message)

```
./jboss-cli.sh --connect --commands="/subsystem=logging/root-logger=ROOT:remove-handler(name='CONSOLE')"
```

11. Настройте firewall, а также рекомендуем использовать nginx.

Установка ЛК

1. Создайте нового пользователя в BGBillingServer, в поле "Полное имя" укажите Пользователь (как ранее помечались действия абонента), в поле "Описание" укажите "Личный кабинет". Придумайте сложные логин и пароль, разрешите все действия.
2. Скачайте <https://bgbilling.ru/download/8.2101/mybgbilling/MyBGBilling.last> соответствующий версии биллинга, например, в папку /opt. Распакуйте архив, оставив имя директории, содержащее файлы архива MyBGBilling.war и поместите эту папку в /opt/wildfly/standalone/deployments:

```
cd /opt
wget https://bgbilling.ru/download/8.2101/mybgbilling/MyBGBilling.last
unzip ./MyBGBilling.last -d /opt/wildfly/standalone/deployments && rm -f MyBGBilling.last
touch /opt/wildfly/standalone/deployments/MyBGBilling.war.dodeploy
```

3. Поменяйте владельца у директории MyBGBilling.war

```
chown -R wildfly:wildfly /opt/wildfly/standalone/deployments/MyBGBilling.war
```

4. Укажите в файле /opt/wildfly/bin/common.sh переменную JAVA_HOME

```
#!/bin/sh -x

JAVA_HOME=/opt/java/jdk14

setModularJdk() {
    "$JAVA" --add-modules=java.se -version > /dev/null 2>&1 && MODULAR_JDK=true || MODULAR_JDK=false
}
```

5. Запустите скрипт синхронизации библиотек сервера биллинга и ЛК (для работы скрипта должен быть запущен сервер биллинга, и если у вас ЛК и биллинг работают на разных серверах или биллинг запущен на порту отличном от 8080, отредактируйте соответствующие напарметры в файле /opt/wildfly/standalone/deployments/MyBGBilling.war/WEB-INF/script/application.xml):

```
chmod 0755 /opt/wildfly/standalone/deployments/MyBGBilling.war/WEB-INF/script/update_libs.sh
/opt/wildfly/standalone/deployments/MyBGBilling.war/WEB-INF/script/update_libs.sh
```

6. Скопируйте файлы конфигурации из MyBGBilling.war/WEB-INF/defaults в MyBGBilling.war/WEB-INF:

```
cp /opt/wildfly/standalone/deployments/MyBGBilling.war/WEB-INF/defaults/*.groovy /opt/wildfly/standalone/deployments/MyBGBilling.war/WEB-INF
```

7. Отредактируйте MyBGBilling.war/WEB-INF/mybgbilling-conf.groovy, укажите параметры подключения к BGBillingServer

```
// BGBillingServer.
// , , BGBillingClient
bgbilling {
    // URL BGBilling
    url = 'http://127.0.0.1:8080/bgbilling/executer'
    //
    user = 'customer'
    //
    password = 'customer'
}
```

8. Попробуйте войти в ЛК через браузер `http://[host[:port]]/my`

Рекомендация для reverse-proxy (Nginx)

Если вы используется reverse-proxy (например, Nginx) и на нём настроен HTTPS, а reverse-proxy работает с Wildfly по HTTP, то в некоторых случаях нужно, чтобы Wildfly знал, что абонент на самом деле использует HTTPS (например, это может быть нужно при отправке редиректа от Wildfly). Для этого рекомендуется внести такие настройки:

1. Указать в reverse-proxy добавление заголовка X-Forwarded-Proto:

```
proxy_set_header X-Forwarded-Proto $scheme;
```

2. Указать Wildfly, чтобы он обрабатывал такой заголовок. Для этого в standalone.xml в элементе http-listener нужно добавить `proxy-address-forwarding="true"`:

```
<http-listener name="default" socket-binding="http" redirect-socket="https" enable-http2="true" proxy-address-forwarding="true"/>
```

Также необходимо, чтобы путь к контексту ЛК совпадал с путем, который установлен через NGINX. Например, и там и там должно быть `/my`.

Настройка личного кабинета

Конфигурация ЛК состоит из трех файлов, расположенных в папке WEB-INF: основного файла конфигурации **mygbilling-conf.groovy**, файла конфигурации меню **mygbilling-menu.groovy**, файла конфигурации платежных систем **mygbilling-payment.groovy**. Файлы конфигурации созданы с использованием синтаксиса [Groovy](#).

- [Основная конфигурация \(mygbilling-conf.groovy\)](#)
- [Конфигурация меню \(mygbilling-menu.groovy\)](#)
- [Конфигурация приема платежей \(mygbilling-payment.groovy\)](#)

Обновление личного кабинета

Для обновления личного кабинета запустите скрипт mygbilling-update.sh:

```
/opt/wildfly/standalone/deployments/MyBGBilling.war/WEB-INF/script/update_libs.sh
```

При обновлении файлы, рядом с которыми есть файл с таким же именем плюс суффикс(расширение) **.orig**, не будут перезаписаны файлом из сборки, вместо них обновятся **.orig**-файлы (см. [Кастомизация личного кабинета](#)). Также при обновлении полностью удаляется и перезаписывается директория MyBGBilling.war/WEB-INF/classes/ru/bitel.

