

Н	Ф	Labs
---	---	------

Руководство администратора

«Единый клиент»

Оглавление

1	Требования к администратору CDI	7
1.1	Требования.....	7
1.2	Обязанности.....	7
1.3	Доступы	7
2	Архитектура системы	8
2.1	Типовая схема развёртывания	8
2.2	Компоненты типовой схемы развёртывания	8
2.3	Взаимосвязи типовой схемы развёртывания	9
2.4	Отказоустойчивое решение (горячий резерв)	9
2.4.1	Работа в режиме горячего резерва	10
2.4.2	Обновление без прерывания обслуживания.....	11
3	Вход в систему (администратор)	12
4	Задачи обработки данных	13
4.1	Группы задач	13
4.1.1	Особенности	14
4.1.2	Результат выполнения.....	14
4.1.3	Идентификация и слияние дубликатов	14
4.1.4	Проверка по черным спискам	15
4.1.5	Массовые атрибуты.....	15
4.1.6	Миграция	15
4.1.7	Повторная обработка контрагентов	15
4.1.8	Подготовка списков для повторной обработки	16
4.1.9	Системные задачи	17
4.1.10	Работа с задачами.....	17
4.1.11	Параметры, общие для всех типов задач	18
4.2	Планировщик задач	18
4.2.1	Запуск задачи по расписанию	18
4.2.2	Остановка выполнения задачи	20
4.3	Взаимодействие с JMS-очередями.....	21
4.3.1	Взаимодействие с JMS-очередями.....	21
4.3.2	Задача очистки хранилища JMS-сообщений.....	21
4.3.3	Задача переобработки сохраненных сообщений из входной очереди.....	21
4.3.4	Задача переправки сообщений в выходную очередь.....	22
4.3.5	Повторная отправка ошибочных JMS сообщений за последние 24 часа	23
4.4	Импорт данных.....	23
4.4.1	Особенности	24
4.4.2	Результат выполнения.....	24
4.4.3	Общие параметры для задач загрузки данных	24
4.4.4	Задача загрузки контрагентов из буферной таблицы БД	25
4.4.5	Задача загрузки договоров из буферной таблицы БД	26
4.4.6	Задача загрузки связей договоров из буферной таблицы БД	27
4.4.7	Задача загрузки связей	28
4.4.8	Задача загрузки контрагентов из CSV-файла	28
4.4.9	Задача загрузки контрагентов из каталога	29
4.4.10	Задача загрузки контрагентов по SFTP	30
4.4.11	Задача загрузки справочников ЕГРЮЛ / ЕГРИП	31
4.4.12	Трансформация данных	32
4.4.13	Активировать исходные записи.....	34
4.4.14	Деактивировать исходные записи	35
4.5	Идентификация и слияние дубликатов.....	36
4.5.1	Идентификация и слияние дубликатов	36
4.5.2	Кросс-сверка ФЛ-ЮЛ	36
4.5.3	Перестроение хэшей полного поиска дубликатов	37
4.5.4	Слияние дубликатов по внешнему списку.....	38
4.6	Повторная обработка	40
4.6.1	Повторная обработка контрагентов	40
4.6.2	Задача перерасчета hidden-флагов атрибутов по списку контрагентов.....	40



4.6.3	Задача полного переслияния контрагентов.....	41
4.6.4	Задача переслияния реквизитов контрагентов.....	43
4.6.5	Разлияние контрагентов по списку MergedId.....	44
4.6.6	Полное разлияние контрагентов по списку результирующих контрагентов	45
4.6.7	Задача восстановления атрибутов по списку HИDов	47
4.6.8	Задача актуализации адресов по КЛАДР	47
4.6.9	Исправьте базу данных в случае неконсистентного состояния	49
4.6.10	Переоткройте закрытых по ошибке контрагентов с сохранением истории	49
4.6.11	Актуализация атрибутов на основе данных staging	50
4.6.12	Закрытие актуальных (результирующих и всех исходных) атрибутов по списку	51
4.6.13	Задача очистки таблиц параметров для массовой переобработки данных	52
4.6.14	Подготовка списков для повторной обработки	52
4.6.15	Описание таблиц	53
4.6.16	Переобработка кодов качества паспортов по CDI-4750.....	54
4.6.17	Переочистка кодов качества паспортов по CDI-4145.....	55
4.6.18	Актуализация статуса невалидности документов	56
4.6.19	Закрытие исходных атрибутов по списку с переслиянием результирующих	57
4.6.20	Переобработка телефонов	58
4.6.21	Задача восстановления КА на версию по списку HИD	60
4.6.22	Задача переочистки реквизитов	61
4.7	Системные задачи	62
4.7.1	Системные задачи	62
4.7.2	allEnginesRebuildByName. Перестроение указанных поисковых индексов	62
4.7.3	Общее описание	63
4.7.4	Актуализация поискового индекса по списку контрагентов	63
4.7.5	Выполнение вспомогательной хранимой процедуры	64
4.7.6	Удаление старых обработанных конфликтов	64
4.7.7	Создание и редактирование системных учетных записей	65
4.7.8	searchEngineOptimize. Оптимизация поисковых индексов	67
4.7.9	graphEngineFullRebuild перестроение графа связей	67
4.7.10	enginesFullRebuild. Полное перестроение поискового индекса	68
4.8	Экспорт данных.....	68
4.8.1	Инкрементальный экспорт эталонных связей	68
4.8.2	Полный экспорт эталонных связей	69
4.9	Диагностика.....	70
4.9.1	Когда использовать.....	70
4.9.2	Как собрать диагностику	70
4.9.3	Сбор логов.....	70
4.9.4	Сбор диагностики по контрагенту.....	71
4.9.5	Сбор диагностики с указанным профилем.....	72
4.9.6	Проверка БД на непротиворечивость	73
4.9.7	Профили диагностики	75
4.9.8	Проверка слитых контрагентов на непротиворечивость	76
4.9.9	Найти и исправить проблемы в золотых карточках.....	77
4.9.10	Сбор диагностики в режиме горячего резерва.....	77
4.9.11	Сбор метрик.....	78
4.9.12	Сбор диагностики по задаче.....	78
4.9.13	Сбор логов за сегодня.....	79
4.9.14	Сбор метрик за предыдущий месяц	80
4.10	Триггеры	81
4.10.1	Триггер.....	81
4.10.2	Список триггеров	81
4.10.3	Работа с триггерами	82
4.10.4	Список триггеров	84
4.11	Возобновляемые задачи	87
4.11.1	Запуск на ночь.....	87
4.12	Непрерываемые задачи	88
4.13	Проверка по черным спискам	88
4.14	Необратимое удаление данных.....	88
4.14.1	Общее описание	88
4.14.2	Особенности задачи	88
4.14.3	Запуск задачи	88
4.14.4	Обработка ошибок	89



4.15	Заккрытие контрагентов	89
4.15.1	Общее описание	89
4.15.2	Особенности задачи	89
4.15.3	Запуск задачи	90
4.15.4	Обработка ошибок	90
4.16	Порционные задачи	90
4.16.1	Конфигурация задачи	90
4.16.2	Работа задачи	91
4.16.3	Подключение порционности	91
4.17	Проверка по ЕГРЮЛ	91
4.17.1	Полная проверка по ЕГРЮЛ	91
4.17.2	Проверка по ЕГРЮЛ контрагентов, требующих актуализации	92
4.18	Массовые атрибуты	92
4.18.1	Массовые атрибуты	92
4.18.2	Поиск массовых атрибутов	92
4.18.3	Проставление популярности по найденным значениям (ФЛ и ЮЛ)	93
4.19	Миграция	93
4.19.1	Автопрогон скриптов выдачи грантов пользователям	93
4.19.2	Инициализация прошлых версий атрибутов	93
4.19.3	Миграция нового поля	94
4.20	Миграция структуры хранения исторических данных	95
4.20.1	Форсирование миграции структуры исторических данных (Фаза 1)	95
4.20.2	Форсирование миграции структуры исторических данных (Фаза 3)	97
4.21	Сбор статистики	98
4.21.1	Описание задачи	98
4.21.2	Использование	99
4.21.3	Примеры запросов	100
4.21.4	Конфигурация задачи	101
4.21.5	Особенности задачи	102
4.21.6	DATA_METRIC	102
4.21.7	DATA_STATISTIC	102
5	Настройка системы	103
5.1	Мэппинг ролей LDAP	103
5.2	Настройки задач во внешней директории	104
5.3	Параметры JMS	104
5.3.1	Общие параметры	105
5.3.2	Специфичные для очередей TIBCO параметры	105
5.3.3	Специфичные для очередей IBM MQ параметры	105
5.4	Параметры LDAP	106
5.5	Параметры SOAP и REST сервисов	108
5.6	Параметры задач	110
5.6.1	Таймауты при запросах к длительным БД процедурам	113
5.7	Параметры задачи дедупликации	113
5.8	Параметры запуска JBOSS	114
5.9	Параметры импорта из буферной области	115
5.10	Параметры модуля мониторинга	115
5.11	Параметры отправки email	117
5.11.1	Механизм отправки писем об ошибках на почту	122
5.11.2	Фильтрация ошибок перед отправкой на почту	123
5.12	Параметры подключения к Фактору	126
5.13	Параметры поискового индекса	129
5.13.1	Неблокирующее перестроение поискового индекса	132
5.14	Параметры сервисов	133
5.14.1	Стратегия решения конфликтов	136
5.15	Параметры синхронизации нод	137
5.16	Параметры экспорта данных	138
6	Работа со справочниками	140
6.1	Добавление в справочник новой записи	140
6.2	Модификация существующей в справочнике записи	140
6.3	Представление с содержимым справочников	141



6.4	Список справочников, которые можно пополнять и редактировать	141
6.5	Примеры использования процедур (Oracle)	141
6.6	Примеры использования процедур (MariaDB).....	142
7	Журналирование и аудит	143
7.1	Журналы Единого клиента	143
7.1.1	Журнал аутентификаций.....	144
7.1.2	Журнал доступа к персональным данным	145
7.1.3	Журнал исключенных ошибок.....	145
7.1.4	Журнал обновлений версий сервера	146
7.1.5	Логи задачи трансформации	146
7.1.6	Лог триггера.....	148
7.1.7	Единый клиент (cdi-soap-stats.log)	149
7.1.8	Трассировка SOAP-запросов	150
7.2	Журналы Фактора	151
7.3	Уровни протоколирования.....	151
7.4	Аудит	152
7.4.1	Поля записи журнала	152
7.4.2	Типы событий	153
8	Установка обновлений	156
8.1	Общее описание	156
8.2	Обновление модулей.....	156
8.2.1	Обновление модуля Единый Клиент.....	156
8.2.2	Обновление модуля Фактор.....	157
8.3	Ручной запуск и остановка.....	157
8.4	Обновление версии Java Runtime Environment.....	157
8.5	Обновление модуля Единый Клиент	158
8.6	Обновление модуля Фактор.....	158
8.7	Автоматическая миграция на новую версию	158
8.7.1	1. При старте ЕК понимает, что нужно мигрировать на новую версию	158
8.7.2	2. Обновляется БД	158
8.7.3	3. Перестраиваются поисковые индексы.....	158
8.7.4	4. Собирается диагностика.....	159
8.7.5	Автоматическая миграция на новую версию (техподдержка)	159
8.8	Инструкция по переходу с JBoss 4 на JBoss 7 для администраторов	161
8.8.1	Инструкция для Linux	161
8.8.2	Инструкция для Windows	163
8.8.3	Настройка datasource	165
8.9	Инструкция по переходу с JBoss 7 на WildFly 8 для администраторов.....	166
8.9.1	Инструкция для Linux (переход на WildFly 8)	166
8.9.2	Инструкция для Windows (переход на WildFly 8).....	169
8.10	Инструкция по переносу дампа БД.....	170
8.10.1	Экспорт.....	170
8.10.2	Импорт.....	171
8.10.3	Создание oracle-директории	172
8.11	Автоматическое создание схемы CDI	173
8.11.1	Настройка WildFly.....	173
8.11.2	Создание новой схемы	173
8.11.3	Пересоздание существующей схемы	174
8.12	Инструкция по переходу с Wildfly 8 на WildFly 10 для администраторов.....	174
8.12.1	Инструкция для Linux (переход на WildFly 10)	174
8.12.2	Инструкция для Windows (переход на WildFly 10).....	177
9	FAQ	180
9.1	Основное (FAQ).....	180
9.1.1	CPU под 100% на задаче.....	180
9.1.2	Индекс неактуальный (Index may be corrupted).....	180
9.1.3	Как переместить прод с одной машины на другую	180
9.1.4	Как поднять бекап базы с PROD на TEST стенде?	181
9.1.5	Что такое "схлопывание" атрибутов?	182
9.2	Менеджер Данных (FAQ)	182
9.2.1	NullPointerException при попытке просмотреть историю в CDM.....	182



9.2.2	Веб-CDM: ошибка "Неактуальная группа дубликатов. Обновите данные или вернитесь к списку групп."	183
9.2.3	Другие ошибки при работе в APM Менеджер данных	183
9.2.4	Запуск CDM на машине, где уже установлена Java 1.8.x	183
9.2.5	Невозможно запустить приложение APM Менеджера данных (Unable to launch application)	185
9.2.6	Не могу зайти в APM Менеджера данных, неверный пароль или недостаточно прав для доступа	186
9.2.7	Не могу зайти в APM Менеджера данных, некорректная версия Java	187
9.2.8	Не могу зайти в APM Менеджера данных, ошибка соединения с сервером	187
9.2.9	Отслеживание логинов в CDM	188
9.3	APM Администратора (FAQ)	188
9.3.1	В IE* нельзя создать триггер - не удается выбрать задачи из списка	188
9.3.2	Изменение параметра oldMaxId	188
9.4	Дополнительно (FAQ)	189
9.4.1	Запросы в БД, соответствующие фильтрам в CDM	189
9.4.2	Как пополнить справочник имен / фамилий?	190
9.5	Инструменты для проверки (FAQ)	191
9.5.1	Как снять Thread Dump	191
9.5.2	Как снять Thread Dump через командную строку в Unix	192
9.5.3	Как можно вызвать метод SOAP-интерфейса?	193
9.5.4	Как можно вызвать REST-метод вручную	200
9.5.5	Как проверить учетную запись LDAP?	203
9.5.6	Как посмотреть JMS, уходящие в Tibco	206
9.5.7	Как послать в Tibco входящее JMS-сообщение	209
9.5.8	Как настроить приветствие на Linux	216
9.5.9	Как по ID в исходной системе понять с кем слит контрагент	217
9.5.10	Как посмотреть JMS, уходящие в RabbitMQ	217
9.6	Запуск задач / триггеров	219
9.6.1	Фазы миграции на _HIST таблицы	219
9.6.2	Что делать, если инкремент упал с ошибкой	222
9.7	FAQ для сопровождающих	223
9.8	Фактор (FAQ)	223
9.8.1	Есть ли doBatchClean для телефонов?	223
9.9	Интеграция (faq)	224
9.9.1	Как передать двух клиентов с одной связью	224
9.10	Обновление справочников (FAQ)	225
9.10.1	Я обновил перечень ПНП на Linux, но он не подхватывается	225
10	Как добавить или изменить новый тип атрибута	226
10.1	Создать новый тип атрибута	226
10.1.1	Добавить новый тип в ЕК	226
10.1.2	Добавить новый тип в Фактор	226
10.1.3	Результат	226



1 Требования к администратору CDI

1.1 Требования

Ниже перечислены требования к администратору системы:

- Базовые знания Oracle (создание БД, прогон скриптов, анализ ошибок).
- Базовые навыки администрирования Java EE (понимание архитектуры контейнер/приложение, архитектуры java приложений).
- Базовые навыки сетевого администрирования / администрирования Windows (например, умение определить почему порт открыт, а соединения не проходят).
- Базовые знания JBOSS (рестарт сервера, анализ логов)
- Технический английский (прочитать и понять текст ошибки JBOSS/Oracle).

1.2 Обязанности

Обязанности администратора системы включают в себя:

- Базовую настройку системы:
 - триггеры выполнения задач;
 - уровни логирования и пр.
- Обеспечение непрерывности работы системы:
 - установка обновлений системы;
 - установка обновлений JRE;
 - мониторинг логов системы;
 - бэкапирование базы и настроек системы;
 - решение текущих проблем по ходу эксплуатации;
 - восстановление системы после сбоев.
- Передачу запросов пользователей в поддержку Единый Клиент и отслеживание их выполнения.

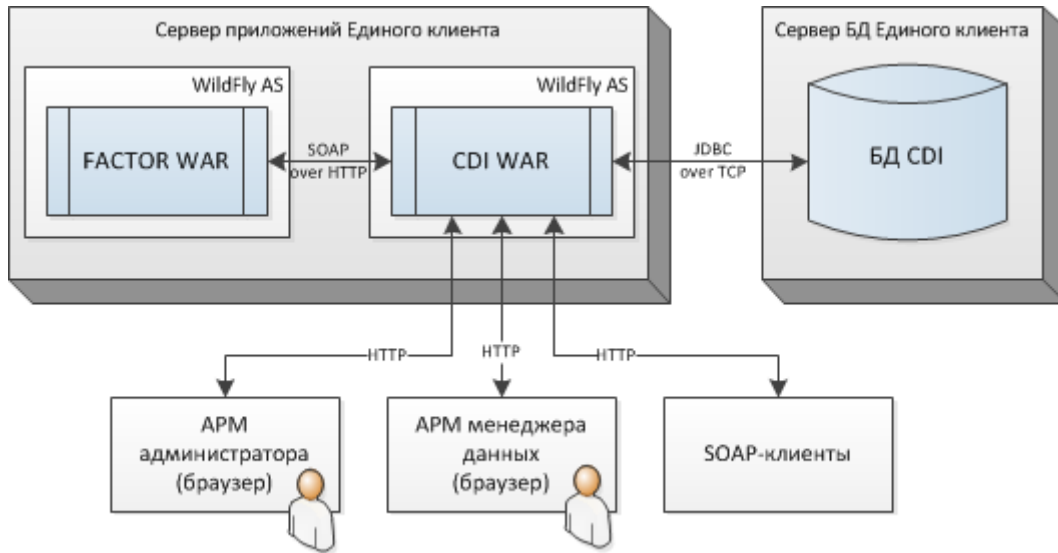
1.3 Доступы

Администратору CDI необходимы права администратора на сервере приложений.

2 Архитектура системы

2.1 Типовая схема развёртывания

На рисунке ниже представлена типовая схема развёртывания системы Единый клиент:



В данном разделе описана схема развёртывания одного экземпляра системы, без обеспечения отказоустойчивости. Реализация отказоустойчивого решения возможна за счет использования [горячего резерва](#).

2.2 Компоненты типовой схемы развёртывания

Компонент схемы	Назначение
АРМ Администратора	Автоматизированное рабочее место администратора системы. Реализовано как тонкий клиент, доступный через браузер.
АРМ МД	Автоматизированное рабочее место менеджера данных. Предоставляет графический интерфейс пользователя. С помощью данного модуля конечные пользователи могут работать с системой. Реализовано как тонкий клиент, доступный через браузер.
SOAP-клиенты	Внешние системы/сервисы. Используют функциональность Единого клиента через предоставляемые системой SOAP-сервисы.
WildFly AS	WildFly Application Server, не ниже версии 8.2.0+. Предоставляет контейнер для работы серверов Единого клиента и ФАКТОР.
CDI WAR	Модуль сервера Единый клиент. Исполняется в контейнере WildFly AS. Обеспечивает выполнение функций системы Единый клиент.
ФАКТОР WAR	Модуль сервера ФАКТОР. Обеспечивает выполнение функций системы ФАКТОР.
БД CDI	База данных Единый клиент. Реализуется на базе СУБД Oracle или MariaDB.

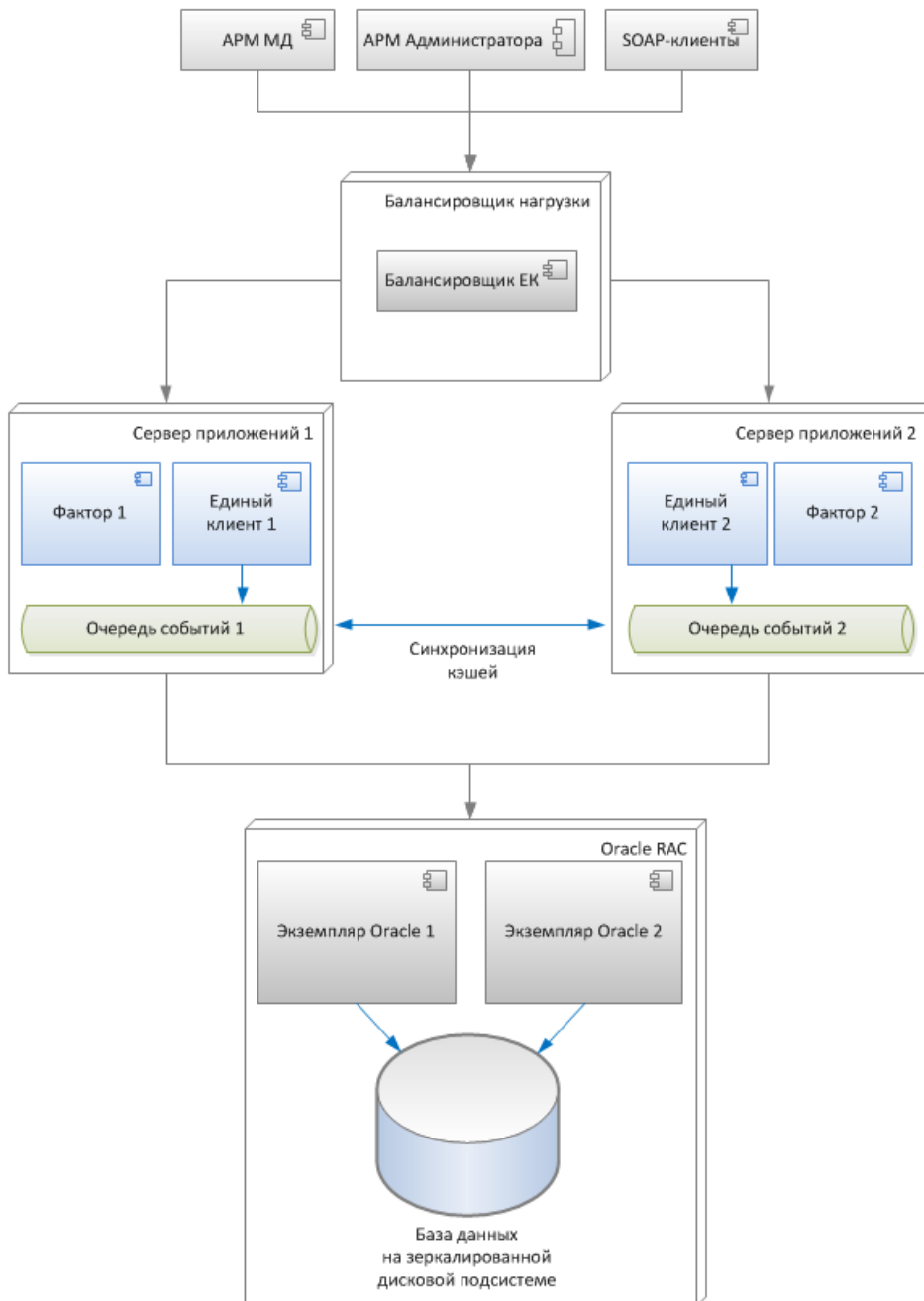
2.3 Взаимосвязи типовой схемы развертывания

Тип взаимосвязи	Компоненты	Назначение
JDBC over TCP	CDI WAR – БД CDI	Сервер Единого клиента взаимодействует с базой данных для работы с информацией о клиентах, находящейся под управлением Единого клиента.
HTTP	Браузер – CDI WAR	Автоматизированные рабочие места администратора и менеджера данных реализованы как тонкий клиент, доступный через браузер.
SOAP over HTTP	CDI WAR – ФАКТОР WAR	Взаимодействие между системами Единый клиент и ФАКТОР реализуется с помощью вызова SOAP-сервисов, предоставляемых системой ФАКТОР.
SOAP over HTTP	SOAP-клиенты – CDI WAR	Взаимодействие между внешними системами/сервисами и системой Единый клиент реализуется с помощью вызова SOAP-сервисов, предоставляемых последней.

2.4 Отказоустойчивое решение (горячий резерв)

Отказоустойчивая схема развертывания системы в варианте с горячим резервом подразумевает два активных экземпляра системы, которые работают одновременно. Запросы от пользователей и внешних систем балансируются между экземплярами, а при отказе одного из них все новые запросы направляются на второй.

Общая схема развертывания системы приведена на рисунке ниже:



2.4.1 Работа в режиме горячего резерва

Сценарий использования выглядит следующим образом:

1. Клиенты (АРМ Менеджера данных, АРМ Администратора, SOAP-клиенты) обращаются к балансировщику нагрузки, который направляет запросы на тот или иной экземпляр ЕК.
2. При необходимости доступа к данным ЕК обращается к кластеру СУБД, реализованному средствами Oracle RAC.
3. Кэши, необходимые для работы ЕК (поисковый индекс и кэш поиска дубликатов) автоматически синхронизируются между экземплярами системы через очередь сообщений.

4. Общие настройки всех экземпляров ЕК хранятся в БД.
5. Запуск периодических задач координируется на уровне СУБД.
6. На физическом уровне для хранения базы данных используется высоконадежный дисковый массив с зеркалированием.

При отказе одного из экземпляров ЕК балансировщик начинает перенаправлять все запросы на второй экземпляр. Когда первый экземпляр восстанавливает работоспособность, балансировщик снова начинает распределять нагрузку между обоими экземплярами.

Для реализации описанной схемы от заказчика требуется:

1. Приобрести дополнительные лицензии ЕК / ФАКТОР.
2. Предоставить два сервера приложений для ЕК.
3. Предоставить и настроить балансировщик нагрузки, поддерживающий прикрепление пользователя к ноде с использованием cookie (Nginx Plus, HAProxy, Tengine).
4. Предоставить два сервера СУБД и дисковую подсистему для БД; установить и настроить Oracle RAC.
5. Настроить мониторинг отдельных компонентов схемы, и выделить необходимые человеческие ресурсы для оперативного реагирования на отказы.

2.4.2 Обновление без прерывания обслуживания

Типичное обновление ЕК при переходе на следующую версию может включать в себя следующие шаги:

1. Обновление ПО ЕК.
2. Обновление ПО Фактора.
3. Миграция схемы БД.
4. Актуализация кэшей (как правило, перестроение поискового индекса).

Обновление ПО (пункты 1 и 2) и актуализация кэшей (пункт 4) может выполняться без прерывания обслуживания (сначала на одном экземпляре ЕК, затем на втором). Миграция схемы БД также может проводиться без остановки сервиса в том случае, если изменения обратно совместимы с предыдущей версией.

В случае же несовместимых изменений схемы базы данных требуется остановка всех экземпляров ЕК для проведения миграции.

3 Вход в систему (администратор)

Страница входа: http://CEPBEP_EK:8080/cdi/.

Пожалуйста, укажите логин и пароль для входа в интерфейс управления.

Программный комплекс «Единый клиент» предназначен для централизованного хранения, обеспечения качества и синхронизации информации о клиентах между различными информационными системами.

Логин

Пароль

Запомнить меня

2005-2018 © HFLabs

Версия продукта: 18.4-SNAPSHOT (5816a1fa7718)
Версия ядра: 18.4-SNAPSHOT (4cb09b282b5c)
ФАКТОР localhost:18080: trunk/ (96710)

Чтобы каждый раз не вводить логин и пароль, поставьте галочку *Запомнить меня*.

Если вход не работает, убедитесь, что в браузере включены куки (cookies).

4 Задачи обработки данных

H F Labs
SOAP
| Пользователь: admin
[Выйти]

Расписание
Задачи
Триггеры
Конфигурация
Аудит

Список задач

Дедупликация и слияние

Действия	Название	Описание
Детали Редактировать Выполнить	findDuplicates	Поиск дубликатов
Детали Редактировать Выполнить	findDuplicatesAndMerge	Поиск дубликатов и слияние
Детали Редактировать Выполнить	findDuplicatesAndMergeAndRebuildGroups	Поиск дубликатов, слияние и перестроение групп дубликатов
Детали Редактировать Выполнить	importExternalDuplicateWithMerge	Загрузка и слияние внешних дубликатов
Детали Редактировать Выполнить	incrementDuplicates	Инкрементальный поиск дубликатов без слияния
Детали Редактировать Выполнить	incrementDuplicatesAndMerge	Инкрементальный поиск дубликатов и слияние
Детали Редактировать Выполнить	merge	Объединение дубликатов
Детали Редактировать Выполнить	rebuildFullDedupHashes	Перестроение хэшей полного поиска дубликатов

Диагностика

Действия	Название	Описание
Детали Редактировать Выполнить	diagnosticLogs	Сбор логов
Детали Редактировать Выполнить	diagnosticMetrics	Сбор метрик

4.1 Группы задач

Задачи обработки данных отвечают за пакетную загрузку и преобразование данных, находящихся под управлением системы. Конкретный набор задач определяется конфигурацией заказчика. Описание типовых задач приведено далее по тексту.

Импорт данных

Группа задач, предназначенных для загрузки данных из внешних источников в Единый клиент.

Импорт данных

Действия	Название	Описание
Детали Редактировать Выполнить	disableStagingRecords	Деактивировать исходные записи
Детали Редактировать Выполнить	enableStagingRecords	Активировать исходные записи
Детали Редактировать Выполнить	physicalBufferIPC	Загрузка инкремента ФЛ
Детали Редактировать Выполнить	transform	Трансформация данных
Детали Редактировать Выполнить	transformWithCustomConfig	Трансформация данных с указанными параметрами

Типовые задачи, которые есть всегда:

Название	Описание	Комментарий
disableStagingRecords	Деактивировать исходные записи	Временно приостановить обработку записей
enableStagingRecords	Активировать исходные записи	Снова запустить обработку записей
transform	Трансформация данных	Если загрузка инкремента падает с ошибкой, можно догрузить в ЕК необработанные данные с помощью данной задачи

Название	Описание	Комментарий
transformWithCustomConfig	Трансформация данных с указанными параметрами	Когда нужно при отработке задачи трансформации обновить запись принудительно.

Общий список задач по загрузке данных (задачи подключаются сотрудниками ХФЛабс)

4.1.1 Особенности

Все задачи импорта данных, которые используют [Трансформация данных](#), могут работать в двух режимах:

- с оповещением поискового индекса КА о трансформации пачек;
- без оповещения поискового индекса.

Режим запуска определяется значением параметра:

Параметр	Тип	Описание	Значение по умолчанию
notifyListeners	Логическое значение	Оповещение поискового индекса КА о трансформации пачек. Если оповещение отключено, после завершения задачи нужно перестроить индекс.	true

4.1.2 Результат выполнения

Если данные загружены успешно, то задачи и триггеры завершаются со статусом "FINISHED".

Если данных для загрузки не было, то задачи и триггеры завершаются со статусом "SKIPPED".

4.1.3 Идентификация и слияние дубликатов

Группа задач, предназначенных для идентификации и слияния дубликатов:

- findDuplicates – [поиск дубликатов](#) по всей базе контрагентов;
- merge – [слияние](#) гарантированных и подтвержденных дубликатов;
- findDuplicatesAndMerge – [поиск дубликатов](#) по всей базе контрагентов с последующим [слиянием](#) гарантированных дубликатов;
- findDuplicatesAndMergeAndRebuildGroups – поиск дубликатов по всей базе контрагентов, слияние гарантированных и перестроение индекса негарантированных групп дублей;
- incrementDuplicatesAndMerge – [инкрементальный поиск дубликатов](#) (дубликаты ищутся среди контрагентов, по которым были изменения после предыдущего поиска дубликатов) с последующим [слиянием](#) гарантированных дубликатов;

- `importExternalDuplicateWithMerge` – слияние дубликатов по внешнему списку;
- `rebuildFullDedupHashes` – перестроение хэшей полного поиска дубликатов.

4.1.4 Проверка по черным спискам

Группа задач, предназначенных для проверки данных по черным спискам:

- `fullBlacklistCheck` – полная проверка контрагентов по черным спискам;
- `actualizeBlacklistCheck` – проверки по черным спискам контрагентов, требующих актуализации. Задача проверяет только новых и измененных клиентов;
- `rebuildBlacklistCheckHashes` – перестроение хэшей для проверки ЧС (принудительно перестраиваются онлайн хэши).

4.1.5 Массовые атрибуты

Группа задач, предназначенных для выявления и проставления популярности массовым атрибутам:

- `findPopularValues` – найти массовые атрибуты;
- `markPopularPhysicalAttributes` – проставить популярность по найденным значениям (ФЛ);
- `markPopularLegalAttributes` – проставить популярность по найденным значениям (ЮЛ);
- `truncatePopularValues` – сбросить накопленную информацию о массовых атрибутах на факторе (сброс индекса).

4.1.6 Миграция

Задачи, предназначенные для первоначальной загрузки данных по историческому объему клиентов, уже хранящихся в системе, в случае добавления нового реквизита

- `innMigrateRequisiteTask` – Миграция поля ИНН
- `kiomigrateRequisiteTask` – Миграция поля КИО

4.1.7 Повторная обработка контрагентов

Группа задач, предназначенных для повторной обработки определенного подмножества контрагентов:

- `addressActualization` – актуализация адресов по КЛАДР;
- `attributeActualizeByStaging` – актуализация атрибутов на основе данных staging;
- `attributeRecoverByHids` – восстановление атрибутов по списку HIDs;
- `attributeCloseActualByHids` – восстановление атрибутов по списку HIDs;
- `hiddenFlagRefineByHids` – пересчет hidden-флагов атрибутов по списку контрагентов;

- partyReopenByHids – переоткрытие контрагентов по списку HIDов
- reMergeFullByPartyHids – полное переслияние по списку HID'ов результирующих контрагентов;
- reMergeRequisiteByPartyHids – переслияние реквизитов по списку HID'ов результирующих контрагентов;
- unMergeByMergedIds – разлияние контрагентов по списку MergedId;
- fullUnmergeByResultHids – полное разлияние контрагентов по списку результирующих контрагентов;
- terroristStatusReClean – переобработка по справочнику террористов.
- phoneExtraFieldsReCleaner – переобработка дополнительных полей телефонов.
- partyRecoverByHids – восстановление КА на версию по списку HID.

4.1.8 Подготовка списков для повторной обработки

Группа задач, предназначенных для заполнения таблиц с параметрами для массовой переобработки данных :

```
fillTaskPartyHidsActual, fillTaskPartyHidsOriginal,
fillTaskPartyHidsResulted, fillAddressActualizationTaskPartyHids,
fillTaskPartyHidsBadAddresses, cleanTaskParameterTables
```

Данные задачи следует выполнять перед запуском задачи [повторной обработки выборочных контрагентов](#).

В результате работы задачи таблица TASK_PARTY_HIDS будет наполнена идентификаторами контрагентов:

- fillTaskPartyHidsActual – все актуальные контрагенты (merged_status 0 или 2);
- fillTaskPartyHidsOriginal – все оригинальные контрагенты (merged_status 0 или 1);
- fillTaskPartyHidsResulted – все результирующие контрагенты (merged_status 2);
- cleanTaskParameterTables – очистка таблиц с параметрами задач массовой переобработки.

Для заполнения таблицы TASK_ACT_PARTY_HIDS предназначены отдельные задачи:

- fillAddressActualizationTaskPartyHids (отображается в АРМ Администратора только, если подключена [Задача актуализации адресов по КЛАДР](#)) – контрагенты, адреса которых должны быть актуализированы по КЛАДР;
- fillTaskPartyHidsBadAddresses – контрагенты с адресами, отмеченными на ручную обработку (mergedStatus 0 или 1), задача не выбирает контрагентов, если все его адреса, отмеченные на ручную обработку, скрыты.

Для заполнения TASK_DOC_PARTY_HIDS:

- fillDocInvalidStatusActTaskPartyHids – все результирующие контрагенты для переобработки статуса актуальности паспортов. Задача отдельная, чтобы переобработка паспортов не блокировала работу других задач

4.1.9 Системные задачи

- `changeJdbcUser` – изменение пользователя безопасности

Задачи, предназначенные для работы с индексом полнотекстового поиска и другими внутрисистемными структурами:

- `deleteOutdatedConflicts` – удаление старых обработанных конфликтов;
- `duplicateGroupFullRebuild` – полное перестроение групп дубликатов для сравнения вручную;
- `enginesFullRebuild` – полное перестроение поискового индекса;
- `graphEngineFullRebuild` – полное перестроение графа связей;
- `conflictEngineFullRebuild` – полное перестроение индекса конфликтов обновления;
- `allEnginesRebuildByName` – перестроение указанного поискового индекса;
- `allEnginesRebuildInvalid` – перестроение всех не прошедших валидацию индексов;
- `searchEngineActualizeByParties` – актуализация поискового индекса по списку контрагентов;
- `searchEngineOptimize` – оптимизация поисковых индексов (граф связей, индексы по контрагентам, конфликты), группы дубликатов не оптимизируются;
- `searchEngineOptimizeByName` – оптимизация указанных поисковых индексов;
- `searchEngineValidate` – валидация всех поисковых индексов;
- `executeDBProcedure` – выполнение хранимой процедуры.

Группа задач, предназначенных для экспорта данных из Единого клиента:

- `fullExportRelations` – полный экспорт эталонных связей;
- `incrementalExportRelations` – инкрементальный экспорт эталонных связей.

Экспорт данных		
Действия	Название	Описание
Детали Редактировать Выполнить	<code>fullExportRelations</code>	Полный экспорт эталонных связей
Детали Редактировать Выполнить	<code>incrementalExportRelations</code>	Инкрементальный экспорт эталонных связей

4.1.10 Работа с задачами

Администратор имеет возможность:

- Просмотреть детальную информацию о задаче;
- Изменить параметры задачи;
- Поставить задачу на выполнение.

Выполнение любой задачи также может быть инициировано автоматически по заданному расписанию (триггеру).

4.1.11 Параметры, общие для всех типов задач

Параметр	Тип	Описание
executionTimeout	Целое	Таймаут в минутах, по истечению которого выполнение задачи будет прервано

4.2 Планировщик задач

Планировщик задач запускается автоматически после старта сервера. Он необходим для выполнения различных задач по загрузке и обработке данных с ручным запуском, а также для выполнения тех задач, для которых сконфигурировано выполнение [по расписанию](#).

The screenshot shows the 'История' (History) page in the JBoss Admin Console. At the top, there are navigation tabs: 'Расписание', 'Задачи', 'Триггеры', 'Конфигурация', and 'Аудит'. Below the tabs, there is a section titled 'История' with a refresh button and a dropdown menu for automatic updates (set to 5 seconds). A table displays the execution history of tasks.

Название	Дата старта	Дата окончания	Статус	Комментарий
Загрузка инкремента ФЛ	22.03.2018 11:56:28	22.03.2018 11:56:28	FINISHED	OK
Обновление метрик данных	22.03.2018 02:00:00	22.03.2018 02:00:00	FINISHED	OK
Загрузка инкремента ФЛ	21.03.2018 18:22:19	21.03.2018 18:22:20	FINISHED	OK
Загрузка инкремента ФЛ	21.03.2018 17:03:30	21.03.2018 17:03:31	FINISHED	OK

На странице отображается таблица с результатами выполнения последних задач (в обратном хронологическом порядке):

- *Название* — наименование задачи на русском языке.
- *Дата старта* и *Дата окончания* — информация о том, когда началось и закончилось выполнение задачи.
- *Статус* — результат выполнения задачи (FINISHED — успешное завершение, ERROR — ошибка, CANCELLED — задача была прервана).
- *Комментарий* — дополнительная информация о результатах выполнения задачи.

4.2.1 Запуск задачи по расписанию

Самый простой способ поставить задачу на выполнение по расписанию — [создать триггер в АРМ Администратора](#). Минус данного подхода в том, что триггер сохранится и будет работать только до перезапуска ЕК.

Если нужно, чтобы триггер сохранился и после перезапуска ЕК, необходимо добавить его в конфигурацию:

1. Посмотреть имя интересующей задачи в АРМ Администратора (страница *Задачи*, столбец *Название*).
2. Остановить Единый клиент.
3. Зайти в архив `{JBOSS_HOME}/server/default/deploy/cdi-web-{version}.war/WEB-INF/classes`.
4. Отредактировать настройки задачи (находятся в файлах `jobs_flow*.xml` или `jobs_import*.xml`) как описано ниже.
5. Упаковать отредактированный файл обратно в архив (`{JBOSS_HOME}/server/default/deploy/cdi-web-{version}.war/WEB-INF/classes`).
6. Запустить Единый клиент.

В файле с настройками необходимо найти задачу (элемент `<bean parent="abstractTaskInstance">`) по ее имени, добавить элемент `<bean parent="abstractTaskTrigger">`, как в примере ниже:

```
<beans>

  <bean parent="abstractTaskInstance">
    <property name="key" value="PhysicalBufferIPC"/>
    <property name="group" value="import"/>
    <property name="description" value="Загрузка инкремента ФЛ"/>
    <property name="performerName" value="bufferUploadTaskPerformer"/>
    <property name="taskParameters">
      <util:map>
        <entry key="sourceSystem" value="bufferConnector"/>
        <entry key="essenceType" value="PHYSICAL"/>
        <entry key="bufferDataReadBlockSize" value="1000"/>
        <entry key="oldMaxIdPropertyName" value="oldMaxId.buffer.physical"/>
      </util:map>
    </property>
  </bean>

  <bean parent="abstractTaskTrigger">
    <property name="triggerKey" value="PhysicalBufferIPCTrigger"/>
    <property name="cronExpression" value="0 0 22 ? * SAT"/>
    <property name="taskChain">
      <util:list>
        <value>PhysicalBufferIPC</value>
      </util:list>
    </property>
  </bean>

</beans>
```

Само расписание запуска настраивается в строке:

```
<property name="cronExpression" value="0 0 22 ? * SAT"/>
```

Используется формат записи утилиты `cron` для UNIX-систем. Выражение состоит из 6 колонок, разделяемых пробелами или табуляторами. Они задают время выполнения (секунда, минута, час, день, месяц, день недели), в них может находиться:

- число;
- список чисел, разделенных запятыми;
- диапазон чисел, разделенных тире;
- символ *;
- символ ?.

```
*****
-----
|||||
|||||---- День недели (MON - SUN)
|||||---- Месяц (1 - 12)
|||||---- День (1 - 31)
```

|| ----- Час (0 - 23)
 | ----- Минута (0 - 59)
 ----- Секунда (0-59)

Примеры:

Выражение	Расшифровка
0 0 12 * * ?	Каждый день в 12:00
0 15 10 ? * *	Каждый день в 10:15
0 15 10 * * ?	Каждый день в 10:15
0 20 13 ? * SUN	Каждое воскресенье в 13:20
0 * 14 * * ?	Каждую минуту в интервале 14:00 – 14:59
0 0/5 14 * * ?	Каждые пять минут в интервале 14:00 – 14:55
0 0/5 14,18 * * ?	Каждые пять минут в интервале 14:00 – 14:55 и 18:00 – 18:55
0 0 8-19 * * ?	Каждый час с 8 до 19
0 0/30 * 2 * ?	Каждые 30 минут 2 числа каждого месяца
0 */10 * ? * *	Каждые 10 минут

4.2.2 Остановка выполнения задачи

Запущенную задачу можно остановить двумя способами:

- **Завершить** - мягкая остановка. Система ждет удобного момента для прерывания (например, пока будет дообработан кусок данных) и останавливает задачу.

Действие доступно только для прерываемых задач.

Запущенные задачи автоматическое обновление через: 5 сек

Группа	Название	Статус	Описание	Комментарий	Действия
system	searchEngineOptimize	PENDING	Оптимизация всех поисковых индексов		Завершить Детали

- **Убить** - жесткая остановка. Система принудительно завершает все процессы задачи. Несохранный прогресс выполнения (например недообработанный кусок данных) будет потерян.

Действие доступно для непрерываемых задач и задач, которые в данный момент завершаются (статус CANCELLING).

Данная остановка может приводить к ошибкам уровня ERROR в логах (они будут высланы на почту).

Запущенные задачи автоматическое обновление через: 5 сек

Группа	Название	Статус	Описание	Комментарий	Действия
dedup	findDuplicatesAndMergeAndRebuildGroups	RUNNING	Поиск дубликатов, слияние и перестроение групп дубликатов		Убить Детали

4.3 Взаимодействие с JMS-очередями

4.3.1 Взаимодействие с JMS-очередями

Группа задач, предназначенных для взаимодействия с JMS-очередями:

4.3.2 Задача очистки хранилища JMS-сообщений

CleanEventsPersistence

4.3.2.1 Описание задачи

Очищает системную таблицу, в которой хранятся JMS-сообщения, в соответствии с указанными условиями фильтрации. Удаляются как входные, так и выходные сообщения.

Рекомендуется выполнять задачу периодически, чтобы не накапливать сообщения в БД до бесконечности (т.к. это может привести к проблемам с производительностью БД или к нехватке дискового пространства на сервере БД).

4.3.2.2 Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
olderThanInDays	Целое	Удалять сообщения, созданные более чем olderThanInDays дней назад	30
successOnly	Логическое значение	Удалять только сообщения со статусом SUCCESS или с любым	true

4.3.3 Задача переобработки сохраненных сообщений из входной очереди

ReprocessIncomingEvents

4.3.3.1 Описание задачи

Переобрабатывает сообщения, поступившие во входную очередь и сохраненные в БД Единого Клиента, в соответствии с указанными условиями фильтрации.

Предназначена для повторной обработки тех сообщений, которые не удалось обработать до этого при выборке из входной очереди. Такие сообщения имеют статус ERROR.

4.3.3.2 Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
minCreateDate	Дата в формате dd.mm.yyyy hh:mi:ss	Нижняя граница (включительно) даты создания сообщений, которые будут переобработаны	

Параметр	Тип	Описание	Значение по умолчанию
maxCreateDate	Дата в формате dd.mm.yyyy hh:mi:ss	Верхняя граница (включительно) даты создания сообщений, которые будут переобработаны	

4.3.4 Задача переотправки сообщений в выходную очередь

RepublishOutgoingEvents

4.3.4.1 Описание задачи

Переотправляет в выходную очередь сообщения в соответствии с указанными условиями фильтрации.

Предназначена для следующих ситуаций:

- Повторная выгрузка сообщений, которые не удалось записать в выходную очередь ранее (такие сообщения имеют статус ERROR),
- Повторная выгрузка сообщений, которые ранее были переданы в выходную очередь, но по каким-либо причинам не дошли до систем-получателей. Такие сообщения в ЕК имеют статус SUCCESS, а факт их недоставки нужно выяснять внешними средствами (ESB, ПО мониторинга внешних систем, и т.п.)

Сообщения пишутся в очередь пакетами размером в packSize штук. После отправки каждого пакета делается пауза в sleepMilliseconds мс.

4.3.4.2 Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
minPublicationDate	Дата в формате dd.mm.yyyy hh:mi:ss	Нижняя граница (включительно) даты создания сообщений, которые будут переотправлены в очередь	
maxPublicationDate	Дата в формате dd.mm.yyyy hh:mi:ss	Верхняя граница (включительно) даты создания сообщений, которые будут переотправлены в очередь	
failedOnly	Логическое значение	Переотправлять только события со статусом ERROR или с любым	true
sleepMilliseconds	Целое	Ждать указанное время перед тем, как отправить следующий пакет сообщений.	50
packSize	Целое	Размер пакета сообщений для записи в очередь	1

4.3.5 Повторная отправка ошибочных JMS сообщений за последние 24 часа

RepublishOutgoingEventsForTheLastDay

4.3.5.1 Описание задачи

Переотправляет в выходную очередь ошибочные события за последние 24 часа.

Отправляет только туда, куда не удалось отправить: если есть 2 JMS-источника и один эти сообщения уже получил, повторно он их не получит.

4.3.5.2 Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
failedOnly	Логическое значение	Переотправлять только события со статусом ERROR или с любым	true
createDateLastInterval	Целое + day	За какой период искать сообщения	1 day

4.4 Импорт данных

Импорт данных

Группа задач, предназначенных для загрузки данных из внешних источников в Единый клиент.