Путь к контексту ЛК

Путь к контексту личного кабинета можно изменить в файле wildfly/standalone/deployments/MyBGBilling.war/WEB-INF/jboss-web.xml:

```
<!-- https://provider.ru/my -->  
<context-root>/my</context-root>
```

Например, чтобы ЛК открывался от корня:

```
<!-- https://my.provider.ru/ -->  
<context-root>/</context-root>
```

При использовании NGINX необходимо, чтобы путь к контексту был одинаковым в NGINX и в самом ЛК.

Восстановление пароля

При нажатии кнопки восстановления пароля через email пользователю в письме приходит ссылка, которая настраивается параметром [конфигурации contract.password.forgot.link](#). Укажите в нем правильный путь до страницы index.xhtml (вместо webexecuter):

```
contract.password.forgot.link=http://provider.ru/my/index.xhtml?action=ChangePassword&mid=contract
```

Описание синтаксиса файлов конфигурации groovy

Группы параметров в конфиге разделяются не точкой, а с помощью вложенных блоков. Значение параметра должно быть правильным Groovy /Java-объектом - **в простом случае строкой**, заключенной в одинарные или двойные кавычки, или **числом**, например:

```
one {  
    two {  
        parameterA = 'value1'  
        three {  
            parameterA = 'value2'  
            parameterB = 100  
        }  
    }  
}
```

Т.е. параметр конфигурации - это один или несколько вложенных блоков, имя параметра и значение после знака =. Данный пример в конфигурации модулей биллинга выглядел бы так:

```
one.two.parameterA=value1
one.two.three.parameterA=value2
one.two.three.parameterB=100
```

Некоторые значения параметров должны быть списками или массивами определенных объектов. Объекты списка заключены в квадратные скобки [] и разделены между собой символом запятой. Например:

```
authentication {
    modes = [
        authenticationMode {
            mode = 'contract'
        },
        authenticationMode {
            mode = 'login'
            module = 'inet'
            moduleId = 1
        }
    ]
}
```

Некоторые значения параметров могут быть ассоциативными массивами (список ключ:значение, map). Связки ключ:значение заключены в квадратные скобки [] и разделены между собой символом запятой. Например:

```
example {
    map = [
        key: 'value',
        key2: 200
    ]
}
```

Также параметры могут быть прописаны как ассоциативный массив, заключенный в круглые скобки (значение **в этом случае** прописывается через ':' (двоеточие), а не через символ '='):

```
authentication {
    modes = [
        authenticationMode( mode: 'contract' ),
        authenticationMode( mode: 'login', module: 'inet', moduleId: 1 )
    ]
}
```

Некоторые значения могут быть динамическими, если использовать замыкания (closure). Т.е., грубо говоря, значением может быть функция, которая будет возвращать нужное значение:

```
status {
    //
    //statusChange = { contract -> return contractInGroup( contract, [1, 2, 3, 4, 20] ) &&
    isCustomer(); }
    //statusChange = { contract -> contractInGroup( contract, [1, 2, 3, 4, 20] ) &&
    isCustomer() }
    statusChange = { isCustomer() }
}
```

В **mygbilling-conf.groovy** и **mygbilling-menu.groovy** в таких замыканиях можно использовать определенный набор методов, аргумент объект contract, а также дополнительные аргументы, специфичные для определенного параметра конфигурации (например, параметры content.kernel.customerTitle и content.kernel.subContractGroup):

```

content {
  kernel {
    // ,
    customerTitle = { contract, contractParameterMap ->

      // ID . ( customerTitle)
      def individualCustomerTitleParamIds = [0, 0, 0, 0, 0];
      // ID . ( customerTitle)
      def corporationCustomerParamIds = [0, 0, 0, 0, 0];

      def paramIds = contract.personType == 1 ? corporationCustomerParamIds :
individualCustomerTitleParamIds;

      String result = contractParameterMap.values().stream()
        .filter{ v -> paramIds.contains( v.entitySpecAttrId ) && notBlankString( v.
toString() ) }

        .findFirst()
        .map{ v -> v.toString() }
        .orElse( null );

      //
      //if( result == null ) {
      //    result = contract.comment;
      //}

      return result;
    }
  }
}

```

В замыканиях можно использовать методы:

- **isCustomer()** или **isUserInRole('customer')** - возвращает true, если в режиме аутентификации, которым воспользовался абонент, не указан параметр **role = 'unauthCustomer'**;
- **contractInGroup(contract, groupIds)** - возвращает true, если переданный в первый аргумент объект-contract содержит в себе одну из групп, указанных списке второго аргумента, например: **contractInGroup(contract, [2, 3, 8, 13])**.

Основная конфигурация (mybgbilling-conf.groovy)

Основная конфигурация личного кабинета состоит из нескольких блоков:

- **bgbilling** - конфигурация подключения к BGBillingServer,
- **authentication** - параметры аутентификации абонента,
- **mail** - параметры почтовой подсистемы (чтобы ЛК мог отправлять письма при необходимости),
- **content** - параметры содержимого страниц.