Импорт данных		
Действия	Название	Описание
Детали Редактировать Выполнить	disableStagingRecords	Деактивировать исходные записи
Детали Редактировать Выполнить	enableStagingRecords	Активировать исходные записи
Детали Редактировать Выполнить	physicalBufferIPC	Загрузка инкремента ФЛ
Детали Редактировать Выполнить	transform	Трансформация данных
Детали Редактировать Выполнить	transformWithCustomConfig	Трансформация данных с указанными параметрами

Типовые задачи, которые есть всегда:

Название	Описание	Комментарий
disableStagingRecords	Деактивировать исходные записи	Временно приостановить обработку записей
enableStagingRecords	Активировать исходные записи	Снова запустить обработку записей
transform	Трансформация данных	Если загрузка инкремента падает с ошибкой, можно догрузить в ЕК необработанные данные с помощью данной задачи
transformWithCustomConfig	Трансформация данных с указанными параметрами	Когда нужно при отработке задачи трансформации обновить запись принудительно.

Общий список задач по загрузке данных (задачи подключаются сотрудниками ХФЛабс)

4.4.1 Особенности

Все задачи импорта данных, которые используют [Трансформация данных](#), могут работать в двух режимах:

- с оповещением поискового индекса КА о трансформации пачек;
- без оповещения поискового индекса.

Режим запуска определяется значением параметра:

Параметр	Тип	Описание	Значение по умолчанию
notifyListeners	Логическое значение	Оповещение поискового индекса КА о трансформации пачек. Если оповещение отключено, после завершения задачи нужно перестроить индекс.	true

4.4.2 Результат выполнения

Если данные загружены успешно, то задачи и триггеры завершаются со статусом "FINISHED".

Если данных для загрузки не было, то задачи и триггеры завершаются со статусом "SKIPPED".

4.4.3 Общие параметры для задач загрузки данных

4.4.3.1 Параметры для задач, использующих трансформацию данных

Если при импорте записей требуется обработка данных Фактором, тогда загрузка данных осуществляется в два этапа:

1. Из источника данные загружаются в Staging;
2. Данные из Staging отправляются в Фактор для стандартизации.

На втором этапе неявным образом выполняется задача [Трансформации данных с указанными параметрами](#).

В задаче импорта можно указать любой из параметров доступных для Трансформации данных.

Если в задаче импорта параметры не заданы, тогда трансформация данных в рамках импорта будет выполняться согласно значениям по умолчанию.

4.4.3.2 Параметры задач загрузки данных, применимые только для импорта данных в Staging

Параметр	Тип	Описание	Значение по умолчанию
importStagingPoolSize	Целое	Размер пула потоков для загрузки из буферной таблицы БД	1, если не задан параметр трансформации poolSize; poolSize, если параметр трансформации poolSize задан.

4.4.4 Задача загрузки контрагентов из буферной таблицы БД

Порционная

4.4.4.1 Описание задачи

Загружает инкремент контрагентов из буферной таблицы и опционально удаляет успешно загруженные данные. Чтобы загружать именно инкремент, а не каждый раз всю таблицу целиком, отслеживает и хранит уникальный целочисленный идентификатор последней успешно загруженной записи (исходя из предположения, что идентификаторы в буферной таблице строго возрастающие).

Задача считается успешно выполненной только в том случае, если удалось успешно загрузить весь инкремент без исключения. В противном случае при следующем запуске задача пытается загрузить весь инкремент заново.

Параметр	Тип	Описание
essenceType	Тип контрагента	Тип загружаемых контрагентов
bufferDataReadBlockSize	Целое	Размер пакета записей для выборки из буферной таблицы
groupByResultParty	Логическое значение	Группировать обновляемых контрагентов по идентификатору результирующего. Умолчательное значение – false. Переключение в true позволяет ускорить импорт больших инкрементов на сильно задублированных базах, но замедляет обработку инкрементов менее чем в сто тысяч контрагентов.
massiveChanging	Логическое значение	Оповещение Фактора изменениями, по умолчанию включено (значение false). Если задать true, то Фактор не будет вызываться с инкрементальными populate

4.4.4.2 Выборка данных

Для выборки инкремента данных из буферной таблицы и удаления загруженных записи задача использует хранимые процедуры СУБД:

```

/**
 * Возвращает курсор на записи инкремента (все записи с идентификатором > old_max_id)
 * @param party_type тип контрагента
 * @param old_max_id значение идентификатора последней успешно загруженной записи на
 данный момент
 * @return new_max_id максимальное значение идентификатора в инкремента
 * @return IRecordset курсор на записи инкремента
 */
get_cdi_buffer (party_type in nvarchar2, old_max_id in number, new_max_id out number) return
sys_refcursor as
Irecordset sys_refcursor;

/**
 * Удаляет все загруженные записи (все записи с идентификатором <= old_max_id)
 * @param party_type тип контрагента

```

```

* @param old_max_id значение идентификатора последней успешно загруженной записи на
данный момент
*/
delete_cdi_buffer (party_type in varchar2, old_max_id in number);

```

4.4.5 Задача загрузки договоров из буферной таблицы БД

4.4.5.1 Описание задачи

Загружает инкремент договоров из буферной таблицы и опционально удаляет успешно загруженные данные. Чтобы загружать именно инкремент, а не каждый раз всю таблицу целиком, отслеживает и хранит уникальный целочисленный идентификатор последней успешно загруженной записи (исходя из предположения, что идентификаторы в буферной таблице строго возрастающие).

Задача считается успешно выполненной только в том случае, если удалось успешно загрузить весь инкремент без исключения. В противном случае при следующем запуске задача пытается загрузить весь инкремент заново.

Параметр	Тип	Описание
bufferDataReadBlockSize	Целое	Размер пакета записей для выборки из буферной таблицы
massiveChanging	Логическое значение	Оповещение Фактора изменениями, по умолчанию включено (значение false). Если задать true, то Фактор не будет вызываться с инкрементальными populate

4.4.5.2 Выборка данных

Для выборки инкремента данных из буферной таблицы и удаления загруженных записи задача использует хранимые процедуры СУБД:

```

/**
* Возвращает курсор на записи инкремента (все записи с идентификатором > old_max_id)
* @param essenceType тип загружаемых данных, по умолчанию AGREEMENT
* @param old_max_id значение идентификатора последней успешно загруженной записи на
данный момент
* @return new_max_id максимальное значение идентификатора в инкремента
* @return IRecordset курсор на записи инкремента
*/
get_cdi_buffer_agreement (essenceType in varchar2, old_max_id in number, new_max_id out
number) return sys_refcursor as
Irecordset sys_refcursor;

/**
* Удаляет все загруженные записи (все записи с идентификатором <= old_max_id)
* @param essenceType тип загружаемых данных, по умолчанию AGREEMENT
* @param old_max_id значение идентификатора последней успешно загруженной записи на
данный момент
*/
delete_cdi_buffer_agreement (essenceType in varchar2, old_max_id in number);

```

4.4.6 Задача загрузки связей договоров из буферной таблицы БД

4.4.6.1 Описание задачи

Загружает инкремент связей договоров из буферной таблицы и опционально удаляет успешно загруженные данные. Чтобы загружать именно инкремент, а не каждый раз всю таблицу целиком, отслеживает и хранит уникальный целочисленный идентификатор последней успешно загруженной записи (исходя из предположения, что идентификаторы в буферной таблице строго возрастающие).

Задача считается успешно выполненной только в том случае, если удалось успешно загрузить весь инкремент без исключения. В противном случае при следующем запуске задача пытается загрузить весь инкремент заново.

Параметр	Тип	Описание
essenceType	Тип контрагента	Тип контрагентов, для которых происходит загрузка договоров
bufferDataReadBlockSize	Целое	Размер пакета записей для выборки из буферной таблицы
massiveChanging	Логическое значение	Оповещение Фактора изменениями, по умолчанию включено (значение false). Если задать true, то Фактор не будет вызываться с инкрементальными populate
useRelationCache	Логическое значение	Если задать true, перед началом итерации по буферу, задача берет таблицу RELATION и записывает в компактном виде себе в память (актуально только для большого количества записей, на построение кэша уходит 20-30 минут, зато итерация ускоряется в разы)

4.4.6.2 Выборка данных

Для выборки инкремента данных из буферной таблицы и удаления загруженных записи задача использует хранимые процедуры СУБД:

```

/**
 * Возвращает курсор на записи инкремента (все записи с идентификатором > old_max_id)
 * @param party_type тип контрагента
 * @param old_max_id значение идентификатора последней успешно загруженной записи на
 данный момент
 * @return new_max_id максимальное значение идентификатора в инкремента
 * @return IRecordset курсор на записи инкремента
 */
get_cdi_buffer_agreement_link (party_type in nvarchar2, old_max_id in number, new_max_id out
number) return sys_refcursor as
Irecordset sys_refcursor;

/**
 * Удаляет все загруженные записи (все записи с идентификатором <= old_max_id)
 * @param party_type тип контрагента
 * @param old_max_id значение идентификатора последней успешно загруженной записи на
 данный момент
 */
delete_cdi_buffer_agreement_link (party_type in varchar2, old_max_id in number);

```

4.4.7 Задача загрузки связей

Невозобновляемая

4.4.7.1 Общее описание

Задача RelationBufferIPC загружает из буфера инкремент связей для всех типов контрагентов.

Порядок контрагентов в буферной таблице CDI_BUFFER_RELATION и в таблице [ТИПОВ СВЯЗЕЙ](#) должен соответствовать.

4.4.7.2 Конфигурация задачи

Создать/Редактировать задачу

Группа import
Название RelationBufferIPC

Параметры задачи

Название параметра	Значение
oldMaxIdPropertyName	oldMaxId.buffer.relation
bufferDataReadBlockSize	1000

Сохранить Отмена

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
bufferDataReadBlockSize	Размер пакета записей для выборки из буферной таблицы	1000

4.4.8 Задача загрузки контрагентов из CSV-файла

Загружает контрагентов из CSV-файла, расположенного в локальной файловой системе сервера приложений.

Параметр	Тип	Описание
sourceFileName	Строка	Полный путь к загружаемому файлу. Для Windows путь необходимо указывать с задвоением обратного слэша, например: C:\upload\client_data.csv
sourceDataEncoding	Строка	Кодировка файла. Примеры: windows-1251, UTF-8.
sourceDataSeparator	Символ	Разделитель полей файла. Примеры: ; α
sourceDataReadBlockSize	Целое	Размер пакета записей для выборки из файла

Параметр	Тип	Описание
rawldIndex	Целое	Порядковый номер поля, в котором расположен уникальный идентификатор исходной клиентской записи. Нумерация начинается с 0.
sourceSystem	Строка	Наименование исходной системы (такое, как оно указано в таблице SYSTEM Единого клиента)
essenceType	Тип контрагента	Тип загружаемых контрагентов

4.4.9 Задача загрузки контрагентов из каталога

Загружает контрагентов из каталога, расположенного в локальной файловой системе сервера приложений. Каталог должен содержать CSV-файлы установленного наименования. Маска наименования задается в конфигурации системы; как правило используется шаблон {код_системы}_{гггг-ММ-дд}_{ЧЧмм}.csv, где:

- {код_системы} – наименование исходной системы;
- {гггг-ММ-дд} – дата формирования файла;
- {ЧЧмм} – время формирования файла.

Обработка каталога не рекурсивная, то есть загружаются только файлы, находящиеся непосредственно в указанной директории, но не в ее поддиректориях.

Каждый файл загружается независимо от остальных. После успешной загрузки очередного файла, система переименовывает его с префиксом success_. Если в процессе загрузки файла произошла ошибка, система переименовывает его с префиксом error_ и переходит к следующему еще не обработанному файлу. При повторном выполнении задачи файлы с префиксами success_ и error_ не обрабатываются.

При загрузке наименование исходной системы определяется по имени файла, а идентификатор клиентской записи внутри этой системы берется из данных (см. параметр rawldIndex).

Параметр	Тип	Описание
sourceFileName	Строка	Полный путь к загружаемому каталогу. Для Windows путь необходимо указывать с задвоением обратного слэша, например: C:\\upload
sourceDataEncoding	Строка	Кодировка файлов в каталоге. Примеры: windows-1251, UTF-8.
sourceDataSeparator	Символ	Разделитель полей файлов в каталоге. Примеры: ; ъ
sourceDataReadBlockSize	Целое	Размер пакета записей для выборки из файла
rawldIndex	Целое	Порядковый номер поля, в котором расположен уникальный идентификатор исходной клиентской записи. Нумерация начинается с 0.

Параметр	Тип	Описание
essenceType	Тип контрагента	Тип загружаемых контрагентов

4.4.10 Задача загрузки контрагентов по SFTP

Загружает контрагентов из каталога, расположенного на SFTP-сервере. Логика работы в целом аналогична задаче [загрузки из локального каталога](#), со следующими отличиями:

- После установления соединения с сервером система периодически проверяет его и при обрыве соединения пытается восстановить подключение (см. параметры `validateLinkInterval`, `reconnectCount` и `reconnectInterval`).
- Перед загрузкой каждого файла система копирует его в локальную файловую систему.

Параметр	Тип	Описание
Параметры доступа по SFTP		
serverHost	Строка	Доменное имя SFTP-сервера
serverPort	Строка	Порт для подключения по SFTP
username	Строка	Имя пользователя для подключения по SFTP
password	Строка	Пароль для подключения по SFTP
connectionTimeout	Целое	Таймаут подключения (в миллисекундах)
validateLinkInterval	Целое	Интервал проверки соединения с SFTP-сервером после обрыва соединения
reconnectCount	Целое	Количество повторных попыток подключения к SFTP-серверу
reconnectInterval	Целое	Интервал между повторными попытками подключения (в миллисекундах)
Параметры загружаемых файлов		
sourceFileName	Строка	Полный путь к загружаемому каталогу на SFTP-сервере.
sourceDataEncoding	Строка	Кодировка файлов в каталоге. Примеры: windows-1251, UTF-8.
sourceDataReadBlockSize	Целое	Размер пакета записей для выборки из файла
sourceDataSeparator	Символ	Разделитель полей файлов в каталоге. Примеры: ; ѝ
rawIdIndex	Целое	Порядковый номер поля, в котором расположен уникальный идентификатор исходной клиентской записи. Нумерация начинается с 0.
essenceType	Тип контрагента	Тип загружаемых контрагентов

4.4.11 Задача загрузки справочников ЕГРЮЛ / ЕГРИП

EgrDictionaryImport
Прерываемая, невозобновляемая
Неблокирующая

4.4.11.1 Описание задачи

Задача нужна для загрузки в ЕК справочников ЕГРЮЛ / ЕГРИП из файла (расширение *.dic).

Справочник брать из [хранилища справочников ЕГР](#) — выбрать самую свежую сборку и загрузить из неё файл с расширением .dic.

4.4.11.2 Конфигурация задачи

По-умолчанию все параметры уже настроены. Нужно только указать sourceFileName и можно запускать.

Для первичной загрузки справочника [поменять параметр](#) massiveChanging

Параметр	Описание	Пример
sourceFileName	Путь к директории со справочником ЕГР	c:\egrul2
sourceDataEncoding	Кодировка справочника	UTF-8
bufferDataReadBlockSize	Размер пакета записей для выборки	1000
poolSize	Максимальное количество потоков	8
essenceType	Тип загружаемых контрагентов	EGR
cipher	Загрузка зашифрованного справочника, по умолчанию включена (значение true)	true
massiveChanging	Оповещение Фактора изменениями, по умолчанию включено (значение false). Если задать true, то Фактор не будет вызываться с инкрементальными populate	false
closeAbsentBranches	Закрывать отсутствующие в инкременте филиалы и агентства (по которым нет связей и ручных правок). По умолчанию true.	true

Создать / Редактировать задачу

Группа import
 Название EgrDictionary/import

Параметры задачи

Название параметра	Значение	Обязательность
massiveChanging	false	
sourceFileName	/replace/it/on/directory/with/files	
sourceDataReadBlockSize	1000	
sourceDataEncoding	UTF-8	
poolSize	8	
essenceType	EGR	

4.4.12 Трансформация данных

transform

4.4.12.1 Описание задачи

Задача нужна для принудительной обработки записей из staging-области.

4.4.12.2 Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
terminateOnErrors	Логическое значение	Прервать выполнение при возникновении ошибок.	false
massiveChanging	Логическое значение	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false
notifyListeners	Логическое значение	Оповещение поискового индекса КА о трансформации пачек Если оповещение отключено, после завершения задачи нужно перестроить индекс <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставлять в запускаемой задаче</i>	true

4.4.12.3 Трансформация данных с указанными параметрами

transformWithCustomConfig

Задача не участвует ни в каких триггерах, только ручной запуск.

4.4.12.3.1 Описание задачи

Задача нужна для принудительного обновления только части записи, когда нужно обработать конкретный атрибут или реквизит.

4.4.12.3.2 Конфигурация задачи

Параметр	Описание	Значение по умолчанию
forceUpdateRequisiteNames	Имена реквизитов (через запятую), которые нужно безусловно обновлять, остальные реквизиты в штатном порядке. Пример: <ul style="list-style-type: none"> NONE – нет принудительного обновления; ALL – принудительно обновляем все реквизиты; SURNAME, BIRTHDATE – только фио и др обновляем принудительно. 	NONE
forceUpdateAttributeTypes	Имена атрибутов (через запятую), которые нужно безусловно обновлять, остальные атрибуты в штатном порядке. Пример: <ul style="list-style-type: none"> NONE – нет принудительного обновления; ALL – принудительно обновляем все атрибуты; PHONE, EMAIL – только телефоны и email обновляем принудительно. 	NONE
forceUpdate	Безусловное обновления контрагента, без учета настроенных правил	true
groupByResultParty	Использовать провайдер записей стейджинга с группировкой по исходным КА одного результирующего	false
terminateOnErrors	Прервать выполнение при возникновении ошибок	false
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false
transformPoolSize	Размер пула потоков для трансформации	16
poolSize	Размер пула потоков для стандартизации	кол-во доступных

Параметр	Описание	Значение по умолчанию
		серверов Фактора
pageSize	Размер одной страницы данных	1000
maxStoredErrors	Максимальное кол-во ошибок для хранения в памяти	1000
cacheSize	Размер кеша страниц	1
notifyListeners	Оповещение поискового индекса КА о трансформации пачек Если оповещение отключено, после завершения задачи нужно перестроить индекс <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставлять в запускаемой задаче</i>	true

4.4.13 Активировать исходные записи

enableStagingRecords

4.4.13.1 Описание задачи

Задача нужна для того, чтобы снова запустить обработку записей, [проигнорированных ранее](#).

4.4.13.2 Конфигурация задачи

Параметр	Описание	Пример
stagingHids	Список hid_staging контрагентов, которые необходимо игнорировать. Значения указываются через запятую	1, 4, 100

Группа import
Название enableStagingRecords

Параметры задачи

Название параметра	Значение	Обязательность
stagingHids	1,2,3,1000	

Для получения списка клиентов, игнорируемых ранее, необходимо выполнить запрос

```
select distinct hid_staging from staging
where
staging_status = 5;
```

4.4.13.3 Технические детали

Задача устанавливает заданному в параметрах списку контрагентов [STAGING_STATUS](#) = 0.

При этом обрабатываются только контрагенты, имеющие STAGING_STATUS = 5, остальные игнорируются

4.4.14 Деактивировать исходные записи

disableStagingRecords

4.4.14.1 Описание задачи

Задача нужна для временной приостановки обработки записей.

Например, в буферной таблице прислали неправильное значение и *Единый клиент* падает с ошибкой при каждой загрузке инкремента из-за одной неверной записи. Исправление данных в исходной системе может занять время, а постоянное падение с ошибкой мешает понять актуальный статус задачи «за исключением этой одной записи». Теперь записи можно временно игнорировать, а после исправления проблемы [снова обрабатывать](#).

4.4.14.2 Конфигурация задачи

Параметр	Описание	Пример
stagingHids	Список hid_staging контрагентов, которые необходимо игнорировать. Значения указываются через запятую	1, 4, 100

Группа import

Название disableStagingRecords

Параметры задачи

Название параметра	Значение	Обязательность
stagingHids	1,2,3,1000	

Сохранить

Отмена

Пример получения hid_staging для исходной записи из системы AL с идентификатором 1

```
select distinct hid_staging from staging
where
author = 'AL'
and
id_record = '1';
```

4.4.14.3 Технические детали

Задача устанавливает заданному в параметрах списку контрагентов [STAGING_STATUS](#) = 5.

Чтобы посмотреть список деактивированных клиентов, достаточно выполнить запрос:

```
select distinct hid_staging from staging
where
staging_status = 5;
```

4.5 Идентификация и слияние дубликатов

4.5.1 Идентификация и слияние дубликатов

Группа задач, предназначенных для идентификации и слияния дубликатов:

- `findDuplicates` – [поиск дубликатов](#) по всей базе контрагентов;
- `merge` – [слияние](#) гарантированных и подтвержденных дубликатов;
- `findDuplicatesAndMerge` – [поиск дубликатов](#) по всей базе контрагентов с последующим [слиянием](#) гарантированных дубликатов;
- `findDuplicatesAndMergeAndRebuildGroups` – поиск дубликатов по всей базе контрагентов, слияние гарантированных и перестроение индекса негарантированных групп дублей;
- `incrementDuplicatesAndMerge` – [инкрементальный поиск дубликатов](#) (дубликаты ищутся среди контрагентов, по которым были изменения после предыдущего поиска дубликатов) с последующим [слиянием](#) гарантированных дубликатов;
- `importExternalDuplicateWithMerge` – слияние дубликатов по [внешнему списку](#);
- `rebuildFullDedupHashes` – [перестроение хэшей](#) полного поиска дубликатов.

4.5.2 Кросс-сверка ФЛ-ЮЛ

```
findCrossRelationsPhysicalAndLegal
```

4.5.2.1 Описание задачи

Задача ищет дубликаты между ФЛ и ИП по [набору правил](#).

Если совпадение найдено – между клиентами проставляется связь.

4.5.2.2 Конфигурация задачи

Конфигурировать задачу не надо.

Она выгружает всех физических лиц и индивидуальных предпринимателей. Невозобновляемая, каждый раз выгружает всю базу

4.5.2.3 Перестроение хэшей кросс-сверки ФЛ-ЮЛ

```
rebuildCrossRelationsPhysicalAndLegalHashes
```

4.5.2.3.1 Описание задачи

Задача перестраивает хеши на Факторе для корректной работы кросс-сверки ФЛ-ЮЛ.

4.5.2.3.2 Конфигурация задачи

Конфигурировать задачу не надо.

4.5.3 Перестроение хэшей полного поиска дубликатов

```
rebuildFullDedupHashes
```

4.5.3.1 Описание задачи

Техническая задача, используется по инструкции сотрудников компании ХФЛабс или после того, как развернут бэкап с прода.

См также: [Как поднять бэкап базы с PROD?](#) Задача входит в этот триггер.

Технические подробности, не видимые заказчику

Задача выгружает все записи из поискового индекса для «Фактора».

На «Факторе» происходит расчет хэшей, но без дальнейшего поиска дублей, загрузки результатов в ЕК и слияния.

Полученные хеши используются для инкрементального дедапа и запросов fuzzySearch

4.5.3.2 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	<p>Тип контрагента, для которых сбрасываются хэши.</p> <ul style="list-style-type: none"> Если параметр не задан, сбрасывает хэши для всех типов. Если указан тип контрагента, сбрасывает только для него Если указано несколько типов, задача упадет с ошибкой 	

4.5.4 Слияние дубликатов по внешнему списку

4.5.4.1 Общее описание

Задача *importExternalDuplicateWithMerge* предназначена для слияния дубликатов, информация о которых приходит из внешних или дополнительных источников.

4.5.4.2 Механизм

Задача вызывает хранимую функцию *get_external_duplicates* с параметром *partyType* (PHYSICAL / INDIVIDUAL / LEGAL).

Функция возвращает курсор на выборку в следующем формате:

Поле	Описание
hid_2	HID второй PARTY
hid_1	HID первой PARTY
match_scope	коэффициент схожести для создаваемых дубликатов
rule	номер сценария для создаваемых дубликатов

Требования к возвращаемым записям:

- hid_1 и hid_2 – идентификаторы результирующих записей (с merged_status 0 или 2).
- hid_1 != hid_2.
- Каждая пара возвращается только один раз.
- Контрагенты в паре – одного типа.

Если записи не были предварительно найдены, то создаётся новая пара в duplicate, иначе обновляется уже существующая запись о дубликате.

В случае, если пара дубликатов была предварительно найдена и отклонена оператором, обновления существующей записи о дубликате не происходит.

4.5.4.3 Пример функции *get_external_duplicates*

В данном примере функция выбирает пары дубликатов из внешней таблицы *external_duplicates*, принудительно проставляя номер сценария 100 и 100% схожести (чтобы дубликаты попали под автоматическое слияние). Привязка к таблицам *staging*, *physical_party* и *merged* производится для того, чтобы получить актуальные идентификаторы для пар, переданных в таблице *external_duplicates*.

```

create or replace
function get_external_duplicates(partyType in nvarchar2) return sys_refcursor as
  IRecordset sys_refcursor;
begin
  case partyType
    when 'PHYSICAL' then
      return get_ext_duplicates_physical();
    when 'LEGAL' then
      return get_ext_duplicates_legal();
  end case;
end;
/

create or replace
function get_ext_duplicates_physical return sys_refcursor as
  IRecordset sys_refcursor;
begin
  open IRecordset for

```

```

select distinct
  least(nvl(m1.hid_final_scion, p1.hid_party), nvl(m2.hid_final_scion, p2.hid_party)) hid_1,
  greatest(nvl(m1.hid_final_scion, p1.hid_party), nvl(m2.hid_final_scion, p2.hid_party)) hid_2,
  100 as rule,
  100 as match_scope
from external_duplicates d
  left join staging s1 on (s1.author = d.system_1 and s1.id_record = to_char(d.external_id_1))
  left join physical_party p1 on (s1.hid_staging = p1.hid_staging)
  left join merged m1 on (m1.hid_party = p1.hid_party)
  left join staging s2 on (s2.author = d.system_2 and s2.id_record = to_char(d.external_id_2))
  left join physical_party p2 on (s2.hid_staging = p2.hid_staging)
  left join merged m2 on (m2.hid_party = p2.hid_party)
where
  s1.party_type = 'PHYSICAL' and s2.party_type = 'PHYSICAL'
  and s1.version = 0 and p1.version = 0 and m1.canceled is null
  and s2.version = 0 and p2.version = 0 and m2.canceled is null
  and p1.hid_party is not null and p2.hid_party is not null
  and nvl(m1.hid_final_scion, p1.hid_party) <> nvl(m2.hid_final_scion, p2.hid_party);
return IRecordset;
end;
/

create or replace
function get_ext_duplicates_legal return sys_refcursor as
  IRecordset sys_refcursor;
begin
  ...
end;
/

```

4.5.4.4 Механизм загрузки дубликатов

Задача вызывает хранимую функцию `get_external_duplicates` с параметром `partyType` (PHYSICAL / INDIVIDUAL / LEGAL).

Функция возвращает курсор на выборку в следующем формате:

Поле	Описание
hid_2	HID второй PARTY
hid_1	HID первой PARTY
match_scope	коэффициент схожести для создаваемых дубликатов
rule	номер сценария для создаваемых дубликатов

Требования к возвращаемым записям:

- hid_1 и hid_2 – идентификаторы результирующих записей (с `merged_status` 0 или 2).
- hid_1 != hid_2.
- Каждая пара возвращается только один раз.
- Контрагенты в паре – одного типа.

Если записи не были предварительно найдены, то создаётся новая пара в `duplicate`, иначе обновляется уже существующая запись о дубликате.

В случае, если пара дубликатов была предварительно найдена и отклонена оператором, обновления существующей записи о дубликате не происходит.

4.6 Повторная обработка

4.6.1 Повторная обработка контрагентов

Группа задач, предназначенных для повторной обработки определенного подмножества контрагентов:

- addressActualization – актуализация адресов по КЛАДР;
- attributeActualizeByStaging – актуализация атрибутов на основе данных staging;
- attributeRecoverByHids – восстановление атрибутов по списку HIDs;
- attributeCloseActualByHids – восстановление атрибутов по списку HIDs;
- hiddenFlagRefineByHids – пересчет hidden-флагов атрибутов по списку контрагентов;
- partyReopenByHids – переоткрытие контрагентов по списку HIDs
- reMergeFullByPartyHids – полное переслияние по списку HIDs'ов результирующих контрагентов;
- reMergeRequisiteByPartyHids – переслияние реквизитов по списку HIDs'ов результирующих контрагентов;
- unMergeByMergedIds – разлияние контрагентов по списку MergedId;
- fullUnmergeByResultHids – полное разлияние контрагентов по списку результирующих контрагентов;
- terroristStatusReClean – переобработка по справочнику террористов.
- phoneExtraFieldsReCleaner – переобработка дополнительных полей телефонов.
- partyRecoverByHids – восстановление КА на версию по списку HID.

4.6.2 Задача перерасчета hidden-флагов атрибутов по списку контрагентов

hiddenFlagRefineByHids

[Возобновляемая](#)

4.6.2.1 Описание задачи

Используется в случае, когда требуется актуализировать hidden-флаги атрибутов контрагентов (например, если в результате переочистки данных у контрагента появилось несколько одинаковых атрибутов, которые должны быть "схлопнуты").

Задача включает в себя перерасчет hidden-флагов только для заданного списка контрагентов.

Работает по золотым карточкам

4.6.2.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;

2. Выяснить идентификаторы контрагентов (hid_party), для которых необходимо выполнить перерасчет;
3. Добавить список данных идентификаторов в таблицу task_party_hids вручную или с помощью вызова одной из [задач](#).

Пример ручного заполнения таблицы:

```
insert into task_party_hids
select hid_final_scion from merged where ...
;
```

4.6.2.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется перерасчета hidden-флагов атрибутов	PHYSICAL
poolSize	Максимальное количество потоков	8
pageSize	Размер пачки	1000
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false

Группа reprocess

Название hiddenFlagRefineByHids

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	8	
pageSize	1000	
partyType	PHYSICAL	

Сохранить

Отмена

4.6.3 Задача полного переслияния контрагентов

reMergeFullByPartyHids

[Возобновляемая](#)

4.6.3.1 Описание задачи

Использовать следует в случае, когда добавляется новое или изменяется существующее правило слияния и требуется актуализировать уже слитые дубликаты. Задача включает в себя переслияние как реквизитов, так и атрибутов контрагентов.

Для атрибутов выбираются все атрибуты исходных клиентов и проводится их слияние.

- если результат не менялся — ничего не выполняется;
- если результат изменился — обновляем атрибут в золотой карточке;
- если победил ранее [схлопнутый](#) атрибут — закрываются старые атрибуты, создается новый.

4.6.3.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Выяснить идентификаторы результирующих контрагентов (hid_party), для которых необходимо выполнить переслияние;
3. Добавить список данных идентификаторов в таблицу task_party_hids вручную или с помощью вызова одной из [задач](#).

Пример ручного наполнения таблицы для полного переслияния объединенных контрагентов:

```
insert into task_party_hids
select hid_party from physical_party
where merged_status = 2 and version = 0 and enddate is null;

commit;
```

4.6.3.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется переслияние	PHYSICAL
poolSize	Максимальное количество потоков	8
pageSize	Размер пачки	1000
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false

Группа reprocess

Название reMergeFullByPartyHids

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	8	
pageSize	1000	
partyType	PHYSICAL	

Сохранить

Отмена

4.6.4 Задача переслияния реквизитов контрагентов

reMergeRequisiteByPartyHids

[Возобновляемая](#)

4.6.4.1 Описание задачи

Использовать следует в случае, когда добавляется новое или изменяется существующее правило слияния для реквизитов и требуется актуализировать уже слитые дубликаты.

Задача включает в себя переслияние только реквизитов контрагентов.

4.6.4.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Выяснить идентификаторы результирующих контрагентов (hid_party), для которых необходимо выполнить переслияние;
3. Добавить список данных идентификаторов в таблицу task_party_hids вручную или с помощью вызова одной из [задач](#).

Пример ручного наполнения таблицы:

```
insert into task_party_hids
select hid_party from physical_party where merged_status = 2 and ...
;
```

4.6.4.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется переслияние	PHYSICAL
poolSize	Максимальное количество потоков	8
pageSize	Размер пачки	1000
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false

Группа reprocess
 Название reMergeRequisiteByPartyHids

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	8	
pageSize	1000	
partyType	PHYSICAL	

Сохранить Отмена

4.6.5 Разливание контрагентов по списку MergedId

unMergeByMergedIds

[Возобновляемая](#)

4.6.5.1 Описание задачи

Если большое количество контрагентов были ошибочно объединены как гарантированные дубликаты, их можно разъединить с помощью данной задачи.

4.6.5.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_merged_ids;
2. Выяснить идентификаторы контрагентов, которые были объединены ошибочно;
3. Определить идентификаторы id_merged соответствующих записей из таблицы merged;
4. Добавить список данных идентификаторов в таблицу task_merged_ids.

Пример:

```
insert into task_merged_ids
select id_merged from merged where hid_party in (
  select ...
)
;
```

4.6.5.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется разливание	PHYSICAL

Параметр	Описание	Значение по умолчанию
unmergeDescription	Комментарий к задаче (например, причина, по которой выполняется разливание)	
poolSize	Максимальное количество потоков	1
pageSize	Размер пачки	1000
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false
preventFurtherMerging	Запретить разлитым контрагентам сливаться в дальнейшем (проставить им Confirmation = 4)	false

Группа reprocess
Название unMergeByMergedIds

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	1	
pageSize	1000	
unmergeDescription	Unmerge description	
partyType	PHYSICAL	

[Сохранить](#) [Отмена](#)

4.6.5.4 Выполнение задачи
Задача запускается из АРМ Администратора.

4.6.6 Полное разливание контрагентов по списку результирующих контрагентов

fullUnmergeByResultHids
Возобновляемая

4.6.6.1 Описание задачи
Золотые карточки разливаются целиком, без промежуточных версий – экономит место в БД.

Удаляет записи, взятые в обработку, из таблицы task_party_hids в любом случае (даже если возникла ошибка при разлинии)

4.6.6.2 Запуск задачи
Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Добавить идентификаторы результирующих контрагентов для разливания в таблицу task_party_hids;

Пример:

```
insert into task_party_hids(hid_party, party_type) values (1, 'PHYSICAL');
commit;
```

4.6.6.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется разливание	PHYSICAL
unmergeDescription	Комментарий к задаче (например, причина, по которой выполняется разливание), сохраняется в истории	
unmergePackLimit	Лимит на количество слитых записей в одной обрабатываемой пачке	1000
poolSize	Максимальное количество потоков	4
pageSize	Размер пачки	50
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false
preventFurtherMerging	Запретить разлитым контрагентам сливаться в дальнейшем (Проставить им Confirmation = 4)	false

Группа reprocess
Название fullUnmergeByResultHids

Параметры задачи

Название параметра	Значение	Обязательность
unmergePackLimit	1000	
poolSize	4	
pageSize	50	
unmergeDescription	Unmerge description	
partyType	PHYSICAL	

Сохранить Отмена

4.6.7 Задача восстановления атрибутов по списку HIDs

attributeRecoverByHids

4.6.7.1 Описание задачи

Данную задачу следует использовать в случае, когда требуется массово восстановить атрибуты контрагентов (например, при некорректном обновлении или ошибочном закрытии).

Задача включает в себя восстановление на указанные версии только для заданного списка атрибутов.

4.6.7.2 Особенности задачи

Перед запуском задачи необходимо:

1. выполнить задачу [cleanTaskParameterTables](#) для очистки таблицы [task_attribute_hids](#);
2. выяснить следующие параметры атрибутов, которые нужно восстановить:
 - a. тип и идентификатор атрибута в ЕК;
 - b. тип и идентификатор контрагента в ЕК;
 - c. дату и время, на которую необходимо восстановить атрибут.
3. добавить эту информацию в таблицу [task_attribute_hids](#) (см. пример ниже).

Пример:

```
INSERT INTO task_attribute_hids (attribute_type, hid_attribute, party_type, hid_party, target_date)
SELECT 'ADDRESS', hid_address, party_type, hid_party, startdate FROM ...
```

4.6.8 Задача актуализации адресов по КЛАДР

addressActualization

[Возобновляемая](#)

4.6.8.1 Описание задачи

Данную задачу следует использовать в случае, когда требуется актуализировать адреса по КЛАДР – учесть переподчинения, переименования, изменения в индексах и кодах КЛАДР.

4.6.8.2 Алгоритм работы задачи

На актуализацию выбираются все адреса всех исходных клиентских записей, вне зависимости от типа клиентской записи или кодов качества/проверки адресов. То есть актуализируются в том числе адреса с ручными правками и отмеченные на ручную обработку. Адреса результатов слияния актуализируются за счёт обновления исходных адресов в соответствии с правилами обновления.

Алгоритм актуализации работает в зависимости от заполненности служебного поля "исходный адрес после препроцессинга" (PREPROCESSED_SOURCE). Поле добавлено в релизе 3.16. Соответственно, заполнено будет для адресов, которые созданы или обновлены после развёртывания данного релиза.

Если адрес помечен как "подтверждённый оператором" (код проверки CONFIRMED_MANUALLY) или поле с адресом после препроцессинга не заполнено, то используется алгоритм частичной актуализации из результатов предыдущего разбора. Во всех остальных случаях используется алгоритм повторного распознавания адреса.

4.6.8.2.1 Частичная актуализация

Актуализации подвергаются индексы (почтовый и КЛАДР), группа полей Страна-Улица, Федеральный округ, коды КЛАДР и ОКАТО. Расширение дома, коды качества и проверки, расширение валидатора возвращаются без изменений.

Актуализация производится путём отсылки **результатов** предыдущего распознавания на повторную стандартизацию.

Если текущая (по предыдущему распознаванию) дата актуальности КЛАДР адреса соответствует КЛАДРу в ФАКТОРе или не удалось хорошо повторно разобрать адрес (т.е. код проверки отличен от VALIDATED, бывает в случае, например, ручного внесения информации, не соответствующей КЛАДР), Фактор возвращает предыдущие результаты распознавания без изменений.

4.6.8.2.2 Повторное распознавание

На стандартизацию отправляется содержимое служебного поля "исходный адрес после препроцессинга".

Обновиться могут все адресные поля, включая расширение дома, коды качества и проверки, расширение валидатора.

Если текущая (по предыдущему распознаванию) дата актуальности КЛАДР адреса соответствует КЛАДРу в ФАКТОРе, Фактор возвращает предыдущие результаты распознавания без изменений.

4.6.8.3 Особенности задачи

Задача оптимизирована для автоматического периодического выполнения с ограничением на максимальное время работы.

В отличие от других задач повторной обработки использует отдельную таблицу для хранения идентификаторов контрагентов (TASK_ACT_PARTY_HIDS), которые должны быть обработаны.

Для инициализации списка контрагентов на обработку необходимо выполнить задачу fillAddressActualizationTaskPartyHids.

Для обработки своего списка заполнить таблицу TASK_ACT_PARTY_HIDS вручную.

4.6.8.4 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется актуализация адресов	PHYSICAL
executionTimeout	Максимальное время работы задачи во время одного запуска (мин)	180
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false

4.6.9 Исправьте базу данных в случае неконсистентного состояния

Выполните задачу mergedPartiesCheckAndRemerge (*Найти и исправить проблемы в золотых карточках*) из раздела *Диагностика*.

4.6.10 Переоткройте закрытых по ошибке контрагентов с сохранением истории

partyReopenByHids

[Возобновляемая](#)

4.6.10.1 Особенности работы

Контрагент восстанавливается на последнюю перед закрытием версию, при этом:

- исходная версия восстанавливается, даже если контрагент на момент закрытия был объединен с другими контрагентами;
- связи не восстанавливаются (если не указан флаг reopenRelations, по умолчанию отключен);
- если контрагент не найден, идентификатор удаляется из таблицы task_party_hids, задача продолжает работу;
- если контрагент активный или объединенный, задача прервется с ошибкой.

4.6.10.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Добавить идентификаторы контрагентов hid для переоткрытия в таблицу task_party_hids;

Пример:

```
insert into task_party_hids(hid_party, party_type) values (1, 'PHYSICAL');
commit;
```

4.6.10.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется восстановление	PHYSICAL
reopenRelations	Переоткрыть связи контрагента?	false
poolSize	Максимальное количество потоков	8
pageSize	Размер пачки	1000
massiveChanging	Оповещение Фактора изменениями.	false

Параметр	Описание	Значение по умолчанию
	Если задать true, то Фактор не будет вызываться с инкрементальными populate	

Группа reprocess
Название partyReopenByHids

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	8	
pageSize	1000	
partyType	PHYSICAL	
reopenRelations	false	

Сохранить Отмена

4.6.11 Актуализация атрибутов на основе данных staging

attributeActualizeByStaging

4.6.11.1 Описание задачи

Данную задачу следует использовать в случае, когда требуется актуализировать атрибуты на основе исходных записей — например, в случае изменения правил стандартизации или обработки отдельных атрибутов.

Актуализируются атрибуты контрагентов, данные о которых хранятся в staging в виде запакowanego XML-документа в поле plain_xml.

4.6.11.2 Алгоритм работы задачи

На актуализацию выбираются контрагенты, идентификаторы которых указаны в таблице task_act_party_hids.

Данные по указанному типу атрибутов отправляются на очистку. В случае если результат обработки отличается от актуального, данные отправляются на трансформацию с применением стандартных правил обновления и параметрами closeMissed = false и skipRequisites = true.

4.6.11.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется актуализация атрибутов	PHYSICAL
attributeType	Тип атрибута	ADDRESS
massiveChanging	Оповещение Фактора изменениями.	false

Параметр	Описание	Значение по умолчанию
	Если задать true, то Фактор не будет вызываться с инкрементальными populate	

4.6.11.4 Инструкция по переобработке адресов, отмеченных на ручную обработку
Для переобработки адресов зайдите в АРМ Администратора и для каждого [типа контрагента](#) выполните шаги:

4.6.11.4.1 1. Заполнить список к переобработке HIDами контрагентов

- Откройте настройки задачи fillTaskPartyHidsBadAdresses.
- Укажите [тип контрагента](#) в параметре procedureParams.
- Запустите задачу.

4.6.11.4.2 2. Запустить переочистку адресов

- Откройте настройки задачи attributeActualizeByStaging.
- Задайте в параметре partyType тот же тип контрагента, что и на шаге 1.
- Запустите задачу.
- Дождитесь окончания выполнения задачи.

4.6.11.5 Инструкция по переобработке адресов, отмеченных на ручную обработку
Для переобработки адресов зайдите в АРМ Администратора и для каждого [типа контрагента](#) выполните шаги:

4.6.11.5.1 1. Заполнить список к переобработке HIDами контрагентов

- Откройте настройки задачи fillTaskPartyHidsBadAdresses.
- Укажите [тип контрагента](#) в параметре procedureParams.
- Запустите задачу.

4.6.11.5.2 2. Запустить переочистку адресов

- Откройте настройки задачи attributeActualizeByStaging.
- Задайте в параметре partyType тот же тип контрагента, что и на шаге 1.
- Запустите задачу.
- Дождитесь окончания выполнения задачи.

4.6.12 Закрытие актуальных (результатирующих и всех исходных) атрибутов по списку

attributeCloseActualByHids

4.6.12.1 Описание задачи

Данную задачу следует использовать в случае, когда требуется массово закрыть актуальные (результатирующие и все их исходные) атрибуты контрагентов (например, при некорректном инкременте).

4.6.12.2 Запуск задачи

1. Выполнить задачу cleanTaskParameterTables. Она очистит таблицы task_attribute_hids.
2. Подготовить список атрибутов для закрытия. Для каждого атрибута выяснить:
 - a. Тип и идентификатор атрибута в ЕК;
 - b. Тип и идентификатор контрагента в ЕК.
3. Записать список в таблицу task_attribute_hids:

```
INSERT INTO task_attribute_hids (attribute_type, hid_attribute, party_type, hid_party)
SELECT 'ADDRESS', hid_address, party_type, hid_party FROM ...
```

4. Запустить задачу attributeCloseActualByHids.

4.6.12.3 Особенности задачи

Задача закрывает только атрибуты у «золотых» и исходных неслитых клиентов.

Игнорирует атрибуты, которые:

- были объединены;
- закрыты;
- не найдены по идентификатору.

Проигнорированные атрибуты задача пишет в лог-файл.

Закрыв «золотой» атрибут, система закрывает все его исходные атрибуты.

4.6.13 Задача очистки таблиц параметров для массовой переобработки данных

cleanTaskParameterTables

4.6.13.1 Описание задачи

Данную задачу следует выполнить перед запуском задачи [повторной обработки выборочных контрагентов](#).

В результате работы задачи будут очищены [таблицы с параметрами задач массовой переобработки](#).

4.6.14 Подготовка списков для повторной обработки

Группа задач, предназначенных для наполнения таблиц с параметрами для массовой переобработки данных :

fillTaskPartyHidsActual, fillTaskPartyHidsOriginal,
fillTaskPartyHidsResulted, fillAddressActualizationTaskPartyHids,
fillTaskPartyHidsBadAddresses, cleanTaskParameterTables

Данные задачи следует выполнять перед запуском задачи [повторной обработки выборочных контрагентов](#).

В результате работы задачи таблица TASK_PARTY_HIDS будет наполнена идентификаторами контрагентов:

- fillTaskPartyHidsActual – все актуальные контрагенты (merged_status 0 или 2);
- fillTaskPartyHidsOriginal – все оригинальные контрагенты (merged_status 0 или 1);
- fillTaskPartyHidsResulted – все результирующие контрагенты (merged_status 2);
- cleanTaskParameterTables – очистка таблиц с параметрами задач массовой переобработки.

Для наполнения таблицы TASK_ACT_PARTY_HIDS предназначены отдельные задачи:

- fillAddressActualizationTaskPartyHids (отображается в АРМ Администратора только, если подключена [Задача актуализации адресов по КЛАДР](#)) – контрагенты, адреса которых должны быть актуализированы по КЛАДР;
- fillTaskPartyHidsBadAddresses – контрагенты с адресами, отмеченными на ручную обработку (mergedStatus 0 или 1), задача не выбирает контрагентов, если все его адреса, отмеченные на ручную обработку, скрыты.

Для наполнения TASK_DOC_PARTY_HIDS:

- fillDocInvalidStatusActTaskPartyHids – все результирующие контрагенты для переобработки статуса актуальности паспортов. Задача отдельная, чтобы переобработка паспортов не блокировала работу других задач

4.6.15 Описание таблиц

Данные таблицы используются для хранения параметров контрагентов, которых необходимо [переобработать](#).

4.6.15.1 Таблица task_party_hids

№	Поле	Описание
1	HID_PARTY	Уникальный идентификатор контрагента в ЕК
2	PARTY_TYPE	Тип контрагента (литера)

4.6.15.2 Таблица task_merged_ids

Таблица используется для хранения списка идентификаторов id_merged записей из таблицы merged, соответствующих контрагентам, которые были ошибочно объединены.

№	Поле	Описание
1	ID_MERGED	Идентификатор id_merged.

4.6.15.3 Таблица task_act_party_hids

Таблица используется для хранения списка идентификаторов контрагентов, для которых нужно выполнить актуализацию.

№	Поле	Описание
1	HID_PARTY	Уникальный идентификатор контрагента в ЕК

4.6.15.4 Таблица task_attribute_hids

№	Поле	Описание
1	ATTRIBUTE_TYPE	Тип атрибута (литера)
2	HID_ATTRIBUTE	Уникальный идентификатор атрибута в ЕК
3	PARTY_TYPE	Тип контрагента (литера)
4	HID_PARTY	Уникальный идентификатор контрагента в ЕК
5	TARGET_DATE	Дата и время, на которое нужно восстановить атрибут

4.6.15.5 Таблица delete_party_hids

Таблица используется для хранения списка идентификаторов контрагентов, которые нужно удалить навсегда.

№	Поле	Описание
1	HID_PARTY	Уникальный идентификатор контрагента в ЕК
2	PARTY_TYPE	Тип контрагента (литера)

4.6.15.6 Таблица close_party_hids

Таблица используется для хранения списка идентификаторов контрагентов, которые нужно закрыть.