Конфигурация подключения к BGBillingServer

```
// BGBillingServer.
// , , BGBillingClient
bgbilling {
    // URL BGBilling
    url = 'http://127.0.0.1:8080/bgbilling/executer'
    //
    user = 'customer'
    //
    password = '123456'
}
```

Параметры идентификации HTTP-соединения

Личному кабинету в некоторых случаях требуется знать базовый URL, по которому абоненты получают доступ к нему. Личный кабинет биллинга может получить это значение из запроса, однако при использовании NGINX значение из запроса может быть не правильным. Поэтому базовый URL следует указать в конфигурации в параметре **baseUrl**.

Также личному кабинету требуется знать IP-адрес абонента, который пользуется им в текущий момент (например, для авторизации по IP-адресу или блокировке при переборе логинов/паролей). Поэтому при использовании NGINX требуется указать HTTP-заголовок в параметре **context.hostHttpRequestHeader**, из которого получать реальный IP-адрес вместо физического IP-адреса HTTP-соединения.

```
context {

    // ( ).
    //baseUrl = 'https://provider.ru/selfcare'
    baseUrl = 'https://my.provider.ru'

    // HTTP-, , X-Real-IP. IP-
    hostHttpRequestHeader = 'X-Real-IP'
}
```

Параметры аутентификации абонента

```
//
authentication {
    // - , captcha
    captchaLoginErrorCount = 5
    // - , captcha
    captchaHostErrorCount = 20
    // - ,
    blockHostErrorCount = 30

    //
    modes = [
        //
        authenticationMode {
            mode = 'contract'
        }
    ]
}
```

Режимов аутентификации может быть несколько - в этом случае в окне логина можно выбрать необходимый. На данный момент поддерживаются 6 режимов аутентификации:

- по номеру договора:

```
authenticationMode {
    mode = 'contract'
}
```

- по логину модуля Inet:

```
authenticationMode {
    module = 'inet'
    mode = 'login'
    // ID
    moduleId = 1
}
```

- по IP-адресу сессии модуля Inet (вход без пароля):

```
authenticationMode {
    module = 'inet'
    mode = 'ip'
    // ID
    moduleId = 1
    //
    role = 'unauthCustomer'
}
```

- по параметру договора "телефон":

```
authenticationMode {
    module = 'kernel'
    mode = 'phoneParam'
    // ID
    parameterId = 1
    //
    username = { s -> s.replaceAll( /^8(.+)$/,'7$1' ) }
}
```

- по параметру договора "Email":

```
authenticationMode {
    module = 'kernel'
    mode = 'emailParam'
    // ID
    parameterId = 2
    //
    username = { s -> s.replaceAll( /^8(.+)$/,'7$1' ) }
}
```

- по текстовому параметру договора:

```
authenticationMode {
    module = 'kernel'
    mode = 'textParam'
    // ID
    parameterId = 3
}
```

При аутентификации по параметру договора в качестве пароля используется пароль к личному кабинету (как и при аутентификации номеру договора). При аутентификации по номеру телефона идет поиск только по введенным цифрам, при этом можно задать преобразование введенного номера в другой вид с помощью параметра **username**; если параметр **username** не указан, то по умолчанию 8 в начале строки заменяется на 7.

Для режима аутентификации можно назначить, чтобы доступ после аутентификации через него был ограничен. Для этого указывается параметр **role = 'unauthCustomer'**. В этом случае, вызов **isUserInRole("customer")** будет возвращать **false**. Ограниченный доступ может быть указан, например, для режима аутентификации по IP-адресу модуля Inet.

Можно разрешить аутентификацию только для определенных групп договоров, указав условие в параметре filter:

```
authenticationMode {
    module = 'inet'
    mode = 'ip'
    //ID
    moduleId = 1
    //
    role = 'unauthCustomer'
    //
    filter = { contract -> contractInGroup( contract, [1, 2, 3, 4, 20] ) }
}
```

Или наоборот, запретить для определенных групп договоров:

```
filter = { contract -> !contractInGroup( contract, [1, 2, 3, 4, 20] ) }
```

Или разрешить по номеру договора:

```
filter = { contract -> contract.title.startsWith( "NK" ) }
```

Или использовать регулярное выражение:

```
filter = { contract -> contract.title.matches( "NK.*" ) }
```

Параметры почтовой подсистемы

```
// SMTP,
mail {
    smtp {
        host = 'smtp.provider.ru'
    }

    from {
        email = 'support@provider.ru'
        name = 'BGBilling'
    }
}
```

Параметры содержимого страниц

Разрешенные фрагменты

Данный блок конфигурации позволяет настраивать, какие фрагменты страницы или какие действия доступны абонентам или группам абонентов. Например, в коде страницы статусов договора есть фрагмент смены статуса:

```
<ui:fragment rendered="#{configuration.get('content.kernel.status.statusChange', true)}">
    ...
</ui:fragment>
```