№	Поле	Описание
1	HID_PARTY	Уникальный идентификатор контрагента в ЕК
2	PARTY_TYPE	Тип контрагента (литера)

4.6.15.7 Таблица task_doubtful_reclean_hids

Таблица используется для хранения списка идентификаторов контрагентов, для которых нужно проставить популярность атрибутам.

№	Поле	Описание
1	PARTY_TYPE	Тип контрагента (литера)
2	HID_PARTY	Уникальный идентификатор контрагента в ЕК
3	TYPES	Тип атрибута (литера) (через запятую когда несколько)

4.6.16 Переобработка кодов качества паспортов по CDI-4750

passportQCReCleaner.edited
Возобновляемая

4.6.16.1 Описание задачи

В релизе 18.5 Единый клиент обзавелся новым кодом качества SER_ISSUE_MISMATCH для паспортов, в которых год выдачи по серии отличается от реального года выдачи больше, чем на 3 года.

Раньше такие паспорта получали хороший код качества. Теперь эти документы — ожидается, что их около 1,5% от всего количества, — «Единый клиент» пометит плохим кодом качества, чтобы привлечь внимание дата-стюардов. Задача passportQCRCleaner.edited создана, чтобы переобработать существующий объем базы паспортов с кодом качества EDITED.

4.6.16.2 Запуск задачи

Задача работает по списку клиентов, что делает ее возобновляемой. Но для запуска нужно выполнить предварительные шаги:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Выполнить задачу fillTaskPartyHidsOriginal, чтобы наполнить таблицу task_party_hids идентификаторами исходных клиентов;

После этого можно выполнять задачу passportQCRCleaner.edited

4.6.16.3 Конфигурация задачи

Настраиваемые параметры:

partyType	Тип контрагента , для которых выполняется переобработка	PHYSICAL
<small>Группа reprocess Название passportQCRCleaner.edited</small>		
Параметры задачи		
Название параметра	Значение	Обязательность
partyType	PHYSICAL	
<input type="button" value="Сохранить"/> <input type="button" value="Отмена"/>		

4.6.17 Переочистка кодов качества паспортов по CDI-4145

passportQCRCleaner.notSure
Возобновляемая

4.6.17.1 Описание задачи

В релизе 17.4 Единый клиент научился определять причину «сомнительности» документов:

- *Плохая серия или номер.* Некорректный формат серии, недействительная серия, некорректный формат номера.
- *Пустая или плохая дата выдачи.* Плохая — значит, сильно в прошлом или из будущего.
- *Плохой код подразделения.* Некорректный формат кода подразделения или несуществующий код подразделения.

До этого он делил документы только на «сомнительные» (NOT_SURE) и «корректные» (EDITED).

Чтобы переобработать существующий объем базы, создана эта задача

4.6.17.2 Запуск задачи

Задача работает по списку клиентов, что делает ее возобновляемой. Но для запуска нужно выполнить предварительные шаги:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Выполнить задачу fillTaskPartyHidsOriginal, чтобы наполнить таблицу [task_party_hids](#) идентификаторами исходных клиентов;

После этого можно выполнять задачу cdi_4145_PassportQCRCleaner

4.6.17.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется переобработка	PHYSICAL

Группа reprocess
Название passportQCRCleaner.notSure

Параметры задачи

Название параметра	Значение	Обязательность
partyType	PHYSICAL	

4.6.18 Актуализация статуса невалидности документов

passportInvalidStatusReCleaner
Возобновляемая

4.6.18.1 Описание задачи

Задача проверяет указанные паспорта по перечню недействительных паспортов (ПНП).

Работает по таблице TASK_DOC_PARTY_HIDS, не блокирует работу систему. Обновляет только признак актуальности (invalid_status).

4.6.18.1.1 Какие проверки для каких типов документов проводятся

Проверка по перечню недействительных документов (INVALID) – проводится только для паспортов РФ (PASSPORT_RU)

Проверка поля EXPIRY_DATE – проводится для всех [типов документов](#). Если дата истечения уже прошла, invalidStatus = OUTDATED.

Если поле EXPIRY_DATE не заполнено производится проверка по дате выдачи:

- DRIVING_LICENSE – водительские удостоверения выдают на 10 лет.
- OVERSEAS_PASSPORT_RU – выдается максимум на 10 лет.

Для отдельных типов документов производится проверка по комбинации даты выдачи и даты рождения:

- PASSPORT_RU – выдается в 14 лет, 20 и 45.
- PASSPORT_USSR – выдается в 16 лет, 20 и 45.

4.6.18.1.2 Актуализация признака недействительности у документов

1. Выполнить одну из задач, включающих полный поиск дубликатов (`findDuplicates` / `findDuplicatesAndMerge` / `findDuplicatesAndMergeAndRebuildGroups`), убедившись, что для неё указан параметр `consumersEnabled = true`. При этом идентификаторы записей, для которых необходимо актуализировать признак актуальности документа, запишутся в таблицу `task_doc_party_hids`.
2. Запустить задачу переобработки `passportInvalidStatusReCleaner` или настроить периодический её запуск, как описано ниже.

Если задача полного поиска дубликатов периодически выполняется в составе триггера, то рекомендуется добавить задачу переобработки в тот же триггер для автоматического запуска.

4.6.18.1.3 Перепроверка всех документов с простановкой признака недействительности

1. Выполнить задачу `fillDocInvalidStatusActTaskPartyHids` (инициализировать список контрагентов для переобработки документов).
2. Запустить задачу переобработки `passportInvalidStatusReCleaner` или настроить периодический её запуск, как описано ниже.

Этот способ имеет смысл использовать при первичной простановке признака недействительности.

4.6.18.1.4 Настройка периодического запуска для задачи актуализации признака недействительности

1. Отредактировать параметры задачи `passportInvalidStatusReCleaner`, добавить параметр `executionTimeout = <длительность одного запуска в минутах>`.
2. Создать [триггер](#), который будет вызывать запуск задачи в желаемый момент времени. Если нужно актуализировать статусы непосредственно после дедапа - можно добавить `passportInvalidStatusReCleaner` в триггер с задачами на дедап.
3. Когда обработка документов будет завершена, выключить или удалить триггер.

Задача переобработки может выполняться одновременно с другими задачами.

4.6.18.2 Запуск задачи по отдельным клиентам

1. Заполнить таблицу `TASK_DOC_PARTY_HIDS` хидами актуальных клиентов, паспорта которых нужно проверить.
2. Выполнить задачу `passportInvalidStatusReCleaner`.

Для проверки паспортов у карточки, объединенной с другой, указываем `hid_party` золотой карточки.

4.6.18.3 Конфигурация задачи

Конфигурировать задачу не надо, она работает только с паспортами физических лиц

4.6.19 Закрытие исходных атрибутов по списку с переслиянием результирующих

```
attributeCloseSourceByHids
```

4.6.19.1 Описание задачи

Данную задачу следует использовать в случае, когда требуется массово закрыть исходные атрибуты клиентов по списку, с переслиянием результирующих если нужно.

4.6.19.2 Запуск задачи

1. Выполнить задачу cleanTaskParameterTables. Она очистит таблицу task_attribute_hids.
2. Подготовить список атрибутов для закрытия. Для каждого атрибута выяснить:
 - a. Тип и идентификатор атрибута в ЕК;
 - b. Тип и идентификатор контрагента в ЕК.
3. Записать список в таблицу task_attribute_hids:

```
INSERT INTO task_attribute_hids (attribute_type, hid_attribute, party_type, hid_party)
SELECT 'ADDRESS', hid_address, party_type, hid_party FROM ...
```

4. Запустить задачу attributeCloseSourceByHids.

4.6.19.3 Особенности задачи

Задача закрывает только атрибуты у исходных (слитых и оригинальных) клиентов.

Игнорирует атрибуты, которые:

- относятся к «золотым» клиентам;
- закрыты;
- не найдены по идентификатору.

Проигнорированные атрибуты задача пишет в лог-файл.

Если атрибут находится в закрытом контрагенте, задача упадет с ошибкой

4.6.20 Переобработка телефонов

phoneExtraFieldsReCleaner

[Возобновляемая](#)

4.6.20.1 Описание задачи

Задача используется для обновления дополнительной информации по телефонам.

Например:

- изменились часовые пояса и нужно переобработать текущие;
- клиент [сменил оператора связи](#), а в «Едином клиенте» остался старый.

Можно выборочно переобработать одного или несколько клиентов, а также создать триггер на периодическую актуализацию всех базы данных.

4.6.20.2 Выборочная переобработка

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids от существующих записей;
2. Выяснить идентификаторы исходных контрагентов (hid_party), для которых необходимо выполнить актуализацию данных;

3. Добавить список данных идентификаторов в таблицу task_party_hids вручную или с помощью вызова одной из [задач](#).

Пример ручного наполнения таблицы:

```

-- Хотим переобработать конкретного человека
insert into task_party_hids(hid_party, party_type) values (1, 'PHYSICAL');

-- Нужно переобработать набор клиентов. Работает только для исходных, поэтому ставим
условие merged_status in (0, 1)
insert into task_party_hids
select hid_party from physical_party where merged_status in (0, 1) and ... ;

```

4.6.20.3 Переобработка всей базы

Если хотите переобработать всю базу, сделайте триггер на задачи:

1. cleanTaskParameterTables – для очистки таблицы task_party_hids;
2. fillTaskPartyHidsOriginal – для заполнения таблицы данными на переобработку;
3. phoneExtraFieldsReCleaner – сама переобработка.

4.6.20.4 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется переслияние	PHYSICAL
massiveChanging	Оповещение Фактора инкрементальными изменениями. При массовой переобработке нужно задать true, а потом запустить полный поиск дубликатов	false

4.6.20.5 Особенности задачи

Задача обрабатывает только атрибуты у исходных (слитых и оригинальных) клиентов.

Игнорирует идентификаторы (без ошибки в лог), которые:

- относятся к «золотым» клиентам;
- закрыты;
- не найдены.

Задача обновляет принудительно сразу все второстепенные поля телефона:

- оператор;
- регион и адрес оператора;
- часовой пояс;
- профиль телефона;
- популярность;
- комментарий.

Но не трогает основные:

- код страны;
- код города;
- номер телефона;
- добавочный.

4.6.21 Задача восстановления КА на версию по списку HID

partyRecoverByHids

[Возобновляемая](#)

4.6.21.1 Описание задачи

Данную задачу следует использовать в случае, когда требуется массово восстановить контрагентов (например, при некорректном обновлении или ошибочном закрытии).

Задача восстановит контрагенты из заданного списка HID-ов, которые были не закрыты в указанный промежуток времени.

4.6.21.2 Особенности работы

Контрагент восстанавливается на версию, существующую в указанный промежуток времени, при этом:

- если контрагент не найден, идентификатор удаляется из таблицы task_party_hids, задача продолжает работу;
- если в указанное время контрагент не существовал, идентификатор удаляется из таблицы task_party_hids, задача продолжает работу;
- если контрагент закрыт или влит в золотой, задача прервется с ошибкой;
- изменений в связях не происходит;
- атрибуты восстанавливаются.

4.6.21.3 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Добавить идентификаторы контрагентов hid для переоткрытия в таблицу task_party_hids;

Пример:

```
insert into task_party_hids(hid_party, party_type) values (1, 'PHYSICAL');
commit;
```

3. Запустить задачу partyRecoverByHids

4.6.21.4 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется восстановление	PHYSICAL
targetDate	Дата, на которую надо восстановить	—

Формат даты – 10.06.2018 00:00:03.

4.6.22 Задача переочистки реквизитов

reCleanRequisiteTask
Возобновляемая

4.6.22.1 Описание задачи

Использовать следует в случае, когда нужно переочистить один или несколько реквизитов.

4.6.22.2 Особенности задачи

Работает через провайдер для получения всех оригинальных контрагентов.

Переочищенные реквизиты обновляются принудительно.

4.6.22.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется задача	PHYSICAL
poolSize	Максимальное количество потоков	8
pageSize	Размер пачки	1000
massiveChanging	Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate	false

4.6.22.4 Логирование

Задача логирует переочистку каждые N записей (по умолчанию pageSize).

```
INFO RequisiteReCleanTask - Processed 1000 'PHYSICAL' parties. Iterate 1000 parties.
```

В итоговой статистике выводится:

```
processed N entities; 0 with errors; took X seconds or X.X ms; speed XX.0 records/sec
```

4.7 Системные задачи

4.7.1 Системные задачи

- `changeJdbcUser` – изменение пользователя безопасности

Задачи, предназначенные для работы с индексом полнотекстового поиска и другими внутрисистемными структурами:

- `deleteOutdatedConflicts` – удаление старых обработанных конфликтов;
- `duplicateGroupFullRebuild` – полное перестроение групп дубликатов для сравнения вручную;
- `enginesFullRebuild` – полное перестроение поискового индекса;
- `graphEngineFullRebuild` – полное перестроение графа связей;
- `conflictEngineFullRebuild` – полное перестроение индекса конфликтов обновления;
- `allEnginesRebuildByName` – перестроение указанного поискового индекса;
- `allEnginesRebuildInvalid` – перестроение всех не прошедших валидацию индексов;
- `searchEngineActualizeByParties` – актуализация поискового индекса по списку контрагентов;
- `searchEngineOptimize` – оптимизация поисковых индексов (граф связей, индексы по контрагентам, конфликты), группы дубликатов не оптимизируются;
- `searchEngineOptimizeByName` – оптимизация указанных поисковых индексов;
- `searchEngineValidate` – валидация всех поисковых индексов;
- `executeDBProcedure` – выполнение хранимой процедуры.

4.7.2 `allEnginesRebuildByName`. Перестроение указанных поисковых индексов

4.7.2.1 Общее описание

Задача `allEnginesRebuildByName` предназначена для перестроения поисковых индексов по списку.

4.7.2.2 Конфигурация задачи

Параметр	Тип	Описание	Примеры значений	По-умолчанию
<code>enginesToRebuild</code>	Строка	Названия индексов, которые будут перестроены (через запятую).	PHYSICAL, GRAPH	
<code>skipOptimize</code>	true / false	Пропустить оптимизацию поискового индекса	true	false

4.7.2.3 Допустимые значения параметра `enginesToRebuild`

- PHYSICAL – физические лица;
- GRAPH – граф связей;

- MANUAL_DUPLICATE_GROUP – группы для ручной проверки дубликатов;
- INDIVIDUAL – индивидуальные предприниматели;
- LEGAL – юридические лица;
- AGREEMENT – договора;
- EGR – ЕГРЮЛ.

4.7.3 Общее описание

Поисковый индекс может быть испорчен (если во время его перестроения возникла ошибка или был остановлен сервер Единого клиента). Это приводит к тому, что пользователи при поиске видят неактуальную информацию. Чтобы исправить ситуацию, необходимо перестроить испорченный индекс. Задача *allEnginesRebuildInvalid* запускает перестроение всех индексов, не прошедших валидацию.

Валидация индексов запускается:

- Автоматически – при старте ЕК.
- Вручную – задача *searchEngineValidate*.

Если какой-то индекс не прошел валидацию, в лог выводится сообщение с уровнем ERROR:

```
ERROR engine.SearchEngineValidator - Invalid 'PHYSICAL' index document count (expected 1000, got 999). Index may be corrupted. Please rebuild ASAP.
```

Как только в логах написало "rebuild ASAP" - индекс помечается как *invalid* и будет перестроен при запуске *allEnginesRebuildInvalid*.

4.7.4 Актуализация поискового индекса по списку контрагентов

searchEngineActualizeByParties

[Возобновляемая](#)

4.7.4.1 Общее описание

Данную задачу следует использовать при необходимости актуализировать поисковый индекс по конкретному списку контрагентов.

4.7.4.2 Запуск задачи

1. Выполнить задачу *cleanTaskParameterTables* для очистки таблицы *task_party_hids*;
2. Добавить идентификаторы контрагентов *hid* для актуализации в таблицу *task_party_hids*;
3. Сконфигурировать и запустить задачу (см. ниже).

Пример:

```
insert into task_party_hids(hid_party, party_type) values (1, 'PHYSICAL');
commit;
```

4.7.4.3 Конфигурация задачи

В задаче конфигурируется только параметр partyType – [тип контрагента](#) (по умолчанию PHYSICAL).

Группа system
Название searchEngineActualizeByParties

Параметры задачи

Название параметра	Значение	Обязательность
partyType	PHYSICAL	

Сохранить Отмена

4.7.5 Выполнение вспомогательной хранимой процедуры

4.7.5.1 Общее описание

Задача *executeDBProcedure* предназначена для запуска произвольной хранимой процедуры.

Позволяет добавить в [цепочку задач](#) выполнение PL/SQL-кода, например, для предварительного отбора данных, с которыми будут работать последующие задачи.

4.7.5.2 Конфигурация задачи

Параметр	Тип	Описание
procedureName	Строка	Название процедуры
procedureParams	Строка	Параметры одной строкой в формате "param1;param2;..."

4.7.6 Удаление старых обработанных конфликтов

```
deleteOutdatedConflicts
```

4.7.6.1 Общее описание

Удаляет обработанные конфликты ([статус](#) != 0) с датой обработки (resolved) старше, чем 30 дней назад от текущей даты.

Запускается по триггеру в 21 по субботам.

4.7.6.2 Конфигурация задачи

Параметр	По умолчанию	Описание
deleteAfterDays	30	Количество дней, которые конфликты хранятся в базе

4.7.7 Создание и редактирование системных учетных записей

changeJdbcUser
Системная

4.7.7.1 Описание задачи

Задача добавляет, изменяет и удаляет in-memory пользователей.

Такие пользователи нужны для настройки интеграции систем, чтобы не проседать по производительности из-за проблем в LDAP.

4.7.7.2 Конфигурация задачи

Параметр	Описание	По умолчанию
username	Имя пользователя, чувствительно к регистру	
passwordHash	Зашифрованный пароль (по алгоритму Scrypt)	
roles	Необходимые роли (разделитель - ",")	
delete	Нужно ли удалить пользователя (по умолчанию false)	false

При удалении пользователя (delete = true) пароль и роли можно не указывать.

При изменении (delete = false) можно указать только пароль или только роли, тогда именно этот параметр и обновится.

4.7.7.3 Примеры

4.7.7.3.1 Создать нового пользователя

Параметр	Значение
username	user1
passwordHash	\$s0\$e0801\$6ZeK6idgRJpL3Za0sezDFA== \$ZpdgXt8jWpU02EuZ9btIhHPiB9A1S2eWOUHqhDx27Co=
roles	ROLE_ADMINISTRATOR, ROLE_PERFORMER
delete	false

4.7.7.3.2 Изменить пароль

Параметр	Значение
username	user1
passwordHash	\$s0\$e0801\$SKg6OZwd84heZhCMn7nvcA== \$MixhOphiPBsGIWGy0sWyLhIVsga082LCL0GEVhRNKWM=
delete	false

4.7.7.3.3 Изменить роли

Параметр	Значение
username	user1

Параметр	Значение
roles	ROLE_OPERATOR
delete	false

4.7.7.3.4 Удалить пользователя

Параметр	Значение
username	user1
delete	true

4.7.7.4 Шифрование пароля через утилиту scryptpass

Утилита `scryptpass` используется для шифрования пароля по алгоритму `SCrypt`.

4.7.7.4.1 Установка

Должна быть определена переменная окружения `JAVA_HOME`.

Распаковать архив `scryptpass.zip`.

4.7.7.4.2 Запуск шифрования паролей указанных в файле
Добавить в файл `scryptpass\data.txt` пароли для шифрования.

Запустить `scryptpass/run.bat`.

На выходе получится файл `scryptpass/rypted.txt`.

4.7.7.4.2.1 Пример:

На вход:

Code Block 1 `scryptpass\data.txt`

```
# Файл в кодировке UTF-8 (без BOM)
# Комментарии копируются как есть
# Можно задавать несколько колонок, разделитель табуляция. Кодировка первая колонка.
Ниже строки для примера.
пароль1
пароль2   неcodируемое_значение_1   неcodируемое_значение_2
```

На выходе:

Code Block 2 `rypted.txt`

```
# Файл в кодировке UTF-8 (без BOM)
# Комментарии копируются как есть
# Можно задавать несколько колонок, разделитель табуляция. Кодировка первая колонка.
Ниже строки для примера.
$S0$e0801$naKBJ92C00i11ERiNIXNfQ==$DgjGHcoeGJYA6b6qoR/PdYTnkWSVFHwfziKcWglVo+0
=
```

```
$s0$e0801$iehrFvIwQ9FNiO574WwkbQ==$OSfM7rTReZJlGfM6zK3UUo7e1igsSzMnKqRXj14CQHg
=некодируемое_значение_1   некодируемое_значение_2
```

4.7.7.4.3 Запуск шифрования паролей из консоли

В консоли ввести:

```
{path}\scryptpass> run password1 password1 password1
```

В этом случае в консоль будут выведены хеши для переданных значений:

```
$s0$e0801$jyKC7AAbZsu8XBcuRHP|pw==$60d4+3vRCO5ZOa5l6x0bw+EK3FIJzfET+wCgU33LCZc
=
$s0$e0801$fqDFHtqwp7MUEFKkZJCUUw==$VBNVjgjqy2QdzZYWUwfGPtwywlbeB197M0Xykr71Z
30=
$s0$e0801$9lqwdWeg2R5exDQui5cHDQ==$cSrlW0H30cmRNf0jeSDExser43b4vc1RZblqAqB6wxw
=
```

4.7.8 searchEngineOptimize. Оптимизация поисковых индексов

4.7.8.1 Общее описание

Задача searchEngineOptimize предназначена для оптимизации поисковых индексов:

- граф связей – GRAPH;
- конфликты обновления – CONFLICT;
- контрагенты – PHYSICAL (и др настроенные [типы контрагентов](#)).

Индекс групп для ручной проверки дубликатов (MANUAL_DUPLICATE_GROUP) в рамках данной задачи не оптимизируется.

4.7.8.2 Конфигурация задачи

Параметр	По умолчанию	Описание
poolSize	2	Максимальное количество потоков

4.7.9 graphEngineFullRebuild перестроение графа связей

Граф связей перестраивается с CONCURRENT стратегией - эта стратегия выставлена по умолчанию для графа, потому что обновление графа происходит в онлайн синхронно и не подвешивает онлайн операции.

Не поддерживается продолжение перестроения после остановки.

4.7.9.1 Конфигурация задачи

Параметр	Тип	Описание	Примеры значений
strategy	Строка	Стратегия перестроения, блокирующая, не блокирующая	BLOCKING, CONCURRENT

Параметр	Тип	Описание	Примеры значений
poolSize	Число	Количество потоков, по умолчанию берет из engine.rebuild.poolSize	

4.7.10 enginesFullRebuild. Полное перестроение поискового индекса

4.7.10.1 Общее описание

Задача enginesFullRebuild предназначена для полного перестроения поисковых индексов по списку.

Без инструкций от команды ХФЛабс запускать не надо, так как может занимать продолжительное время, если в системе много типов сущностей: физические лица, юридические лица, ЕГРЮЛ... Вместо нее обычно используется дочерняя задача [allEnginesRebuildByName](#), чтобы не тратить время на перестроение ненужных индексов.

4.7.10.2 Конфигурация задачи

Параметр	Тип	Описание	Примеры значений	По-умолчанию
skipOptimize	true / false	Пропустить оптимизацию поискового индекса	true	false

4.8 Экспорт данных

Группа задач, предназначенных для экспорта данных из Единого клиента:

- fullExportRelations — полный экспорт эталонных связей;
- incrementalExportRelations — инкрементальный экспорт эталонных связей.

Экспорт данных		
Действия	Название	Описание
Детали Редактировать Выполнить	fullExportRelations	Полный экспорт эталонных связей
Детали Редактировать Выполнить	incrementalExportRelations	Инкрементальный экспорт эталонных связей

4.8.1 Инкрементальный экспорт эталонных связей

incrementalExportRelations

Непрерываемая

Остановка задачи приведет к некорректному результату в RELATION_EXPORT, поэтому после нужно обязательно выполнить Полный экспорт эталонных связей.

Актуализирует таблицу [RELATION_EXPORT](#) по изменениям в связях, прошедшим с момента последнего успешного запуска задачи.

4.8.1.1 Концепция задачи:

В результате работы задачи данные таблицы RELATION_EXPORT становятся такими, как если бы был запущен полный экспорт связей. При этом вместо алгоритма "очистить старое, вставить актуальное заново" используется алгоритм "внести изменения в существующее"

4.8.1.2 Правила работы:

1. Удаляет закрытые связи из таблицы и связи, которые стали указывать на одну сущность.
2. Выгружает новые актуальные связи с уникальным сочетанием типа связи и типов связанных объектов и исторических идентификаторов и порядка объектов во взаимосвязи. Концы связи всегда представляют актуальные объединенные сущности, если связь указывает на сущность, которая была объединена.
3. Обновляет связи, один или оба конца которых стали указывать на новую объединенную сущность.

4.8.2 Полный экспорт эталонных связей

fullExportRelations

[Непрерываемая](#)

Полностью очищает таблицу [RELATION_EXPORT](#) и потом заполняет ее актуальными связями с учетом объединения клиентов, договоров и других сущностей. Используется для первичного заполнения таблицы [RELATION_EXPORT](#) или ее полного перестроения. Для актуализации связей после загрузки инкремента используйте задачу [инкрементального экспорта связей](#).

4.8.2.1 Правила заполнения:

1. Выгружаются только актуальные связи с уникальным сочетанием типа связи и типов связанных объектов и исторических идентификаторов и порядка объектов во взаимосвязи.
2. Если любая из сторон связи – контрагент, который был влит, то вернуть объединенного контрагента.
3. Не выгружать связи, оба конца которых указывают на одну сущность.

4.8.2.2 Задача прошла, таблица пустая

Задача работает по поисковому индексу – графу связей. Если после выполнения задачи таблица [RELATION_EXPORT](#) пуста, хотя связи в системе есть: перестройте граф и повторно запустите задачу:

1. Выполнить задачу graphEngineFullRebuild
2. Выполнить задачу fullExportRelations

4.9 Диагностика

4.9.1 Когда использовать

По просьбе сотрудника техподдержки.

4.9.2 Как собрать диагностику

1. В АРМ Администратора зайдите на вкладку *Задачи*, раздел *Диагностика*.

Действия	Название	Описание
Детали Редактировать Выполнить	diagnosticLogs	Сбор логов
Детали Редактировать Выполнить	diagnosticMetrics	Сбор метрик
Детали Редактировать Выполнить	diagnosticParty	Сбор диагностики по контрагенту
Детали Редактировать Выполнить	diagnosticProfiles	Сбор диагностики с указанным профилем
Детали Редактировать Выполнить	diagnosticTaskStat	Сбор диагностики по задаче
Детали Редактировать Выполнить	mergedPartiesCheck	Найти проблемы в золотых карточках
Детали Редактировать Выполнить	mergedPartiesCheckAndRemerge	Найти и исправить проблемы в золотых карточках
Детали Редактировать Выполнить	runDBCheckers	Проверка БД на непротиворечивость

2. Соберите диагностику:

- [Логи](#) (diagnosticLogs).
- [Информация по контрагенту](#) (diagnosticParty).
- [Диагностика по профилю](#) (diagnosticProfiles).
- [Проверка БД на непротиворечивость](#) (runDBCheckers).
- [Проверка на непротиворечивость и исправление проблем](#) (mergedPartiesCheckAndRemerge).

3. Архив собранной информации ЕК автоматически отправит по электронной почте администратору и в техподдержку (после [настройки](#)).

Копию архива сохранит на сервере во временной директории ({TEMP}/diagnostic.{date}_{time}).

4. Иногда случается, что логи увеличиваются и архив не помещается во вложение письма. Тогда «Единый клиент» выбирает все файлы до превышения размера в 100Мб и этот архив отправляет в письме. Полный архив диагностики сохраняется во временной директории {TEMP} с постфиксом _full.

5. В папке {TEMP} «Единый клиент» хранит диагностики только за последние три дня (начиная с версии 18.14). Все что старше — удаляется, чтобы беречь ресурсы.

6. Название архива и его месторасположение можно посмотреть в панели администратора

```
– diagnosticClusterTaskPerformer: processed 0 entities; 0 with errors; took 7 seconds or 7.39 sec;
diagnostic saved to C:\cdi\tmp\diag_10.0.18.47_181008_2355.zip; diagnostic sent to
cdi_qa@hflabs.ru
```

4.9.3 Сбор логов

4.9.3.1 Общее описание

Задача diagnosticLogs предназначена для сбора логов ЕК и Фактора в диапазоне дат [lowerBound, upperBound].

4.9.3.2 Конфигурация задачи

Группа diagnostic
Название diagnosticLogs

Параметры задачи

Название параметра	Значение	Обязательность
profiles	logs	
params	lowerBound=;upperBound=	

Сохранить Отмена

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
sendEmail	Признак необходимости отправки диагностики по почте	true
params	Диапазон дат, задается в формате гггг-мм-дд. По умолчанию даты не заданы, тогда будут собраны только текущие логи.	lowerBound=;upperBound=
runInCluster	Для сбора в режиме горячего резерва	true
sendTo	Если нужно временно перекрыть настройку «кому писать письмо» из конфигурации	

4.9.3.3 Пример

params	Что собирает диагностика
lowerBound=2015-08-08	Логи с 08.08.2015 по текущую дату.
lowerBound=2015-08-08;upperBound=2015-08-11	Логи с 08.08.2015 по 11.08.2015

4.9.4 Сбор диагностики по контрагенту

4.9.4.1 Общее описание

Задача diagnosticParty предназначена для сбора информации по конкретному контрагенту или по группе контрагентов.

4.9.4.2 Конфигурация задачи

Создать/Редактировать задачу

Группа diagnostic
Название diagnosticParty

Параметры задачи

Название параметра	Значение	Обязательность
sendTo	nikolaig@hflabs.ru	
profiles	sql.party,mapdb-state	
params	hid=4447	

Сохранить Отмена

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
sendEmail	Признак необходимости отправки диагностики по почте	true
params	HID конкретного контрагента или группы контрагентов, указанных через ","	hid=1024,1025,1026

4.9.5 Сбор диагностики с указанным профилем

4.9.5.1 Общее описание

Задача diagnosticProfiles собирает [диагностику по профилю](#).

4.9.5.2 Конфигурация задачи

Группа diagnostic
Название diagnosticProfiles

Параметры задачи

Название параметра	Значение	Обязательность
runInCluster	true	
profiles	config,versions	

Сохранить Отмена

Параметр	Описание	Значение по умолчанию
runInCluster	Сбор диагностик и в режиме горячего резерва	true
profiles	Список профилей,	config,db.connection,logs,search,sql.counts,sql.db,sql.metadata,versions

Параметр	Описание	Значение по умолчанию
	указанных через ","	

4.9.6 Проверка БД на непротиворечивость

4.9.6.1 Общее описание

Задача *runDBCheckers* проверяет целостность и непротиворечивость бизнес-объектов системы. Проверка производится на уровне данных в таблицах БД.

Найденные проблемы записывает в таблицу TMP\$PROBLEMS и отправляет по электронной почте администратору.

4.9.6.2 Конфигурация задачи

По умолчанию осуществляется проверка всех таблиц системы (запуск без параметров).

Группа diagnostic
Название runDBCheckers

Параметры задачи

Название параметра	Значение	Обязательность
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	

Если нужно запустить определенную проверку (по просьбе сотрудника из ХФЛабс), настройте параметр `procedureParams`.

Группа diagnostic
Название runDBCheckers

Параметры задачи

Название параметра	Значение	Обязательность
<input type="text" value="procedureParams"/>	<input type="text" value="1,2,10,5"/>	
<input type="text"/>	<input type="text"/>	

Параметр	Описание	Пример
<code>procedureParams</code>	Список проверок , указанных через ","	1,2,10,5

4.9.6.3 Проверка БД на непротиворечивость (техподдержка, checker)

4.9.6.3.1 Проверки

Проверяется наличие проблем:

Тип проблемы	Описание
0	Атрибут есть в исходном, нет в МА
1	Атрибут есть в слитом, нет в МА Если он что-то вернет, проанализировать и тогда применять <code>fix_ma.zip</code> (См. CDI-2348)


Тип проблемы	Описание
2	Атрибута нет в слитом, есть в МА
3	Атрибута нет в исходном, есть в МА (менее реальный случай, чем 2)
4	Атрибут есть в исходном с merged_status = 0, МА не закрыт
5	Атрибут в слитом контрагенте - исходный атрибут в МА См. BINBANK-268 , там один атрибут в слитом был завязан на другой атрибут в том же самом слитом контрагенте
6	Атрибут есть в итоговом контрагенте (открытый), но все МА закрыты
7	Атрибут есть в итоговом контрагенте (открытый), но все исходные закрыты
8	Одна исходная запись влита в несколько результирующих. Поиск идентификаторов результирующих записей по исходным, для которых есть некорректность в слиянии. По результирующим можно снять диагностику и разбираться дальше, что с ними делать. Если максимальное количество результатов слияния равно двум, то далее стоит выполнить ЭТОТ СКРИПТ . После реализации CDI-2863 чекер неактуален
9	Контрагент имеет статус 1, при этом связки MERGED нет, или она закрыта
10	Результирующий атрибут состоит из одного исходного, но их RAW_ID не совпадают

Тип проблемы записывается в поле PROBLEM_TYPE в формате {тип_объекта}_{тип_проблемы}. Пример:

HID_PARTY	PROBLEM_TYPE	HID_ATTR
1	PHONE_0	11

Говорит нам о том, что у исходного контрагента (куда-то влитого на данный момент) с hid = 1 есть телефон с hid = 11, которого нет в МА.

4.9.6.3.2 Дополнительная информация о проблеме

 Только для проблем 2 и 3 возвращается не hid_party и hid_attr, а данные из таблицы MERGED_ATTRIBUTE.

HID_PARTY	PROBLEM_TYPE	HID_ATTR
исходный	0	атрибут исходного
итоговый	1	атрибут итогового
ma.id_merged	2	ma.final_hid
ma.id_merged	3	ma.original_hid
исходный	4	атрибут исходного
итоговый	5	атрибут итогового
итоговый	6	атрибут итогового
итоговый	7	атрибут итогового

HID_PARTY	PROBLEM_TYPE	HID_ATTR
итоговый	8	<i>null</i>
исходный	9	<i>null</i>
итоговый	10	атрибут итогового

Старый алгоритм исправления базы (нынче замененный на новую [модную задачу](#)):

1. Проверьте базу на целостность:
runDBCheckers.
2. Подготовьте слитых контрагентов для переслияния:
executeDBProcedure с параметром procedureName = fix_pkg.prepare_for_remerge.
3. Переслейте физических или юридических лиц, или индивидуальных предпринимателей:
reMergeFullByPartyHids с параметром partyType = PHYSICAL или LEGAL, или INDIVIDUAL.

4.9.7 Профили диагностики

Задачи диагностики предоставляют возможность сбора следующей информации:

Профиль	Параметры	Какую информацию собирает
logs	lowerBound upperBound	Журналы ЕК и Фактора в диапазоне дат [lowerBound, upperBound]. Оба параметра опциональные, задаются в формате гггг-мм-дд.
sql.party	hid	Информация по конкретному контрагенту или по группе контрагентов, указанных через ","
mapdb-state	hid	Состояние mapdb-индекса по контрагенту. Обычно собирается в паре с sql.party
graph-state	hid	Состояние графа связей по контрагенту.
cdi.jvm		Информация о JVM Единого клиента: дампы потоков, параметры обработки HTTP-запросов, использование ресурсов ОС
config		Конфигурационные файлы
db.connection		Информация о соединении с БД
factor.jvm		Информация о JVM Фактора: дампы потоков, параметры обработки HTTP-запросов, использование ресурсов ОС.
search		Статистика по поисковому индексу
sql.common		Системные таблицы
sql.counts		Количество контрагентов
sql.db		Физические характеристики БД (нужны права GRANT SELECT_CATALOG_ROLE TO <user>;)
sql.duplicate		Общая статистика по дубликатам

Профиль	Параметры	Какую информацию собирает
sql.duplicate.physical		Статистика по дубликата (физические лица)
sql.metadata		Метаданные БД
sql.quality.physical		Статистика по качеству данных (физические лица)
versions		Версии ЕК, Фактора и справочников Фактора
sql.checker		Проверка целостности и согласованности бизнес-объектов системы
sql.stat.task	task=task_key	Информация обо всех запусках указанной задачи. task_key - ключ задачи как в админке, например, findDuplicates
user.env		Информации о браузере
user.event	lowerBound upperBound	Информации о событиях в веб-интерфейсе в диапазоне дат [lowerBound, upperBound]. Оба параметра опциональные, задаются в формате гггг-мм-дд.
metadata		Метаинформация о модели данных для автоматической генерации документации

4.9.8 Проверка слитых контрагентов на непротиворечивость

4.9.8.1 Общее описание

Задача *mergedPartiesCheck* проверяет целостность и непротиворечивость бизнес-объектов системы.

Найденные проблемы записывает в таблицу TMP\$PROBLEMS и отправляет по электронной почте администратору.

4.9.8.2 Конфигурация задачи

По умолчанию осуществляется проверка всех таблиц системы (запуск без параметров) на всех типах контрагентов.

4.9.8.3 Особенности работы

Задачу необходимо запускать «в ночь», когда больше ничего не работает.

Если во время проверки БД идет активное изменение данных (трансформация или ручные правки операторов), возможны ложно-положительные результаты.

4.9.8.4 Поиск неконсистентности "Контрагент закрыт, связь – нет"

4.9.8.4.1 Общее описание

Задачи *findInconsistentRelations* и *findInconsistentRelationsAndClose* проверяют бизнес-объекты системы на неконсистентность:

Контрагент закрыт, но его связи открыты

Задача *findInconsistentRelations* только записывает найденные проблемы в таблицу *task_relation_ids*, а задача *findInconsistentRelationsAndClose* находит и сразу закрывает некорректные связи.

4.9.8.4.2 Конфигурация задачи

По умолчанию осуществляется проверка всех таблиц системы (запуск без параметров) для всех типов контрагентов.

4.9.9 Найти и исправить проблемы в золотых карточках

4.9.9.1 Общее описание

Задача *mergedPartiesCheckAndRemerge* проверяет целостность бизнес-объектов системы и автоматически исправляет проблемы.

4.9.9.2 Конфигурация задачи

По умолчанию осуществляется проверка всех таблиц системы (запуск без параметров) на всех типах контрагентов.

4.9.10 Сбор диагностики в режиме горячего резерва

Задачи диагностики содержат параметр **runInCluster**, который отвечает за одновременный сбор диагностической информации с разных экземпляров ЕК.

4.9.10.1 Значения параметра runInCluster :

Значение параметра	Описание
false	Сбор диагностики локально
true	Сбор диагностики с разных экземпляров ЕК.

4.9.10.2 Значения параметра runInCluster в задачах диагностики по умолчанию:

Задача	Значение по умолчанию	Описание
diagnosticParty	false	Задача предназначена для сбора информации по конкретному контрагенту или по группе контрагентов. Значение параметра runInCluster настраивается в task-performer-diagnostic.xml.
diagnosticLogs	true	Задача предназначена для сбора логов ЕК и Фактора в диапазоне дат. Значение параметра runInCluster настраивается в task-performer-diagnostic.xml.
diagnosticProfiles	true	Задача предназначена для сбора информации по любому профилю диагностики . Значение параметра runInCluster настраивается в jobs_system.xml или через интерфейс Администратора.

При запуске задачи с runInCluster = true формируется событие, которое уходит на вторую ноду и запускает там сбор диагностики. В истории это будет один запуск задачи. Задача

не дожидается завершения сбора информации на второй ноде, она только отгружает событие.

4.9.11 Сбор метрик

4.9.11.1 Общее описание

Задача `diagnosticMetrics` предназначена для сбора анонимной статистики по использованию ЕК.

Не всегда Единый клиент приносит дата-стюардам только хорошие эмоции. Иногда что-то неудобно в интерфейсе или непонятно, как работает та или иная возможность. К сожалению, пользователи редко делятся своими впечатлениями. Поэтому мы добавили в Единый клиент сбор статистики использования мелких «фич» вроде поиска на списке клиентов, фильтрации на столбцах и тому подобного.

Статистика полностью анонимная (не записываем даже имя пользователя) и деперсонализированная (не сохраняем никакие данные о клиентах)

4.9.11.2 Конфигурация задачи

Группа `diagnostic`
Название `diagnosticMetrics`

Параметры задачи

Название параметра	Значение	Обязательность
<input type="text" value="profiles"/>	<input type="text" value="user.event,user.env"/>	
<input type="text" value="params"/>	<input "="" type="text" value="lowerBound=;upperBound="/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<code>profiles</code>	Профиль диагностики	<code>user.event,user.env.</code>
<code>params</code>	Диапазон дат, задается в формате гггг-мм-дд. По умолчанию даты не заданы.	<code>lowerBound=;upperBound=</code>
<code>attachLogs</code>	Нужно ли собирать вместе с метриками логи	<code>false</code>

4.9.11.3 Пример

<code>params</code>	Что собирает диагностика
<code>lowerBound=2015-08-08</code>	Метрики с 08.08.2015 по текущую дату.
<code>lowerBound=2015-08-08;upperBound=2015-08-11</code>	Метрики с 08.08.2015 по 11.08.2015

4.9.12 Сбор диагностики по задаче

4.9.12.1 Общее описание

Задача `diagnosticTaskStat` предназначена для сбора информации по задаче:

- Когда запускалась;
- Сколько раз;
- Сколько было успешно, сколько нет;
- Сколько времени занимало – можно быстро проверить, нет ли деградации.

4.9.12.2 Конфигурация задачи

Создать/Редактировать задачу

Группа diagnostic
Название diagnosticTaskStat

Параметры задачи

Название параметра	Значение
<input type="text" value="profiles"/>	<input type="text" value="sql.stat.task"/>
<input type="text" value="params"/>	<input type="text" value="task=findDuplicates"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Настраиваемые параметры:

Параметр	Значение	Описание	Значение по умолчанию
params	task=<название задачи>	<p>Название задачи, по которой нужна диагностика. Берется из общего списка задач, колонка «Название».</p> <p>Можно указать только одну задачу</p>	<p>task=findDuplicates</p> <p>↓</p> <p>Соберет диагностику по задаче findDuplicates</p>

4.9.13 Сбор логов за сегодня

4.9.13.1 Общее описание

Задача diagnosticLogsForToday аналогична задаче diagnosticLogs и собирает логи ЕК и Фактора только за сегодня.

4.9.13.2 Конфигурация задачи

Создать/Редактировать задачу

Группа diagnostic
Название diagnosticLogsForToday

Параметры задачи

Название параметра	Значение	Обязательность
sendEmail	true	
profiles	logs	

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
sendEmail	Признак необходимости отправки диагностики по почте	true
runInCluster	Для сбора в режиме горячего резерва	true
profiles	Профиль диагностики	logs
sendTo	Если нужно временно перекрыть настройку «кому писать письмо» из конфигурации	

4.9.14 Сбор метрик за предыдущий месяц

4.9.14.1 Общее описание

Задача diagnosticMetricsForPreviousMonth аналогична задаче diagnosticMetrics и предназначена для сбора анонимной статистики по использованию ЕК только за предыдущий месяц.

Статистика полностью анонимная (не записываем даже имя пользователя) и деперсонализированная (не сохраняем никакие данные о клиентах)

4.9.14.2 Конфигурация задачи

Группа diagnostic
Название diagnosticMetricsForPreviousMonth

Параметры задачи

Название параметра	Значение	Обязательность
profiles	user.event.previous.month,user.env	

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
profiles	Профиль диагностики	user.event.previous.month,user.env
attachLogs	Нужно ли собирать вместе с метриками логи	false

4.10 Триггеры

Действия	Название	Дата следующего запуска	Cron выражение
Редактировать Удалить Выполнить	afterDatabaseBackupTrigger		
Редактировать Удалить Выполнить Выключить	deleteOutdatedConflictsTrigger	24.03.2018 21:00:00	0 0 21 ? * SAT
Редактировать Удалить Выполнить Включить	detectPopularAttributes		
Редактировать Удалить Выполнить	rebuildAllIndexesTrigger		
Редактировать Удалить Выполнить Выключить	refreshDataStatisticTrigger	23.03.2018 2:00:00	0 0 2 * * ?
Редактировать Удалить Выполнить Выключить	searchEngineOptimizeTrigger	24.03.2018 22:00:00	0 0 22 ? * SAT

Создать триггер

Страница *Триггеры* необходима для отслеживания настроенных в системе триггеров, времени их последнего и следующего запуска, а также ручного выполнения.

4.10.1 Триггер

Триггер - задача или группа задач, запускающаяся автоматически по заранее настроенному расписанию (каждые 10 минут, каждый час, раз в день в 9 вечера, раз в неделю и так далее).

В триггере есть возможность указать цепочки последовательно запускающихся задач.

4.10.1.1 Пример

Необходимо загружать в систему новые данные и сразу искать дубликаты с уже существующими в системе клиентами.

1. Можно вручную, через раздел *Задачи*, выполнить:

- *Загрузка инкремента*
- *Инкрементальный поиск дубликатов и слияние*

Но для этого надо выполнить одну задачу, дождаться ее завершения, а потом выполнить вторую.

2. Можно настроить 2 разных триггера, подстроив расписание так, чтобы они не пересекались:

- *Загрузка инкремента* - каждый час с 7 утра до 7 вечера
- *Инкрементальный поиск дубликатов и слияние* - каждый день в 21.00

3. А можно просто настроить триггер *Загрузка инкремента + Инкрементальный поиск дубликатов и слияние* и его расписание (например, выполнять раз в час)

В таком случае при запуске триггера сначала запустится первая задача из цепочки, дождется ее завершения и сразу запустит другую, дождется ее завершения и сразу запустит следующую...

4.10.2 Список триггеров

На данной странице можно увидеть:

- *Действия* - действия, которые можно выполнить с триггером.
- *Название* - название триггера
- *Дата следующего запуска* - если пустая, то для триггера не настроено расписание

- *Сноп-выражение* - сноп-выражение, по которому настроен автоматический запуск триггера

4.10.3 Работа с триггерами

Ниже перечислены возможные операции по работе с триггерами.

Все изменения действуют только в рамках сессии и после рестарта сервера настройки будут возвращены к исходному состоянию.

4.10.3.1 Запуск триггеров

Триггеры стартуют автоматически, время запуска настраивают сотрудники ХФЛабс.

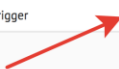
Но цепочки задач можно запускать и вручную. Для этого достаточно нажать *Выполнить* около нужного триггера.

Автоматический запуск триггера можно отключать и включать обратно. Допустим, что нас не устраивает дата следующего запуска задачи *Оптимизация поискового индекса*.

Список триггеров

Действия	Название	Дата следующего запуска	Сноп выражение
Редактировать Удалить Выполнить	afterDatabaseBackupTrigger		
Редактировать Удалить Выполнить Выключить	deleteOutdatedConflictsTrigger	24.03.2018 21:00:00	0 0 21 ? * SAT
Редактировать Удалить Выполнить Включить	detectPopularAttributes		
Редактировать Удалить Выполнить	rebuildAllIndexesTrigger		
Редактировать Удалить Выполнить Выключить	refreshDataStatisticTrigger	23.03.2018 2:00:00	0 0 2 * * ?
Редактировать Удалить Выполнить Выключить	searchEngineOptimizeTrigger	24.03.2018 22:00:00	0 0 22 ? * SAT

Создать триггер

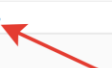


Нажимаем на кнопку *Выключить*, отключая таким образом автоматический запуск триггера.

Список триггеров

Действия	Название	Дата следующего запуска	Сноп выражение
Редактировать Удалить Выполнить	afterDatabaseBackupTrigger		
Редактировать Удалить Выполнить Выключить	deleteOutdatedConflictsTrigger	24.03.2018 21:00:00	0 0 21 ? * SAT
Редактировать Удалить Выполнить Включить	detectPopularAttributes		
Редактировать Удалить Выполнить	rebuildAllIndexesTrigger		
Редактировать Удалить Выполнить Выключить	refreshDataStatisticTrigger	23.03.2018 2:00:00	0 0 2 * * ?
Редактировать Удалить Выполнить Выключить	searchEngineOptimizeTrigger	24.03.2018 22:00:00	0 0 22 ? * SAT

Создать триггер



Смотрим на колонку *Дата следующего запуска* - она пуста. А кнопка *Выключить* изменилась на *Включить*.