Соответственно можно в конфигурации запретить всем менять статус договора из личного кабинета:

```
content {
  kernel {
    ...

    // status.xhtml
    status {
      //
      statusChange = false
    }
    ...
  }
  ...
}
```

Можно разрешить только тем, кто был аутентифицирован по логину/паролу (в конфигурации по умолчанию установлен этот вариант):

```
statusChange = { isUserInRole( "customer" ) }
```

Разрешить только аутентифицированным по логину/паролу физ. лицам:

```
statusChange = { contract -> isUserInRole( "customer" ) && contract.getPersonType() == 0 }
```

Или разрешить только аутентифицированным по логину/паролу определенным группам договоров:

```
statusChange = { contract -> isUserInRole( "customer" ) && contractInGroup( contract, [1, 2, 3, 4, 20] ) }
```

Название контрагента в верхней части страницы

По умолчанию в шапке страницы название или имя контрагента не отображается. За отображение названия (или имени) отвечает параметр **content.kernel.customerTitle**. В конфигурации можно указать, чтобы отображался комментарий договора:

```
content {
  kernel {

    // ,
    customerTitle = { contract, contractParameterMap -> contract.comment }

    ...
  }
}
```

Или же отобразить параметр договора, в зависимости от типа лица договора (физ. лицо или юр. лицо):

```
content {
  kernel {

    // ,
    customerTitle = { contract, contractParameterMap ->

      // ID . ( customerTitle)
      def individualCustomerTitleParamIds = [33, 0, 0, 0, 0];
      // ID . ( customerTitle)
      def corporationCustomerParamIds = [10, 0, 0, 0, 0];

      def paramIds = contract.personType == 1 ? corporationCustomerParamIds :
individualCustomerTitleParamIds;

      String result = contractParameterMap.values().stream()
        .filter{ v -> paramIds.contains( v.entitySpecAttrId ) && notBlankString( v.
toString() ) } }
```

```

        .findFirst()
        .map{ v -> v.toString() }
        .orElse( null );

        return result;
    }
    ...
}
...
}

```

Группировка субдоговоров в меню

Если субдоговоров у данного договора меньше 10 - они отображаются прямо в меню. В этом случае можно сортировать и группировать список субдоговоров:

```

content {
    kernel {

        // ( )
        subContractGroup = { subContractList ->

            subContractList
            .stream()
            .sorted({ a,b -> a.title.compareTo(b.title) })
            .collect( Collectors.groupingBy{ contract ->

                //
                if( contractInGroup( contract, [1, 2, 3, 4, 20] ) ) {
                    return "contract.sub.group.01.internet";
                } else if( contractInGroup( contract, [5, 6, 7, 8, 9] ) ) {
                    return "contract.sub.group.02.phone";
                } else {
                    return "contract.sub.group.99.other";
                }
            }

            // -
            return "";
        })
        .entrySet()
        .stream()
        .sorted({ a,b -> a.key.compareTo(b.key) })
        .collect( Collectors.toList() );

    }
    ...
}
...
}

```

В примере при группировке используются строки вида "contract.sub.group.01.internet". Число в данном случае используется для сортировки групп, а само название группы должно быть прописано в Locale_ru_RU.properties по ключу:

```

contract.sub.group.01.internet=
contract.sub.group.02.phone=
contract.sub.group.99.other=

```


Конфигурация меню (mygbilling-menu.groovy)

Данный файл конфигурации возвращает дерево пунктов меню для договора. Выглядит конфигурация, например, так:

```
menu {
    //
    children = [

        //
        menu( page: "kernel/news", icon: "fa-newspaper-o", title: "menu.news" ),

        // +
        menu( page: "kernel/notificationsEx", subPage: "notifications", icon: "fa-envelope-o",
            title: "menu.notifications", badge: "#{notificationBean.
getUnreadCount()}", badgeUpdate: "#{notificationBean.populate()}",
            show: isCustomer() ),

        // ( )
        menu( page: "kernel/notifications", icon: "fa-envelope-o", title: "menu.notifications",
            show: !isCustomer() ),

        //
        menu( page: "kernel/balance", icon: "fa-rub", title: "menu.balance" ),

        //
        menu( page: "kernel/limit", icon: "fa-umbrella", title: "menu.limit" ),

        //
        menu( page: "kernel/tariffOptions", icon: "fa-cogs", title: "menu.tariffOptions", show:
isCustomer() ),

        //
        menu( icon: "fa-briefcase", title: "menu.contract" ) {
            children = [
                //
                menu( page: "kernel/status", title: "menu.status" ),
                //
                menu( page: "kernel/tariffs", title: "menu.tariffs", show: isCustomer()
),
                //
                menu( page: "kernel/additionalActions", title: "menu.
additionalActions", show: isCustomer() ),
                //
                menu( page: "kernel/documents", title: "menu.documents", show:
isCustomer() ),
                // ( )
                menu( page: "plugins/documents/documents", title: "menu.documents" ),
                //
                menu( module: "bill", page: "modules/bill/bill", title: "menu.bill",
show: isCustomer() ),
                //
                menu( page: "kernel/notes", title: "menu.notes", show: isCustomer() ),
                //
                menu( page: "kernel/password", title: "menu.password", show:
isCustomer() )
            ]
        },

        //
        menu( module:"inet", icon:"fa-globe", title:"menu.inet" ) {
            children = [
                //
                menu( page: "modules/inet/sessions", title: "menu.inet.sessions" ),
                //
                menu( page: "modules/inet/traffics", title: "menu.inet.traffics" ),
                //
                menu( page: "modules/inet/password", title: "menu.inet.password", show:
isCustomer() )
            ]
        }
    ]
}
```

```

        },

        //
        menu( module:"tv", page:"modules/tv/tv", icon:"fa-tv", title:"menu.tv" ),

        //
        menu( page: "plugins/helpdesk/helpdesk", icon: "fa-wrench", title: "menu.helpdesk",
              badge: "#{helpdeskBean.getUnreadTopicCount()}",
              show: isCustomer() )

        badgeUpdate: "#{helpdeskBean.populateTopics()}",

    ]
}