Это означает, что данный триггер можно выполнить только вручную, автоматически он работать не будет.

Список триггеров

Действия	Название	Дата следующего запуска	Cron выражение
Редактировать Удалить Выполнить	afterDatabaseBackupTrigger		
Редактировать Удалить Выполнить Выключить	deleteOutdatedConflictsTrigger	24.03.2018 21:00:00	0 0 21 ? * SAT
Редактировать Удалить Выполнить Включить	detectPopularAttributes		
Редактировать Удалить Выполнить	rebuildAllIndexesTrigger		
Редактировать Удалить Выполнить Выключить	refreshDataStatisticTrigger	23.03.2018 2:00:00	0 0 2 * * ?
Редактировать Удалить Выполнить Включить	searchEngineOptimizeTrigger		0 0 22 ? * SAT

Создать триггер

Если кнопки включения/выключения нет вообще - автоматический запуск для данного триггера не настроен.

4.10.3.2 Создание триггеров

Все изменения действуют только в рамках сессии и после рестарта сервера настройки будут возвращены к исходному состоянию, а созданные вручную триггеры — удалены

Для создания нового триггера нажимаем кнопку *Создать триггер*.

Список триггеров

Действия	Название	Дата следующего запуска	Cron выражение
Редактировать Удалить Выполнить	afterDatabaseBackupTrigger		
Редактировать Удалить Выполнить Выключить	deleteOutdatedConflictsTrigger	24.03.2018 21:00:00	0 0 21 ? * SAT
Редактировать Удалить Выполнить Включить	detectPopularAttributes		
Редактировать Удалить Выполнить	rebuildAllIndexesTrigger		
Редактировать Удалить Выполнить Выключить	refreshDataStatisticTrigger	23.03.2018 2:00:00	0 0 2 * * ?
Редактировать Удалить Выполнить Включить	searchEngineOptimizeTrigger		0 0 22 ? * SAT

Создать триггер

В открывшейся форме заполняем параметры:

- *Название* - уникальный код триггера
- *Cron выражение* - cron-выражение, по которому будет настроен автоматический запуск триггера (секунда, минута, час, день, месяц, день недели указанные через пробел)
- *Продолжать при ошибке* - флаг для продолжения выполнения следующей задачи при наличии ошибок
- *Список задач к выполнению* - из предлагающегося списка выбрать нужные задачи для выполнения

Срон триггер

Название:

Срон выражение:

Продолжать при ошибках
 Продолжать при конфликтах

Выберите список задач к выполнению:

- Проверка по черным спискам
 - actualizeBlacklistCheck
 - fullBlacklistCheck
 - rebuildBlacklistCheckHashes
- Дедупликация и слияние
 - findDuplicates
 - findDuplicatesAndMerge
 - findDuplicatesAndMergeAndRebuildGroups
 - importExternalDuplicateWithMerge
 - incrementDuplicates

Уникальный код триггера (только латинские символы и цифры)
 секунда, минута, час, день, месяц, день недели
 Продолжать выполнение следующей задачи при наличии ошибок
 Продолжать выполнение следующей задачи при наличии конфликтов выполнения

После задания параметров, нажимаем кнопку *Сохранить*.

Срон триггер

Название:
 Срон выражение:
 Продолжать при ошибках
 Продолжать при конфликтах

Выберите список задач к выполнению:

- rebuildBlacklistCheckHashes
- Дедупликация и слияние
 - findDuplicates
 - findDuplicatesAndMerge
 - findDuplicatesAndMergeAndRebuildGroups
 - importExternalDuplicateWithMerge
 - incrementDuplicates
 - incrementDuplicatesAndMerge
- Диагностика
 - diagnosticLogs
 - diagnosticMatrix

Уникальный код триггера (только латинские символы и цифры)
 секунда, минута, час, день, месяц, день недели
 Продолжать выполнение следующей задачи при наличии ошибок
 Продолжать выполнение следующей задачи при наличии конфликтов выполнения

Новый триггер появится в списке.

Список триггеров

Действия	Название	Дата следующего запуска	Срон выражение
Редактировать Удалить Выполнить	afterDatabaseBackupTrigger		
Редактировать Удалить Выполнить Выключить	deleteOutdatedConflictsTrigger	24.03.2018 21:00:00	0 0 21 ? * SAT
Редактировать Удалить Выполнить Включить	detectPopularAttributes		
Редактировать Удалить Выполнить Выключить	findDuplicatesTrigger	23.03.2018 10:00:00	0 * 10 10-18 ? **
Редактировать Удалить Выполнить	rebuildAllIndexesTrigger		
Редактировать Удалить Выполнить Выключить	refreshDataStatisticTrigger	23.03.2018 2:00:00	0 0 2 * * ?
Редактировать Удалить Выполнить Выключить	searchEngineOptimizeTrigger	24.03.2018 22:00:00	0 0 22 ? * SAT

[Создать триггер](#)

4.10.3.3 Редактирование триггеров

Для изменения параметров нажимаем кнопку *Редактировать* около нужного триггера.

Можно менять любые поля, кроме названия. Если необходимо изменить название, нужно удалить старый триггер и создать новый

4.10.3.4 Удаление триггеров

Для удаления достаточно нажать кнопку *Удалить* около нужного триггера.

4.10.4 Список триггеров

По умолчанию в системе подключены триггеры:

Название	Список задач	Описание	Расписание работы
afterDatabaseBackupTrigger	<ol style="list-style-type: none"> 1. Полное перестроение графа связей (раздел <i>Системные</i>). 2. Полное перестроение поискового индекса (раздел <i>Системные</i>). 3. Перестроение хэшей полного поиска дубликатов (раздел <i>Дедупликация и слияние</i>). 4. Перестроение хэшей полной проверки ЧС (раздел <i>Проверка по черным спискам</i>). 5. Полное перестроение групп дубликатов для сравнения вручную (раздел <i>Системные</i>). 6. Полное перестроение индекса конфликтов обновления (раздел <i>Системные</i>). 7. Перестроение хэшей полной проверки по ЕГРЮЛ (раздел <i>Дедупликация и слияние</i>). 	Обновление информации в поисковых индексах после поднятия БД из бэкапа. Запускается вручную.	

Название	Список задач	Описание	Расписание работы
deleteOutdatedConflictsTrigger	Удаление неактуальных конфликтов	Удаление обработанных конфликтов старше, чем 30 дней назад	Каждую субботу в 21:00
detectPopularAttributesTrigger	<ol style="list-style-type: none"> 1. Найти массовые атрибуты (раздел <i>Массовые атрибуты</i>). 2. Проставить популярность по найденным значениям (ФЛ) (раздел <i>Массовые атрибуты</i>). 3. Проставить популярность по найденным значениям (другие типы контрагентов) 	Выявление и проставление популярности массовым атрибутам.	Каждое первое воскресенье месяца в 12:00
diagnosticLogsForTodayTrigger	Отправить логи за сегодня на указанную в настройках почту.		Каждый день в 23:55
diagnosticMetricsForPreviousMonthTrigger	Отправить веб-метрики за предыдущий месяц на указанную в настройках почту.		Каждое первое число месяца в 02:40
rebuildAllIndexesTrigger	<ol style="list-style-type: none"> 1. Полное перестроение графа связей (раздел <i>Системные</i>). 2. Полное перестроение поискового индекса (раздел <i>Системные</i>). 3. Полное перестроение 	Перестроение поисковых индексов. Запускается вручную.	

Название	Список задач	Описание	Расписание работы
	групп дубликатов для сравнения вручную (раздел <i>Системные</i>).		
searchEngineOptimizeTrigger	Оптимизация поисковых индексов	Оптимизирует поисковый индекс. Запускается автоматически и раз в неделю.	Каждую субботу в 22:00
refreshDataStatisticTrigger	Сбор метрик по задаче refreshDataStatistic		Каждый день в 02:00

4.11 Возобновляемые задачи

Задачи, помеченные признаком *Возобновляемая*, могут быть остановлены в процессе выполнения.

При повторном запуске ранее обработанные данные будут пропущены, и задача продолжит работу с места, где была остановлена (при условии, что в таблицу TASK_PARTY_HIDS не было внесено изменений).

4.11.1 Запуск на ночь

Задаче можно настроить параметр executionTimeout – таймаут в минутах, по истечению которого выполнение задачи будет прервано.

Если нужно, чтобы задача выполнялась по ночам, с 00:00 до 05:00, нужно:

1. Нажать «*Редактировать*» около описания задачи.
2. Установить параметр executionTimeout = 300 (5ч * 60 = 300 минут)
3. Создать [триггер](#) с cron-выражением

```
0 0 0 ? * *
```

Таким образом, задача будет работать по ночам, не мешая дневным бизнес-процессам.

При вычислении таймаута нужно учесть, что задача остановится не сразу, по истечении времени она дообработывает пачку данных, которую взяла последней, а потом только остановится. Если время остановки критично, лучше брать запас в полчаса.

4.12 Непрерываемые задачи

Задачи, помеченные признаком *Непрерываемые*, могут быть остановлены в процессе выполнения только [принудительно](#) (жесткая остановка).

Результат выполнения задачи может оказываться в непригодном для использования состоянии, поэтому нужен обязательный перезапуск задачи.

4.13 Проверка по черным спискам

Группа задач, предназначенных для проверки данных по черным спискам:

- `fullBlacklistCheck` — полная проверка контрагентов по черным спискам;
- `actualizeBlacklistCheck` — проверки по черным спискам контрагентов, требующих актуализации. Задача проверяет только новых и измененных клиентов;
- `rebuildBlacklistCheckHashes` — перестроение хэшей для проверки ЧС (принудительно перестраиваются онлайн хэши).

4.14 Необратимое удаление данных

4.14.1 Общее описание

Задача `PartyDelete` полностью удаляет данные из ЕК.

Нужна для исполнения федерального закона по удалению персональных данных после окончания срока их необходимого хранения.

Данные удаляются полностью, восстановить их будет нельзя!

4.14.2 Особенности задачи

1. Задача работает только по закрытым оригинальным контрагентам — открытые игнорируются.
2. Задача удаляет всех закрытых слитых контрагентов, которые относятся к удаляемым оригинальным.
3. Если закрываемый клиент был разлит из открытого результирующего, то в версиях результирующего все равно останется информация о реквизитах/атрибутах удаляемого КА

4.14.3 Запуск задачи

1. Выяснить идентификаторы и типы контрагентов (`hid_party`), которые необходимо удалить.
2. Добавить список данных идентификаторов в таблицу `delete_party_hids` вручную.
3. Запустить задачу.

Пример ручного наполнения таблицы:


```
insert into delete_party_hids(hid_party, party_type) values (1004, 'PHYSICAL');
commit;
```

Действия	Название	Описание
Детали Редактировать Выполнить	PartyClose	Закрытие контрагентов
Детали Редактировать Выполнить	PartyDelete	Необратимое удаление данных

4.14.4 Обработка ошибок

Задача падает с ошибкой только в том случае, когда в разных потоках пришло удаление на две исходные записи одной «золотой» карточки.

Задача удаляет контрагентов пачками по 1000 записей. Если происходит ошибка при удалении (нет прав, недоступна БД, невозможно удалить запись из-за нарушения ограничений БД и т.п.) хотя бы одной записи из пачки, то откатывается вся пачка, но задача продолжает работать.

Не считаются ошибками и игнорируются следующие записи в таблице:

- HID-ы открытых клиентов. При обнаружении HID'ы записываются в лог, удаляются из delete_party_hids.
- HID-ы «золотых» клиентов. При обнаружении HID'ы записываются в лог, удаляются из delete_party_hids.
- Ненайденные идентификаторы. При обнаружении удаляются из delete_party_hids, в лог не записываются.
- Строки с несуществующим типом контрагента (например, 'PH' вместо 'PHYSICAL'). Не удаляются из delete_party_hids, в лог не записываются.

4.15 Закрытие контрагентов

PartyClose

Возобновляемая, конфликтная

4.15.1 Общее описание

Задача реализует закрытие контрагентов по списку.

4.15.2 Особенности задачи

Задача работает только по оригинальным контрагентам — результирующие игнорируются.

У контрагента обязательно должен быть непустой HID_STAGING, иначе запись будет пропущена.

4.15.3 Запуск задачи

1. Выяснить идентификаторы и типы контрагентов (hid_party), которые необходимо удалить.
2. Добавить список данных идентификаторов в таблицу close_party_hids вручную.
3. Запустить задачу.

Пример ручного наполнения таблицы:

```
insert into close_party_hids(hid_party, party_type) values (1004, 'PHYSICAL');
commit;
```

4.15.4 Обработка ошибок

Если происходит ошибка при закрытии (нет прав, недоступна БД, невозможно закрыть запись из-за нарушения ограничений БД и т.п.) хотя бы одной записи из пачки, то откатывается вся пачка, но задача продолжает работать.

Не считаются ошибками и игнорируются следующие записи в таблице:

- HID-ы закрытых, несуществующих, разлитых контрагентов — удаляются из close_party_hids, в лог не записываются.
- HID-ы «золотых» контрагентов — удаляются из close_party_hids, в лог записываются как skipped.
- Строки с несуществующим типом контрагента (например, 'PH' вместо 'PHYSICAL') — не удаляются из close_party_hids, в лог не записываются.

4.16 Порционные задачи

Иногда исходные системы выгружают в Единый клиент миллионы записей за раз. Например, когда добавляется новое поле в модель данных клиента. Тогда Единый клиент несколько дней обрабатывает инкремент, а затем еще несколько дней отдает данные системам-потребителям. Получается, что информация сильно запаздывает.

Чтобы сгладить эффект от «массовых» инкрементов, Единый клиент научился ограничивать количество записей в отдельном инкременте и общее количество записей, загружаемое за сутки. Он загружает данные маленькими порциями и быстрее отдает их потребителям.

4.16.1 Конфигурация задачи

При подключении порционной загрузки в задачах импорта появляются два новых свойства:

- maxChunkIncrement - максимальное количество записей за один импорт
- maxDailyIncrement - максимальное количество записей за день

Параметры применяются сразу, без рестарта сервера

Создать/Редактировать задачу

Группа import
 Название BufferDDWH

Параметры задачи

Название параметра	Значение	Обязательность
essenceType	PHYSICAL	
sourceSystem	bufferConnector	
oldMaxIdPropertyName	oldMaxId	
bufferDataReadBlockSize	1000	
maxChunkIncrement	10000	
maxDailyIncrement	200000	

Сохранить Отмена

4.16.2 Работа задачи

Во время своей работы задача создает конфигурационный параметр "importedToday" (АРМ Администратора – Конфигурация – Параметры импорта из буфера, рядом с "oldMaxId").

В нем записывается импортированное "сегодня" количество. Значение свойства имеет формат "dd.mm.yyy:N" где "N" - общее количество импортированных за день данных, а "dd.mm.yyy" - дата этого количества.

Параметры

Параметры импорта из буферной области

Название параметра	Значение
importedToday	10.02.2016:3
oldMaxId	20915

Сохранить

4.16.3 Подключение порционности

Переход с обычной на порционную загрузку инкремента – платная услуга. Если хотите включить ее, напишите нам.

4.17 Проверка по ЕГРЮЛ

Группа задач, предназначенных для проверки по ЕГРЮЛ данных из Единого клиента:

- egrCheck – полная проверка по ЕГРЮЛ;
- egrCheckActualize – проверка по ЕГРЮЛ контрагентов, требующих актуализации;
- rebuildEgrCheckHashes – перестроение хэшей полной проверки по ЕГРЮЛ.

4.17.1 Полная проверка по ЕГРЮЛ

egrCheck

Задача проверяет все компании ЮЛ по ЕГРЮЛ по [правилам идентификации дубликатов](#).

Не считаются дублями:

- Разные филиалы одной и той же компании.
- Филиалы и головные организации.

4.17.2 Проверка по ЕГРЮЛ контрагентов, требующих актуализации

egrCheckActualize

Задача проверяет по ЕГРЮЛ только компаний, требующие актуализации:

- созданы после [полной проверки](#);
- обновлены;
- отредактированы.

Находит дубли между компаниями и карточками ЕГРЮЛ, но не внутри компаний и не внутри ЕГРЮЛ.

Использует [типовые правила идентификации дубликатов](#). При этом филиалы и головные организации не считаются дублями.

Конфигурировать задачу не нужно.

4.18 Массовые атрибуты

4.18.1 Массовые атрибуты

Группа задач, предназначенных для выявления и проставления популярности массовым атрибутам:

- findPopularValues – [найти массовые атрибуты](#);
- markPopularPhysicalAttributes – [проставить популярность](#) по найденным значениям (ФЛ);
- markPopularLegalAttributes – проставить популярность по найденным значениям (ЮЛ);
- truncatePopularValues – сбросить накопленную информацию о массовых атрибутах на факторе (сброс индекса).

4.18.2 Поиск массовых атрибутов

findPopularValues

[Непрерываемая](#)

4.18.2.1 Описание задачи

ЕК проходит весь поисковый индекс и вычисляет часто встречающиеся атрибуты. Все найденные хиды контрагентов записываются в таблицу `task_doubtful_reclean_hids` с указанием типа атрибута на переобработку.

4.18.3 Проставление популярности по найденным значениям (ФЛ и ЮЛ)

```
markPopularPhysicalAttributes
```

```
markPopularLegalAttributes
```

[Возобновляемая](#)

4.18.3.1 Описание задачи

ЕК забирает контрагентов из таблицы `task_doubtful_reclean_hids` и отправляет в Фактор для проставления [кода популярности](#).

4.19 Миграция

4.19.1 Автопрогон скриптов выдачи грантов пользователям

У ряда заказчиков на отдельных платформах надо выдавать определенным пользователям гранты в БД.

Для этого в «Едином клиенте» можно создать специальный скрипт с грантами, который будет автоматически прогоняться:

1. после применения скрипта миграции;
2. только на избранных платформах, потому что на остальных этих пользователей может не быть.

При отсутствии пользователей, которым выдают гранты, скрипт не упадет с ошибкой, и миграция пройдет успешно.

Для создания скриптов используются стандартные возможности WildFly – описаны [здесь](#), в разделе «Дополнительные возможности».

Скрипты можно добавлять как для Oracle, так и для MariaDB.

4.19.2 Инициализация прошлых версий атрибутов

```
pastValuesInit
```

Возобновляемая

4.19.2.1 Описание задачи

Генерации прошлых значений для ФИО и документов на основе текущей базы.

4.19.2.2 Запуск задачи

Задача работает по списку клиентов, что делает ее возобновляемой. Но для запуска нужно выполнить предварительные шаги:

1. Выполнить задачу cleanTaskParameterTables для очистки таблицы task_party_hids;
2. Выполнить задачу fillTaskPartyHidsActual, чтобы наполнить таблицу task_party_hids идентификаторами актуальных клиентов;

После этого можно выполнять задачу pastValuesInit.

4.19.2.3 Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента , для которых выполняется генерация данных	PHYSICAL
poolSize	Максимальное количество потоков	2
pageSize	Размер пачки	1000

4.19.3 Миграция нового поля

fooMigrateRequisiteTask

4.19.3.1 Описание задачи

Данную задачу следует использовать в случае, когда добавляется новый реквизит и необходимо выполнить первоначальную загрузку данных по историческому объему клиентов, уже хранящихся в системе

4.19.3.2 Особенности задачи

Перед запуском задачи необходимо:

1. Создать буферную таблицу, содержащую в себе значение мигрируемого поля, а также поля, идентифицирующие клиента (*Выполняется на стороне Заказчика*)

Поле	Тип	Описание
pt_external_id	VARCHAR2(60 byte)	Идентификатор клиента в исходной системе
pt_source_system_cd	CHAR(2 byte)	Код исходной системы
pt_migrate_field	VARCHAR2(50 byte)	Мигрируемое поле

2. Добавить в ЕК задачу миграции (*Выполняется специалистами ХФЛабс*)

4.19.3.3 Конфигурация задачи

Если первоначальная загрузка выполняется без очистки данных, в задаче конфигурируется только параметр partyType – тип контрагента,

Создать/Редактировать задачу

Группа reprocess
Название innMigrateRequisiteTask

Параметры задачи

Название параметра	Значение	Обязательность
essenceType	PHYSICAL	

Сохранить Отмена

Если первоначальная загрузка выполняется с очисткой данных, в задаче конфигурируются параметры:

- partyType – тип контрагента;
- mappingName – название меппинга, по которому будет проводиться стандартизация;
- massiveChanging – оповещение Фактора изменениями, по умолчанию включено (значение false).

Создать/Редактировать задачу

Группа reprocess
Название kioMigrateRequisiteTask

Параметры задачи

Название параметра	Значение	Обязательность
essenceType	LEGAL	
mappingName	demo-legal-soep	

Сохранить Отмена

4.20 Миграция структуры хранения исторических данных

4.20.1 Форсирование миграции структуры исторических данных (Фаза 1)

migratePhase1

[Возобновляемая](#)

4.20.1.1 Описание задачи

Принудительное заполнение _ACT таблиц актуальными данными.

Входит в миграцию на исторические версии таблиц.

4.20.1.2 Запуск задачи

Задача возобновляемая, никаких предварительных шагов делать не надо. Просто запустить задачу.

При необходимости настроить триггер запуска по ночам и указать параметр [executionTimeout](#).

4.20.1.3 Алгоритм работы

Задача проверяет таблицу migration_phase, какая фаза у таблицы:

- PHASE_1 → значит, мигрируем
- PHASE_2 → эта таблица отмигрирована, не трогаем

Если нужна миграция, то задача копирует из версионной таблицы все нулевые версии. Только копирует, из исходника ничего не удаляет! Посмотрим на примере таблицы

STAGING:

4.20.1.3.1 Было до миграции

STAGING

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING_ACT

ID_STAGING	HID_STAGING	VERSION

4.20.1.3.2 Стало после

STAGING (ничего не удаляли, вообще не менялась)

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING_ACT (скопировали сюда актуальную версию)

ID_STAGING	HID_STAGING	VERSION
1026	1024	0

Если таблица уже отмигрирована, больше ее не трогаем. Так как система и так при любом изменении контрагента заполняет _ACT таблицу. Эта задача нужна лишь для того, чтобы ускорить процесс.

Последить за прогрессом можно с помощью запроса:

```
select * from migration_phase;
```

Все таблицы, которые еще в PHASE_1, ожидают миграции

4.20.1.4 Конфигурация задачи

Конфигурировать задачу не надо, она работает «из коробки».

Но есть параметры, которые можно добавить по необходимости (если сказал сотрудник ХФЛабс)

Параметр	Описание	Значение по умолчанию
tableNameToMigrate	<ul style="list-style-type: none"> Если задан – будет мигрировать только указанную таблицу Если не задан – все, что ещё не отмигрировано Список таблиц смотреть селектом: <pre>select * from migration_phase;</pre>	
logPeriod	Параметр, определяющий частоту логирования прогресса. Будет логироваться каждая N пачка	100
pageSize	Размер пачки	1000
poolSize	Количество одновременных потоков	16

4.20.2 Форсирование миграции структуры исторических данных (Фаза 3)

migratePhase3
Возобновляемая

4.20.2.1 Описание задачи

Задача удаляет нулевые версии данных в _HIST таблицах.

На первых этапах миграции мы просто копировали нулевые версии в отдельную таблицу. На этом этапе мы удаляем дублирование данных, возвращая размеры БД в исходное состояние.

4.20.2.2 Запуск задачи

Задача возобновляемая, никаких предварительных шагов делать не надо. Просто запустить задачу.

При необходимости настроить триггер запуска по ночам и указать параметр [executionTimeout](#).

4.20.2.3 Алгоритм работы

Задача проверяет таблицу migration_phase, какая фаза у таблицы:

- PHASE_3 → значит, мигрируем
- PHASE_4 → эта таблица отмигрирована, не трогаем

Если нужна миграция, то задача удаляет из версионной таблицы все нулевые версии. Посмотрим на примере таблицы **STAGING**:

4.20.2.3.1 Было до миграции STAGING_HIST

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

4.20.2.3.2 Стало после
STAGING_HIST (удалили нулевую версию)

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2

Если таблица уже отмигрирована, больше ее не трогаем.

4.20.2.4 Конфигурация задачи

Конфигурировать задачу не надо, она работает «из коробки».

Но есть параметры, которые можно добавить по необходимости (если сказал сотрудник ХФЛабс):

Параметр	Описание	Значение по умолчанию
tableNameToMigrate	<ul style="list-style-type: none"> Если задан – будет мигрировать только указанную таблицу Если не задан – все, что ещё не отмигрировано <p>Список таблиц смотреть селектом: select * from migration_phase;</p>	
logPeriod	Параметр, определяющий частоту логирования прогресса. Будет логироваться каждая N пачка	100
pageSize	Размер пачки	1000
poolSize	Количество одновременных потоков	16

4.21 Сбор статистики

refreshDataStatistic

4.21.1 Описание задачи

«Единый клиент» умеет собирать в отдельной таблице статистику по качеству данных. Эти показатели можно выгрузить, чтобы построить график или составить отчет. С этой функцией проще следить за изменением качества данных в исходных системах и в целом по базе.

Задача refreshDataStatistic собирает и сохраняет метрики из таблицы [DATA_METRIC](#), которые формируются, как запросы к расширенному поиску. Накапливает посчитанные значения в таблице [DATA_STATISTIC](#) для дальнейшего анализа.

4.21.2 Использование

Для работы задачи необходимо заполнить таблицу DATA_METRIC. Например, добавим следующую строку:

ID_METRIC	DESCRIPTION	PARTY_TYPE	QUERY
1	Хороший паспорт	PHYSICAL	? Passport.qualityCode = EDITED and Passport.qualifierType = PASSPORT_RU

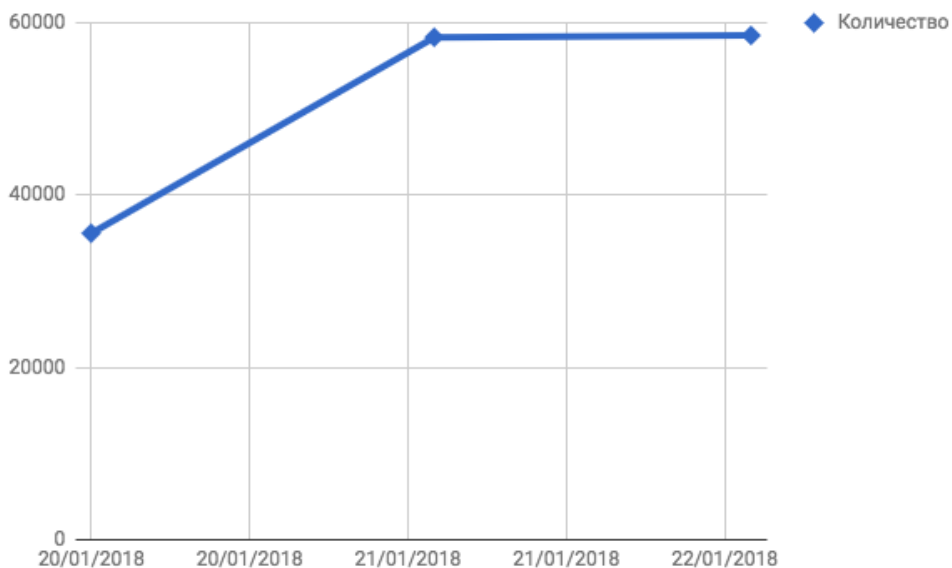
Это означает, что задача будет брать запрос из QUERY колонки и использовать его для поиска по индексу PHYSICAL контрагентов (аналогично расширенному поиску). А получившееся количество сохранять в таблицу DATA_STATISTIC. После успешного выполнения задачи в таблице DATA_STATISTIC появится строка:

ID_METRIC	PARAM	COUNT_TS	COUNT
1		2018-01-20 02:00:00	35583

Далее можно настроить триггер refreshDataStatisticTrigger на выполнение этой задачи по необходимому расписанию для накопления статистики. При ежедневном расчете через три дня таблица DATA_STATISTIC будет выглядеть следующим образом:

ID_METRIC	PARAM	COUNT_TS	COUNT
1		2018-01-20 02:00:00	35583
1		2018-01-21 02:00:00	58332
1		2018-01-22 02:00:00	58543

Как видно, 20 января произошел резкий прирост контрагентов с хорошим паспортами. Возможно, это связано с добавлением новой системы или обновлением алгоритмов обработки. А график по этим данным будет выглядеть, например, так:



4.21.3 Примеры запросов

Запросы работают аналогично [расширенному поиску](#) и могут обращаться к разным поисковым индексам. Дополнительно реализована поддержка итераторов.

4.21.3.1 Типы контрагентов

Название индекса для контрагентов совпадает с типом. Примеры:

```
-- Посчитать количество физических лиц с хорошим или подтвержденным ФИО:
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)
VALUES (1, 'Good FIO', 'PHYSICAL', '? PhysicalParty.fullNameQC = (EDITED or
CONFIRMED_MANUALLY));

-- Посчитать количество юридических лиц с хорошим ИНН
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)
VALUES (2, 'Good INN', 'LEGAL', '? LegalParty.innQC = GOOD');
```

В результате выполнения задачи таблица DATA_STATISTIC будет наполняться данными:

ID_METRIC	PARAM	COUNT_TS	COUNT
1		2018-01-20 02:00:00	35
2		2018-01-20 02:00:00	584
1		2018-01-21 02:00:00	31
2		2018-01-21 02:00:00	675

Аналогично можно составлять запросы по договорам, ЕГРЮЛ и другим типам контрагентов.

4.21.3.2 Группы дубликатов

Для сбора статистики по группам дубликатов необходимо использовать тип индекса MANUAL_DUPLICATE_GROUP. Например:

```
-- Посчитать количество не полностью обработанных групп дублей с количеством карточек
больше 2
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)
VALUES (3, 'Duplicates group >=3 and < 100', 'MANUAL_DUPLICATE_GROUP', '?
ManualDuplicateGroup.partyType = PHYSICAL and ManualDuplicateGroup.partyCount >= 3 and
ManualDuplicateGroup.progress < 100');
```

Важно указать в запросе необходимый тип контрагентов для поиска. В примере выше — это физлица, поэтому ManualDuplicateGroup.partyType = PHYSICAL.

Чтобы посчитать количество контрагентов, находящихся в «серой зоне» (присутствующих в группах дубликатов), одним запросом не обойтись, т.к. поисковый индекс не позволяет производить агрегирующих операций. Если считать, что доля контрагентов в «серой зоне», находящихся в группах дубликатов с более 10 контрагентов, составляет не более 0,5%, можно получить приблизительное количество. Составить запросы вида:

```
? ManualDuplicateGroup.partyType = PHYSICAL and ManualDuplicateGroup.partyCount = N,
где N от 2 до 10, и просуммировать. Таким образом получить размер серой зоны с
точность 99.5%.
```

4.21.3.3 Конфликты

Аналогично можно обращаться к конфликтам:

```
-- Посчитать общее количество конфликтов
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)
VALUES (4, 'Conflicts ALL', 'CONFLICT', '? *.*');
```

4.21.3.4 Использование плейсхолдера для итератора в запросе

В запросы расширенного поиска в таблицу DATA_METRIC можно добавлять плейсхолдеры для справочных значений. При этом, при запуске задачи для каждого значения, которое может принимать справочное поле, будет создан отдельный подзапрос и добавлен в DATA_STATISTIC. Значение справочника, для которого создан подзапрос, пишется в поле Param.

Синтаксис запроса:

```
? PhysicalParty.gender = (%iter%); --Вместо задания значения задаем ему значение %iter%
```

Примеры:

```
-- Посчитать количество физических лиц каждого поля
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)
VALUES (1, 'GENDER DISTRIBUTION', 'PHYSICAL', '? PhysicalParty.gender = (%iter%)');
```

В результате выполнения задачи таблица DATA_STATISTIC будет наполняться данными:

ID_METRIC	PARAM	COUNT_TS	COUNT
1	MALE	2018-01-21 02:00:00	492
1	FEMALE	2018-01-21 02:00:00	584
1	UNKNOWN	2018-01-21 02:00:00	31

Итерация возможна по всем значениям, для которых имеются справочники

Особенности использования:

- Поддерживается только один итератор. При наличии более 1го запись в DATA_STATISTIC не добавляется, в лог пишется предупреждение.
- Если в качестве итератора выбрано поле, которое не является справочным, например фамилия, то сбор статистики не происходит, в лог пишется предупреждение.

4.21.4 Конфигурация задачи

Для стандартного запуска задачу конфигурировать не нужно.

При специальном запросе поддержки, можно настраивать следующие параметры:

Параметр	Описание	Значение по умолчанию
poolSize	Максимальное количество потоков	1

4.21.5 Особенности задачи

Задача не конфликтующая.

4.21.6 DATA_METRIC

4.21.6.1 Столбцы

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ID_METRIC	NUMBER(*)	No	null	1	Идентификатор записи
DESCRIPTION	VARCHAR2(300)	No	null	2	Описание метрики
PARTY_TYPE	VARCHAR2(50)	No	null	3	Поисковый индекс для осуществления запроса
QUERY	VARCHAR2(1500)	No	null	4	Поисковый запрос

4.21.6.2 Индексы

INDEX_NAME	UNIQUENESS	COLUMNS
PK_DATA_METRIC	UNIQUE	ID_METRIC

4.21.7 DATA_STATISTIC

4.21.7.1 Столбцы

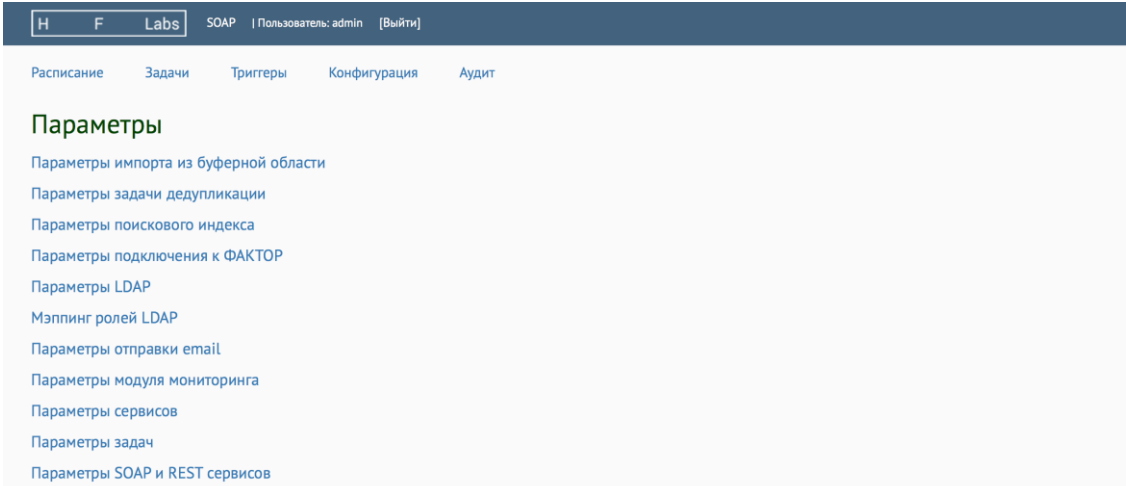
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ID_METRIC	NUMBER(*)	No	null	1	Идентификатор метрики
COUNT_TS	TIMESTAMP(6)	No	null	2	Дата и время расчета метрики
PARAM	VARCHAR(300)	Yes	null	3	Параметр метрики для метрик с итератором
COUNT	NUMBER(*)	No	null	4	Значение метрики

4.21.7.2 Индексы

INDEX_NAME	UNIQUENESS	COLUMNS
UK_DATA_STATISTIC	-	ID_METRIC, COUNT_TS

5 Настройка системы

Страница *Конфигурация* необходима для расширенной настройки системы. Здесь можно просмотреть параметры системы и, при необходимости, отредактировать их.



The screenshot shows the 'Конфигурация' (Configuration) page in the system administrator interface. At the top, there is a navigation bar with 'H F Labs' and 'SOAP | Пользователь: admin [Выйти]'. Below the navigation bar, there are tabs for 'Расписание', 'Задачи', 'Триггеры', 'Конфигурация', and 'Аудит'. The 'Конфигурация' tab is active, displaying a section titled 'Параметры' (Parameters). Under this section, there is a list of parameter categories: 'Параметры импорта из буферной области', 'Параметры задачи дедупликации', 'Параметры поискового индекса', 'Параметры подключения к ФАКТОР', 'Параметры LDAP', 'Мэппинг ролей LDAP', 'Параметры отправки email', 'Параметры модуля мониторинга', 'Параметры сервисов', 'Параметры задач', and 'Параметры SOAP и REST сервисов'.

Если вы изменяете какие-либо настройки, то, чтобы они вступили в силу, необходимо [остановить и затем снова запустить](#) систему.

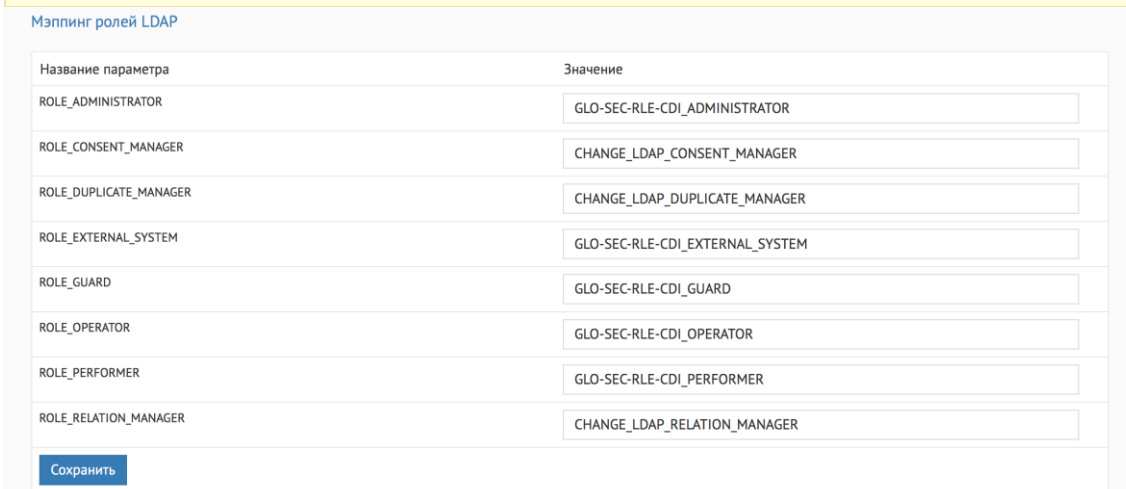
Описание стандартных настраиваемых параметров:

Вне интерфейса администратора настраиваются:

- [Параметры запуска JBOSS](#)
- [Настройки задач во внешней директории](#)

5.1 Мэппинг ролей LDAP

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!



The screenshot shows the 'Мэппинг ролей LDAP' (LDAP Role Mapping) configuration page. It features a table with two columns: 'Название параметра' (Parameter Name) and 'Значение' (Value). The table contains the following entries:

Название параметра	Значение
ROLE_ADMINISTRATOR	GLO-SEC-RLE-CDI_ADMINISTRATOR
ROLE_CONSENT_MANAGER	CHANGE_LDAP_CONSENT_MANAGER
ROLE_DUPLICATE_MANAGER	CHANGE_LDAP_DUPLICATE_MANAGER
ROLE_EXTERNAL_SYSTEM	GLO-SEC-RLE-CDI_EXTERNAL_SYSTEM
ROLE_GUARD	GLO-SEC-RLE-CDI_GUARD
ROLE_OPERATOR	GLO-SEC-RLE-CDI_OPERATOR
ROLE_PERFORMER	GLO-SEC-RLE-CDI_PERFORMER
ROLE_RELATION_MANAGER	CHANGE_LDAP_RELATION_MANAGER

At the bottom left of the table, there is a blue button labeled 'Сохранить' (Save).

Название параметра	Описание параметра
ROLE_ADMINISTRATOR	группа LDAP, соответствующая системной роли администратор
ROLE_GUARD	группа LDAP, соответствующая системной роли офицер информационной безопасности
ROLE_OPERATOR	группа LDAP, соответствующая системной роли оператор (read-only)
ROLE_OPERATOR_CONSENT	группа LDAP, соответствующая системной роли оператор согласий (read-only)
ROLE_PERFORMER	группа LDAP, соответствующая системной роли операционист
ROLE_DUPLICATE_MANAGER	группа LDAP, соответствующая системной роли менеджер дубликатов
ROLE_EXTERNAL_SYSTEM	группа LDAP, соответствующая системной роли Исходная система. Позволяет работать с онлайн-интерфейсами.
ROLE_RELATION_MANAGER	группа LDAP, соответствующая системной роли верификатор связей .

Если значение – `-${ldap.roles...}`, то группа не настроена, такой роли в системе нет.

5.2 Настройки задач во внешней директории

Параметры задач, которые брались из `rom.xml` на этапе сборки, теперь вынесены в `task.properties`:

- Вместо конкретных значений в описании триггеров и задач отображаются плейсхолдеры, например - `-${task(task.import.sftp.sourceDataSeparator)}`
- На этапе выполнения задачи эти плейсхолдеры заменяются значениями из внешнего файла `task.properties`.
- `task.properties` хранится во внешней директории, и именно там надо менять параметры. Изменения будут видны при очередном запуске задания/триггера без необходимости перезагружать `cdi`.
- Файлы типа `properties` хранятся в кодировке ISO-8859-1, поэтому все нелатинские символы должны быть представлены в виде юникода. Например, символ разделителя бомбочка (⚡) хранится как `\u00A4`

5.3 Параметры JMS

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

5.3.1 Общие параметры

Название параметра	Значение по умолчанию	Описание
<code>jms.connectionFactory.name</code>	-	Название <code>connectionFactory</code> , если используется.
<code>jms.events.destination.isTopic</code>	<code>false</code>	Выходная очередь – <code>topic</code> ?
<code>jms.events.destination.name</code>	-	Название выходной очереди.
<code>jms.events.enabled</code>	<code>false</code>	Рассылка сообщений в выходную очередь включена?
<code>jms.jaxws.recovery.interval</code>	60000	Если соединение с шиной пропадает, то раз в <code>jms.jaxws.recovery.interval</code> мс CDI пытается возобновить соединение.
<code>jms.soap.destination.isTopic</code>	<code>false</code>	Входная очередь – <code>topic</code> ? Если <code>false</code> , используется очередь. (Очередь – один получатель, топик – много)
<code>jms.soap.destination.name</code>	-	Название входной очереди.
<code>jms.soap.enabled</code>	<code>false</code>	Получение сообщений из входной очереди включено?
<code>jms.events.attributes.defaultValues</code>	<code>false</code>	Возвращать в обратный поток согласия по умолчанию?
<code>jms.events.changeDetailsEnabled</code>	<code>false</code>	Возвращать ли результат в детализированном виде

5.3.2 Специфичные для очередей TIBCO параметры

Название параметра	Описание	Пример
<code>jms.tibco.url</code>	URL шины.	<code>tcp://host:7222</code>
<code>jms.tibco.user</code>	Логин для авторизации.	
<code>jms.tibco.password</code>	Пароль для авторизации.	

5.3.3 Специфичные для очередей IBM MQ параметры

Название параметра	Описание	Пример
<code>jms.ibmmq.url</code>	URL шины.	<code>mq.domain</code>
<code>jms.ibmmq.port</code>	Порт.	1414
<code>jms.ibmmq.user</code>	Логин для авторизации.	
<code>jms.ibmmq.password</code>	Пароль для авторизации.	
<code>jms.ibmmq.channel</code>	Канал подключения	<code>SYSTEM.ADMIN.SVRCONN</code>
<code>jms.ibmmq.queueManager</code>	Менеджер очередей	<code>CDI.QM</code>

5.4 Параметры LDAP

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Параметры LDAP

Название параметра	Значение
ldap.account	ldap
ldap.connect.timeout	60000
ldap.domain	intranet.hflabs.ru
ldap.groupSearchBase	ou=CDI,ou=Groups,dc=intranet,dc=hflabs,dc=ru
ldap.maxActive	8
ldap.maxIdle	8
ldap.maxWait	-1
ldap.password	DeiCiephua8u
ldap.read.timeout	60000
ldap.searchBase	dc=intranet,dc=hflabs,dc=ru
ldap.searchFilter	(sAMAccountName={0})
ldap.testOnBorrow	true
ldap.testWhileIdle	true
ldap.timeBetweenEvictionRunsMillis	60000
ldap.url	ldap://intranet.hflabs.ru:389
ldap.warnDurationMs	100

Сохранить

Название параметра	Значение по умолчанию	Описание
ldap.url	ldap://hostname:port	Hostname и порт сервера LDAP
ldap.account	-	Логин системной учетной записи
ldap.domain	-	Имя домена локальной сети
ldap.groupSearchBase	-	Ветка, в которой хранятся учетные записи для ролей
ldap.maxActive	50	Максимальное количество активных соединений одного вида (чтение/запись)
ldap.maxIdle	10	Максимальное количество бездействующих соединений одного типа (чтение/запись) в пуле

Название параметра	Значение по умолчанию	Описание
ldap.maxWait	-1	Максимальная длительность (миллисек) ожидания свободного соединения пока не будет выброшено исключение Если равно -1, ждать бесконечно долго, пока не будет свободного соединения
ldap.password	-	Пароль системной учетной записи
ldap.searchBase	-	Ветка, в которой хранятся учетные записи пользователей (поиск ведется в данной ветке и всех дочерних)
ldap.searchFilter	(sAMAccountName={0})	Атрибут, в котором хранится имя пользователя (как он его указывает при входе в систему)
ldap.testOnBorrow	true	Булевый параметр (true/false), определяющий, должен ли объект пройти валидацию сканером (если таковой есть) прежде чем будет взят из пула
ldap.testWhileIdle	true	Булевый параметр (true/false), указывающий, что необходимо проводить валидацию простаивающих потоков пула с интервалом ldap.timeBetweenEvictionRunsMillis
ldap.timeBetweenEvictionRunsMillis	60000	Количество миллисекунд между запусками потока-сканера
ldap.connect.timeout	60000	Таймаут операции соединения с LDAP-сервером в миллисекундах
ldap.read.timeout	60000	Таймаут операции чтения ответа от LDAP-сервера в миллисекундах
ldap.warnDurationMs	100	Пороговое значение длительности запроса к LDAP при превышении которого выводится WARN-запись в журнал cdi-security.log Изменения применяются без перезапуска CDI

Сам процесс использования учеток, чтобы было понятно, что к чему.

(1) ldap.url определяет сервер, к которому будут идти запросы.

(2) `Ldap.account + Ldap.domain + Ldap.password` определяют служебного системного пользователя ЕК на этом сервере. Он может находиться где угодно в дереве AD, не принципиально. Главное, чтобы у него не истекал пароль и чтобы он мог делать поиски по дереву.

Далее, какой-то пользователь пытается залогиниться в `cdi` с использованием доменной своей учётки (3).

ЕК, используя свой аккаунт (2) делает запрос к серверу (1): «а найди-ка мне пользователя с учёткой (3), который находится у тебя в поддереве из моей настройки `Ldap.searchBase`». В этом поддереве должна быть именно учётка пользователя (3), а для служебной учётки ЕК (2) это не требуется. `Ldap.searchFilter` – это в каком именно поле записи AD искать то, что пользователь ввёл в поле `login`.

Если учётка нашлась, делается второй запрос к серверу LDAP: «дай мне список всех групп, которые находятся в `Ldap.groupSearchBase` и в которые входит пользователь, который пытается залогиниться». Далее по именам групп производится мапирование на наши роли. То есть «ролевые» группы должны лежать все вместе где-то в дереве AD, параметр `Ldap.groupSearchBase` должен указывать на это место.

5.5 Параметры SOAP и REST сервисов

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Название параметра	Значение по умолчанию	Описание
<code>webservice.error.logMode</code>	FULL	Если FULL, то в <code>cdi.log</code> записывается текст ошибочного запроса и вызванная им ошибка. Если SHORT и включена трассировка запросов к веб-сервисам, то в <code>cdi.log</code> записывается ошибка и идентификатор полного сообщения в <code>soap-trace.log</code> . Если SHORT и не включена трассировка запросов к веб-сервисам, то в <code>cdi.log</code> записывается только ошибка.
<code>webservice.stats.periodInSeconds</code>	1800	Периодичность вывода в лог статистики по вызову веб-сервисов (в секундах).
<code>webservice.response.attributes.defaultValues</code>	false	Возвращать в ответах согласия по умолчанию?
<code>webservice.concurrent.numberOfRetries</code>	3	Количество попыток повторной обработки контрагента при получении ошибки о параллельной модификации.