```

У каждого объекта-пункта меню есть набор параметров:

- **module** - модуль, если данный пункт относится к модулю, наследуется дочерними пунктами;
- **moduleId** - ID модуля (необязательно, если указан **module**, то подставляется автоматически), наследуется дочерними пунктами. Можно использовать, если одинаковые модули нужно показывать по разному;
- **page** - страница, без .html;
- **subPage** - подстраница;
- **icon** - иконка;
- **title** - название пункта меню (ключ для Locale.properties);
- **badge** - счетчик, указывается JSF-вызов метода, который вернет число;
- **badgeUpdate** - JSF-вызов метода, который нужно произвести для обновления счетчика
- **show** - показывать пункт или нет (если не указан, то показывать)
- **children** - список дочерних пунктов меню

Список дочерних пунктов меню **children** задается как массив, поэтому в данном случае следует разделять элементы запятыми.

Используя параметр **show**, можно ограничивать использование пунктов меню для групп договоров:

```

menu( moduleId: 210, page: "modules/tv/tv", icon: "fa-tv", title: "menu.tv",
      show: contractInGroup( contract, [1, 2, 3, 4, 20] ) )

```

При необходимости список дочерних пунктов меню можно определить как переменную и добавлять пункты в этот список, используя условия:

```

menu {
    //
    def firstLevel = [];
    children = firstLevel;

    //
    firstLevel << menu( page: "kernel/news", icon: "fa-newspaper-o", title: "menu.news" )

    // /
    if( isCustomer() ) {
        // +
        firstLevel << menu( page: "kernel/notificationsEx", subPage: "notifications", icon: "fa-
envelope-o",
                           title: "menu.notifications", badge: "#{notificationBean.
getUnreadCount()}", badgeUpdate: "#{notificationBean.populate()}" )

        } else {
            //
            firstLevel << menu( page: "kernel/notifications", subPage: "", icon: "fa-envelope-o", title:
"menu.notifications" )

        }
    //
    firstLevel << menu( page: "kernel/balance", icon: "fa-rub", title: "menu.balance" )

    // /
    if( isCustomer() ) {
        //

```

```
firstLevel << menu( page: "kernel/limit", icon: "fa-umbrella", title: "menu.limit" )
```

```
//
```

```
firstLevel << menu( page: "kernel/tariffOptions", icon: "fa-cogs", title: "menu.tariffOptions" )
```

```
...
```

Конфигурация приема платежей (mybgbilling-payment-groovy)

Если протокол платежной системы поддерживает передачу URL для возврата абонента после платежа в магазин, то ЛК сам передает данный URL. Если же URL прописывается в админке платежной системы, то

- в случае, если указывается только один URL, укажите путь к странице payment.xml (она находится там же, где и index.xhtml):

```
https://my.provider.ru/payment.xhtml
```

- в случае, если указывается два URL - для удачного и не удачного платежа, то укажите соответственно:

```
https://my.provider.ru/payment.xhtml?result=success
https://my.provider.ru/payment.xhtml?result=failure
```

В файле **mybgbilling-payment.groovy** настраивается, какие платежные системы будут присутствовать при проведении оплаты из личного кабинета. Часто значения по умолчанию не требуют изменений. ЛК сам создаст список провайдеров из платежных модулей и при проведении оплаты отобразит те из них, модули которых подключены к данному договору.

```
paymentConfig {

    /* true -  /,  providers. */
    replaceProviders = false
    /* Card */
    showCard = false
    /* email/ */
    needReceiptContacts = true

    providers = [

    ]

    /* , , -
    . */
    payments = [

    ]
}
```

Если вы хотите, чтобы дополнительно отображался вариант с активацией карты оплаты модуля Card, укажите в конфигурации **showCard=true**.

При необходимости к текущим платежным системам вы можете добавить кнопку-ссылку:

```
paymentConfig {

    /* true -  /,  providers. */
    replaceProviders = false
    /* Card */
    showCard = false
    /* email/ */
    needReceiptContacts = true

    providers = [

        //
        provider {
            id = "externalLinkQiwi"
            title = "Qiwi"
            image = 'static/images/logos/qiwi.png'
            config = config {
                url = 'https://qiwi.com/payment/form.action?provider=297'
            }
        }
    ]
}
```

```

]

payments = [

]
}

```

Обратите внимание на параметр **replaceProviders**. Если он указан **true**, то не будет автоматической подгрузки провайдеров из платежных модулей - будут только те провайдеры, которые указаны в **providers**. Т.е. если его установить в true в конфигурации, что представлена выше, то при проведении платежа будет доступна только эта внешняя ссылка, даже если к договору подключены какие-то платежные модули.

Также для некоторых случаев с помощью **providers** можно переопределить конфигурацию провайдера, например, указать для Яндекс.Денег, какие именно типы платежей можно принимать:

```

paymentConfig {

    /* true -  /,  providers. */
    replaceProviders = false
    /*    Card */
    showCard = false
    /*    email/    */
    needReceiptContacts = true

    providers = [

        // .
        provider {
            // .
            moduleId =
            config = config {
                // ( ). : PC: .;AC: ;MC: ;GP: ;WM: WebMoney;SB:
                //paymentTypes = 'PC:payment.yamoney.PC;AC:payment.yamoney.AC;MC:payment.
yamoney.MC;GP:payment.yamoney.GP;WM:payment.yamoney.WM;SB:payment.yamoney.SB'
                paymentTypes = 'PC:payment.yamoney.PC'
            }
        }
    ]

    payments = [

    ]
}

```

Параметр **payments** предназначен для указания режимов оплаты и их порядка на странице **вручную**. Если в этом параметре указаны какие-то режимы, то при проведении платежа будут доступны **только они** (при дополнительном условии, что соответствующий модуль подключен к договору):

```

paymentConfig {

    /* true -  /,  providers. */
    replaceProviders = false
    /*    Card */
    showCard = false
    /*    email/    */
    needReceiptContacts = true

    providers = [

    ]

    payments = [

        payment {
            title = 'payment.bankCard'
            image = 'static/images/logos/visa_mastercard.png'
            providerId = 'yamoney'
            type = 'bankCard'
        }
    ]
}

```

```

        config = config {
            paymentTypes = 'AC:payment.yamoney.AC'
        }
    },

    payment {
        title = 'payment.yamoney'
        image = 'static/images/logos/yamoney.png'
        providerId = 'yamoney'
        type = 'bankCard'
        config = config {
            paymentTypes = 'PC:payment.yamoney.PC'
        }
    },

    payment {
        title = 'payment.card'
        image = 'static/images/logos/card.png'
        providerId = 'card'
    }
}
]
}

```

Еще один пример:

```

paymentConfig {

    /* true - /, providers. */
    replaceProviders = false
    /* Card */
    showCard = false
    /* email/ */
    needReceiptContacts = true

    providers = [

        provider {
            id = "externalLinkQiwi"
            title = "Qiwi"
            image = 'static/images/logos/qiwi.png'
            config = config {
                url = 'https://qiwi.com/payment/form.action?provider=297'
            }
        }
    ]

    payments = [

        payment {
            title = 'payment.bankCard'
            image = 'static/images/logos/visa_mastercard.png'
            providerId = 'yamoney'
            type = 'bankCard'
            config = config {
                paymentTypes = 'AC:payment.yamoney.AC'
            }
        },

        payment {
            title = 'payment.yamoney'
            image = 'static/images/logos/yamoney.png'
            providerId = 'yamoney'
            type = 'bankCard'
            config = config {
                paymentTypes = 'PC:payment.yamoney.PC'
            }
        },

        payment {

```

```
        providerId = 'externalLinkQiwi'  
    }  
]  
}
```

Параметр **needReceiptContacts** указывает, нужно ли абоненту указывать email или телефон перед проведением оплаты (введенную информацию далее можно использовать при создании электронного чека).

Кастомизация личного кабинета

- [Редактирование файлов](#)
- [Вход в личный кабинет с другой страницы](#)
- [Ссылка на страницу](#)
- [Ограничение доступа](#)
- [Активация модуля](#)

Редактирование файлов

Файлы конфигурации, а также **.xhtml**-файлы можно редактировать. Однако, чтобы при очередном обновлении отредактированные файлы не перезаписались файлами из сборки, необходимо создать рядом файл с именем исходного файла плюс суффикс(расширение) **.orig**. Например, перед редактированием файла просто создайте его копию добавив окончание **.orig**. В этом случае при обновлении файл из сборки запишется не в исходный файл, а в файл **.orig**, а исходный, отредактированный вами останется без изменений.

Вход в личный кабинет с другой страницы

Можно создать форму входа в личный кабинет биллинга, например, на странице вашего сайта. Для этого URL (action) формы должен указывать на `index.xhtml`, а также параметры для аутентификации:

- **authMode** - режим авторизации из `mygbilling-conf.groovy`, число (порядковый номер начиная с нуля из списка `authentication.modes`) или строка (значение поля `authenticationMode.mode`, если эти значения уникальны). Если поле не указано, то используется первый режим аутентификации из списка;
- **username** - номер договора для `authenticationMode.mode=contract` или логин Inet для `authenticationMode.mode=login` (`authenticationMode.module=inet`);
- **password** - пароль для username.