Название параметра	Значение по умолчанию	Описание
webservice.concurrent.delayBetweenRetries	1000	Задержка между попытками повторной обработки при получении ошибки о параллельной модификации.
webservice.response.relations.maxCountByParty	2000	Ограничение на количество возвращаемых связей в <u>SOAP-методах</u> (getByHid, getByRawId, getRelations)
webservice.mergeEnabled	true	Разрешено ли слияния и дедубликация в soap методах? Если false – то метод работает как простой save.
webservice.logging.filters	search=WARN:1s,CRITICAL:2s; getByHID=WARN:1s,CRITICAL:2s; getByRawID=WARN:1s,CRITICAL:2s; fuzzySearch=WARN:5s,CRITICAL:10s; saveAndMerge=WARN:2s,CRITICAL:5s; save=WARN:2s,CRITICAL:5s; saveRelations=WARN:1s,CRITICAL:2s; other=WARN:5s,CRITICAL:15s	<p>Настройки перекрывают умолчательное поведение, заданное в webservice.logging.defaultBehaviour.</p> <p>Для каждого вида запросов можно конкретно указать таймаут и тип реакции.</p> <p>Пример: search=WARN:1s,CRITICAL:5s; other=WARN:5s</p> <p>Расшифровывается так:</p> <ul style="list-style-type: none"> • если вызвали "search", то <ul style="list-style-type: none"> ○ если он длился больше 5 секунд – CRITICAL ○ если больше 1 секунды – WARN ○ иначе умолчательная реакция (обычно SKIP) • для остальных методов <ul style="list-style-type: none"> ○ если метод длился больше 5 секунд – WARN ○ иначе умолчательная реакция <p>На данный момент при превышении значения CRITICAL особое поведение не предусмотрено. Однако, если среднее значение</p>

Название параметра	Значение по умолчанию	Описание
		<p>выполнения запроса превышает CRITICAL, в лог выводится ошибка</p> <p>[SoapLongopsAnalyzer#Flusher] ERROR SoapLongopsAnalyzer - Average processing time of 'name' is x.xx sec (critical=x sec)</p> <p>Изменения данных параметров вступает в силу без рестарта ЕК.</p>
webservice.logging.sampling.periodInMs	100	<p>Период сбора статистики при автоматической jvmt-трассировке долгих запросов для выявления причины тормозов. Чем меньше период, тем точнее трассировка, но при малых значениях возможна избыточная нагрузка на приложение – не рекомендуем изменять дефолтное значение.</p> <p>Чтобы выключить jvmt-трассировку долгих запросов, установите отрицательное значение (например, -1).</p>

5.6 Параметры задач

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Параметры задач

Название параметра	Значение
task.deduplication.cacheSize	<input type="text" value="10"/>
task.deduplication.freeSpaceRatio	<input type="text" value="1.25"/>
task.deduplication.pageSize	<input type="text" value="1000"/>
task.deduplication.poolSize	<input type="text" value="-1"/>
task.finishedTaskDescriptorExpirationPeriod	<input type="text" value="12h"/>
task.merge.pageSize	<input type="text" value="500"/>
task.merge.poolSize	<input type="text" value="-1"/>
task.merge.threshold	<input type="text" value="5000"/>
task.multiclean.timeout	<input type="text" value="10"/>
task.reMerge.maxOriginalPartiesCount	<input type="text" value="1000"/>
task.sql.connector.timeout	<input type="text" value="2h"/>
task.sql.dbProcedure.timeout	<input type="text" value="24h"/>
task.transform.poolSize	<input type="text" value="-1"/>
task.triggers.scheduleOnStartup	<input type="text" value="true"/>

Название параметра	Значение по умолчанию	Описание
task.buffer.upload.properties	-	Расположение файла с параметром oldMaxId (параметр теперь хранится в <u>параметрах импорта из буферной области</u> , поэтому можно не заполнять)
task.deduplication.cacheSize	10	Максимальный размер очереди асинхронной загрузки пакетов контрагентов (где размер пакета равен task.deduplication.pageSize) из хранилища
task.deduplication.pageSize	10000	Размер пакета для выборки кандидатов
task.deduplication.poolSize	-1	Количество потоков для выборки кандидатов Если равно -1, устанавливается равным количеству процессоров (ядер)*2
task.deduplication.freeSpaceRatio	1.25	Коэффициент свободного места, которое должно быть при очередном запуске задачи полной дедупликации Если значение <1, то проверка свободного места отключена

Название параметра	Значение по умолчанию	Описание
task.finishedTaskDescriptorExpirationPeriod	12h	Время "протухания" дескрипторов задач после их завершения Формат значений: "1 day", "35 min", "20h" и т.д.
task.merge.pageSize	500	Размер пакета (контрагентов) при слиянии (за один раз объединяется N групп дубликатов, общее количество контрагентов в которых не превышает pageSize)
task.merge.poolSize	8	Количество потоков слияния Если равно -1, устанавливается равным количеству процессоров (ядер)*2
task.merge.threshold	100	Максимальный размер группы дубликатов (количество контрагентов в группе) для слияния (все группы размером больше либо равно максимального будут проигнорированы) Если threshold <= 0, то размер группы не ограничен
task.multiclean.timeout	10	Время ожидания (в минутах) появления свободного сервиса очистки, при превышении которого задача завершается
task.reMerge.maxOriginalPartiesCount	1000	Максимальное число оригинальных контрагентов для страницы переслияния
task.transform.poolSize	16	Количество потоков трансформации. Если равно -1, устанавливается равным количеству процессоров (ядер)*2
task.triggers.scheduleOnStartup	true	Активировать триггеры по расписанию на старте приложения
task.physical.household.relationTypeld	8	Идентификатор типа связи для задачи поиска "Родственник" физ.лиц (при подключении поиска домохозяйств)
task.physical.household.hhRelationTypeld	500	Идентификатор типа связи для задачи поиска "Домохозяйств"

Название параметра	Значение по умолчанию	Описание
		физ.лиц (при подключении поиска домохозяйств)
task.sql.dbProcedure.timeout	24h	Таймаут для выполнения хранимых процедур и sql-файлов, варианты записи значения см в дочерней странице
task.sql.connector.timeout	2h	Таймаут для коннекторов, которые возвращают данные (допустим, не удалось подключиться к БД – коннектор отваливается), варианты записи значения см в дочерней странице

5.6.1 Таймауты при запросах к длительным БД процедурам

У нас есть два механизма, выполняющие потенциально долгие SQL-вызовы:

- коннекторы для загрузки из буферной таблицы
- перформеры, запускающие всякие процедуры

Если такой запрос повиснет – то [бизнес-процессы встают](#).

В обоих случаях ввели таймауты. Дефолтные значения:

- для коннекторов 2 часа
- для sqlPerformer'ов 24 часа

5.6.1.1 Изменение таймаута

Изменять значения можно через [параметры задачи](#)

- task.sql.dbProcedure.timeout - таймаут для выполнения хранимых процедур и sql-файлов
- task.sql.connector.timeout - таймаут для коннекторов, которые возвращают данные где значение может выглядеть, например, так:

- 5 days
- 10 ms.
- 1h
- 60 SEC

Можно писать слитно или через пробел, с точкой и без на конце, капсом или нормально.

5.7 Параметры задачи дедупликации

Для каждого [типа контрагента](#) после запуска задачи *Поиска дубликатов* сохраняется размер файлов с данными дедупликации, который был использован.

Напр. для физических лиц:

Название параметра	Описание
DeDuplicationTask.PHYSICAL.in.fileSize	Размер IN файла, в который выгружаются данные по контрагентам для отправки в Фактор
DeDuplicationTask.PHYSICAL.out.fileSize	Размер OUT файла с парами дубликатов, который приходит от Фактора

Данные параметры используются для расчета необходимого свободного места на диске при дальнейших запусках задачи *Поиска дубликатов*, когда проверка включена (`task.deduplication.freeSpaceRatio >= 1`).

5.8 Параметры запуска JBOSS

При старте система проверяет наличие следующих параметров запуска для JVM:

Имя параметра	Обязательный?	Значение по умолчанию	Описание
<code>cdi.root.folder</code>	нет * (но рекомендуется)		Выделенный каталог Единого клиента на сервере приложений.
<code>cdi.dedup.folder</code>	да *	<code>{cdi.root.folder}/dedup</code>	Каталог для временных файлов задачи поиска дубликатов.
<code>cdi.search.folder</code>	да *	<code>{cdi.root.folder}/search</code>	Каталог поискового индекса Единого клиента.
<code>cdi.properties.folder</code>	нет	<code>{cdi.root.folder}/conf</code>	Каталог с дополнительными настройками Единого клиента.
<code>cdi.services.state.mapdb.folder</code>	нет	<code>{cdi.root.folder}/mapdb-state</code>	Каталог для синхронизации событий при настроенном Горячем резерве.
<code>user.timezone</code>	нет * (но рекомендуется)		Часовой пояс, выставлен в GMT+4.

* Если указан параметр `cdi.root.folder`, то параметры `cdi.dedup.folder` и `cdi.search.folder` можно не указывать — в этом случае для них будут использоваться значения по умолчанию. Рекомендуется именно так и поступать.

Параметры настраиваются в:

- `JBOSS_CDI/bin/standalone.conf.bat` для Windows

- JBOSS_CDI/bin/standalone.conf для Linux

У пользователя HFL_USER должны быть права на чтение и запись для всех вышеперечисленных каталогов.

При использовании отдельных пользователей для каждой из служб, необходимые уровни доступа выглядят так:

Каталог	Доступ для CDI	Доступ для FACTOR
cdi.root.folder	полный	не требуется
cdi.properties.folder	полный	не требуется
cdi.dedup.folder	полный	полный
cdi.search.folder	полный	не требуется
cdi.services.state.mapdb.folder	полный	не требуется

5.9 Параметры импорта из буферной области

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Параметры импорта из буферной области

Название параметра	Значение
oldMaxId.buffer.physical	<input type="text" value="21839"/>

Название параметра	Значение по умолчанию	Описание
oldMaxId	0	Последний (максимальный) идентификатор

5.10 Параметры модуля мониторинга

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Название параметра	Значение по умолчанию	Описание
monitoring.applicationHealth.badHealthRefreshPeriodInSeconds	600	Как часто (в секундах) логировать

Название параметра	Значение по умолчанию	Описание
		состояние приложения, находящегося в критическом состоянии.
monitoring.applicationHealth.badHealthThreshold	0.95	Если отношение занятой памяти к общему её объему больше чем установлено значение, состояние приложения считается критическим.
monitoring.applicationHealth.goodHealthThreshold	0.85	Если отношение занятой памяти к общему её объему меньше чем установлено значение, состояние приложения считается хорошим.
monitoring.applicationHealth.refreshPeriodInSeconds	10	Как часто (в секундах) проверять состояние приложения.
monitoring.applicationHealth.http.httpPort	8080	Http-порт, на котором ведется статистика активных подключений.
monitoring.applicationHealth.http.httpsPort	8443	Https-порт, на котором ведется статистика

Название параметра	Значение по умолчанию	Описание
		активных подключений.
monitoring.applicationHealth.http.connectionsToLogThreshold	100	Число активных подключений, при превышении которого в лог будет записываться информация о размере входящей очереди сообщений.
monitoring.applicationHealth.http.workerQueueSizeToLogThreshold	10	Число очередей приложения, при превышении которого в лог будет записываться информация о размере входящей очереди сообщений.

5.11 Параметры отправки email

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Параметры делятся на параметры для рассылки диагностической информации, оповещений автомиграции приложения, мониторинга логов (ошибок, происходящих в системе) и общие. Что нужно настроить исходно, [читать тут](#).

Название параметра	Значение по умолчанию / Пример	Описание
Общие параметры		
mail.from	automation@hflabs.ru	Имя почтового ящика, с которого будет

Название параметра	Значение по умолчанию / Пример	Описание
		производится я рассылка
mail.host	smtp.gmail.com	Адрес smtp сервера, с которого будет производится я рассылка
mail.debug	false	Флаг, отвечающий за отправку сообщений DEBUG-уровня
mail.user	automation@hflabs.ru	Логин для подключения к серверу рассылки
mail.password	***	Пароль для подключения к серверу рассылки
mail.port	-1	Порт для отправки почты (если отличается от умолчательно го для протокола)
mail.protocol	smtps	Протокол для отправки почты (smtp / smtps)
Диагностическая информация		
mail.diagnostic.enabled	false	Флаг, отвечающий за отправку диагностики на почту
mail.diagnostic.to	log.catcher@hflabs.ru	Список получателей (разделитель - ";") собранный диагностики
mail.diagnostic.subject	Единый клиент {nodeld} - диагностика	Формат заголовка сообщения о

Название параметра	Значение по умолчанию / Пример	Описание
		собранный диагностики
mail.diagnostic.body	Профили: {profiles}. Параметры: {parameters}.	Содержание сообщения о собранной диагностики
mail.diagnostic.defaultProfiles	config,db.connection,logs,search,sql.counts,sql.metadata,versions	Умолчательный набор профилей
Автоматическая миграция приложения		
mail.migration.enabled	false	Флаг, отвечающий за отправку оповещений миграции на почту
mail.migration.to	log.catcher@hflabs.ru	Список получателей (разделитель - ";") оповещений миграции
mail.migration.subject.rebuildIndexes	Единый клиент {nodeId} обновляется на версию {newVersion}	Формат заголовка оповещения о перестроении индексов
mail.migration.body.rebuildIndexes	Обновление с версии {oldVersion} на {newVersion}. Будут перестроены индексы ({indexes}).	Содержание оповещения о перестроении индексов
mail.migration.subject.diagnostic	Единый клиент {nodeId} обновился на версию {newVersion}	Формат заголовка оповещения об окончании миграции
mail.migration.body.diagnostic	Обновление с версии {oldVersion} на {newVersion}. Диагностика во вложении ({profiles}).	Содержание оповещения об окончании миграции
Автоматический мониторинг логов		
mail.logging.enabled	false	Флаг, отвечающий за отправку сообщений об ошибках на почту

Название параметра	Значение по умолчанию / Пример	Описание
mail.logging.to	log.catcher@hflabs.ru	Список получателей (разделитель - ";") сообщений об ошибках
mail.logging.subject.single	CDI Error Occurred @ \${cdi.customer.name} ({errorName})	Формат заголовка сообщения об одиночной ошибке
mail.logging.subject.multiple	CDI Errors Occurred @ \${cdi.customer.name} during {reportDate} ({lastTime} - {reportTime})	Формат заголовка агрегированного сообщения об ошибках, произошедших за определенный период
mail.logging.bufferSize	10	Количество блоков информации, предшествующих ошибке
mail.logging.accumulatePeriod	3600	Частота оповещения в секундах
mail.logging.skipEventPredicate 1(2-5)	errorClass.equals('IncorrectVersionException')	<p><u>Фильтр событий ошибок</u> к отправке по почте. Если ошибка попадает под шаблон, то оповещение о ней не будет выслано.</p> <p>Изменения данных параметров вступают в</p>

Название параметра	Значение по умолчанию / Пример	Описание
		силу без рестарта ЕК.
Оповещение о необходимости проверить связи с ЧС		
mail.notify.blacklist.enabled	false	Флаг, отвечающий за отправку оповещений на почту
mail.notify.blacklist.partyLinksCountThreshold	100	В теле письма всегда приводится общее количество найденных связей и ссылка на соответствующий поисковый запрос. Когда количество найденных связей < данного порогового значения, то в теле письма дополнительно указывается полный список контрагентов с прямыми ссылками на их карточки.
mail.notify.blacklist.to	log.catcher@hflabs.ru	Список получателей (разделитель - ";") сообщений

В теме письма можно использовать шаблоны, значения которых должны быть заданы в файле standalone.conf.bat для Windows-серверов или standalone.conf для Unix.

Пример:

- `mail.diagnostic.subject = Единый клиент {nodeld} - диагностика.`
- В `standalone.conf.bat` задан параметр `-Dcdi.app.nodeld=MY_NODE_1.`
- Тема отправленного письма – «Единый клиент MY_NODE_1 - диагностика».

5.11.1 Механизм отправки писем об ошибках на почту

5.11.1.1 Механизм отправки писем об ошибках на почту (технические подробности)

При возникновении ошибки «Единый клиент» через 10 сек отправляет на почту письмо с информацией о ней.

С этого же момента начинается формирование суммарного отчета по событиям лога за период:

- длина периода по умолчанию час (настраивается через параметр `mail.logging.accumulatePeriod`);
- в отчете накапливается информация о всех последующих ошибках данного типа (кроме первой, она в отчет не попадает), отдельных писем по ним приходит не будет.

При возникновении нового типа ошибки:

- через 10 сек отправляется на почту письмо о ней
- уже существующий отчет также начинает накапливать информацию по последующим подобным ошибкам.

Суммарный отчет отправляется на почту в конце периода, вся накопленная информация об ошибках при этом сбрасывается.

Если за период в отчете не было накоплено ошибок, то он не будет отправлен.

Если первые ошибки разных типов произошли в одни и те же 10 сек, то приходит не отдельное письмо по каждой, а мини-агрегирующий отчет за 10 сек.

5.11.1.2 Пример 1

В течении часа произошло:

- 10:00 – 2 ошибки `NotAuthorizedLocalUserAlert`
- 10:05 – 2 ошибки `IllegalArgumentException`
- 10:10 – 2 ошибки `AccessDeniedException`

На почту придет:

- 10:00 – отдельное письмо про `NotAuthorizedLocalUserAlert`
- 10:05 – отдельное письмо про `IllegalArgumentException`
- 10:10 – отдельное письмо про `AccessDeniedException`
- 11:00 – суммарный отчет за период 10:00 – 11:00, в котором:

<code>NotAuthorizedLocalUserAlert</code>	1
<code>IllegalArgumentException</code>	1
<code>AccessDeniedException</code>	1

5.11.1.3 Пример 2

В течении часа произошло:

- 10:05 – 2 ошибки NotAuthorizedLocalUserAlert
- 10:05 – ошибка IllegalArgumentException
- 10:05 – 2 ошибки AccessDeniedException
- 10:36 – ошибка AccessDeniedException

На почту придет:

- 10:05 – суммарный мини-отчет за период 10:05 – 10:05, в котором:

NotAuthorizedLocalUserAlert	1
IllegalArgumentException	1
AccessDeniedException	1

- 11:05 – суммарный отчет за период 10:05 – 11:05, в котором:

NotAuthorizedLocalUserAlert	1
AccessDeniedException	2

5.11.2 Фильтрация ошибок перед отправкой на почту

5.11.2.1 Зачем это

Мы умеем [МОНИТОРИТЬ ОШИБКИ](#) в логах с прода: система упала → нам пришло письмо → мы быстро все починили / завели баг в исходную систему.

Но быстро чиним только мы. Если проблема в исходной системе, фикса можно ждать месяцами. А письма все идут и идут... Поэтому мы научились игнорировать неинтересные нам ошибки:

- Система упала;
- На почту письмо НЕ пришло;
- cdi.log не замусорило → заскипанные ошибки вынесены в отдельный лог (начиная с релиза 16.12)

5.11.2.2 Как делать

Пополнять параметры skipEventPredicate1, skipEventPredicate2... Можно сразу через админку – параметры есть в конфигурации email.

Если меняем с релизом, то с помощью скрипта миграции. Сначала узнаем, какие параметры заняты (не null):

```
select * from config_property
where source = 'mail.properties'
and name like 'mail.logging.skipEventPredicate%';
```

Потом пополняем. Например, система AL постоянно шлет закрытие уже закрытых атрибутов. Мы не хотим читать спам от AL, но хотим узнать, если другая система начнет делать также. Скрипт миграции:

```
begin
  UPDATE config_property SET value = 'message matches "(?s).*Error closing
attribute.*sourceSystem AL.*"' WHERE name = 'mail.logging.skipEventPredicate1';
```

```
exception when dup_val_on_index then null;
end;
/
```

Кавычки экранируем другими кавычками.

Если хотим проверить скрипт и надо его через SQLdeveloper исполнить, то предварительно необходимо выполнить `set define off`, иначе ругается на амперсанд. В миграции этого не нужно.

5.11.2.3 Механизм фильтрации (технические подробности)

Для удобного использования информации об ошибке над ней формируется объект-обертка:

```
private static class LoggingEventContext {
    /** Категория логгера */
    private final String logger;
    /** Логируемое сообщение (без исключения) */
    private final String message;
    /** Объект исключения (может быть пустым) */
    private final Throwable error;
    /** Класс исключения без пакета (getSimpleName) длл удобного сравнения */
    private final String errorClass;
    /** Сообщение исключения вместе со стектрейсом одной строкой*/
    private final String errorMsg;
}
```

По каждому полю можно задавать условия фильтрации (предикаты), которые представляют из себя выражения [SpEL](#).

Под предикаты отведено несколько параметров (в разделе *Параметры отправки email*) – `mail.logging.skipEventPredicate1(2-5)`. Значения параметров подтягиваются сразу после сохранения, без рестарта ЕК.

Параметры отправки email

Название параметра	Значение
<code>mail.logging.skipEventPredicate1</code>	<input type="text" value="logger.contains('ru.hflabs.task')"/>
<code>mail.logging.skipEventPredicate2</code>	<input type="text" value="message.contains('Not enough free disk space')"/>
<code>mail.logging.skipEventPredicate3</code>	<input type="text"/>
<code>mail.logging.skipEventPredicate4</code>	<input type="text"/>
<code>mail.logging.skipEventPredicate5</code>	<input type="text"/>

Сохранить

Все параметры суммируются по условию "или". В параметре можно задать несколько предикатов через точку с запятой, они тоже будут суммироваться по "или".

Чтобы задать несколько предикатов через условие "и" следует использовать оператор `&&`.

5.11.2.4 Примеры предикатов

Обертка ошибки нехватки места при перестроении индекса:

```
logger = "ru.hflabs.task.manager.TaskWorker";
message = "Task 'graphEngineRebuildTaskPerformer' finished with error: Can't rebuild GRAPH"
```

```

engine. Not enough free disk space: required 4146243 (4.15 MB) but got only 2273280 (2.27 MB)";
error = "java.lang.IllegalStateException: Can't rebuild GRAPH engine. Not enough free disk space:
required 4146243 (4.15 MB) but got only 2273280 (2.27 MB)";
errorClass = "IllegalStateException";
errorMsg = "java.lang.IllegalStateException: Can't rebuild GRAPH engine. Not enough free disk
space: required 4146243 (4.15 MB) but got only 2273280 (2.27 MB)
  at
ru.hflabs.cdi.engine.EnginesRebuildProcessor.checkFreeSpace(EnginesRebuildProcessor.java:255
)
  at
ru.hflabs.cdi.engine.EnginesRebuildProcessor.rebuildEngine(EnginesRebuildProcessor.java:186)
  at
ru.hflabs.cdi.engine.EnginesRebuildProcessor.executeRebuild(EnginesRebuildProcessor.java:108)
  at
ru.hflabs.cdi.engine.EnginesRebuildController.rebuildCustomEngines(EnginesRebuildController.jav
a:78)
  ...";

```

Фильтрация:

- по категории логгера

```
mail.logging.skipEventPredicate1 = logger.contains('ru.hflabs.task')
```

- по сообщению исключения вместе со стектрейсом

```
mail.logging.skipEventPredicate1 = errorMsg matches '(?s).*EnginesRebuildProcessor.*'
```

Встроенный флаг "(?s)" включает учет перевода строки для "."

- по объекту исключения

```
mail.logging.skipEventPredicate1 = error instanceof T(java.lang.IllegalStateException)
```

- несколько предикатов через оператор "и", по логируемому сообщению и классу исключения

```
mail.logging.skipEventPredicate1 = message.contains('Not enough free disk space') &&
errorClass.equals('IllegalStateException')
```

- несколько предикатов через оператор "или"

```
mail.logging.skipEventPredicate1 = logger.contains('ru.hflabs.task')
mail.logging.skipEventPredicate2 = message.contains('Not enough free disk space');
errorClass.equals('IllegalStateException')
```

5.12 Параметры подключения к Фактору

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Название параметра	Значение по умолчанию	Описание
factor.blacklist.cdi.context	/factor-service-{customer}/services/CdiBlackListService	WSDL сервиса поиска по Черным спискам
factor.blacklist.cdi.timeout	600000	Тайм-аут ожидания ответа от Фактора во время поиска по Черным спискам, в миллисекундах
factor.clean.context	/factor-service-{customer}/services/CleanService	WSDL CleanService
factor.clean.packetSize	200	Размер пакета на стандартизацию / трансформацию
factor.clean.timeout	1200000	Тайм-аут ожидания ответа от Фактора во время стандартизации / трансформации (для каждого пакета), в миллисекундах
factor.dedup.context	/factor-service-{customer}/services/DedupService	WSDL сервиса поиска дубликатов
factor.dedup.full.threadsCount	-1	Количество потоков для выполнения полной дедубликации в Факторе (автоматически, если указан "-1"). Эти же настройки используются для оффлайн проверки по ЧС / ЕГРЮЛ.
factor.dedup.full.threadsPriority	-1	Приоритет потоков для

Название параметра	Значение по умолчанию	Описание
		<p>выполнения полной дедубликации в Факторе (автоматически, если указан "-1").</p> <p>Эти же настройки используются для оффлайн проверки по ЧС / ЕГРЮЛ.</p> <p>Все приоритеты настраиваются в диапазоне [1, 10], приоритет по умолчанию (если не указан или указан отрицательный): 5</p>
factor.dedup.incremental.queueSize	10000	<p>Размер очереди на асинхронную передачу изменений в Фактор для инкрементального поиска дубликатов. Очередь измеряется в операциях, одна операция = пакет измененных данных (коллекция контрагентов при трансформации, один контрагент при операциях через АРМ МД)</p>
factor.dedup.timeout	600000	<p>Тайм-аут ожидания ответа от Фактора во время поиска дубликатов, в миллисекундах</p>
factor.factor.context	/factor-service-{customer}/services/FactorService	WSDL Фактора
factor.factor.timeout	600000	<p>Тайм-аут ожидания ответа от Фактора</p>

Название параметра	Значение по умолчанию	Описание
factor.host	http://host:port	Базовый URL Фактора в формате http://host:port (используется для поиска дубликатов)
factor.hosts	http://host:port	Перечень URL Фактора для пакетной очистки (может быть несколько), разделитель - ";"
factor.ws.numberOfRetries	1	Количество попыток повторного вызова Фактора при наличии сетевой ошибки
factor.ws.delayBetweenRetries	1000	Задержка в мс между попытками повторного вызова Фактора при наличии сетевой ошибки
factor.ws.chunkingMode	ALWAYS	<p>Режим использования Chunked Encoding при обращении к серверу:</p> <ul style="list-style-type: none"> • ALWAYS — всегда использовать ; • REMOTE — использовать только для удаленных серверов; • NEVER — никогда не использовать . <p>При возникновении ошибок "Marshalling Error: Error writing request body to</p>

Название параметра	Значение по умолчанию	Описание
		server" рекомендуется переключить в REMOTE.

5.13 Параметры поискового индекса

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Название параметра	Значение по умолчанию	Описание
engine.rebuild.freeSpaceRatio	1.5	Коэффициент свободного места, которое должно быть при очередном запуске задачи перестроения индекса Если значение <1, то проверка свободного места отключена
engine.rebuild.pageSize	10000	Размер страницы (количество контрагентов) при перестроении индекса
engine.rebuild.poolSize	-1	Количество потоков для перестроения индекса Если равно -1, устанавливается по формуле: $\min(\text{количество процессоров (ядер)} * 2, 8)$
engine.rebuild.skipOptimize	false	Пропускать ли оптимизацию при перестроении
engine.rebuild.sortedIterator	true	Добавлять ли сортировку в итератор по актуальным патям. По умолчанию true, поменять можно без ребута ЕК. Параметр можно менять для ускорения работы поискового индекса, но это надо тестировать на конкретной базе. Оптимальное значение по умолчанию. Сортировка в этом итераторе добавлялась, чтобы при загрузке атрибутов по пачке клиентов делать это быстрее, потому что попадаем в соседние листы индекса.

Название параметра	Значение по умолчанию	Описание
engine.rebuild.strategy	BLOCKING	<p>Стратегия перестроения индекса:</p> <ul style="list-style-type: none"> • BLOCKING – во время перестроения блокируются все модификации индекса, рядом строится новый свежий индекс, которым потом подменяем старый; • CONCURRENT – состоит из двух шагов (применима только для индексов контрагентов; граф / группы дублей / конфликты перестраиваются по BLOCKING алгоритму): <ul style="list-style-type: none"> • в существующем индексе обновляются все документы, индекс оптимизируется; • поиск неконсистентных документов, которых в индекс быть не должно: проверяются все документы старше, чем начало полного перестроения – по ним отдельно проходит проверка актуальности и те, которых быть не должно, удаляются. <p>Если у вас нет проблем с онлайн-потоками, то эффективнее использовать BLOCKING. Преимущества – она быстрее и требует меньше места. Минусы – запоздания в попадании данных в индекс + ограничение очереди (пока строит индекс, все изменения по клиентам складываются в очередь. В search ответе вернутся только после окончания перестроения)</p> <p>Если онлайн идет постоянно, используйте CONCURRENT</p>
engine.search.bufferSize	1000	<p>Размер буфера коммита изменений поискового индекса на диск. Все операции из очереди группируются в памяти, пока не будет заполнен данный буфер. Буфер измеряется в контрагентах</p>

Название параметра	Значение по умолчанию	Описание
		<p>(1 контрагент = 1 документ поискового индекса)</p> <p>Если выставить "-1", то буфер вообще не будет создаваться, каждый контрагент будет сразу передаваться в поисковый индекс.</p> <p>Однако это увеличит время работы метода save, которое будет включать не только создание, но и оповещение индекса.</p> <p>Параметр действует по всей системе, не только в SOAP, поэтому на загрузке из буферной области или внешних файлах это также может негативно сказаться.</p>
engine.search.commitInterval	5000	<p>Частота коммита изменений поискового индекса на диск.</p> <p>Все операции из очереди группируются в памяти, пока не сработает таймер коммита.</p> <p>Измеряется в миллисекундах. Не рекомендуется устанавливать менее 1000 мс.</p>
engine.search.maxBufferedDocs	1000	<p>Максимальное количество документов в памяти до сброса на диск</p>
engine.search.needOptimizeWhenOpen	false	<p>Проводить оптимизацию индекса при старте системы?</p>
engine.search.queueSize	20000	<p>Размер очереди операций по модификации индекса.</p> <p>Если очередь полностью заполняется, на все последующие операции модификации индекса возвращается ошибка.</p>
engine.search.indexDirectoryType	AUTO	<p>Способ чтения файлов индекса (тип директории – AUTO / MMAP / SIMPLE / NIO / RAM) для всех индексов, кроме групп дубликатов.</p> <p>Можно переключать с AUTO (который на современных серверах всегда MMAP) на NIO. Это замедлит скорость отдельно взятых процессов, но в сумме</p>

Название параметра	Значение по умолчанию	Описание
		дает большую стабильность сервера и прогнозируемость.
engine.search.groups.indexDirectoryType	AUTO	Способ чтения файлов индекса (тип директории – AUTO / MMAP / SIMPLE / NIO / RAM) для групп дубликатов. Если есть проблемы с MMAP, то в первую очередь стоит поменять доступ на NIO для групп дубликатов.
engine.search.supportLegacyNumericFields	true	Флаг для переключения формата полей, которые пишутся в индекс на новый Lucene 6. Если false, то все запросы начинают работать по новым полям Lucene 6, старый формат полей не пишется в индекс.
engine.search.useNIOForIterateIndex	false	Флаг использования доступов через NIO для итераторов по всему индексу, в частности, при выгрузке кандидатов для поиска дубликатов.
engine.validation.maxAcceptableMismatch	-1	Максимально допустимое расхождение при валидации между индексом и БД. По умолчанию -1, на любое расхождение будет ERROR-сообщение в лог.

5.13.1 Неблокирующее перестроение поискового индекса

```
engine.rebuild.strategy = CONCURRENT
```

5.13.1.1 Стратегия перестроения

Состоит из двух шагов:

1. Обновление всех документов в существующем индексе, оптимизация.
2. Поиск неконсистентных документов, которых в индексе быть не должно. Например, контрагент когда-то закрылся, но в индексе до сих пор фигурирует. Проверяются все документы старше, чем начало полного перестроения – по ним отдельно проходит проверка актуальности и те, которых быть не должно, удаляются.
Для быстрой работы, в каждом документе хранится timestamp его добавления / обновления.

Данная стратегия применима только для индексов контрагентов. Граф / группы дублей / конфликты перестраиваются по старому алгоритму.

5.13.1.2 Возобновляемость

Процесс перестроения **возобновляемый**, каждые 500 тыс документов сохраняются данные о последнем хиде и дате последней активности в файл `{cdi.root.folder}/work/{PartyType}.rebuildState.properties`.

Если повторно запустить перестроение того же индекса в течении суток, то процесс продолжится с последнего хида, о чем будет написано в лог.

```
INFO ConcurrentEnginesRebuildProcessor - The last rebuild state (DD.MM.YYYY HH:MI:SS) is found. Start rebuilding from hid XXX.
```

Если позднее чем через сутки, то начнется сначала. Так сделано, чтобы спустя месяц не начать старый процесс.

```
INFO ConcurrentEnginesRebuildProcessor - The last rebuild state is too old: DD.MM.YYYY HH:MI:SS, 24 h max allowed. Start rebuilding from scratch.
```

5.13.1.3 Требования к месту

На время перестроения размер индекс увеличивается в 3,2 раза от первоначального размера, максимум достигается на этапе оптимизации. Заказчикам, которые будут использовать данный способ перестроения, нужно как минимум увеличивать `engine.rebuild.freeSpaceRatio` с 1.5 до 2.

5.13.1.4 Рекомендации по запуску

Процесс значительно загружает БД, поскольку активно читает данные по всем контрагентам, поэтому желательно все равно запускать перестроение в неактивное рабочее время, чтобы минимизировать возможные эффекты на онлайн нагрузку. Или, если же нужно запустить в рабочее время, то **уменьшать количество потоков до 8-12** в зависимости от производительности конкретной БД заказчика.

Для быстрой смены в параметрах задачи можно напрямую задать `poolSize` без перегрузки ЕК, который будет использоваться вместо умолчательного значения `engine.rebuild.poolSize`.

5.14 Параметры сервисов

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Название параметра	Значение по умолчанию	Описание
<code>services.actualityDate.truncateTime</code>	true	Если true, то для даты актуальности используется тип данных DATE, иначе – TIMESTAMP (для переключения на

Название параметра	Значение по умолчанию	Описание
		TIMESTAMP нужна перепоставка релиза).
services.applicationMigration.enabled	false	Флаг включения миграции версий приложения
services.applicationMigration.rebuildInvalidIndexes	false	Флаг автоматического перестроения невалидных индексов на миграции приложения
services.cache.refreshIntervalInMin	60	Периодичность сбора кэша справочников (в минутах).
services.dao.versioning.defaultPhase	—	Умолчательная фаза миграции для таблиц, отсутствующих в MIGRATION_PHASE
services.deduplication.autoMatchScope	98	Пороговое значения коэффициента похожести для гарантированных дубликатов по умолчанию.
services.deduplication.autoMatchScopeByPartyType	—	Пороговое значения коэффициента похожести для гарантированных дубликатов для конкретных типов PARTY (заполняется только в случае, если значения разные, например - PHYSICAL:85;LENDER:95).
services.deduplication.byPartyHidsFullScanIteratorThreshold	2000000	Максимальный размер контрагентов для итерирования фулсканом по дубликатам при инкрементальной загрузке
services.deduplication.matchScopeThreshold	0	Минимальный порог похожести дубликатов для создания дубликата (все, что ниже — уже не дубли, а домохозяйства)
services.deduplication.matchScopeThresholdByPartyType	PHYSICAL : 50	Минимальный порог похожести дубликатов для создания дубликата по типам контрагентов
services.deduplication.maxDuplicatesForGroupsCalculation	2000000	Максимальный размер множества дублей, загружаемых в память, для построения группы дубликатов.

Название параметра	Значение по умолчанию	Описание
services.lifecycleEvents.persistent	true	Сохранять ли в таблице logging события старта-остановки приложения
services.merge.attribute.mergePackSize	50	Допустимое количество атрибутов, которые можно сливать "каждый с каждым" (при превышении делить на группы)
services.merge.findCycles	true	Если false, то поиск циклов не производится. Группа дубликатов отмечается как ошибочная при подозрении на цикл. Если true, то определяются атрибуты-победители без поиска циклов. Если в результате есть несовпадение ожидаемого и фактического количества атрибутов, происходит поиск циклов.
services.pastValues.enabled	false	Включить сервис использования прошлых атрибутов для поиска дублей
services.relation.enabled	false	Включить сервис расчета графа взаимосвязей.
services.sql.paramsLoggingEnabled	true	Флаг записи параметров sql-запросов в cdi.log.
services.sqlProfiler.thresholdInSeconds	60	Пороговое значение для логирования sql-запросов. Запрос, выполняющийся дольше установленного значения, будет сохранен в логе.
services.state.mapdb.useMmap	true	Использовать mmap для работы с локальной базой (файловая), в которой хранятся даты последнего обновления индекса. Если false, то mmap не используется — используется обычный доступ к файлам
services.suggestions.enabled	false	Флаг включения подсказок для адресов

Название параметра	Значение по умолчанию	Описание
services.suggestions.token	{api-ключ}	Ключ доступа к облачным подсказкам
services.suggestions.upperCase	true	Флаг перевода значений из подсказок в верхний регистр при подстановке
services.suggestions.url	https://dadata.ru/api/v2	URL сервиса подсказок. При использовании standalone-подсказок заменить на: http://{ip}:{port}/suggestions/api/4_1/rs.
services.transform.closeMissedAttributes	false	Закрывать ли при обновлении атрибуты, которые отсутствуют в обновлении?
services.transform.maxSourcePartiesToUpdateResult	-1	Максимальное количество исходных контрагентов, при котором можно обновлять результирующую карточку
services.transform.updateConflictStrategy	DISABLED	Стратегия мониторинга конфликтующих изменений . Для подключения выставить в значение SKIP_UPDATE_AND_SAVE.
services.transform.updateConflictStrategyByPartyType		Стратегия мониторинга конфликтующих изменений по типам контрагентов

5.14.1 Стратегия решения конфликтов

При [мониторинге подозрительных изменений](#) (параметр services.transform.updateConflictStrategy в админке) доступны стратегии:

Название	Стратегия
DISABLED	Конфликты не искать, обновление по стандартным правилам
SKIP_UPDATE_AND_SAVE	Конфликты записывать, конфликтующие обновления блокировать.
UNMERGE_AND_APPLY	Исходную, на которую пришло конфликтное обновление, разлить и обновление применить.

После смены стратегии перезапустите «Единый клиент».

5.15 Параметры синхронизации нод

Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!

Параметры синхронизации нод

Название параметра	Значение
nodeSync.distributed.members	server-1, server-2
nodeSync.distributed.port	5071
nodeSync.jms.enabled	true
nodeSync.jms.queue.name	cdi.nodeSync
nodeSync.jms.receive.connectionFactory.jndiName	nodeSyncRemoteConnectionFactory
nodeSync.jms.receive.maxInterval	300000
nodeSync.jms.receive.password	cdi
nodeSync.jms.receive.recoveryInterval	30000
nodeSync.jms.receive.username	sync
nodeSync.jms.send.connectionFactory.jndiName	jmsConnectionFactory

Сохранить

Название параметра	Значение по умолчанию	Описание
nodeSync.jms.enabled	true	Флаг включения синхронизации меж нодами
nodeSync.jms.queue.name	cdi.nodeSync	Название очереди
nodeSync.jms.send.connectionFactory.jndiName	jmsConnectionFactory	Название JMS-фабрики для отправки сообщений на вторую ноду
nodeSync.jms.receive.connectionFactory.jndiName	nodeSyncRemoteConnectionFactory	Название JMS-фабрики для получения сообщений со второй ноды
nodeSync.jms.receive.username	sync	Логин авторизации для соединения с очередью второй ноды
nodeSync.jms.receive.password	cdi	Пароль авторизации для соединения с

Название параметра	Значение по умолчанию	Описание
		очередью второй ноды
nodeSync.jms.receive.recoveryInterval	30000	Интервал ожидания между попытками возобновить соединение (в миллисекундах). Значение увеличивается экспоненциально с коэффициентом 1.5 (т.е. проверки будут 30 сек → 45 сек → 68 сек и т.д. до 5 мин).
nodeSync.jms.receive.maxInterval	300000	Максимальный интервал ожидания между попытками возобновить соединение (в миллисекундах).
nodeSync.distributed.members	localhost	Список нод через запятую, которые образуют кластер
nodeSync.distributed.port	5071	Порт общения меж нодами для пересылки информации о блокировке в режиме разбора буфера

5.16 Параметры экспорта данных

Параметры экспорта данных

Название параметра	Значение
incrementalExportRelations.lastDate	10.02.2015 02:23:47

Сохранить

Название параметра	Значение по умолчанию	Описание
<code>incrementalExportRelations.lastDate</code>	пустое	Дата последнего запуска экспорта эталонных связей

При запуске задачи *fullExportRelations* (Полный экспорт эталонных связей) параметр сбрасывается.

Задача *incrementalExportRelations* (Инкрементальный экспорт эталонных связей) обновляет параметр на дату последнего запуска.

6 Работа со справочниками

Пакет `reference_pkg` позволяет пополнять справочники.

С помощью пакета `reference_pkg` можно пополнять бизнес-классификаторы, как [общие](#) (например, типы адресов, телефонов, документов), так и специфичные для конкретного заказчика (например, уровень образования, тип занятости, род деятельности организации). При пополнении справочников может потребоваться доработка правил стандартизации (это зависит от конфигурации заказчика, подробности можно уточнить в службе технической поддержки).

6.1 Добавление в справочник новой записи

Для добавления новых записей в справочники предназначена процедура `create_reference_record`.

Параметр	Обязательность	Значение по умолчанию	Описание
<code>p_ref_type</code>	Да		Тип справочника
<code>p_code</code>	Да		Литера справочного значения
<code>p_ordinal</code>	Да		Числовой код справочного значения
<code>p_label</code>	Да		Краткое описание справочного значения
<code>p_description</code>	Нет	null	Развернутое описание справочного значения
<code>p_party_type</code>	Нет	null	Типы контрагентов (через запятую), для которых может использоваться значение. Если не задан, то значение доступно всем типам контрагентов. Пример: домашний телефон может быть только у контрагентов-физических лиц (PHYSICAL), а мобильный — у любых контрагентов (в том числе юридических лиц и банков).
<code>p_author</code>	Нет	HFLabs-CDI	Автор последнего изменения

6.2 Модификация существующей в справочнике записи

Для модификации существующих в справочнике записей предназначена процедура `update_reference_record`. Изменять можно любые поля, кроме типа справочника и литеры справочного значения.

Параметр	Обязательность	Значение по умолчанию	Описание
<code>p_ref_type</code>	Да		Тип справочника
<code>p_code</code>	Да		Литера справочного значения

Параметр	Обязательность	Значение по умолчанию	Описание
p_ordinal	Нет	Исходное значение поля в обновляемой записи	Числовой код справочного значения
p_label	Нет	Исходное значение поля в обновляемой записи	Краткое описание справочного значения
p_description	Нет	Исходное значение поля в обновляемой записи	Развернутое описание справочного значения
p_party_type	Нет	Исходное значение поля в обновляемой записи	Типы контрагентов (через запятую), для которых может использоваться значение. Если не задан, то значение доступно всем типам контрагентов.
p_author	Нет	Исходное значение поля в обновляемой записи	Автор последнего изменения

6.3 Представление с содержимым справочников

Представление `reference_view` содержит всю информацию, существующую в справочниках. Модель данных аналогична таблице [REFERENCE](#).

6.4 Список справочников, которые можно пополнять и редактировать

- [Общие классификаторы](#).
- Реквизиты, значения которых являются справочными (набор таких реквизитов определяется конфигурацией заказчика).

6.5 Примеры использования процедур (Oracle)

Добавление нового типа телефона, который может использоваться для любого типа контрагентов.

```
BEGIN
reference_pkg.create_reference_record(
  p_ref_type => 'PHONE_TYPE',
  p_code     => 'NEW_TYPE',
  p_ordinal  => 10,
  p_label    => 'Новый',
  p_description => 'Новый тип телефона',
  p_author   => 'ivanov'
```

```
);
END;
```

Изменение существующего типа адреса – после изменений адрес с указанным типом будет доступен только юридическим лицам.

```
BEGIN
reference_pkg.update_reference_record(
  p_ref_type => 'ADDRESS_TYPE',
  p_code     => 'LOCATION',
  p_party_type => 'LEGAL'
);
END;
```

6.6 Примеры использования процедур (MariaDB)

Добавление нового типа телефона, который может использоваться для любого типа контрагентов.

```
call reference_pkg__create_reference_record('PHONE_TYPE', 'NEW_TYPE', 100, 'Новый', 'Новый
тип телефона', null, 'ivanov');
```

Изменение существующего типа адреса – после изменений адрес с указанным типом будет доступен только юридическим лицам.

```
call reference_pkg__update_reference_record('ADDRESS_TYPE', 'LOCATION', null, null, null, 'LEGAL',
null);
```

7 Журналирование и аудит

7.1 Журналы Единого клиента

Журналы расположены на сервере приложений в каталоге JBOSS_CDI_HOME/standalone/log.

Все логи кроме cinsole.log и cdi-lifecycle.log ротируются на ежедневной основе с добавлением суффикса YYYY-MM-DD к расширению лога.