Например, для такой конфигурации:

```
modes = [  
  authenticationMode {  
    mode = 'contract'  
  },  
  
  authenticationMode {  
    mode = 'login'  
    module = 'inet'  
    moduleId = 13  
  },  
  
  authenticationMode {  
    mode = 'ip'  
    module = 'inet'  
    moduleId = 13  
    role = 'unauthCustomer'  
    filter = { contract -> contractInGroup( contract, [4, 8, 15, 16, 23, 42] ) }  
  }  
]
```

можно использовать такую форму для аутентификации по номеру договора:

```
<form action="lk.provider.ru/index.xhtml">  
  <input type="text" name="username">  
  <input type="password" name="password">  
  <input type="submit">  
</form>
```

Для аутентификации по логину Inet:

```
<form action="lk.provider.ru/index.xhtml">  
  <input type="hidden" name="authMode" value="1">  
  <input type="text" name="username">  
  <input type="password" name="password">
```



```
<input type="submit">
</form>
```

Для аутентификации по IP-адресу:

```
<form action="lk.provider.ru/index.xhtml">
  <input type="hidden" name="authMode" value="ip">
  <input type="submit">
</form>
```

Или, для последнего случая, просто ссылкой:

Аутентификация по IP-адресу без страницы логина

```
<a href="lk.provider.ru/index.xhtml?authMode=ip"> </a>
```

Ссылка на страницу

В ссылке на личный кабинет можно указывать конкретную страницу (ядра и helpdesk), для этого в URL нужно указать параметр **page** со значением страницы, как она прописана в **mygbilling-menu.groovy**. Например:

```
<a href="lk.provider.ru/index.xhtml?page=plugins/helpdesk/helpdesk"></a>
```

Данный параметр можно совместить с аутентификацией с другой страницы (например, аутентификацией по IP-адресу):

```
<a href="lk.provider.ru/index.xhtml?authMode=ip&page=kernel/payment"></a>
```

Страница будет открыта, только если к ней разрешен доступ в **mygbilling-menu.groovy**.

Ограничение доступа

Для некоторых режимов аутентификации можно ограничить доступ к различным действиям и пунктам меню. Это может быть удобно, например, для режима аутентификации по IP-адресу. Для этого нужно указать в режиме аутентификации **role = 'unauthCustomer'**:

```
modes = [
  ...
  authenticationMode {
    mode = 'ip'
    module = 'inet'
    moduleId = 13
    role = 'unauthCustomer'
    //filter = { contract -> contractInGroup( contract, [4, 8, 15, 16, 23, 42] ) }
  }
]
```

В этом случае, если абонент аутентифицируется через данный режим, в **mygbilling-menu.groovy** и **mygbilling-conf.groovy** вызов **isCustomer()** будет возвращать **false**. Соответственно, например, данный пункт меню в этом случае будет недоступен абоненту:

```
//
menu( page: "kernel/tariffOptions", icon: "fa-cogs", title: "menu.tariffOptions", show:
isCustomer() ),
```

Т.к. у него параметр **show** зависит от **isCustomer()**. А изменение лимита запрещено (даже если пункт меню "Лимит" доступен), т.к. в **mygbilling-conf.groovy** указана зависимость:

```
// limit.xhtml
limit {
    //
    limitChange = { contract -> (contract.balanceMode == (byte)1) && isCustomer() }
}
```

Активация модуля

Вы можете указать дополнительные действия для активации модуля. Если на договоре модуль отсутствует, но доступно связанное доп. действие активации модуля - будет доступен пункт меню модуля, при нажатии на который абонент увидит форму выполнения доп. действия. Для привязки действия к активации модуля укажите **activateModuleActions** в блоке **context**:

```
context {
    ...
    // TV (mid=15, actionId=475603953)
    activateModuleActions[15] = [475603953]
}
```