Журнал	Описание	условие создания
console.log	Консольный вывод джавы. Перезатируется при рестарте приложения.	всегда при старте jboss
server.log	Системные сообщения и сообщения от сервера JBOSS.	всегда при старте jboss
cdi.log	Основной журнал работы Единого клиента.	всегда при старте cdi
cdi-security.log	Журнал попыток аутентификации пользователей в системе. По умолчанию логируются только неуспешные попытки и долгие запросы к Active Directory	всегда при старте cdi
cdi-soap-stats.log	Журнал статистики вызовов SOAP/REST-методов. Пустой, если нет онлайн-вызовов	всегда при старте cdi
cdi-health.log	Журнал состояния памяти приложения. Содержит treaddump, собранный при превышении минимального порога доступной приложению памяти.	всегда при старте cdi
cdi-jms.log	Журнал событий, относящихся к очередям: горячего резерва, jms-интерфейсов.	если подключен jms
cdi-distributed.log	Журнал синхронизации веб-интерфейса (только для горячего резерва)	если подключен горячий резерв
cdi-soap-trace.log	Журнал трассировки SOAP/REST-запросов. Содержит все присланные SOAP-запросы и ответы на них. Подключается по запросу на короткое время.	по умолчанию отключен
cdi-data-access.log	Журнал получения персональных данных внешними потребителями. Чтобы подключить, напишите в службу поддержки.	по умолчанию отключен
cdi-minor-errors.log	Журнал ошибок, исключенных из основного лога.	всегда при старте cdi
cdi-soap-longops.log	Журнал трассировки SOAP/REST-запросов, выполняющихся дольше максимального порога. Настройка критичного времени ответа – webservice.logging.filters	всегда при старте cdi
cdi-lifecycle.log	Обновления версий сервиса и события старта/стопа приложения. Лог не пересоздается.	При первом старте cdi

7.1.1 Журнал аутентификаций

7.1.1.1 Стандартное журналирование
По умолчанию в cdi-security.log записываются:

1. Неудачные попытки авторизации:
{Timestamp} {Имя потока} WARN authentication - Authentication failure. Principal: {логин пользователя}. Cause by: {причина ошибки} (Source: {UI - при авторизации в веб-интерфейсе CDI, API - при авторизации на веб-сервисах} ;RemoteAddress: {IP-адрес, с которого производилась попытка логина})
2. Сообщения о нехватке прав для выполнения действия:
{Timestamp} {Имя потока} WARN authorization - Недостаточно прав на {Запрошенное действие}. Требуются права: {Необходимые права}. Текущие роли: {Роли пользователя} (Source: {UI - при авторизации в веб-интерфейсе CDI, API - при авторизации на веб-сервисах} ;RemoteAddress: {IP-адрес, с которого производилась попытка логина})
3. Долгие запросы к Active Directory:
{Timestamp} {Имя потока} WARN LongopsLdapBindAuthenticator - Long running LDAP request {длительность запроса}. Username: {логин пользователя}, WS-Operation: {название метода веб-интерфейса}, WS-MessageID: {ссылка на журнал cdi-soap-trace.log}

Примеры всех типов сообщений:

```
2017-08-24 10:30:01,123 [AsyncApplicationEventListenerExecutor-1] WARN authentication -
Authentication failure. Principal: 'cdiadmin'. Cause by: BadCredentialsException: LDAP 52e (Source:
API;RemoteAddress: 10.0.63.1)
2017-08-24 10:30:01,123 [AsyncApplicationEventListenerExecutor-2] WARN authorization -
Недостаточно прав на FilterInvocation: URL: /ui/. Требуются права:
PERM_WEB_CDM_EXECUTE. Текущие роли: ROLE_EXTERNAL_SYSTEM (Source:
UI;RemoteAddress: 10.0.63.1)
2017-08-24 10:30:01,123 [default task-10] WARN LongopsLdapBindAuthenticator - Long running
LDAP request [250 ms]. Username: cdiadmin, WS-Operation: save, WS-MessageID: 1
```

7.1.1.2 Расширенное журналирование


Для журналирования всех попыток аутентификации поменяйте уровень журналирования с INFO на TRACE для AuthEventListener в файле JBOSS_CDI_HOME\standalone\tmp\vfs\tmp*** (с наибольшей датой актуальности)\cdi-web-*.war\WEB-INF\classes\logback.xml

```
<logger name="ru.hflabs.cdi.interceptor.logging.listener.AuthEventListener" level="TRACE"
additivity="false">
  <appender-ref ref="SECURITY"/>
</logger>
```

Формат записи об успешной аутентификации:

```
{Timestamp} TRACE {логин пользователя} - Authentication success. Granted authority: {Роли LDAP, которые выданы данному пользователю} (Source: {UI - при авторизации в APM Администратора, RMI - при авторизации в APM Менеджера данных} ;RemoteAddress:{IP-адрес, с которого производилась попытка логина})
```

Перезапуск JBoss при этом не требуется, журналирование будет автоматически включено в течение нескольких секунд.

 Журналирование успешных попыток аутентификации будет автоматически выключено при перезапуске сервиса ЕК.

Включать данное журналирование на длительное время не рекомендуется, т. к. это снижает быстродействие системы, и существенно увеличивает объем журналов. Чтобы отключить журналирование успешных попыток, достаточно заменить в указанной настройке уровень протоколирования TRACE на INFO, либо перезапустить JBoss.

7.1.2 Журнал доступа к персональным данным

7.1.2.1 Доступ к персональным данным

Попытки доступа к клиентской информации журналируются в следующем формате: {Timestamp} User {логин пользователя} IP {IP-адрес клиента} - {Интерфейс + способ доступа} - returned {hid}

7.1.2.2 Примеры

Пример 1. Вызов метода save через API (веб-сервис PartyWS) вернул информацию по контрагенту с идентификатором 1024:

```
2016-01-01 00:00:01,000 User 'external_system' IP '192.168.0.55' - PartyWS.save - returned [1024]
```

Пример 2. Вызов метода search через API (веб-сервис PartyWS) вернул информацию по контрагентам с идентификаторам 1024, 1025 и 1026:

```
2016-01-01 00:00:01,000 User 'external_system' IP '192.168.0.55' - PartyWS.search [Иванов] - returned [1024,1025,1026]
```

Пример 3. Вызов метода search через API (веб-сервис PartyWS) вернул ошибку:

```
2016-01-01 00:00:01,000 User 'external_system' IP '192.168.0.55' - PartyWS.search [PhysicalParty.surname=Иванов] - returned [exception]
```

Пример 4. Дата-стюард открыл карту контрагента через веб-интерфейс:

```
2016-01-01 00:00:01,000 User 'IvanovPS' IP '192.168.0.79' - WebUI.getParty - returned [1024]
```

7.1.3 Журнал исключенных ошибок

Ошибки, исправление которых лежит вне нашей зоны ответственности (исходные системы, интеграция, баги в сервере приложений), исключаются по [набору условий](#) из основного лога, чтобы не мешать поддержке.

Чтобы не потерять их совсем, они сохраняются в журнал "исключенных" ошибок — cdi-minor-errors.log.

7.1.3.1 Особенности

Лог создается всегда, даже если условия фильтрации не заданы.

Включен в сбор диагностики.

Форматирование и информация об ошибке остается такой же, как было в основном логге – cdi.log.

7.1.4 Журнал обновлений версий сервера

Журнал отслеживает:

- миграции БД ЕК на новую версию;
- старт приложения с указанием версии ядра и версии релиза заказчика;
- остановка приложения.

Для анализа проблемы нужно знать, в каком релизе она появилась. Но как отследить, когда было установлено обновление? Раньше можно было подсмотреть в базе, когда выполнялись скрипты автомиграции. Но что, если скриптов не было, а обновление было? А если горячий резерв и одну ноду обновили на неделю раньше другой? Оставалось лишь уповать на долгосрочную память...

Теперь это можно проверить в логге:

```
2017-07-10 16:02:11,841 INFO ApplicationMigrationListener - Application migration from 17.6-SNAPSHOT to 17.7-SNAPSHOT revision 454bef1c338b started
2017-07-10 16:02:30,098 INFO ApplicationMigrationListener - Application migration from 17.6-SNAPSHOT to 17.7-SNAPSHOT revision 454bef1c338b finished
```

Лог неротируемый – не будет обновляться ежедневно

7.1.5 Логи задачи трансформации

7.1.5.1 Логирование результатов

После завершения задачи трансформации в логах пишется отдельно:

- BUFFER - сколько записей было обработано на уровне STAGING
- TRANSFORM - сколько записей было обработано на этапе трансформации

Трансформация умеет распознавать события:

- CREATE
- UPDATE_ORIGINAL
- UPDATE_MERGED
- CLOSE
- SKIP
- ERROR

```
-- Создание контрагентов
BUFFER: processed 4 entities (create: 4); 0 with errors; took 1 seconds or 205,5 ms; speed 4.0 records/sec
TRANSFORM: processed 4 entities (create: 3, skip: 1); 0 with errors; took 5 seconds or 5,428 s; speed 0.8 records/sec
```

```

-- Обновление влитого контрагента
BUFFER: processed 3 entities (update: 1, skip: 2); 0 with errors; took 1 seconds or 89,84 ms; speed
3.0 records/sec
TRANSFORM: processed 1 entities (update_merged: 1); 0 with errors; took 1 seconds or 590,7 ms;
speed 1.0 records/sec

-- Обновление исходного, никуда не влитого контрагента + закрытие влитого (и разлиание)
BUFFER: processed 2 entities (update: 2); 0 with errors; took 1 seconds or 51,14 ms; speed 2.0
records/sec
TRANSFORM: processed 3 entities (update_original: 1, close: 1, unmerged: 1); 0 with errors; took 1
seconds or 825,9 ms; speed 3.0 records/sec

```

Загрузка связей тоже логируется, в таком формате:

```

INFO interceptor.TaskPerformerStatsInterceptor - FINISHED task 'PhysicalRelationBufferIPC
[relationUploadTaskPerformer]'; processed 1 entities (create: 1); 0 with errors; took 1 seconds or
503,8 ms; speed 1.0 records/sec
INFO interceptor.TaskPerformerStatsInterceptor - FINISHED task 'PhysicalRelationBufferIPC
[relationUploadTaskPerformer]'; processed 1 entities (close: 1); 0 with errors; took 1 seconds or
670,5 ms; speed 1.0 records/sec

```

7.1.5.2 Задача дошла до конца без ошибок

- АРМ Администратора, статус - *FINISHED*
- АРМ Администратора, комментарий - *ОК*
- Логи

```

INFO transform.TransformTask - Transformed 200 'PHYSICAL' parties from system 'AL'.
Error count: 0
INFO transform.TransformTask - Transformed 400 'PHYSICAL' parties from system 'BT'.
Error count: 0
INFO interceptor.TaskPerformerStatsInterceptor - FINISHED task
'bufferUploadTaskPerformer'; processed 400 entities (update: 0, skip: 0, create: 400, close:
0); 0 errors; took 3 seconds

```

7.1.5.3 Задача дошла до конца, но были ошибки

- АРМ Администратора, статус - *ERROR*
- АРМ Администратора, комментарий:

```

Task finished with 2 errors:
Error while cleansing records with IDs [7028, 7030]: Unexpected cleansed records count
(expected 1, got 0)
Ошибка трансформации
see more in logs

```

- Логи

```

INFO interceptor.TaskPerformerStatsInterceptor - ERROR task 'PhysicalBufferIPC
[bufferUploadTaskPerformer]';
BUFFER: processed 1 entities (create: 1); 0 with errors; took 1 seconds or 19,26 ms; speed
1.0 records/sec

```

```

TRANSFORM: processed 1 entities (error: 1); 1 with errors; took 1 seconds or 192,6 ms;
speed 1.0 records/sec
ERROR manager.TaskWorker - Task 'bufferUploadTaskPerformer' finished with error: Task
finished with 1 errors

```

7.1.5.4 Задача остановилась из-за ошибок

- АРМ Администратора, статус - *ERROR*
- АРМ Администратора, комментарий:

```

Stop task due errors:
Error while cleansing records with IDs [7028, 7030]: Unexpected cleansed records count
(expected 1, got 0)
Ошибка трансформации
see more in logs

```

- Логи

```

INFO transform.TransformTask - Transformed 200 'PHYSICAL' parties from system 'AL'.
Error count: 0
INFO transform.TransformTask - Transformed 400 'PHYSICAL' parties from system 'BT'.
Error count: 0
INFO interceptor.TaskPerformerStatsInterceptor - FINISHED task
'bufferUploadTaskPerformer'; processed 400 entities (update: 0, skip: 0, create: 400, close:
0); 400 errors; took 3 seconds

```

7.1.5.5 Подсчет ошибок

Кол-во ошибок считается:

- для очистки - по каждой записи.
- для трансформации - по каждой записи.
- Ошибки вставки в стейджинг - добавляются к общему кол-ву

7.1.6 Лог триггера

7.1.6.1 Все задачи прошли успешно

В логе есть старт и финиш триггера, а также его шаги в виде [n/m]

```

INFO task.TaskTriggerExecutor - Trigger 'rebuildAllIndexesTrigger' STARTED manually
INFO task.TaskTriggerExecutor - [1/3] step 'graphEngineFullRebuild' of trigger
'rebuildAllIndexesTrigger' STARTED
INFO task.TaskLauncherTemplate - Task graphEngineFullRebuild FINISHED
INFO task.TaskTriggerExecutor - [2/3] step 'enginesFullRebuild' of trigger 'rebuildAllIndexesTrigger'
STARTED
INFO task.TaskLauncherTemplate - Task enginesFullRebuild FINISHED
INFO task.TaskTriggerExecutor - [3/3] step 'duplicateGroupFullRebuild' of trigger
'rebuildAllIndexesTrigger' STARTED
INFO task.TaskLauncherTemplate - Task duplicateGroupFullRebuild FINISHED
INFO task.TaskTriggerExecutor - Trigger 'rebuildAllIndexesTrigger' finished. All task executed
successfully.

```

7.1.6.2 Задача упала, триггер остановился
В логе мы видим количество не запущенных задач:

```
INFO task.TaskTriggerExecutor - Trigger 'dailyIncrementLoadWithDedupTrigger' STOPPED on
errors in task 'bufferUploadTaskPerformer'
INFO task.TaskTriggerExecutor - Trigger 'dailyIncrementLoadWithDedupTrigger' finished. 1 tasks
successful, 1 tasks failed, 4 tasks not started.
```

7.1.6.3 Задача упала, триггер продолжил работу
В логе можно увидеть, что триггер прошел успешно, но N задач прошло успешно, а M упали с ошибкой

```
INFO task.TaskTriggerExecutor - Trigger 'dailyIncrementLoadWithDedupTrigger' finished. 5 tasks
successful, 1 tasks failed.
```

7.1.7 Единый клиент (cdi-soap-stats.log)

В журнале содержится агрегированная информация о вызовах SOAP-методов.

7.1.7.1 Формат:

```
{date} '{time}' {endpoint} {method} call {call_qty} failed {failed_qty} avg,ms {avg_qty}
max,ms {max_qty} p90,ms {p90_qty} p95,ms {p95_qty} p99,ms {p99_qty} min,ms
{min_qty} qps {qps_qty}
```

7.1.7.2 Описание колонок:

Колонка	Значение	Описание
1	date	Дата сбора статистики.
2	time	Время сбора статистики с точностью до минут. Отображается в одинарных кавычках.
3	endpoint	Название endpoint'а. Присутствует только если подключен входящий JMS. По нему определяется тип потока: SOAP/JMS или SOAP/HTTP. Если SOAP/JMS, тогда наименование endpoint'а дополняется информацией о потоке (в примере ниже — JMS-поток от системы CFT, диапазон clientId [80-89]).
4	method	Название метода.
6	call_qty	Число вызовов.
8	failed_qty	Число вызовов, завершившихся с ошибкой.
10	avg_qty	Среднее время обработки успешных вызовов, в мс.
12	max_qty	Максимальное время обработки успешных вызовов, в мс.
14	p90_qty	Время обработки, в которое укладывается 90% успешных вызовов, в мс.

16	p95_qty	Время обработки, в которое укладывается 95% успешных вызовов, в мс.
18	p99_qty	Время обработки, в которое укладывается 99% успешных вызовов, в мс.
20	min_qty	Минимальное время обработки успешных вызовов, в мс.
22	qps_qty	Количество вызовов в секунду.

Колонки 5,7,9,11,13,15,17,19,21 содержат фиксированный текст.

Примечание: Если запись логов происходит не по расписанию, а при закрытии приложения, тогда вместо 21 и 22 колонок будет строка типа: (forced close at 5-03-2018 11:26:52).

7.1.7.3 Пример:

```
2018-03-23 '12:31' soapOverJms_CFT_80_89 save      call 1  fail 0  avg,ms 3050  max,ms
3050  p90,ms 3050  p95,ms 3050  p99,ms 3050  min,ms 3050  qps 0.00
2018-03-23 '13:02' soap_PartyWS      checkBlacklist  call 2  fail 0  avg,ms 1598  max,ms 2689
p90,ms 2689  p95,ms 2689  p99,ms 2689  min,ms 506  qps 0.00
2018-03-23 '13:26' soap_PartyWS      search      call 5  fail 0  avg,ms 5   max,ms 7
p90,ms 7   p95,ms 7   p99,ms 7   min,ms 0
(forced close at 23-03-2018 13:26:52)
```

7.1.7.4 Настройки:

По умолчанию собирается раз в 15 минут.

Для изменения частоты записи требуется изменить параметр `webservice.stats.periodInSeconds` в *APM Администратора – Конфигурация – Параметры SOAP и REST сервисов*.

7.1.8 Трассировка SOAP-запросов

Временно включить журналирование SOAP-запросов:

1. Найти файл `logback.xml` текущей сборки ЕК. Местоположение:

- JBoss: `JBOSS_CDI_HOME\standalone\tmp\vfs\tmp***`(с наибольшей датой актуальности)\`cdi-web-*.war\WEB-INF\classes\logback.xml`
- WildFly 8: `WILDFLY_CDI_HOME\standalone\tmp\vfs\tmp\tmp***`(с наибольшей датой актуальности)\`cdi-web-*.war*\WEB-INF\classes\logback.xml`
- WildFly 10: `WILDFLY_CDI_HOME\standalone\tmp\vfs\tmp\tmp***`(с наибольшей датой актуальности)\`content-***\WEB-INF\classes\logback.xml`

2. Найти настройку `ExtendedLoggingFeature_TRACE`:

```
<!-- Trace SOAP and REST messages [OFF/INFO] -->
<logger name="org.apache.cxf.feature.ExtendedLoggingFeature_TRACE" level="OFF"
additivity="false">
  <appender-ref ref="SOAP_TRACE" />
</logger>
```

3. Изменить `level` с `OFF` на `INFO`:

```

<!-- Trace SOAP and REST messages [OFF/INFO] -->
<logger name="org.apache.cxf.feature.ExtendedLoggingFeature_TRACE" level="INFO"
additivity="false">
  <appender-ref ref="SOAP_TRACE" />
</logger>

```

Перезапуск JBoss / WildFly не требуется, журналирование будет автоматически включено в течение нескольких секунд.

 Журналирование будет автоматически выключено при перезапуске сервиса ЕК.

Включать журналирование на длительное время не рекомендуется, т. к. это снижает быстродействие системы и существенно увеличивает объем журналов.

Чтобы отключить журналирование, достаточно заменить в указанной настройке уровень протоколирования INFO на OFF, либо перезапустить JBoss/WildFly.

7.2 Журналы Фактора

Журналы расположены на сервере приложений в каталоге JBOSS_FACTOR_HOME/standalone/log:

Журнал (8.13+)	Описание	Журнал (8.5+ до 8.13)	Журнал (до 8.5)
server.log	Системные сообщения и сообщения от сервера JBOSS.	server.log	
factor.log	Основной журнал работы Фактора.	factor.log	
factor-blacklist.log	Журнал поиска по черным спискам	factor-blacklist.log	blacklist.log
factor-blacklist-online.log	Журнал онлайн поиска по черным спискам		
factor-blacklist-offline.log	Журнал полного поиска по черным спискам		
factor-stats.log	Журнал статистики вызовов SOAP-методов.	factor-stats.log	factor-soap-stats.log
factor-dedup.log	Журнал поиска дубликатов.	factor-dedup.log	dedup.log
factor-longops.log	Журнал трассировки SOAP-запросов, выполняющихся дольше максимального порога.		
cdi-recode.log	Журнал перекодировок.	cdi-recode.log	recode.log

7.3 Уровни протоколирования

В системе используется 4 уровня протоколирования.

ERROR – сообщения об ошибках;
 WARN – предупреждения и сообщения об ошибках;
 INFO – ошибки, предупреждения и информационные сообщения;
 DEBUG – детальная информация о ходе процесса обработки данных, в т.ч. информационные сообщения, предупреждения и сообщения об ошибках.

Для повседневного использования в штатном режиме рекомендуется уровень протоколирования WARN или INFO.

Для отладки и отслеживания причин возникновения проблем – уровень DEBUG.

7.4 Аудит

Экран аудита предназначен для мониторинга действий, производимых пользователями в системе Единый Клиент.

Активность пользователей

Параметры отбора

Дата: Автор:

Записей: Событие:

ID	Дата	Автор	Событие	Описание
115183	22.03.2018 21:15:13	admin	authentication	Authentication failure. Principal: 'admin'. Cause by: Bad credentials (Source: UI;RemoteAddress: 10.0.18.1)
115182	22.03.2018 20:14:46	system	start	CDI application [CDI Demo 18.3-SNAPSHOT (c075c32b6e59, core be24a49545cc)] [10.0.18.11] started in 99 s.
115181	22.03.2018 20:11:33	system	stop	CDI application [CDI Demo 18.3-SNAPSHOT (8cf3072a91ab, core 982ea85b9699)] [10.0.18.11] stopped.
115180	22.03.2018 19:48:13	admin_performer	findLatestObjectByHID	Запрошена информация по контрагенту, HID '111056'
115179	22.03.2018 19:48:13	admin_performer	findLatestObjectByHID	Запрошена информация по контрагенту, HID '111056'
115178	22.03.2018 18:57:19	admin_performer	findLatestObjectByHID	Запрошена информация по контрагенту, HID '81227'
115177	22.03.2018 18:57:19	admin_performer	findLatestObjectByHID	Запрошена информация по контрагенту, HID '81227'
115176	22.03.2018 18:16:12	admin_performer	findLatestObjectByHID	Запрошена информация по контрагенту, HID '111050'
115175	22.03.2018 18:16:10	admin_performer	findLatestObjectByHID	Запрошена информация по контрагенту, HID '111050'

Администратор может фильтровать события по следующим параметрам:

- **Дата** - выбор по дате наступления события либо по диапазону дат;
- **Автор** - пользователь системы Единый Клиент, произведший действие;
- **Событие** - системное имя произошедшего события.

Также можно указать параметр **Записей** – количество отображаемых событий.

7.4.1 Поля записи журнала

Каждая запись журнала содержит следующие поля:

- **ID** - идентификатор события;
- **Дата** - дата и время наступления события
- **Автор** - пользователь системы Единый Клиент, произведший действие;
- **Событие** - системное имя произошедшего события;
- **Описание** - подробности о событии, информация об ошибках, заполняется в зависимости от конкретного типа события.

В столбце Описание, как правило, указывается уникальный идентификатор объекта, над которым было произведено действие.

7.4.2 Типы событий

В системе определены следующие типы событий:

Обозначение	Событие	Комментарий
start	Запуск приложения	CDI application [version] started.
stop	Остановка приложения	CDI application [version] stopped.
authentication	Неудачные попытки аутентификации в системе	Authentication failure
authorization	Сообщения о нехватке прав для выполнения действия	Запрошен доступ к объекту '{атрибут}'; необходимы права '{роль}'
findObjectByHID	Просмотр карточки контрагента	Запрошена информация по контрагенту, HID '{идентификатор_контрагента}'
findLatestObjectByHID		
getParty		
create	Создание атрибута контрагента	Создание атрибута '{атрибут}', HID контрагента '{идентификатор_контрагента}'
edit	Редактирование контрагента	Редактирование контрагента/атрибута '{атрибут}' HID '{идентификатор_контрагента}'
postpone	Откладывание атрибута или реквизита контрагента	Откладывание реквизита/атрибута '{атрибут}' контрагента, HID '{идентификатор_контрагента}', флаг '{флаг}';
close	Закрытие атрибута контрагента	Закрытие атрибута '{атрибут}', HID '{идентификатор_контрагента}' (HID контрагента '{идентификатор_контрагента}')
delete	Удаление контрагента	Удален исходный/результатирующий контрагент, HID '{идентификатор_контрагента}'
manualRemove	Удаление прошлого значения	Удаление прошлого значения атрибута '{атрибут}', ID значения '{идентификатор_атрибута}'

Обозначение	Событие	Комментарий
	атрибута контрагента	
setRelationStatus	Выставление статуса связи	Выставление статуса связи '{идентификатор_контрагента}', ID связи %s
resolveConflict	Работа с конфликтами	Отклонить обновление/обновить/разлить и обновить контрагента HID '{идентификатор_контрагента}', ID стейджинга '{идентификатор_стейджинга}'
getOriginalWithWinnerFlags	Просмотр исходной карточки контрагента из золотой карточки	Запрошена информация по контрагенту, HID '{идентификатор_контрагента}'
fetchParties	Просмотр карточек контрагентов из группы дубликатов	Запрос информации по контрагентам из дубликата, HID дубликата '{идентификатор_контрагента}'
mergeGroup	Запуск слияния группы дубликатов	Запуск слияния группы дубликатов, HIDы контрагентов '{идентификатор_контрагента}'
performDuplicate	Работа с группами дубликатов	Подтверждение/отклонение/откладывание пары дубликатов, HID '{идентификатор_контрагента}'- '{идентификатор_контрагента}'
performUnmerge	Разъединение контрагентов	Разъединение контрагентов, объединенный HID '{идентификатор_контрагента}', отделяемый HID '{идентификатор_контрагента}'
getDiff	Сравнение контрагентов из групп дубликатов	Запрошена информация по контрагенту, HID '{идентификатор_контрагента}', '{идентификатор_контрагента}'
performRecover	Восстановление контрагента на дату	Восстановление контрагента, HID '{идентификатор_контрагента}' на дату '{дата}'
performRecoverRequisite	Восстановление реквизитов контрагента на дату	Восстановление реквизитов контрагента, HID '{идентификатор_контрагента}' на дату '{дата}'
performRecoverAttribute	Восстановление атрибута контрагента на дату	Восстановление атрибута '{атрибут}', HID '{идентификатор_контрагента}' на дату '{дата}'

Обозначение	Событие	Комментарий
propertiesChange	Изменение свойства конфигурации	Добавлено/изменено/удалено свойство '{свойство}'

8 Установка обновлений

Данная инструкция содержит стандартный порядок действий, выполняемый при обновлении Единого Клиента.

При необходимости выполнения дополнительных действий предоставляется отдельная подробная инструкция по обновлению.

8.1 Общее описание

Перед началом обновления системы необходимо сделать [резервную копию](#).

В случае сбоя следует восстановить систему по [инструкции](#).

Система, как видно из схемы в разделе [Архитектура системы](#), состоит из двух независимых модулей (Единый Клиент, ФАКТОР), работающих под управлением сервера приложений JBOSS AS (JBOSS Application Server), каждый в своём экземпляре. Серверы приложений JBOSS сконфигурированы как системные службы Windows, настроен автоматический старт при загрузке операционной системы. Модули могут быть обновлены независимо один от другого.

Обновление Единого Клиента может повлечь за собой обновление схемы базы данных. В таком случае будет предоставлен набор миграционных скриптов.

Типичный состав обновления:

1. Файл модуля Единый Клиент:
 - cdi-web-{CUSTOMER}-{VERSION}.war
2. Файл модуля ФАКТОР:
 - factor-service-{CUSTOMER}.war
3. Миграционные SQL-скрипты.
4. Сопроводительная документация с описанием обновления (Release Notes).

Любой из пунктов 1-3 может отсутствовать, если обновление соответствующего компонента не производится.

8.2 Обновление модулей

8.2.1 Обновление модуля Единый Клиент

1. Сделать резервные копии предыдущей сборки Единого Клиента cdi-web-{CUSTOMER}-{VERSION}.war и БД Единого Клиента. Резервная копия должна располагаться **вне** директории {JBOSS_CDI_HOME}\standalone\deployments и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис Единого Клиента (HFLABS CDI).

4. Удалить старый war-файл Единого Клиента из директории {JBOSS_CDI_HOME}\standalone\deployments.
5. Скопировать новую сборку Единого Клиента в директорию deployments.
6. Запустить сервис Единого Клиента (HFLABS CDI).
7. Проверить, что в APM Администратора отображается корректная версия продукта (номер версии указан в Release Notes).

8.2.2 Обновление модуля Фактор

1. Сделать резервную копию предыдущей сборки Фактора factor-service-{CUSTOMER}.war. Резервная копия должна располагаться **вне** директории {JBOSS_HOME_FACTOR}\standalone\deployments и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис HFLABS FACTOR.
4. Удалить старый war-файл Фактора из директории {JBOSS_HOME_FACTOR}\standalone\deployments.
5. Скопировать новую сборку Фактора в директорию deployments.
6. Запустить сервис HFLABS FACTOR.
7. Проверить, что в APM Администратора отображается корректная версия Фактора (номер версии указан в Release Notes).

8.3 Ручной запуск и остановка

Если запуск обновлённого модуля в режиме веб-службы завершился неудачно, может понадобиться запустить JBOSS вручную. При этом необходимо удостовериться, что пользователь, с правами которого производится запуск JBOSS, имеет доступ на запись в файлы лога JBOSS.

Запуск осуществляется с помощью bat-файла {JBOSS_HOME}\bin\run.bat (или {JBOSS_HOME}\bin\run.sh в Linux).

Основной лог работы выводится в консоль, для остановки приложения достаточно нажать Ctrl+C.

Для остановки JBOSS можно также воспользоваться bat-файлом

{JBOSS_HOME}\bin\shutdown.bat (или аналогичный sh-файл для Linux) с параметром -S .

Пример:

```
C:\server\jboss-cdi\bin>shutdown.bat -S
```

8.4 Обновление версии Java Runtime Environment

Обновление версии JRE на [рабочих станциях пользователей](#) не обязательно, кроме случая, когда это требуется [Release Notes](#).

Тем не менее, рекомендуется обновлять версию JRE на рабочих станциях пользователей где-то раз в полгода.

8.5 Обновление модуля Единый Клиент

1. Сделать резервные копии предыдущей сборки Единого Клиента `cdi-web-{CUSTOMER}-{VERSION}.war` и БД Единого Клиента. Резервная копия должна располагаться **вне** директории `{JBOSS_CDI_HOME}\standalone\deployments` и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис Единого Клиента (HFLABS CDI).
4. Удалить старый war-файл Единого Клиента из директории `{JBOSS_CDI_HOME}\standalone\deployments`.
5. Скопировать новую сборку Единого Клиента в директорию `deployments`.
6. Запустить сервис Единого Клиента (HFLABS CDI).
7. Проверить, что в АРМ Администратора отображается корректная версия продукта (номер версии указан в Release Notes).

8.6 Обновление модуля Фактор

1. Сделать резервную копию предыдущей сборки Фактора `factor-service-{CUSTOMER}.war`. Резервная копия должна располагаться **вне** директории `{JBOSS_HOME_FACTOR}\standalone\deployments` и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис HFLABS FACTOR.
4. Удалить старый war-файл Фактора из директории `{JBOSS_HOME_FACTOR}\standalone\deployments`.
5. Скопировать новую сборку Фактора в директорию `deployments`.
6. Запустить сервис HFLABS FACTOR.
7. Проверить, что в АРМ Администратора отображается корректная версия Фактора (номер версии указан в Release Notes).

8.7 Автоматическая миграция на новую версию

При переходе на новую версию ЕК автоматически обновляет БД, поисковые индексы и собирает диагностику:

8.7.1 1. При старте ЕК понимает, что нужно мигрировать на новую версию

8.7.2 2. Обновляется БД

ЕК изменяет схему данных, создает новые и удаляет устаревшие объекты.

8.7.3 3. Перестраиваются поисковые индексы

ЕК определяет, какие индексы нужно перестроить.

Запускает перестроение индексов. Список индексов отправляет администратору по почте:

Code Block 3 Тема: Единый клиент обновляется на версию 3.22

Обновление с версии 3.21 на 3.22.

Будут перестроены индексы (GRAPH, PHYSICAL, MANUAL_DUPLICATE_GROUP).

8.7.4 4. Собирается диагностика

ЕК собирает стандартную диагностику (профили mail.diagnostic.defaultProfiles).

Отправляет диагностику администратору по почте:

Code Block 4 Тема: Единый клиент обновился на версию 3.22

Обновление с версии 3.21 на 3.22. Диагностика во вложении (config,db.connection,logs,search,sql.counts,sql.metadata,versions).

8.7.5 Автоматическая миграция на новую версию (техподдержка)

8.7.5.1 Включить автоматическую миграцию

В [параметрах сервисов](#) установить services.applicationMigration.enabled = true.

8.7.5.2 Перестраивать невалидные индексы при миграции

У невалидного индекса не совпадает количество документов с БД. Приложением с таким индексом работает, но индекс неполный.

Прежде чем перестраивать, лучше разобраться в причинах.

Чтобы перестраивать невалидные индексы автоматически, включите [параметр сервиса](#) services.applicationMigration.rebuildInvalidIndexes = true. Тогда после запуска новой версии ЕК придется ждать, пока ЕК проверит все индексы. Это может занять больше часа.

8.7.5.3 Задать индексы для автоматической миграции

8.7.5.3.1 Где настраивается

Общий список — в cdi-web\src\main\resources\task-migration.xml:

```

<!-- Список индексов, которые нужно перестраивать всем при миграции приложения -->
<util:map id="searchEnginesToRebuildOnMigration.core">
  <entry key="3.21-SNAPSHOT">
    <util:list>
      <value>MANUAL_DUPLICATE_GROUP</value>
      <value>LEGAL</value>
    </util:list>
  </entry>
  <entry key="3.20-SNAPSHOT">
    <util:list>
      <value>GRAPH</value>
    </util:list>
  </entry>
</util:map>

```

Чтобы расширить список, добавьте заказчику конфиг `cdi-customer\cdi-web\src\main\resources\task-migration-customer.xml`:

```
<!-- Расширение списка индексов, которые нужно перестраивать при миграции приложения
-->
<util:map id="searchEnginesToRebuildOnMigration.custom">
  <entry key="3.21-SNAPSHOT">
    <util:list>
      <value>AGREEMENT</value>
    </util:list>
  </entry>
</util:map>
```

8.7.5.3.2 Как формируется список индексов для перестроения

Старая версия ЕК, (`{{cdi.root.folder}}/conf/version.properties`), сравнивается в алфавитном порядке с каждым ключом элемента (entry key) из `searchEnginesToRebuildOnMigration`. Все что больше – объединяется в множество и передается перестройщику индекса.

Если на сервере отсутствует файл со старой версией ЕК, то в список добавляются все индексы, указанные в конфигах, без учета версий.

Если включем флаг `services.applicationMigration.enabled = true`, список пополняется невалидными индексами.

8.7.5.4 Настроить оповещение по почте

Задать [параметры отправки email](#), блок *Автоматическая миграция приложения*.

8.7.5.5 Логирование

Обновления версий сервера пишутся в отдельный лог – `cdi-lifecycle.log` (помимо версий туда логируются и события старта/остановки приложения).

```
2017-07-10 16:02:11,841 INFO ApplicationMigrationListener - Application migration from 17.6-SNAPSHOT to 17.7-SNAPSHOT revision 454bef1c338b started
2017-07-10 16:02:30,098 INFO ApplicationMigrationListener - Application migration from 17.6-SNAPSHOT to 17.7-SNAPSHOT revision 454bef1c338b finished
```

В диагностике лог также собирается – каждый для своей ноды.

8.8 Инструкция по переходу с JBoss 4 на JBoss 7 для администраторов

8.8.1 Инструкция для Linux

В данной инструкции предполагается, что CDI запускается от имени пользователя CDI, а Фактор от имени пользователя FACTOR. Старые сервера JBoss находятся в домашних директориях соответствующих пользователей в директориях jboss.

8.8.1.1 Установочный пакет

Установочные пакеты JBoss поставляются совместно с системой в архивах:

- JBoss Единый клиент: jboss-7.1.1.Final-8080.zip
- JBoss Фактор: jboss-7.1.1.Final-18080.zip

8.8.1.2 Создание резервной копии предыдущей инсталляции JBoss

Остановить службы CDI и Фактор, заархивировать предыдущие инсталляции JBoss (CDI и Фактор), после успешной архивации удалить директории, в которых они располагались.

Перед архивированием можно удалить ненужное содержимое из директорий jboss (например: логи, дампы памяти).

```
# service cdi stop
# service factor stop

# zip -rT /home/cdi/old-jboss-cdi /home/cdi/jboss
# zip -rT /home/factor/old-jboss-factor /home/factor/jboss

# rm -r -f /home/cdi/jboss
# rm -r -f /home/factor/jboss
```

8.8.1.3 Установка JBOSS

1. Распаковать архивы с JBoss в каталог /home:

```
# unzip jboss-7.1.1.Final-8080.zip -d /home/cdi
# unzip jboss-7.1.1.Final-18080.zip -d /home/factor

# mv /home/cdi/jboss* /home/cdi/jboss
# mv /home/factor/jboss* /home/factor/jboss
```

2. Убедиться, что у соответствующих пользователей есть права доступа к каталогам JBOSS_CDI_HOME и JBOSS_FACTOR_HOME, в том числе и на редактирование. Самый простой способ сделать это — переназначить владельцев директорий:

```
# chown -R cdi:cdi /home/cdi/jboss
# chown -R factor:factor /home/factor/jboss
```

3. Назначить права на запуск исполняемых файлов:

```
# chmod +x /home/factor/jboss/bin/*.sh
# chmod +x /home/factor/jboss/bin/**/*.sh
# chmod +x /home/cdi/jboss/bin/*.sh
# chmod +x /home/cdi/jboss/bin/**/*.sh
```

8.8.1.4 Настройка сервисов

1. Скопировать файлы:

- bin/init.d/cdi.sh в /etc/init.d/cdi,
- bin/init.d/jboss-as-cdi.conf в /etc/jboss-as/ (если директория jboss-as отсутствует – создать ее).

```
# cp /home/cdi/jboss/bin/init.d/cdi.sh /etc/init.d/cdi
# mkdir /etc/jboss-as
# cp /home/cdi/jboss/bin/init.d/jboss-as-cdi.conf /etc/jboss-as/
```

2. Выполнить аналогичные действия для JBoss, где будет разворачиваться Фактор. Скопировать скрипты запуска и конфигурационный файл:

```
# cp /home/factor/jboss/bin/init.d/factor.sh /etc/init.d/factor
# cp /home/factor/jboss/bin/init.d/jboss-as-factor.conf /etc/jboss-as/
```

8.8.1.5 Настройка datasource

В файле %JBASS_CDI_HOME%/standalone/deployments/cdi-oracle-ds.xml заполнить реквизиты для подключения к БД.

```
[...]
<datasource jndi-name="cdi-dataSource" pool-name="cdi-dataSource" use-java-context="true">
[...]
```

```
<connection-url>jdbc:oracle:thin:@host:port:sid</connection-url>
[...]
```

```
<security>
  <user-name>username</user-name>
  <password>password</password>
</security>
[...]
```

8.8.1.6 Настройка datasource для заказчиков, использующих зашифрованный пароль к БД

Для зашифрования пароля выполнить следующую команду (из директории JBoss CDI):

```
java -cp ./modules/org/picketbox/main/picketbox-4.0.7.Final.jar;./modules/org/jboss/logging/main/jboss-logging-3.1.0.GA.jar org.picketbox.datasource.security.SecureIdentityLoginModule пароль_для_шифрования
```

В файле `standalone/configuration/standalone.xml` Jboss CDI добавить в блок `security-domains` следующий код:

```
<security-domain name="EncryptedPassword">
  <authentication>
    <login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule"
      flag="required">
      <module-option name="username" value="username"/>
      <module-option name="password" value="encrypted_password"/>
    </login-module>
  </authentication>
</security-domain>
```

В файле `standalone/deployments/cdi-oracle-ds.xml` Jboss CDI вместо

```
<user-name>username</user-name>
<password>password</password>
```

ИСПОЛЬЗОВАТЬ

```
<security-domain>EncryptedPassword</security-domain>
```

8.8.2 Инструкция для Windows

Под пользователем `NFL_USER` подразумевается пользователь, от имени которого запускались службы CDI и ФАКТОР на JBoss 4

8.8.2.1 Установочный пакет

Установочные пакеты JBoss поставляются совместно с системой в архивах:

- JBoss Единый клиент: `jboss-7.1.1.Final-8080.zip`
- JBoss Фактор: `jboss-7.1.1.Final-18080.zip`

8.8.2.2 Создание резервной копии предыдущей инсталляции JBoss

Остановить службы CDI и Фактор, заархивировать предыдущие инсталляции JBoss (CDI и Фактор), после успешной архивации удалить директории, в которых они располагались.

8.8.2.3 Установка JBOSS

Распакуйте архивы с JBoss в директорию, в которой они располагались ранее.

8.8.2.4 Установка переменных окружения

Для пользователя HFL_USER удалить следующие переменные окружения, если они существуют (Компьютер -> Свойства -> Дополнительные параметры системы -> Переменные среды):

```
JBOSS_CDI_HOME
JBOSS_FACTOR_HOME
CDI_ROOT_DIR
CDI_CONF_DIR
```

8.8.2.5 Удаление старых системных служб запуска JBOSS 4

Удалите старые системные службы:

```
sc delete {название службы Единого Клиента, обычно "cdi"}
sc delete {название службы Фактор, обычно "factor"}
```

8.8.2.6 Создание системных служб запуска JBOSS 7

Установите системные службы (команду следует выполнять из директории bin соответствующего jboss):

```
cd jboss-7.1.1.Final-8080\bin
service.bat install

cd jboss-7.1.1.Final-18080\bin
service.bat install
```

8.8.2.7 Установка параметра cdi.root.dir

В файл jboss-7.1.1.Final-8080/bin/standalone.conf.bat добавить строку:

```
set "JAVA_OPTS=%JAVA_OPTS% -Dcdi.root.folder={путь до корневой директории cdi}"
```

8.8.2.8 Настройка datasource

В файле %JBOSS_CDI_HOME%/standalone/deployments/cdi-oracle-ds.xml заполнить реквизиты для подключения к БД.

```
[...]
<datasource jndi-name="cdi-datasource" pool-name="cdi-datasource" use-java-context="true">
[...]
<connection-url>jdbc:oracle:thin:@host:port:sid</connection-url>
[...]
<security>
  <user-name>username</user-name>
  <password>password</password>
</security>
[...]
```

8.8.2.9 Настройка datasource для заказчиков, использующих зашифрованный пароль к БД

Для зашифрования пароля выполнить следующую команду (из директории JBoss CDI):

```
java -cp ./modules/org/picketbox/main/picketbox-
4.0.7.Final.jar;./modules/org/jboss/logging/main/jboss-logging-3.1.0.GA.jar
org.picketbox.datasource.security.SecureIdentityLoginModule пароль_для_шифрования
```

В файле `standalone/configuration/standalone.xml` Jboss CDI добавить в блок `security-domains` следующий код:

```
<security-domain name="EncryptedPassword">
  <authentication>
    <login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule"
flag="required">
      <module-option name="username" value="username"/>
      <module-option name="password" value="encrypted_password"/>
    </login-module>
  </authentication>
</security-domain>
```

В файле `standalone/deployments/cdi-oracle-ds.xml` Jboss CDI вместо

```
<user-name>username</user-name>
<password>password</password>
```

ИСПОЛЬЗОВАТЬ

```
<security-domain>EncryptedPassword</security-domain>
```

8.8.3 Настройка datasource

8.8.3.1 Настройка datasource

В файле `%JBOSS_CDI_HOME%/standalone/deployments/cdi-oracle-ds.xml` заполнить реквизиты для подключения к БД.

```
[...]
<datasource jndi-name="cdi-datasource" pool-name="cdi-datasource" use-java-context="true">
[...]
```

```
<connection-url>jdbc:oracle:thin:@host:port:sid</connection-url>
[...]
```

```
<security>
  <user-name>username</user-name>
  <password>password</password>
</security>
[...]
```

8.8.3.2 Настройка datasource для заказчиков, использующих зашифрованный пароль к БД

Для зашифрования пароля выполнить следующую команду (из директории JBoss CDI):

```
java -cp ./modules/org/picketbox/main/picketbox-4.0.7.Final.jar;./modules/org/jboss/logging/main/jboss-logging-3.1.0.GA.jar org.picketbox.datasource.security.SecureIdentityLoginModule пароль_для_шифрования
```

В файле `standalone/configuration/standalone.xml` Jboss CDI добавить в блок `security-domains` следующий код:

```
<security-domain name="EncryptedPassword">
  <authentication>
    <login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule"
      flag="required">
      <module-option name="username" value="username"/>
      <module-option name="password" value="encrypted_password"/>
    </login-module>
  </authentication>
</security-domain>
```

В файле `standalone/deployments/cdi-oracle-ds.xml` Jboss CDI вместо

```
<user-name>username</user-name>
<password>password</password>
```

ИСПОЛЬЗОВАТЬ

```
<security-domain>EncryptedPassword</security-domain>
```

8.9 Инструкция по переходу с JBoss 7 на WildFly 8 для администраторов

8.9.1 Инструкция для Linux (переход на WildFly 8)

В данной инструкции предполагается, что CDI запускается от имени пользователя `cdi`, а Фактор от имени пользователя `factor`. Старые сервера JBoss находятся в домашних директориях соответствующих пользователей в директориях `jboss`.

8.9.1.1 Установочный пакет

Установочные пакеты WildFly (новое название сервера приложений) поставляются совместно с системой в архивах:

`wildfly-8.2.0.Final-8080.zip` — для «Единого клиента».

`wildfly-8.2.0.Final-18080.zip` — для «Фактора».

8.9.1.2 Создание резервной копии предыдущей инсталляции JBoss

1. Остановить службы cdi и factor.
2. Удалить ненужное содержимое из директорий jboss (например: логи, дампы памяти).
3. Заархивировать предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

```
# service cdi stop
# service factor stop

# zip -rT /home/cdi/old-jboss-cdi /home/cdi/jboss
# zip -rT /home/factor/old-jboss-factor /home/factor/jboss

# mv /home/cdi/jboss* /home/cdi/jboss_old
# mv /home/factor/jboss* /home/factor/jboss_old
```

8.9.1.3 Установка WildFly

1. Распаковать архивы с WildFly в каталог /home:

```
# unzip wildfly-8.2.0.Final-8080.zip -d /home/cdi
# unzip wildfly-8.2.0.Final-18080.zip -d /home/factor

# mv /home/cdi/wildfly* /home/cdi/jboss
# mv /home/factor/wildfly* /home/factor/jboss
```

2. Убедиться, что у соответствующих пользователей есть права доступа к каталогам jboss «Единого клиента» и jboss «Фактора», в том числе на редактирование. Самый простой способ сделать это – переназначить владельцев директорий:

```
# chown -R cdi:cdi /home/cdi/jboss
# chown -R factor:factor /home/factor/jboss
```

3. Назначить права на запуск исполняемых файлов:

```
# chmod +x /home/factor/jboss/bin/*.sh
# chmod +x /home/factor/jboss/bin/**/*.sh
# chmod +x /home/cdi/jboss/bin/*.sh
# chmod +x /home/cdi/jboss/bin/**/*.sh
```

8.9.1.4 Настройка сервисов

1. Скопировать файлы:

bin/init.d/cdi.sh в /etc/init.d/cdi,
bin/init.d/jboss-as-cdi.conf в /etc/jboss-as/

```
# cp /home/cdi/jboss/bin/init.d/cdi.sh /etc/init.d/cdi
# cp /home/cdi/jboss/bin/init.d/jboss-as-cdi.conf /etc/jboss-as/
```

2. Выполнить аналогичные действия для JBoss, где будет разворачиваться «Фактор». Скопировать скрипты запуска и конфигурационный файл:

```
# cp /home/factor/jboss/bin/init.d/factor.sh /etc/init.d/factor
# cp /home/factor/jboss/bin/init.d/jboss-as-factor.conf /etc/jboss-as/
```

8.9.1.5 Настройка параметров доступа к БД системы

Скопировать файл `cdi-oracle-ds.xml` или `cdi-mariadb-ds.xml` из инсталляции JBoss для «Единого клиента» в директорию `/home/cdi/jboss/standalone/deployments/`

Оракл

```
# cp /home/cdi/jboss_old/standalone/deployments/cdi-oracle-ds.xml
/home/cdi/jboss/standalone/deployments
```

Mariadb

```
# cp /home/cdi/jboss_old/standalone/deployments/cdi-mariadb-ds.xml
/home/cdi/jboss/standalone/deployments
```

При необходимости настроить [шифрование пароля к БД](#).

8.9.1.6 Настройка параметров запуска JBoss

1. Скопировать файл `standalone.conf` из предыдущей инсталляции JBoss для «Единого клиента» и «Фактора».

```
# cp /home/cdi/jboss_old/bin/standalone.conf /home/cdi/jboss/bin/
# cp /home/factor/jboss_old/bin/standalone.conf /home/factor/jboss/bin/
```

2. Отредактировать `standalone.conf` «Единого клиента»:
- Удалить параметр `-ea`.
 - Проверить параметры `-Xms` и `-Xmx` — должно быть не менее `6g`.
3. Проверить параметры `-Xms` и `-Xmx` в `standalone.conf` «Фактора» — должно быть не менее `8g`.

8.9.1.7 Настройка горячего резерва (при необходимости)

Отредактировать файл `/home/cdi/jboss/standalone/configuration/standalone.xml`.

Внести изменения в раздел `socket-binding-group`:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-
offset="{jboss.socket.binding.port-offset:0}">
[...]
```



```
<outbound-socket-binding name="node-sync-binding">
  <remote-destination host="{доменное имя второго экземпляра ЕК}" port="{http порт
второго экземпляра ЕК}"/>
</outbound-socket-binding>
</socket-binding-group>
```

Примечание: в параметре port элемента remote-destination учитывается смещение порта. Обычно он равен 8080, задается так:

```
<outbound-socket-binding name="node-sync-binding">
  <remote-destination host="{доменное имя второго экземпляра ЕК}"
port="{${jboss.http.port:8080}"/>
</outbound-socket-binding>
```

Но если на втором экземпляре ЕК настроено смещение, т. е. запускается с параметром напр. -Djboss.socket.binding.port-offset=8, то для текущего экземпляра он будет доступен как:

```
<outbound-socket-binding name="node-sync-binding">
  <remote-destination host="{доменное имя второго экземпляра ЕК}"
port="{${jboss.http.port:8088}"/>
</outbound-socket-binding>
```

8.9.1.8 Удаление установленного JBoss

Удалить директории, в которых располагались предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

```
# rm -r -f /home/cdi/jboss_old
# rm -r -f /home/factor/jboss_old
```

8.9.2 Инструкция для Windows (переход на WildFly 8)

8.9.2.1 Установочный пакет

Установочные пакеты WildFly (новое название сервера приложений) поставляются совместно с системой в архивах:

wildfly-8.2.0.Final-8080.zip — для «Единого клиента».

wildfly-8.2.0.Final-18080.zip — для «Фактора».

8.9.2.2 Создание резервной копии предыдущей инсталляции JBoss

1. Остановить службы CDI и Фактор.
2. Заархивировать предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

8.9.2.3 Удаление старых системных служб запуска JBoss

```
sc delete cdi
sc delete factor
```

8.9.2.4 Установка WildFly

Распаковать архивы с WildFly в директорию, где ранее лежал JBoss.

8.9.2.5 Создание системных служб запуска WildFly

Установить системные службы (команду следует выполнять из директории \bin\service соответствующего WildFly):

```
cd wildfly-8.2.0.Final-8080\bin\service
service.bat install
```

```
cd ..\..\..\wildfly-8.2.0.Final-18080\bin\service
service.bat install
```

Примечание. Если возникает ошибка " ERROR: The service already exists ", то перед установкой системных служб следует перезагрузить ОС.

8.9.2.6 Настройка параметров доступа к БД системы

Скопировать файл cdi-oracle-ds.xml или cdi-mariadb-ds.xml из инсталляции JBoss для «Единого клиента» в директорию wildfly-8.2.0.Final-8080\standalone\deployments. При необходимости настроить [шифрование пароля к БД](#).

8.9.2.7 Настройка параметров запуска WildFly

1. В директорию wildfly-8.2.0.Final-8080\bin скопировать файл standalone.conf.bat из предыдущей инсталляции JBoss для «Единого клиента».
2. Отредактировать standalone.conf.bat:
 - a. Удалить параметр -ea.
 - b. Проверить параметры -Xms и -Xmx — должно быть не менее 6g.
3. В директорию wildfly-8.2.0.Final-18080\bin скопировать файл standalone.conf.bat из предыдущей инсталляции JBoss для «Фактора».
4. Проверить параметры -Xms и -Xmx — должно быть не менее 8g.

8.9.2.8 Удаление установленного JBoss

Удалить директории, в которых располагались предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

8.10 Инструкция по переносу дампа БД

8.10.1 Экспорт

1. [Найти подходящую oracle-директорию или создать новую](#)
2. Запустить команду для экспорта:

```
<ORACLE_HOME>/bin/expdp <USERNAME>/<PASSWORD>
FLASHBACK_TIME=systimestamp SCHEMAS=<schema_name>
directory=<ORACLE_DIRECTORY(из шага 1)> dumpfile=<Имя файла дампа>.DMP
logfile=<Любое имя для лога процедуры экспорта>.log
```

где

- <ORACLE_HOME> — путь, куда установлен Oracle на сервере, откуда экспортируем БД

- FLASHBACK_TIME=systimestamp – параметр, отвечающий за консистентность полученного дампа на конкретный момент времени (требуется, если во время дампа БД может меняться)
- <USERNAME>/<PASSWORD> – имя пользователя для EK (например, CDI_TEST) и его пароль.
Не рекомендуется использовать для этих целей пользователя с правами sysDBA
- <ORACLE_DIRECTORY> – название oracle-директории из шага 1.

После успешного экспорта в директории появится файл дампа схемы БД и лог.

8.10.2 Импорт

1. Пересоздать пользователя в БД, куда необходимо импортировать дампы:
 - a. DROP USER <username> CASCADE
 - b. [Создать пользователя БД](#)
2. [Найти подходящую oracle-директорию или создать новую](#)
3. Скопировать файл дампа в найденную или созданную на шаге 2 директорию.
4. Переключиться в директорию с дампом:
 - a. на Linux дополнительно необходимо переключиться в пользователя oracle:

```
cd <path_to_directory>
su - oracle
```

- b. На Windows достаточно обладать правами администратора.
5. Запустить команду для импорта:

```
<ORACLE_HOME>/bin/impdp <USERNAME>/<PASSWORD>
directory=<ORACLE_DIRECTORY(из шага 2)> dumpfile=<Имя файла дампа>.DMP
logfile=<Любое имя для лога процедуры импорта>.log remap_schema=<OLD_SCHEMA
(как в дампе):<NEW_SCHEMA> remap_tablespace=<OLD_SCHEMA (как в
дампе):<NEW_SCHEMA>,<OLD_SCHEMA (как в дампе)>_idx:<NEW_SCHEMA>_idx
table_exists_action=REPLACE
```

где

- <ORACLE_HOME> - путь, куда установлен Oracle на сервере.
 - <USERNAME>/<PASSWORD> - имя пользователя, обладающего таким же набором прав, как и пользователь, из-под которого создавался дамп БД и его пароль.
 - <ORACLE_DIRECTORY> - название oracle-директории, из шага 2.
 - <OLD_SCHEMA> - название схемы, которое было в старой БД (откуда экспортировали дамп)
 - <NEW_SCHEMA> - название схемы в БД, куда импортируется дамп
6. После успешного окончания импорта.

- a. Выполнить скрипт `<CDI_mercurial_root>\cdi-services\src\main\schema\schema_oracle_tuning.sql`, предварительно заменив в нем `"${jdbc.username}"` на имя пользователя БД.
 - b. [Собрать статистику по схеме в целом](#)
 - c. [Перестроить все индексы в схеме](#)
7. Чтобы ЕК корректно работал с импортированной схемой, необходимо также перестроить граф связей и поисковый индекс (запустить последовательно задачи `graphEngineFullRebuild` и `enginesFullRebuild` в веб-интерфейсе администратора ЕК).

8.10.3 Создание oracle-директории

1. Поиск подходящей директории:

```
SELECT owner, directory_name, directory_path FROM all_directories;
```

Необходимо, чтобы у пользователя, из-под которого будет выполняться экспорт/импорт, были права на чтение и запись в найденную директорию. В противном случае при старте задачи будет показана ошибка:

```
Connected to: Oracle Database 11g Release 11.2.0.3.0 - 64bit Production
ORA-39002: invalid operation
ORA-39070: Unable to open the log file.
ORA-39087: directory name DUMPDIR is invalid
```

2. Создание oracle-директории:

- a. Создать директорию в файловой системе, куда будет скопирован дамп
- b. В случае linux-сервера после создания директории необходимо сделать владельцем директории пользователя "oracle":

```
chown oracle <path_to_directory>
```

- c. Подключиться к БД с ролью sysDBA (пользователь system) любой программой для выполнения запросов (SQL Developer, sqlplus, sql workbench)
- d. Выполнить скрипт:

```
CREATE OR REPLACE DIRECTORY <directory_name> AS
'<path_to_directory>';
GRANT WRITE ON DIRECTORY <directory_name> TO <username>;
GRANT READ ON DIRECTORY <directory_name> TO <username>;
```

где

- `<directory_name>` - любое имя директории, например, "EXPORT_DIR"
- `<path_to_directory>` - полный путь до директории, созданной на шаге 2а, например, "C:/tmp/export"
- `<username>` - имя пользователя для ЕК (например, CDI_TEST)

Если дампы с прода подняли на тесте, то необходимо в веб-интерфейсе в "Конфигурация" в пункте "Параметры отправки email" изменить настройки mail.from и mail.logging.subject.single на настройки для тестового ЕК

8.11 Автоматическое создание схемы CDI

8.11.1 Настройка WildFly

В конфигурационном файле standalone.conf для Linux-серверов или standalone.conf.bat (Windows-сервера) указать параметры, задающие логин и пароль пользователя с правами SYSDBA, а также connection-string БД.

Windows:

Code Block 5 standalone.conf.bat

```
:: Init Database
set "JAVA_OPTS=%JAVA_OPTS% -Djdbc.dbo.username=dbaAdminName -
Djdbc.dbo.password=dbaAdminPass -Djdbc.url=jdbc:oracle:thin:@server:port:sid"
```

Linux:

Code Block 6 standalone.conf

```
# Init Database
JAVA_OPTS="$JAVA_OPTS -Djdbc.dbo.username=dbaAdminName -
Djdbc.dbo.password=dbaAdminPass -Djdbc.url=jdbc:oracle:thin:@server:port:sid"
```

8.11.2 Создание новой схемы

Запустить WildFly с параметром -Djdbc.create.

Windows:

```
standalone.bat -Djdbc.create
```

Linux:

```
sh standalone.sh -Djdbc.create
```

Схема создается с учетом скриптов миграции. Например, «17.12.1.demo». Это позволяет добавлять скрипты автомиграции в текущую сборку (see [CDI-4417](#))

8.11.3 Пересоздание существующей схемы

Запустить WildFly с параметром -Djdbc.recreate.

Windows:

```
standalone.bat -Djdbc.recreate
```

Linux:

```
sh standalone.sh -Djdbc.recreate
```

8.12 Инструкция по переходу с Wildfly 8 на WildFly 10 для администраторов

8.12.1 Инструкция для Linux (переход на WildFly 10)

В данной инструкции предполагается, что CDI запускается от имени пользователя cdi, а Фактор от имени пользователя factor. Старые сервера JBoss находятся в домашних директориях соответствующих пользователей в директориях jboss.

8.12.1.1 Установочный пакет

Установочные пакеты WildFly поставляются совместно с системой в архивах:

wildfly-10.1.0.Final-8080.zip — для «Единого клиента».

wildfly-10.1.0.Final-8080-HotReserve.zip — для «Единого клиента» с горячим резервированием.

wildfly-10.1.0.Final-18080.zip — для «Фактора».

8.12.1.2 Создание резервной копии предыдущей инсталляции JBoss

1. Остановить службы cdi и factor.
2. Удалить ненужное содержимое из директорий jboss (например: логи, дампы памяти).
3. Заархивировать предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

```
service cdi stop  
service factor stop
```

```
zip -rT /home/cdi/old-jboss-cdi /home/cdi/jboss  
zip -rT /home/factor/old-jboss-factor /home/factor/jboss
```

```
mv /home/cdi/jboss /home/cdi/jboss_8  
mv /home/factor/jboss /home/factor/jboss_8
```

8.12.1.3 Установка WildFly

1. Распаковать архивы с WildFly в каталог /home:

```
unzip wildfly-10.1.0.Final-8080.zip -d /home/cdi
unzip wildfly-10.1.0.Final-18080.zip -d /home/factor

mv /home/cdi/wildfly* /home/cdi/jboss
mv /home/factor/wildfly* /home/factor/jboss
```

2. Убедиться, что у соответствующих пользователей есть права доступа к каталогам jboss «Единого клиента» и jboss «Фактора», в том числе на редактирование. Самый простой способ сделать это — переназначить владельцев директорий:

```
chown -R cdi:cdi /home/cdi/jboss
chown -R factor:factor /home/factor/jboss
```

3. Назначить права на запуск исполняемых файлов:

```
find /home/{factor,cdi}/jboss/ -type d -exec chmod 755 {} \;
find /home/{factor,cdi}/jboss/ -type f -exec chmod 644 {} \;
find /home/{factor,cdi}/jboss/ -type f -name "*.sh" -exec chmod 755 {} \;
```

8.12.1.4 Настройка параметров доступа к БД системы

Скопировать файл cdi-oracle-ds.xml или cdi-mariadb-ds.xml из инсталляции JBoss для «Единого клиента» в директорию /home/cdi/jboss/ standalone/deployments/

Oracle

```
cp /home/cdi/jboss_8/standalone/deployments/cdi-oracle-ds.xml
/home/cdi/jboss/standalone/deployments
```

Mariadb

```
cp /home/cdi/jboss_8/standalone/deployments/cdi-mariadb-ds.xml
/home/cdi/jboss/standalone/deployments
```

При необходимости настроить [шифрование пароля к БД](#).

8.12.1.5 Настройка параметров запуска JBoss

1. Скопировать файл standalone.conf из предыдущей инсталляции JBoss для «Единого клиента» и «Фактора».

```
cp /home/cdi/jboss_8/bin/standalone.conf /home/cdi/jboss/bin/
cp /home/factor/jboss_8/bin/standalone.conf /home/factor/jboss/bin/
```

2. Добавить параметры запуска в каждый из них:

```
# Management blocking timeout (for long deployments)
JAVA_OPTS="$JAVA_OPTS -Djboss.as.management.blocking.timeout=3600 -
Djava.net.preferIPv4Stack=true"
```

8.12.1.6 Опциональные настройки

8.12.1.6.1 Горячий резерв

Отредактировать файл `/home/cdi/jboss/standalone/configuration/standalone.xml`.

Внести изменения в раздел `socket-binding-group`:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-
offset="${jboss.socket.binding.port-offset:0}">
  [...]
  <outbound-socket-binding name="node-sync-binding">
    <remote-destination host="{доменное имя второго экземпляра EK}" port="{http порт
второго экземпляра EK}"/>
  </outbound-socket-binding>
</socket-binding-group>
```

Примечание: в параметре `port` элемента `remote-destination` учитывается смещение порта. Если на втором экземпляре EK настроено смещение и он запускается с параметром напр. `-Djboss.socket.binding.port-offset=8`, то для текущего экземпляра он будет доступен как:

```
<outbound-socket-binding name="node-sync-binding">
  <remote-destination host="{доменное имя второго экземпляра EK}" port="8088"/>
</outbound-socket-binding>
```

8.12.1.6.2 Безопасное соединение (SSL)

Перенести [настройки доступа к хранилищу ключей](#) из `/home/cdi/jboss_8/standalone/configuration/standalone.xml` в файл `/home/cdi/jboss/standalone/configuration/standalone.xml`.

В новый конфигурационный файл

`/home/cdi/jboss/standalone/configuration/standalone.xml` скопировать блок `security-realm name="HttpsSecuredRealm"`, удостовериться, что для `https`-коннектора установлен именно этот `security-realm`:

```
<management>
  <security-realms>
    ...
    <security-realm name="HttpsSecuredRealm">
      <server-identities>
        <ssl>
          <keystore path="keystore.jks" relative-to="jboss.server.config.dir" keystore-
password="qwerty" alias="selfsigned"/>
        </ssl>
      </server-identities>
    </security-realm>
  </security-realms>
```



```

</management>
...
<subsystem xmlns="urn:jboss:domain:undertow:3.1">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" socket-binding="http"/>
    <https-listener name="https" socket-binding="https" security-realm="HttpsSecuredRealm"/>
    ...
  </server>
  ...
</subsystem>

```

Скопировать в директорию /home/cdi/jboss/standalone/configuration/ файлы *.jks, на которые ссылается конфигурационный файл.

8.12.1.6.3 Доступ к внешней БД для Фактора

Скопировать datasource из предыдущей инсталляции:

```

cp /home/factor/jboss_8/standalone/deployments/factor-ds.xml
/home/factor/jboss/standalone/deployments

```

8.12.1.7 Удаление предыдущей инсталляции JBoss

Удалить директории, в которых располагались предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

```

rm -rf /home/cdi/jboss_8
rm -rf /home/factor/jboss_8

```

8.12.2 Инструкция для Windows (переход на WildFly 10)

8.12.2.1 Установочный пакет

Установочные пакеты WildFly поставляются совместно с системой в архивах:

wildfly-10.1.0.Final-8080.zip — для «Единого клиента».

wildfly-10.1.0.Final-8080-HotReserve.zip — для «Единого клиента» с горячим резервированием.

wildfly-10.1.0.Final-18080.zip — для «Фактора».

8.12.2.2 Создание резервной копии предыдущей инсталляции WildFly

1. Остановить службы CDI и Фактор.
2. Заархивировать предыдущие инсталляции WildFly для «Единого клиента» и «Фактора».

8.12.2.3 Удаление старых системных служб запуска WildFly

```

sc delete cdi
sc delete factor

```

8.12.2.4 Установка WildFly

Распаковать архивы с WildFly в директорию, где ранее лежал WildFly 8.

8.12.2.5 Создание системных служб запуска WildFly

Установить системные службы (команду следует выполнять из директории \bin\service соответствующего WildFly):

```
cd wildfly-10.1.0.Final-8080\bin\service
service.bat install
cd ..\..\..\wildfly-10.1.0.Final-18080\bin\service
service.bat install
```

Примечание. Если возникает ошибка "ERROR: The service already exists ", то перед установкой системных служб следует перезагрузить ОС.

8.12.2.6 Настройка параметров доступа к БД системы

Скопировать файл cdi-oracle-ds.xml или cdi-mariadb-ds.xml из инсталляции WildFly для «Единого клиента» в директорию wildfly-10.1.0.Final-8080\standalone\deployments. При необходимости настроить [шифрование пароля к БД](#).

8.12.2.7 Настройка параметров запуска WildFly

1. В директорию wildfly-10.1.0.Final-8080\bin скопировать файл standalone.conf.bat из предыдущей инсталляции WildFly для «Единого клиента».
2. В директорию wildfly-10.1.0.Final-18080\bin скопировать файл standalone.conf.bat из предыдущей инсталляции WildFly для «Фактора».
3. Добавить параметры запуска в каждый из них:

```
: Management blocking timeout (for long deployments)
set "JAVA_OPTS=%JAVA_OPTS% -Djboss.as.management.blocking.timeout=3600 -
Djava.net.preferIPv4Stack=true"
```

8.12.2.8 Настройка горячего резерва (при необходимости)

Отредактировать файл wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml.

Внести изменения в раздел socket-binding-group:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-
offset="{jboss.socket.binding.port-offset:0}">
  [...]
  <outbound-socket-binding name="node-sync-binding">
    <remote-destination host="{доменное имя второго экземпляра ЕК}" port="{http порт
второго экземпляра ЕК}"/>
  </outbound-socket-binding>
</socket-binding-group>
```

Примечание: в параметре port элемента remote-destination учитывается смещение порта. Если на втором экземпляре ЕК настроено смещение и он запускается с параметром напр. -Djboss.socket.binding.port-offset=8, то для текущего экземпляра он будет доступен как:

```
<outbound-socket-binding name="node-sync-binding">  
  <remote-destination host="{доменное имя второго экземпляра EK}" port="8088"/>  
</outbound-socket-binding>
```

8.12.2.9 Удаление предыдущей инсталляции WildFly

Удалить директории, в которых располагались предыдущие инсталляции WildFly для «Единого клиента» и «Фактора».

9 FAQ

9.1 Основное (FAQ)

9.1.1 CPU под 100% на задаче

Q: От мониторинга пришло сообщение, что CPU выжрано под 100%. А через час-другой его отпускает (когда завершается задача поиска дублей, например)

A: Можно попробовать изменить параметр poolSize:

1. Узнать сколько ядер на сервере (обычно количество потоков выставляется по количеству ядер.).
2. Узнать, сколько тредов работает (можно посмотреть по логам. В cdi.log пишется, сколько тредов запущено).
3. Уменьшить на четверть-треть.
4. Проверить нагрузку

Для уменьшения количества тредов:

1. Нажать "Редактировать" у выбранной задачи.
2. Изменить значение параметра poolSize. Если его нет — добавить параметр
3. Сохранить изменения.
4. Запустить задачу

9.1.2 Индекс неактуальный (Index may be corrupted)

Q: Автоматический мониторинг логов сообщает об ошибке:

```
... Invalid 'PhysicalParty' index document count (expected 2, got 0). Index may be corrupted. Please rebuild ASAP.
```

A: Поисковый индекс неактуален. Пользователи могут видеть неактуальные данные, во время слияния возможны ошибки.

В АРМ Администратора обновите индексы:

1. В Едином клиенте — [rebuildAllIndexesTrigger](#) на закладке [Триггеры](#).
2. В Факторе — [Поиск дубликатов](#) или [Поиск дубликатов и слияние](#) на закладке [Задачи](#).

9.1.3 Как переместить прод с одной машины на другую

1. Запустить машину, выделенную для переноса прод-сервера.
2. Настроить ОС по [инструкции](#).
3. Выполнить перезагрузку после изменения параметров ОС.
4. Остановить сервисы ЕК и Фактор.
5. Скопировать файлы cdi-oracle-ds.xml и cdi-web-alphas-17.6-SNAPSHOT.war со старой машины (действующий прод) из папки /opt/cdi/jboss/standalone/deployments на новую в соответствующую папку.

6. Скопировать файл factor-service-alphas.war со старой машины из папки /opt/factor/jboss/standalone/deployments на новую в соответствующую папку.
7. Скопировать standalone.conf ЕК и Фактора со старой машины на новую.
8. Удалить индекс и кэш фактора с новой машины: удалить папку /opt/cdi/search и /opt/factor/jboss/standalone/data/dedup-data.ws.
9. Остановить онлайн поток запросов save, saveAndMerge, close, closeAttribute в ЕК на проде (дождаться, пока в очереди не останется сообщений).
10. Выполнить команду "sudo fsync".
11. Скопировать индексы и кеш фактора на новую машину:
 - Скопировать папку /opt/cdi/search, задать владельцем пользователя cdi_user и раздать на нее права RWX.
 - Скопировать папку /opt/factor/jboss/standalone/data/dedup-data.ws, задать владельцем пользователя cdi_user и раздать на нее права RWX.
12. Запустить службы CDI и Factor на новой машине.
13. Подождать 5 минут, чтобы службы ЕК и Фактор полностью запустились и могли принимать онлайн-поток.
14. Остановить продуктовую машину.
15. Подменить имя и IP адреса машины.
16. Выполнить рестарт новой машины.

9.1.4 Как поднять бекап базы с PROD на TEST стенде?

Q: Хотим развернуть бекап базы с PROD на тестовом стенде. Как правильно сделать восстановление из бекапа, чтобы система корректно работала?

A: Для этого необходимо:

1. Остановить сервис Единого клиента.
2. Развернуть бекап, **исключив указанные ниже таблицы**
3. **Удалить старый поисковый индекс.**
4. Запустить сервис Единого клиента.
5. В APM Администратора перейти за закладку [Триггеры](#) и выполнить триггер [afterDatabaseBackupTrigger](#).

Список таблиц, которые **нельзя** восстанавливать из бэкапа на TEST (в них содержатся PROD настройки)

- CONFIG_PROPERTY
- TASK_PARAMETER
- TASK_LOCK
- JMS_EVENT

9.1.5 Что такое "схлопывание" атрибутов?



Q: В правилах слияния атрибутов написано, что они могут "схлопываться" во время создания и обновления клиента. Что это значит?



A: Когда клиент создается в системе, ко всем его атрибутам применяют правила слияния. Если какие-то атрибуты оказываются идентичными с точки зрения правил слияния, то они "схлопываются" — из нескольких одинаковых атрибутов остается видимым только один, остальные скрываются. Это помогает избежать дублирования информации.

Скрытый атрибут



- помечается в БД специальным флагом `hidden=1`;
- не отображается в АРМ Менеджера данных;
- не передается через SOAP;
- передается в представлениях с признаком `is_deleted=1`.


Например, создается клиент с двумя одинаковыми телефонами. Если бы схлопывания не было, то в АРМ Менеджера данных были бы видны дубли

Автор	Тип	Информация	Код качества
GM:547	Домашний	74957896655	
GM:547	Домашний	74957896655	

Но сейчас мы увидим только один телефон

Автор	Тип	Информация	Код качества
GM:547	Домашний	74957896655	

Второй все равно будет в системе, не закрыт, но отмечен флагом `hidden`.

На скрытый атрибут можно присылать обновление. В таком случае атрибуты снова пересливаются и флаг `hidden` перерасчитывается, что дает возможность снова сделать атрибут *видимым* при его изменении.

Например, был контрагент с двумя одинаковыми телефонами, первый из которых оказался скрыт по правилам слияния. На скрытый телефон приходит обновление, и он изменяется на другой. Таким образом, после обновления у контрагента оказывается два разных телефона с точки зрения правил слияния, и ранее скрытый телефон становится видимым (флаг `hidden=0`).

9.2 Менеджер Данных (FAQ)

9.2.1 NullPointerException при попытке просмотреть историю в CDM

9.2.1.1 Ошибка

9.2.1.2

9.2.1.3 Возможные причины

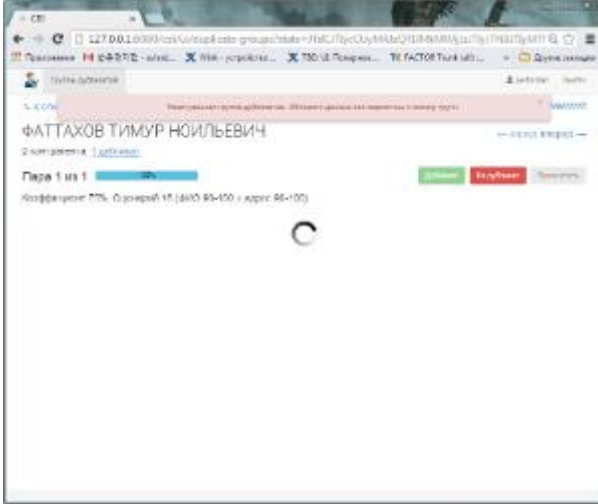
- Ожидаемое поведение для заказчиков, у которых отключено хранение blob-ов истории [CDI-4572](#) - Отключить хранение blob-ов в таблице истории **ЗАКРЫТ**.

9.2.1.4 Что делать

- Смириться и проверить, что для заказчика спрятана ссылка на CDM.

9.2.2 Веб-CDM: ошибка "Неактуальная группа дубликатов. Обновите данные или вернитесь к списку групп."

9.2.2.1 Ошибка



9.2.2.2 Возможные причины

- Группы дубликатов формируются из пачек по 2 млн (задача `duplicateGroupFullRebuild`), поэтому на большой базе иногда один контрагент может попасть в разные группы, транзитивные между собой. После объединения одной из групп, другие перестают быть актуальными и возникает ошибка "Неактуальная группа дубликатов. Обновите данные или вернитесь к списку групп."

9.2.2.3 Что делать

- **Лечится только перестроением индексов, задача `duplicateGroupFullRebuild`.**
-

9.2.3 Другие ошибки при работе в АРМ Менеджер данных

9.2.3.1 Возможные причины

- Недоступность сервисов.
- Системная ошибка.

9.2.3.2 Что делать

1. Прочитать сообщение об ошибке в АРМ Менеджер данных.
2. Попытаться найти корневую причину ошибки в журналах `cdm.log` и `cdi.log`.
3. Отправить скриншот и журналы ЕК (все) и ФАКТОР (кроме `cdi.log`) в службу технической поддержки HFLabs.

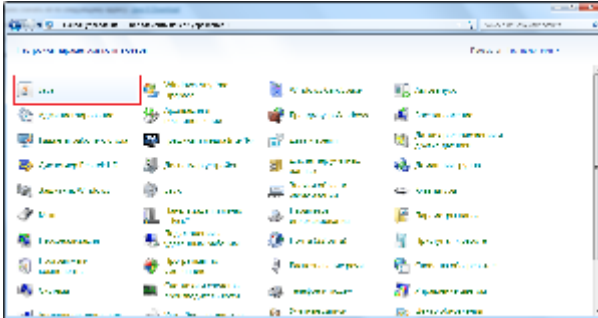
9.2.4 Запуск CDM на машине, где уже установлена Java 1.8.x

Для работы CDM необходим JRE версии 1.7.x. Необходимо скачать его по следующему адресу: [Java 6 Download](#)

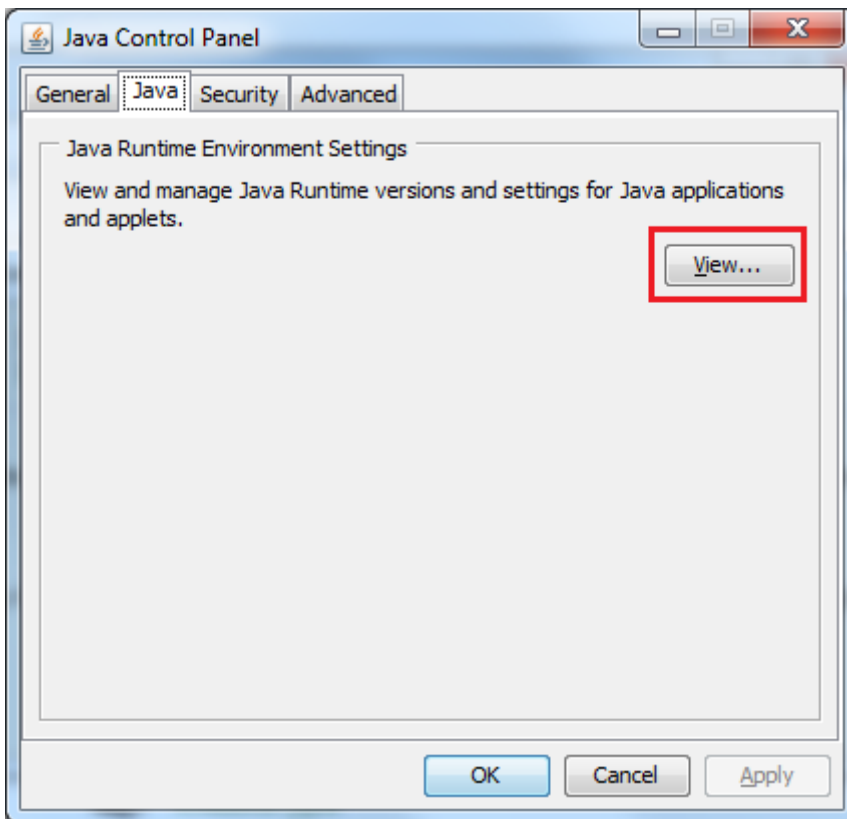
Установка производится в соответствии с инструкциями инсталлятора.

Если на машине ранее был установлен JRE версии 1.8.x, необходимо произвести следующие действия:

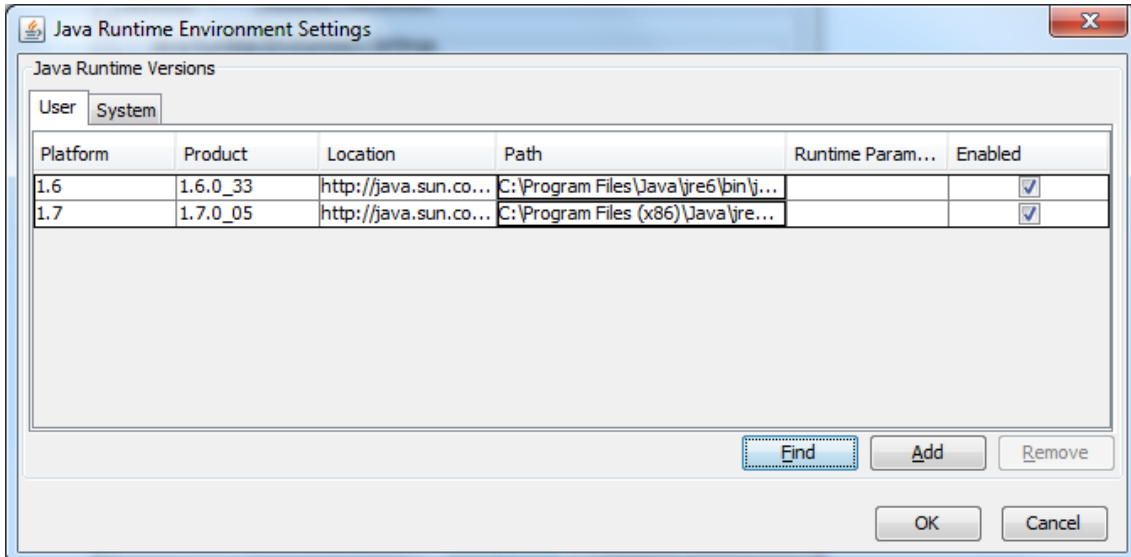
- Открыть Панель управления и выбрать пункт меню "Java":



- В открывшемся окне выбрать вкладку "Java" и нажать кнопку "View":



- После нажатия кнопки появится следующее окно:



- В этом окне необходимо снять чекбоксы "Enable" напротив всех версий Java, кроме 1.7. После этого необходимо нажать кнопку Ok и можно запускать CDM.
- Если во время выполнения указанных действий был запущен любой браузер - перезапустить его.

9.2.5 Невозможно запустить приложение APM Менеджера данных (Unable to launch application)

9.2.5.1 Ошибка



9.2.5.2 Возможные причины

- Не обновилось приложение APM Менеджер данных на рабочей станции пользователя.

9.2.5.3 Что делать

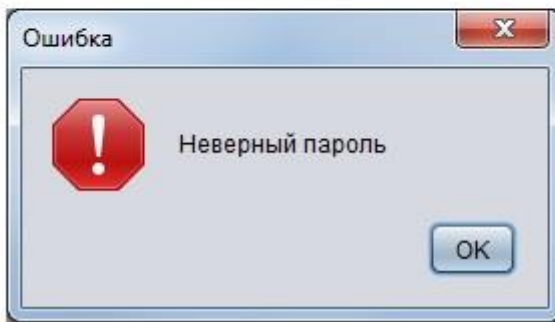
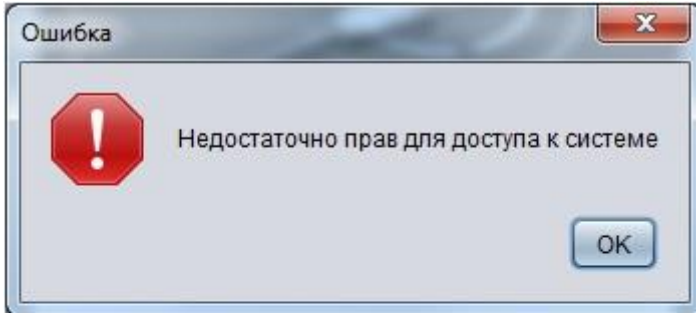
1. Зайти под учетной записью с правами Администратора.
2. Закрыть Менеджер данных.
3. Очистить кеш Java:
Пуск > Панель управления > Java > вкладка General > Settings... > Delete files... > установить обе галочки, нажать ОК.

4. Очистить кеш браузера:

Для IE: Сервис > Свойства обозревателя > вкладка Общие > Удалить... > галочка Временные файлы Интернета > Удалить.

9.2.6 Не могу зайти в АРМ Менеджера данных, неверный пароль или недостаточно прав для доступа

9.2.6.1 Ошибки



9.2.6.2 Возможные причины

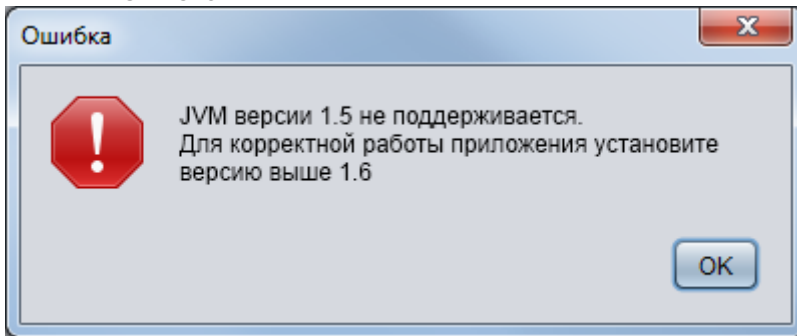
- Неверный пароль.
- Недостаточно прав.
- Пользователь заблокирован.
- Истек срок действия пароля и т.д.

9.2.6.3 Что делать

1. Прочитать текст сообщения об ошибке.
2. Действовать по обстоятельствам.
3. Отправить журналы ЕК в службу технической поддержки HFLabs.

9.2.7 Не могу зайти в APM Менеджера данных, некорректная версия Java

9.2.7.1 Ошибка



9.2.7.2 Возможные причины

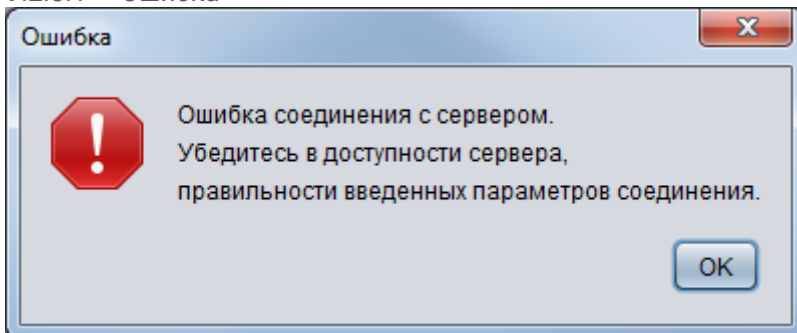
- Некорректная версия Java.

9.2.7.3 Что делать

1. Проверить, что установлена JRE поддерживаемой версии.
2. Если установлено несколько JRE, выбрать подходящую:
Пуск > Панель управления > Java > вкладка Java > View... > выбрать соответствующий JRE, нажать OK. С остальных JRE отметку снять.
3. Перезапустить браузер.

9.2.8 Не могу зайти в APM Менеджера данных, ошибка соединения с сервером

9.2.8.1 Ошибка



9.2.8.2 Возможные причины

- Неправильные параметры соединения.
- Недоступность сервиса.
- Некорректные настройки JVM.

9.2.8.3 Что делать

1. Убедиться, что указано корректное имя сервера и порт.
2. Проверить доступность указанного порта (telnet).
3. Проверить доступность сервисов.
4. Отправить журналы EK в службу технической поддержки HFLabs.

9.2.9 Отслеживание логинов в CDM

Успешные логины выводятся в cdi-security.log

Пример:

```
2018-02-19 11:09:50,145 [AsyncApplicationEventListenerExecutor-2] INFO authentication -
Authentication success. Principal: 'admin_performer'; Granted authority: 'ROLE_PERFORMER'
(Source: RMI;RemoteAddress: 92.168.0.103)
```

Source: RMI означает, что логин из Менеджера данных.

9.3 АРМ Администратора (FAQ)

9.3.1 В IE* нельзя создать триггер - не удается выбрать задачи из списка

9.3.1.1 Ошибка



9.3.1.2 Возможные причины

- Не включен JavaScript.

9.3.1.3 Что делать

1. В настройках открыть "Свойства браузера" (Internet Options).
2. Выбрать вкладку "Безопасность" (Security) и нажать на ней кнопку "Другой..." (Custom level...).
3. После открытия диалогового окна "Параметры безопасности - зона местной интранета" (Security Settings - Internet Zone), посмотреть раздел "Сценарии" (Scripting).
4. В разделе "Активные сценарии" (Active Scripting) выбрать пункт "Включить" (Enable).
5. Сохранить изменения и обновить страницу.

9.3.2 Изменение параметра oldMaxId

Q: Если остановить службы CDI и Factor исправить в конфигурации значение параметра oldMaxId в параметрах импорта из буферной области и заново запустить службы, как себя поведет приложение?

A: В зависимости от того, увеличите или уменьшите вы его значение, получите разные последствия:

1. Уменьшите - система пропустит уже загруженные ранее записи и выполнит загрузку только изменившихся записей.

2. Увеличите - система выполнит загрузку только изменившихся записей начиная с указанного oldMaxId.

В обоих случаях ошибок быть не должно.

9.4 Дополнительно (FAQ)

9.4.1 Запросы в БД, соответствующие фильтрам в CDM

Q: Как выполнить выборки из базы, соответствующие фильтрам в интерфейсе CDM?

A: При фильтрации записей в CDM в базе данных выполняются следующие запросы (запросы выводят количество записей, подходящих под установленный фильтр):

Code Block 7 Ручная обработка: реквизиты

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
WHERE (
  (PP.fullname_qc = 'NOT_SURE')
  OR (PP.inn_qc = 'NOT_SURE')
  OR (PP.birthdate_qc = 'NOT_SURE')
)
AND PP.version = 0
AND PP.enddate is null
AND PP.merged_status IN ( 0, 2 );
```

Code Block 8 Ручная обработка: адреса

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN address A on PP.hid_party = A.hid_party
WHERE (
  (A.validationCode NOT IN ('VALIDATED', 'NOT_VALIDATED_EMPTY_SOURCE',
  'NOT_VALIDATED_FOREIGN', 'CONFIRMED_MANUALLY'))
  OR (A.qualityCode NOT IN ('GOOD', 'UNDEF_05'))
)
AND PP.version = 0
AND PP.enddate is null
AND A.version = 0
AND A.enddate is null
AND PP.merged_status IN ( 0, 2 );
```

Code Block 9 Ручная обработка: телефоны

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN phone P on PP.hid_party = P.hid_party
WHERE (
  (P.qualityCode NOT IN ('GOOD', 'GOOD_CITY_CONFLICT', 'GOOD_REGION_CONFLICT',
  'GOOD_REPLACED_CODE', 'GOOD_REPLACED_NUMBER', 'GOOD_REPLACED_CODE_NUMBER',
  'CONFIRMED_MANUALLY'))
)
)
```

```

AND PP.version = 0
AND PP.enddate is null
AND P.version = 0
AND P.enddate is null
AND PP.merged_status IN ( 0, 2 );

```

Code Block 10 Ручная обработка: email

```

SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN email E on PP.hid_party = E.hid_party
WHERE (
(E.qualityCode = 'NOT_SURE')
)
AND PP.version = 0
AND PP.enddate is null
AND E.version = 0
AND E.enddate is null
AND PP.merged_status IN ( 0, 2 );

```

Code Block 11 Ручная обработка: документы

```

SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN document D on PP.hid_party = D.hid_party
WHERE (
(D.qualityCode = 'NOT_SURE')
)
AND PP.version = 0
AND PP.enddate is null
AND D.version = 0
AND D.enddate is null
AND PP.merged_status IN ( 0, 2 );

```

9.4.2 Как пополнить справочник имен / фамилий?

В сложных ФИО возможен неправильный разбор Фактором, так как в наших справочниках может не быть одного из компонентов. Чтобы исправить ситуацию, есть два варианта:

1. Поправить вручную в АРМ Менеджера данных. Если считаете ФИО корректными и часто встречающимися, пришлите его нам, добавим разбор в следующих релизах.
2. Пополнить справочник в Факторе.

Предпочтительно использовать первый вариант.

Но, если данные проходят через Фактор и уходят сразу в несколько систем, используйте пользовательские справочники.

9.4.2.1 Как добавить имя в справочник Фактора

Допустим, мы хотим добавить в справочник женское имя «АЙЫР-СААНА»

Необходимо пополнить [справочник имен](#)

```
АЙЫР-СААНА;1;;
```

После чего рестартануть фактор (займет минуту-две).

Для горячего резерва повторить на обеих нодах.

По аналогии пополняются:

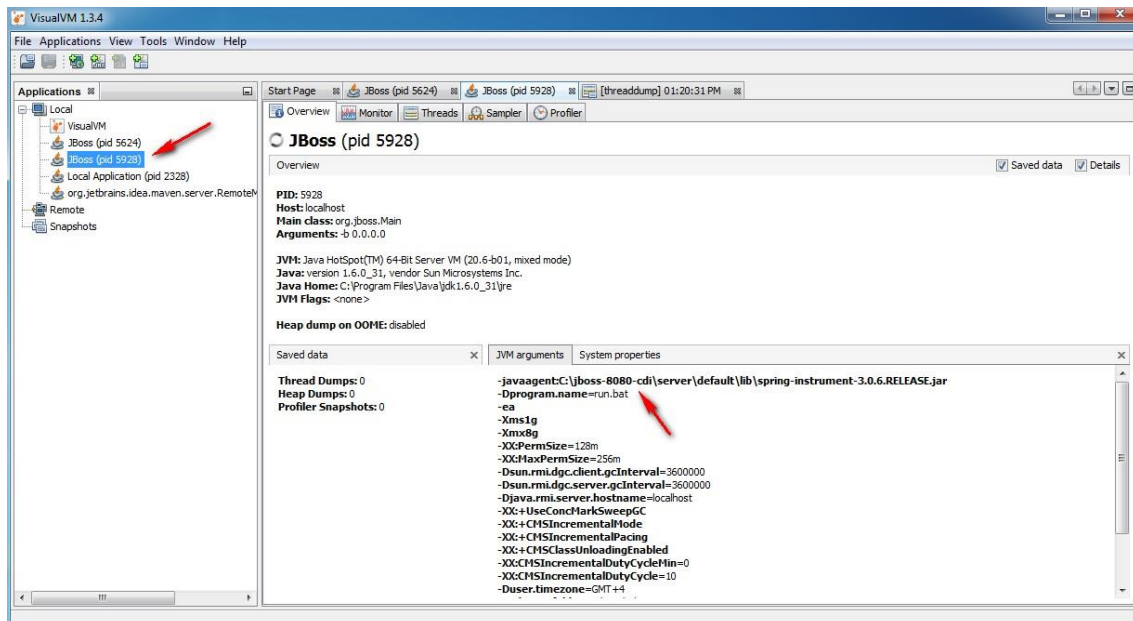
- [справочник фамилий](#);
- [справочник отчетств](#).

9.5 Инструменты для проверки (FAQ)

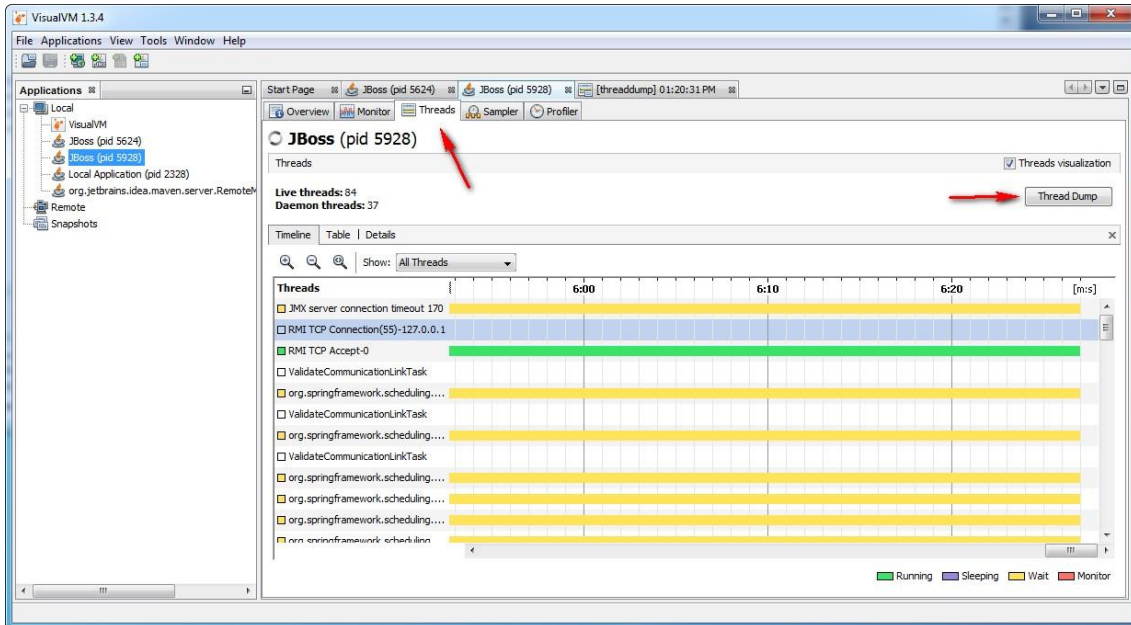
9.5.1 Как снять Thread Dump

Чтобы снять thread dump, необходимо:

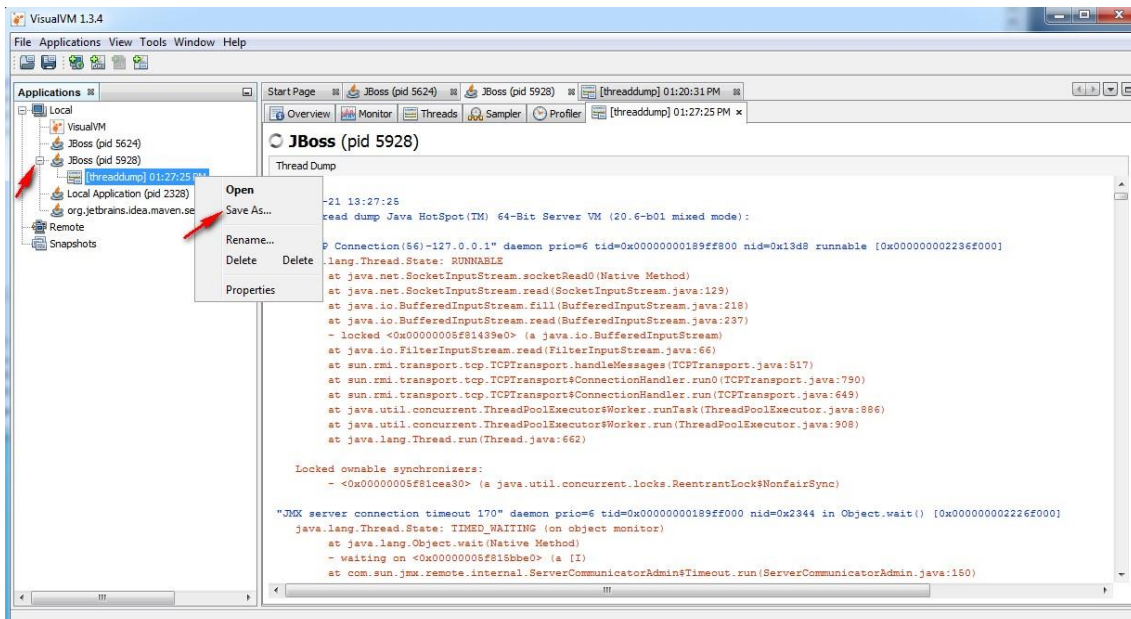
- Запустить [Java VisualVM](#) (находится в папке bin)
- Распаковать архив и запустить приложение
- В левой области под названием “Applications” выбрать jboss. Их там 2 – cdi и factor. Понять, где какой, можно в описании приложения



- Перейти на закладку “Threads” и нажать на кнопку “Thread Dump”.