Создание собственной страницы

Controller

Создайте в директории **WEB-INF/dyn** файл `MyNotificationBean.java` с содержимым:

```
import java.io.Serializable;

import javax.annotation.PostConstruct;
import javax.enterprise.context.RequestScoped;
import javax.enterprise.context.SessionScoped;
import javax.faces.view.ViewScoped;
import javax.inject.Inject;
import javax.inject.Named;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import ru.bitel.mybgbilling.kernel.common.AbstractBean;
import ru.bitel.mybgbilling.kernel.common.inject.BGInject;
import ru.bitel.mybgbilling.kernel.common.inject.BGInjection;

import ru.bitel.mybgbilling.kernel.contract.NotificationBean;
import ru.bitel.bgbilling.common.BGException;
import ru.bitel.bgbilling.kernel.contract.api.common.bean.ContractNotification;
import ru.bitel.bgbilling.kernel.contract.api.common.service.ContractNotificationService;

@Named
@ViewScoped // 1.
@BGInjection // 2.
public class MyNotificationBean
    extends AbstractBean
    implements Serializable
{
    private static final Logger logger = LoggerFactory.getLogger( MyNotificationBean.class );

    @BGInject // 2.
    private ContractNotificationService contractNotificationService;

    private List<ContractNotification> contractNotificationList;

    @Override
    protected void init() // 3.
        throws BGException
    {
        logger.info( "init" );

        populate();
    }

    public void populate() // 4.
    {
        logger.info( "populate" );

        contractNotificationList = contractNotificationService.contractNotificationList( getContractId() );
    }

    public List<ContractNotification> getContractNotificationList() // 5.
    {
        return contractNotificationList;
    }

    public void markRead( int id ) // 6.
        throws BGException
    {
        contractNotificationService.contractNotificationMarkRead( getContractId(), id );

        populate();
    }
}
```



```

dateTime}"> <!-- 6. -->
dateStyle="medium" />
markRead( notification.id }}" event="click" /> <!-- 6. -->
escape="false" /> <!-- 6. -->
form" class="form-inline"> <!-- 5. -->
update-target="page-content-data">#{msg['action.delete']}
{myNotificationBean.deleteContractNotification( notification.id }}"
page-content-data" /> <!-- 8. --> <!-- 9. -->
</ui:define>
</ui:composition>
<h:outputText value="#{notification.
    <f:convertDateTime type="both"
</h:outputText>
#{notification.title}
<f:ajax listener="#{myNotificationBean.
    </a>
</form>
    <div class="collapse" id="collapse#{notification.id}">
        <div class="well well-sm">
            <h:outputText value="#{notification.message}"
            <br /> <br />
        <form jsf:id="notificationDeleteForm" role="
        <button class="btn btn-primary confirm"
        <f:ajax listener="#"
            event="click" render=":
        </button>
    </form>
    </div>
</div>
</div>
</ui:repeat>
</div>
</div>
</ui:define>
</ui:composition>

```

1. Внутри блока `<ui:define name="page-content-header">` находится заголовок текущей страницы, например, "Новости" или "Уведомления".
2. Внутри блока `<ui:define name="page-content-data">` находится сама страница (её содержимое).
3. Атрибут `jsf:rendered` указывает, нужно ли отображать данный элемент.
4. Тэг `ui:repeat` осуществляет обработку списка `#{myNotificationBean.contractNotificationList}`, создавая новую переменную `#{notification}`, которая доступна внутри этого тэга-цикла.
5. Для того, чтобы работал вызов методов контроллера, а также тэг `f:ajax`, форма должна быть JSF-компонентом. Чтобы форма стала JSF-компонентом, достаточно указать атрибут `jsf:id`.
6. Обычно для отображения переменной достаточно запись вида `#{notification.title}`, однако, если необходимо форматирование, например, даты, то нужно использовать тэг `h:outputText`. Если содержимое переменной не нужно экранировать перед выводом (например, если внутри переменной-строки есть запись вида `"
"` и её нужно вывести как перевод строки, а не как строку `"
"`), то также используется тэг `h:outputText` с атрибутом `escape="false"`.
7. Тэг `f:ajax` с атрибутом `event="click"` внутри ссылки указывает, что на событие клика нужно вызвать метод `#{myNotificationBean.markRead(notification.id)}`. Обновление страницы здесь не происходит (только js на то же события клика меняет класс элемента для смены иконки).
8. Тэг `f:ajax` с атрибутом `event="click"` внутри кнопки указывает, что на событие нажатия кнопки нужно вызвать метод `#{myNotificationBean.deleteContractNotification(notification.id)}`.
9. Тэг `f:ajax` с атрибутом `render="page-content-data"` указывает, что после выполнения запроса нужно обновить область с `id page-content-data` (в данном случае это область внутри блока `<ui:define name="page-content-data">`).

Меню

Для того, чтобы страница отобразилась, на нее должна указывать специальная ссылка. Чтобы добавить ссылку в главное меню, нужно указать в **WEB-INF/mybbilling-menu.groovy** (также см. [Конфигурация меню](#)):

```

menu( page: "provider/myNotifications", subPage: "", icon: "fa-envelope-o", title: "menu.notifications", show:
isCustomer() ),

```

- `page` - путь к html странице, без суффикса `.html` (по аналогии уже с существующими пунктами меню);

- **subPage** - на странице могут быть подстраницы, в этом случае внутри xhtml-файла присутствует фильтр по subPage;
- **icon** - имя fontawesome-иконки;
- **title** - название пункта меню, как есть или ключ к Locale.properties;
- **show** - в каком случае отображать пункт меню, true - всегда, isCustomer() - только если абонент аутентифицирован по паролю.

Или можно указать ссылку прямо на странице:

```
<form jsf:id="linkForm" role="form" class="form-inline">
  <a jsf:action="#{navigationBean.setPage('provider/myNotifications', 0)}" update-target="page-wrapper">
    <i class="fa fa-rub fa-fw"></i>
    <f:ajax render=":page-wrapper" />
  </a>
</form>
```

В виде кнопки:

```
<form jsf:id="linkForm" role="form" class="form-inline">
  <button class="btn btn-primary confirm" update-target="page-wrapper">
    <f:ajax listener="#{navigationBean.setPage('provider/myNotifications', 0)}"
      event="click" render=":page-wrapper" />
  </button>
</form>
```

Содержание

[Раскрыть все](#) [Свернуть все](#)