- Полученный дамп откроется в окне VisualVM, а также появится в дереве приложений. В дереве приложений необходимо нажать на «+» около Jboss-a, щелкнуть правой кнопкой мыши по дампу и нажать «Сохранить как».



9.5.2 Как снять Thread Dump через командную строку в Unix

Если нет возможности использовать [Java VisualVM](#), то для снятия дампа потоков необходимо:

- Найти id процесса. Для этого необходимо выполнить команду

```
ps aux | grep java
```

- В результате выполнения этой команды вы увидите запущенные java процессы в виде:


```
cdi 29271 0.4 50.6 10181244 8501280 ? Sl Jun19 10:52 /usr/java/jdk1.6.0_29/bin/java -
Dprogram.name=run.sh -server -javaagent:/home/cdi/jboss/server/default/lib/spring-instrument-
3.0.5.RELEASE.jar -ea -XX:PermSize=256m -XX:MaxPermSize=256m -Xms1g -Xmx8g -
XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode -XX:+CMSIncrementalPacing -
XX:CMSIncrementalDutyCycleMin=0 -XX:CMSIncrementalDutyCycle=10 -
XX:+CMSClassUnloadingEnabled -Djava.io.charset=Cp1251 -Dfile.encoding=Cp1251 -
Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -
Djava.rmi.server.hostname=dev-yota -Duser.timezone=GMT+4 -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=p -
Xrunjdp:transport=dt_socket,address=5005,server=y,suspend=n -Djava.net.preferIPv4Stack=true -
Djava.endorsed.dirs=/home/cdi/jboss/lib/endorsed -classpath
/home/cdi/jboss/bin/run.jar:/usr/java/jdk1.6.0_29/lib/tools.jar org.jboss.Main -b 0.0.0.0 -c default -
Dcdi.root.folder=/home/cdi -Dcdi.dedup.folder=/tmp
```

- Выполнить команду "kill -QUIT <ID>". В данном примере эта команда будет выглядеть как:

```
kill -QUIT 29271
```

После выполнения команды в console.log запишется thread dump.

9.5.2.1 Второй вариант

Командой вида

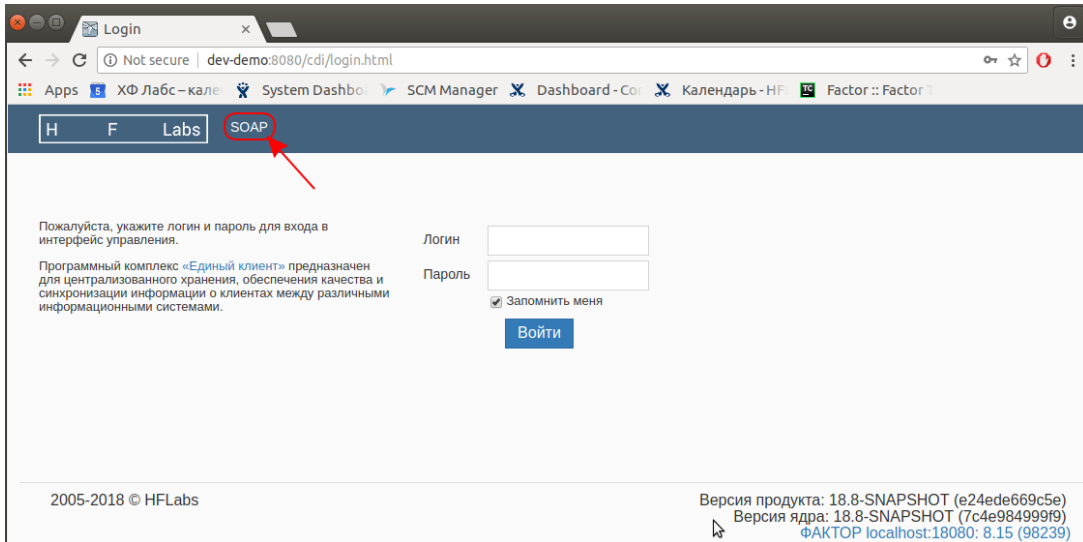
```
jstack -F 29271 > /opt/cdi/jstack.log
```

где 29271 — pid java-процесса. По умолчанию выводит в консоль, можно вывод перенаправить в файл (как в примере и сделано). Флаг -F — это принудительное соединение с джава-процессом.

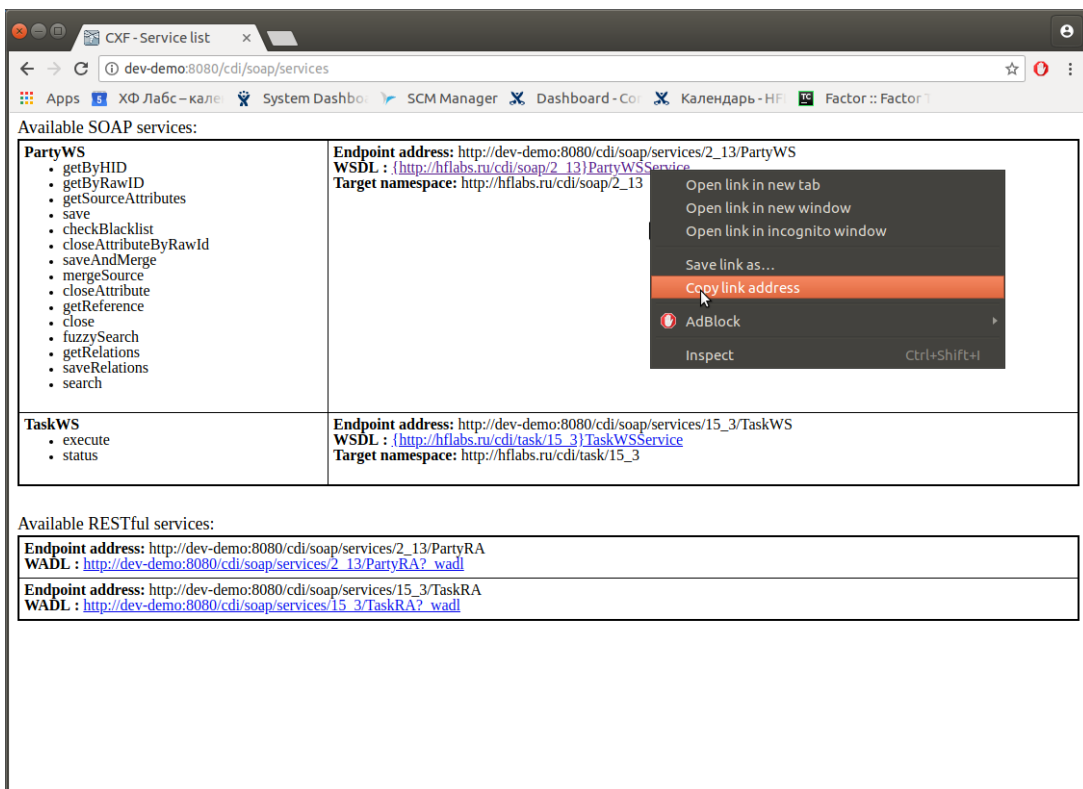
9.5.3 Как можно вызвать метод SOAP-интерфейса?

Чтобы проверить работу [онлайн-интерфейсов](#) вручную, можно использовать инструмент [SOAP UI](#):

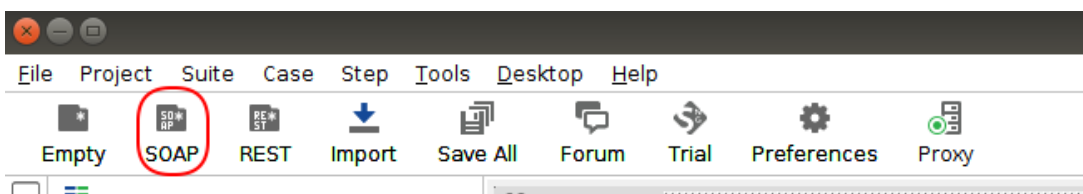
1. Открыть APM Администратора, нажать на ссылку SOAP.



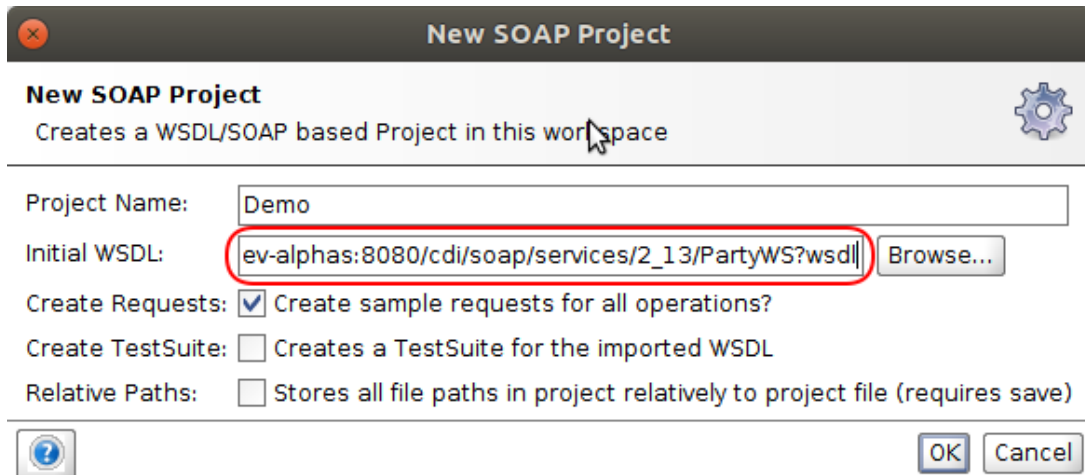
2. Система отображает все доступные пользователю сервисы. Нажать на ссылку WSDL у необходимого сервиса правой кнопкой мыши и выбрать copy link address или перейти по этой ссылке и скопировать адрес:



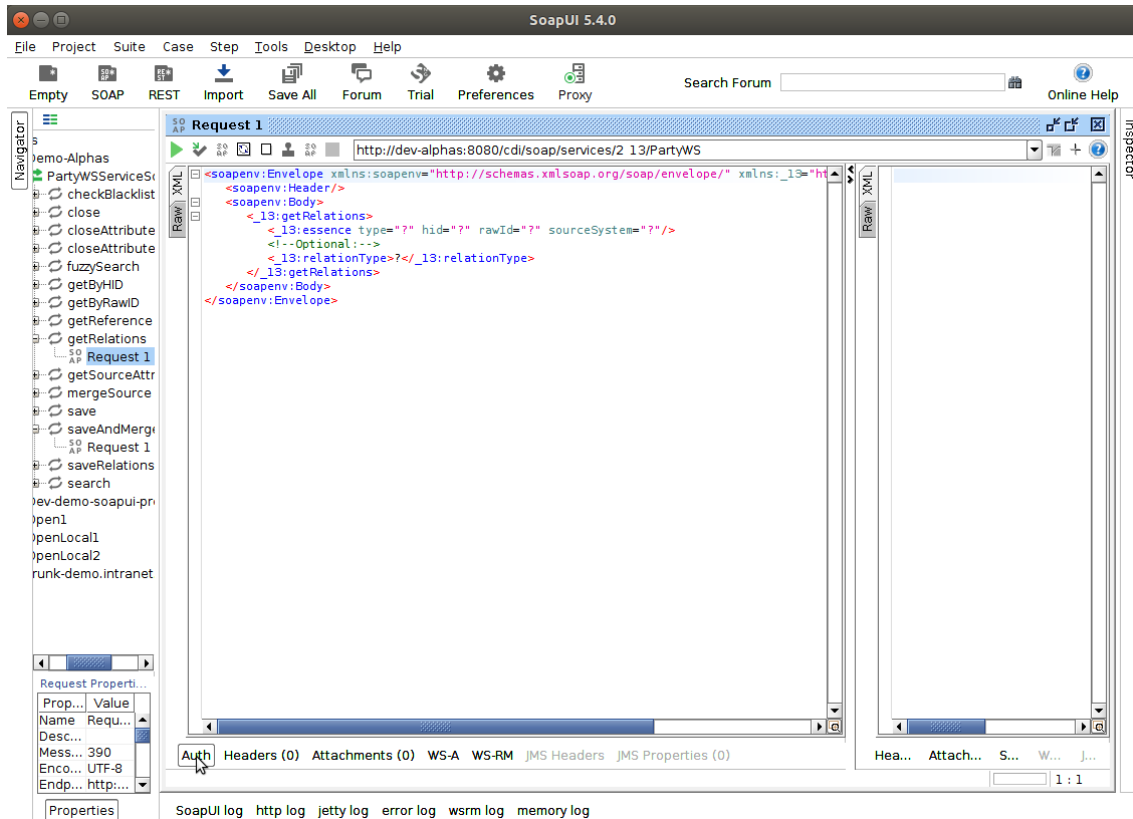
3. В SOAP UI нажать на кнопку Create a new soapUi Project



4. В открывшемся окне необходимо задать имя новому проекту и ввести ссылку на WSDL, полученную на шаге 2. После чего нажать "Ок". Будет создан новый проект.



После выбора запроса нажмите на кнопку Authorization слева внизу и авторизируйтесь



Как правило у учётной записи [операциониста](#) есть необходимые права (ROLE_EXTERNAL_SYSTEM), однако в отдельных случаях потребуется специальная учётная запись. Подробнее смотрите в спэйсе заказчика.

Необходимо также выбрать предварительную авторизацию (authenticate preemptively).

The screenshot shows the SoapUI 5.4.0 interface. The main window displays a SOAP request and its response. The request is a `<saveAndMerge>` operation with a `<party>` element containing various fields like `fullNameRawSource`, `birthdate`, `qualifierType`, `issueDate`, `issueAuthority`, `documentSeries`, `documentNumber`, and `departmentCode`. The response is a `<saveAndMergeResponse>` with a `<party type="PHYSICAL">` element containing fields like `surname`, `name`, `patronymic`, `gender`, `fullNameOC`, `fullNameAuthor`, `fullNameRawSou`, `foreignSurname`, `foreignName`, `foreignPatrony`, `foreignFullNam`, `foreignFullNam`, `birthdate`, `birthdateAuth`, `birthdateOC`, `birthdateRawSou`, `maritalStatus`, `maritalStatusA`, `inn`, `innAuthor`, `innOC`, `innRawSource`, `snils`, `snilsOC`, `snilsAuthor`, `snilsRawSource`, `dependantsCoun`, `dependantsCoun`, `monthlyMainInc`, `monthlyMainInc`, `monthlyIncomeP`, `monthlyIncomeP`, `firstAccountDa`, and `firstAccountDa`.

The interface also shows an Authorization dialog box with the following fields:

- Authorization: Basic
- Username: performer
- Password: *****
- Domain: (empty)
- Pre-emptive auth: Authenticate pre-emptively

The status bar at the bottom indicates a response time of 232ms (4186 bytes) and the time 19:20.

Теперь необходимо отредактировать шаблон запроса (в данном случае [search](#))

The screenshot shows the SoapUI 5.4.0 interface with a SOAP request template for a search operation. The request is a `<_13:search>` operation with the following structure:

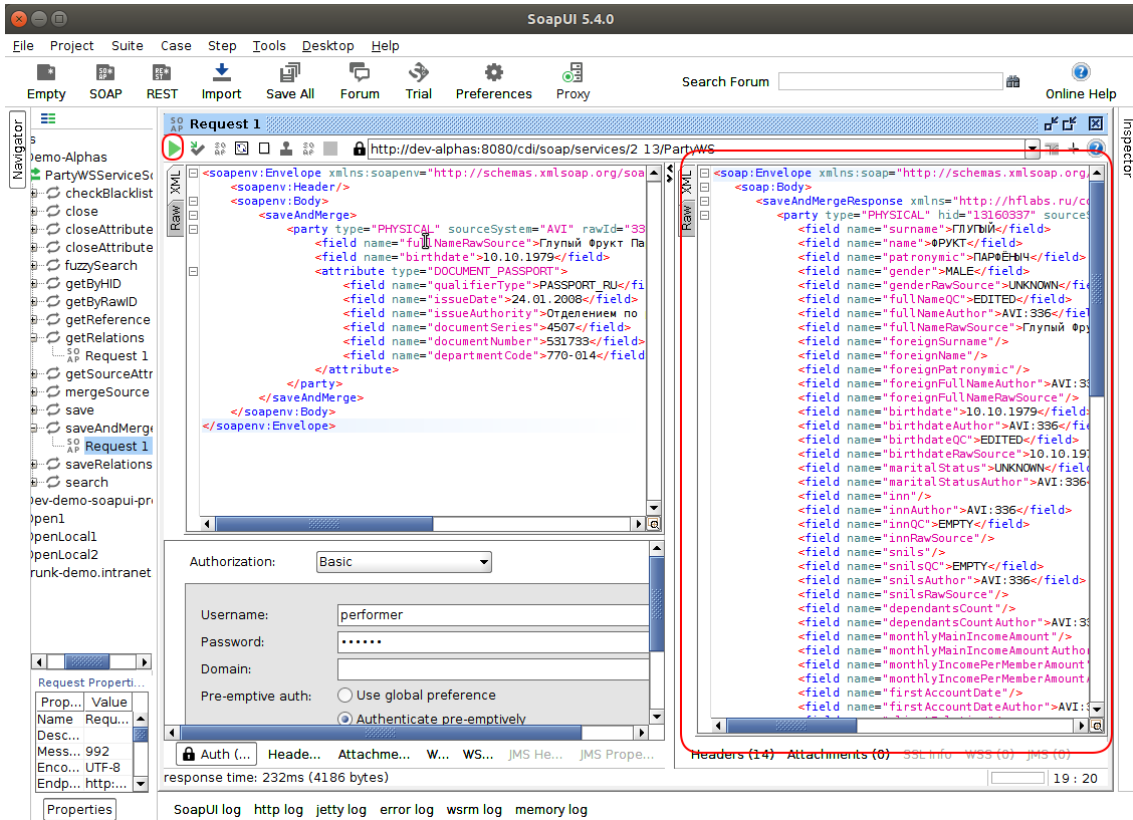
```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:_13="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <_13:search>
      <_13:query?></_13:query>
      <!-- Optional:-->
      <_13:restriction?></_13:restriction>
      <!-- Optional:-->
      <_13:partyType?></_13:partyType>
      <!-- Optional:-->
      <_13:include>
        <!-- 1 or more repetitions:-->
        <_13:partyInfo?></_13:partyInfo>
      </_13:include>
    </_13:search>
  </soapenv:Body>
</soapenv:Envelope>

```

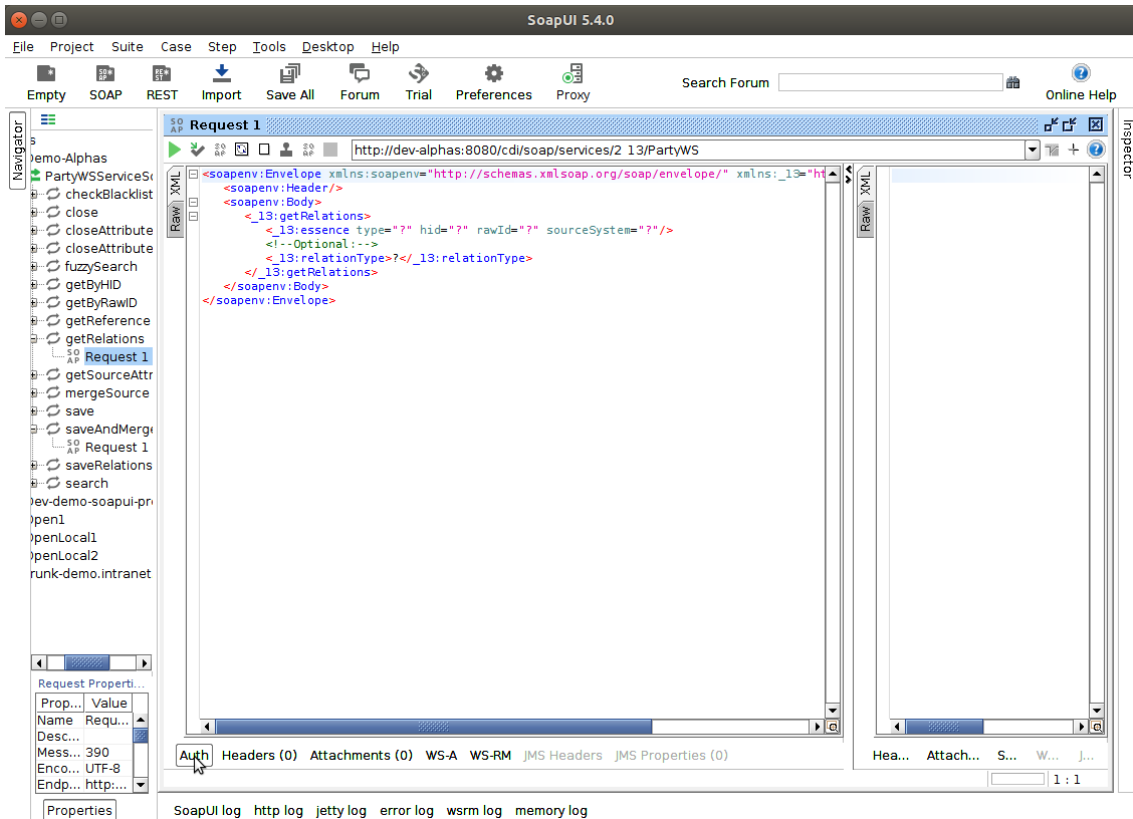
Его необходимо отредактировать согласно [документации](#) (выше приведена ссылка на пример запроса search) и нажать кнопку submit request

В правой части мы получим ответ



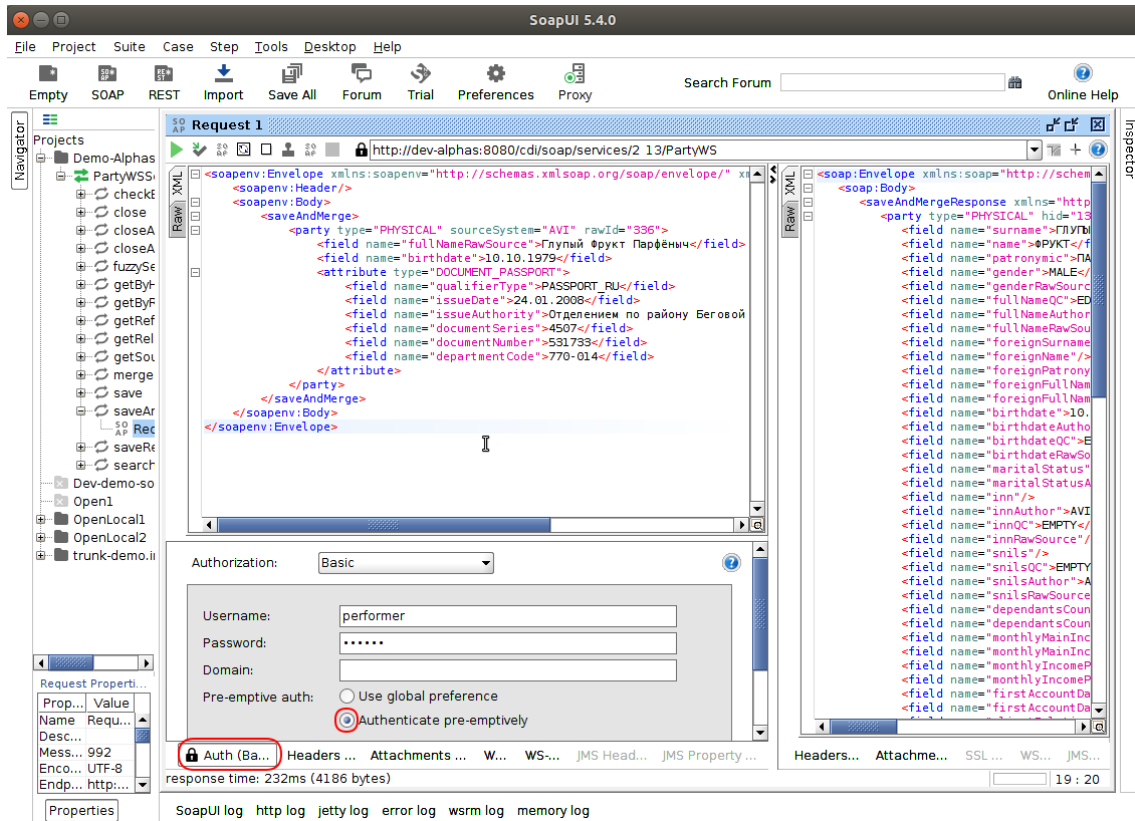
9.5.3.1 Авторизация запроса в SoapUI

После выбора запроса нажмите на кнопку Authorization слева внизу и авторизируйтесь



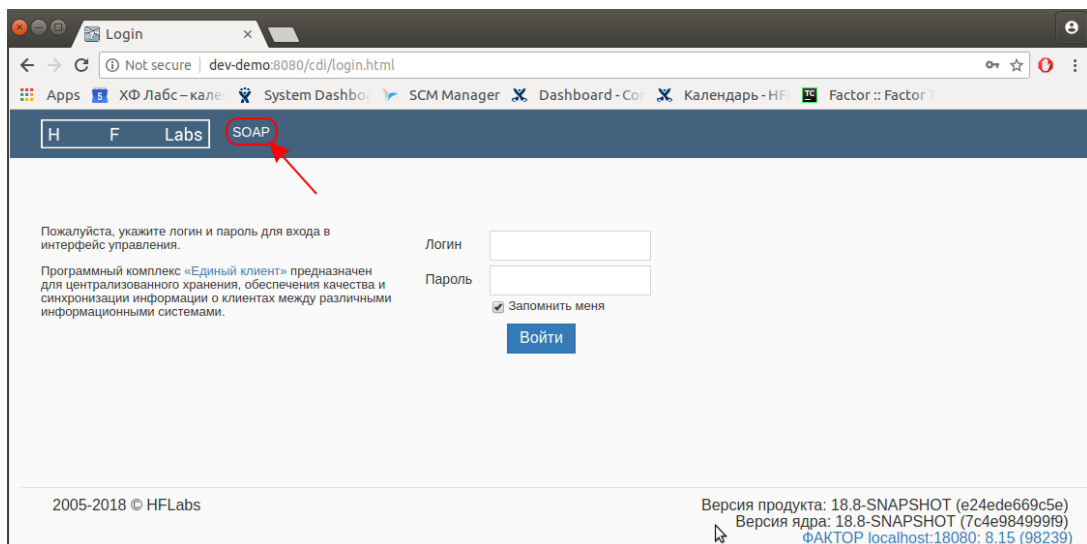
Как правило у учётной записи [операциониста](#) есть необходимые права (ROLE_EXTERNAL_SYSTEM), однако в отдельных случаях потребуется специальная учётная запись. Подробнее смотрите в спэйсе заказчика.

Необходимо также выбрать предварительную авторизацию (authenticate preemptively).

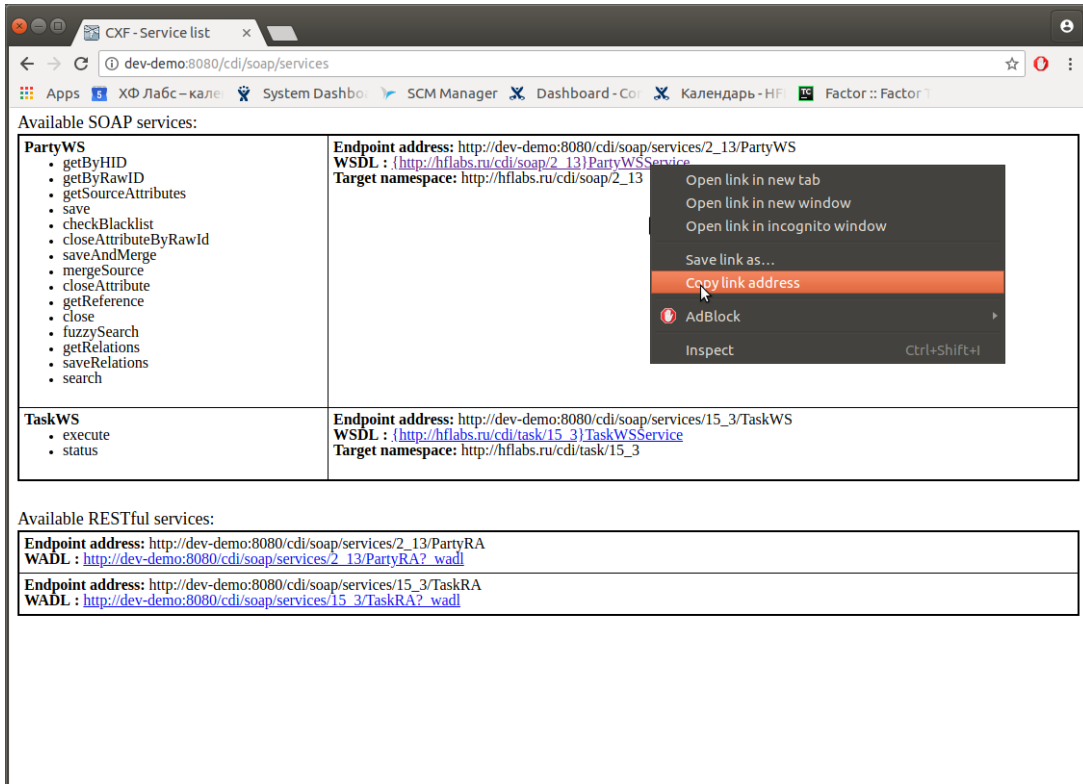


9.5.3.2 Создание проекта в SoapUI

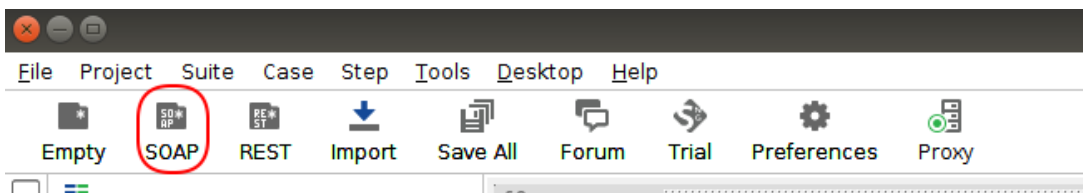
1. Открыть APM Администратора, нажать на ссылку SOAP.



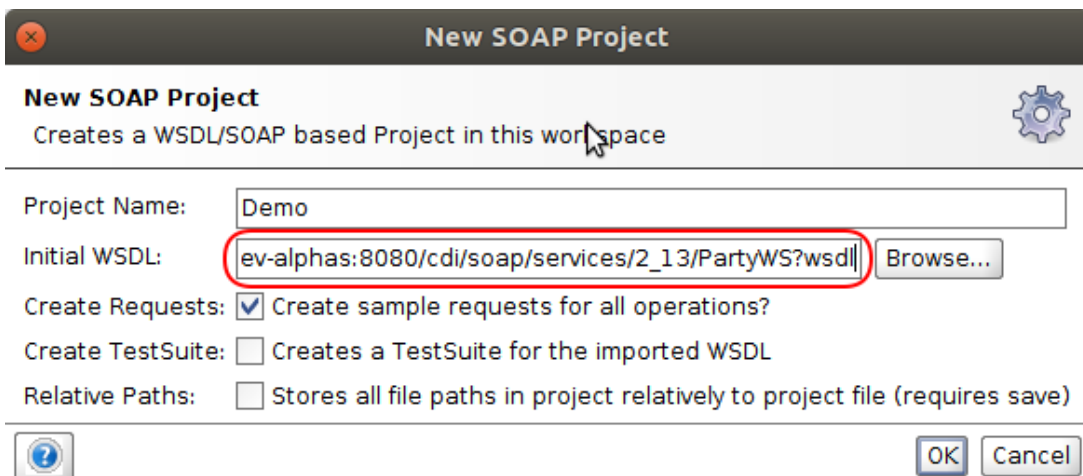
2. Система отображает все доступные пользователю сервисы. Нажать на ссылку WSDL у необходимого сервиса правой кнопкой мыши и выбрать copy link address или перейти по этой ссылке и скопировать адрес:



3. В SOAP UI нажать на кнопку Create a new soapUi Project



4. В открывшемся окне необходимо задать имя новому проекту и ввести ссылку на WSDL, полученную на шаге 2. После чего нажать "Ок". Будет создан новый проект.



9.5.4 Как можно вызвать REST-метод вручную

Для ручного вызова [REST-методов](#) различных интерфейсов ЕК и Фактора можно использовать SoapUI. Описание на примере стенда dev-demo.

9.5.4.1 Методы ЕК

1. Открыть APM Администратора, нажать на ссылку SOAP.

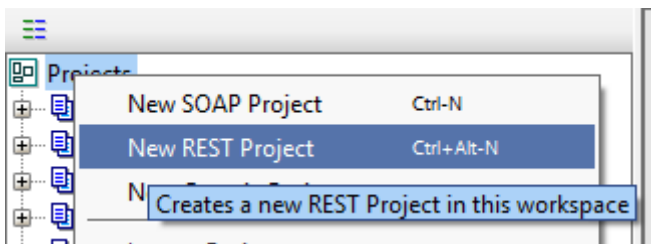


2. Скопировать ссылку на WADL нужного REST-интерфейс.

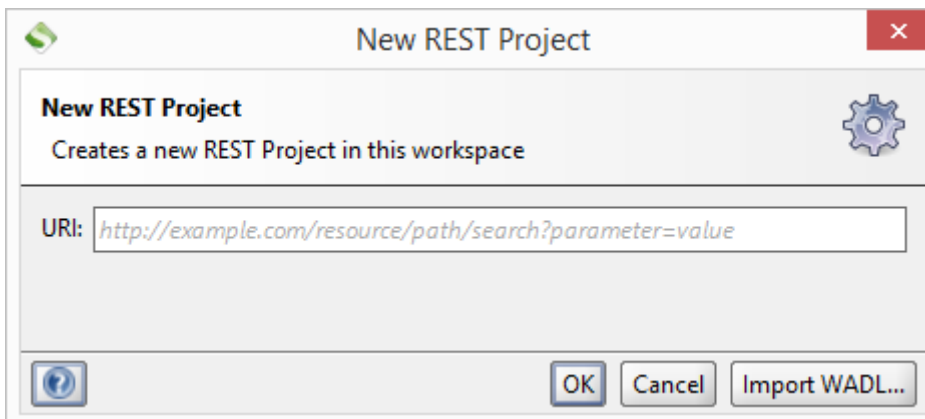
Available RESTful services:

Endpoint address: http://dev-demo:8080/cdi/soap/services/15_3/TaskRA
WADL : http://dev-demo:8080/cdi/soap/services/15_3/TaskRA? wadl
Endpoint address: http://dev-demo:8080/cdi/soap/services/2_13/PartyRA
WADL : http://dev-demo:8080/cdi/soap/services/2_13/PartyRA? wadl

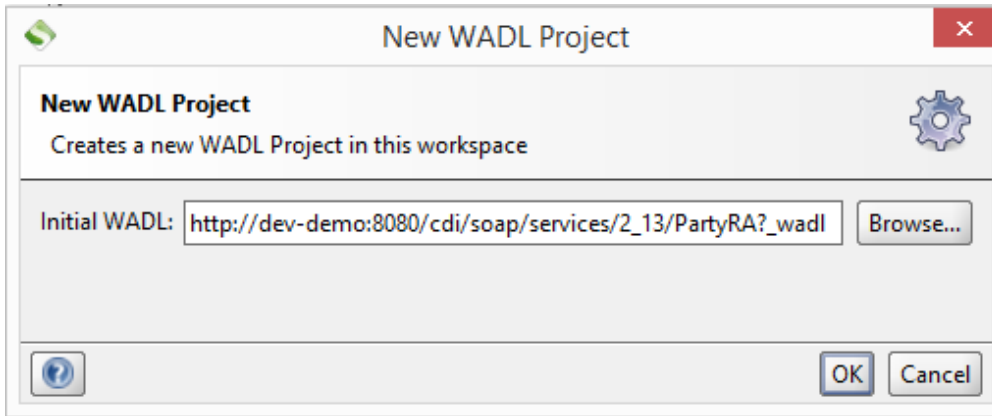
3. В SoapUI создать новый REST-проект.



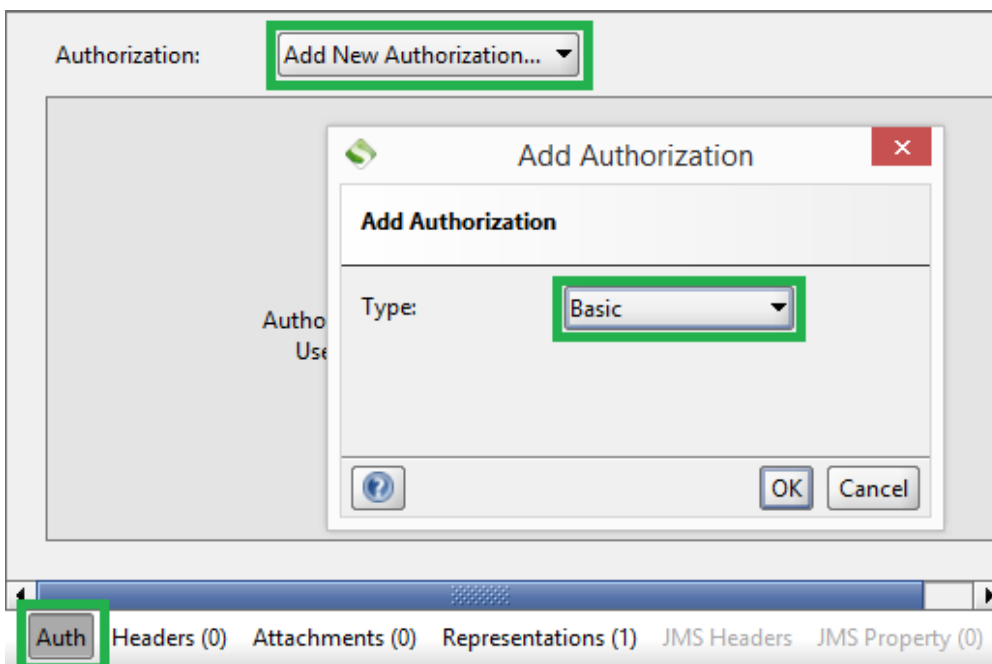
4. В появившемся диалоге нажать на кнопку «Import WADL...».



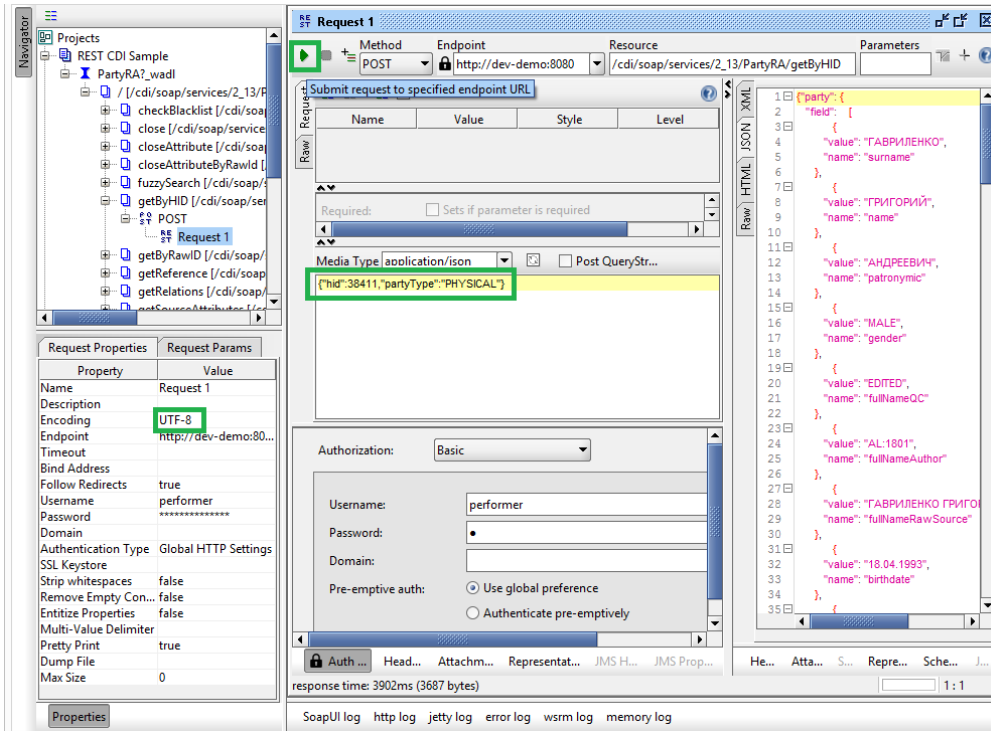
5. Вставить скопированную ранее ссылку и нажать «OK».



6. Указать данные пользователя с правами операциониста (или вебсервиса, если эта роль настроена отдельно) для использования в HTTP Basic Auth.



7. Выбрать нужную кодировку, заполнить данные запроса, нажать Submit.



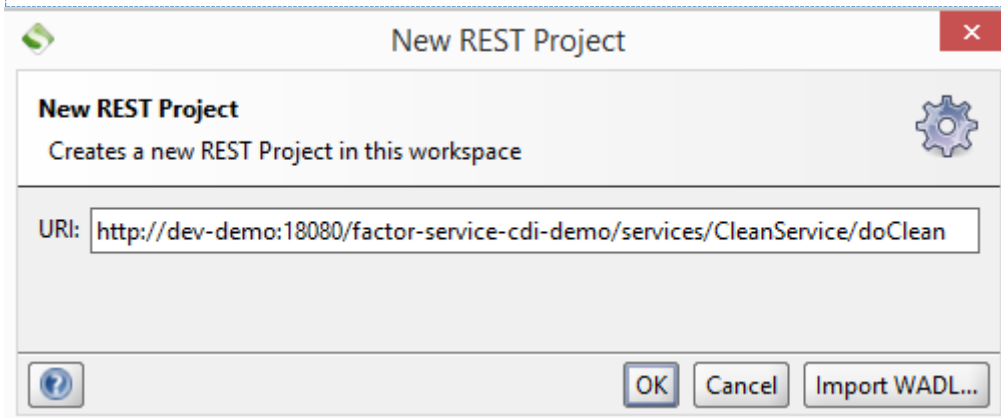
9.5.4.2 Методы Фактор

У Фактора нет WADL и авторизации, поэтому чуть по-другому. Зато можно отправлять как POST-, так и GET-запросы.

1. В SoapUI создать новый REST-проект.
2. В появившемся окне указать ссылку REST-сервиса Фактора.

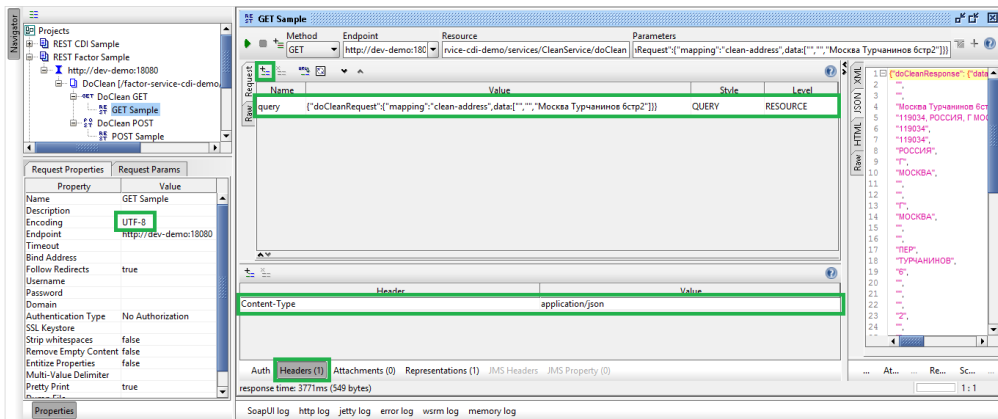
```
-- Шаблон
http://{ip}:{port}/factor-service-{customer}/services/CleanService/{method}

-- Например, для заказчика cdi-demo и метода doClean
http://dev-demo:18080/factor-service-cdi-demo/services/CleanService/doClean
```

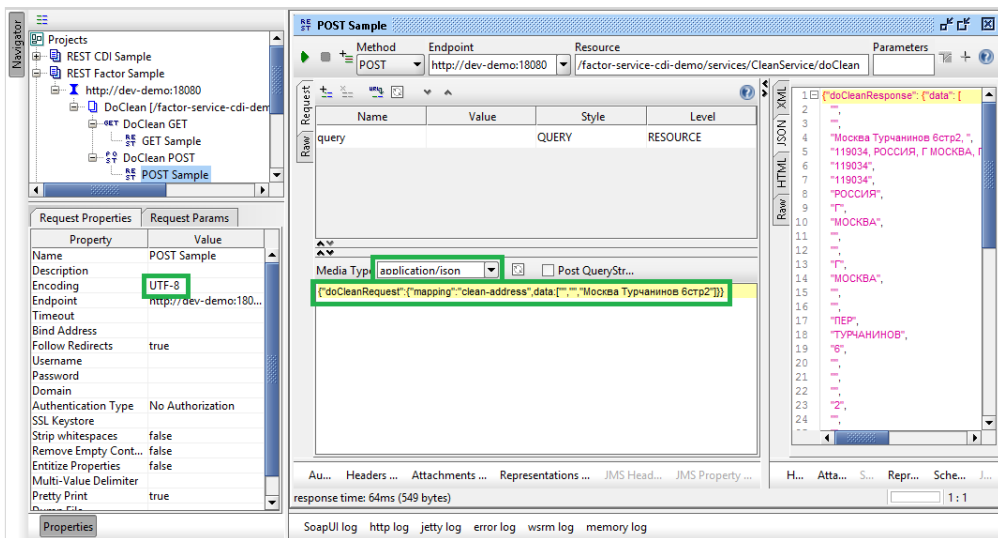


3. Для GET-запроса надо указать кодировку, добавить заголовок Content-Type=application/json, указать параметр query (можно делать через таблицу параметров, можно указывать в Parameters справа сверху). В приведённом примере значение параметра query такое: «{"doCleanRequest":{"mapping":"clean-

address",data:["", "", "Москва Турчанинов бстр2"]}}». После нажатия Submit должен прийти ответ от Фактора.



4. Для POST-запроса надо указать те же параметры, но немного иначе.

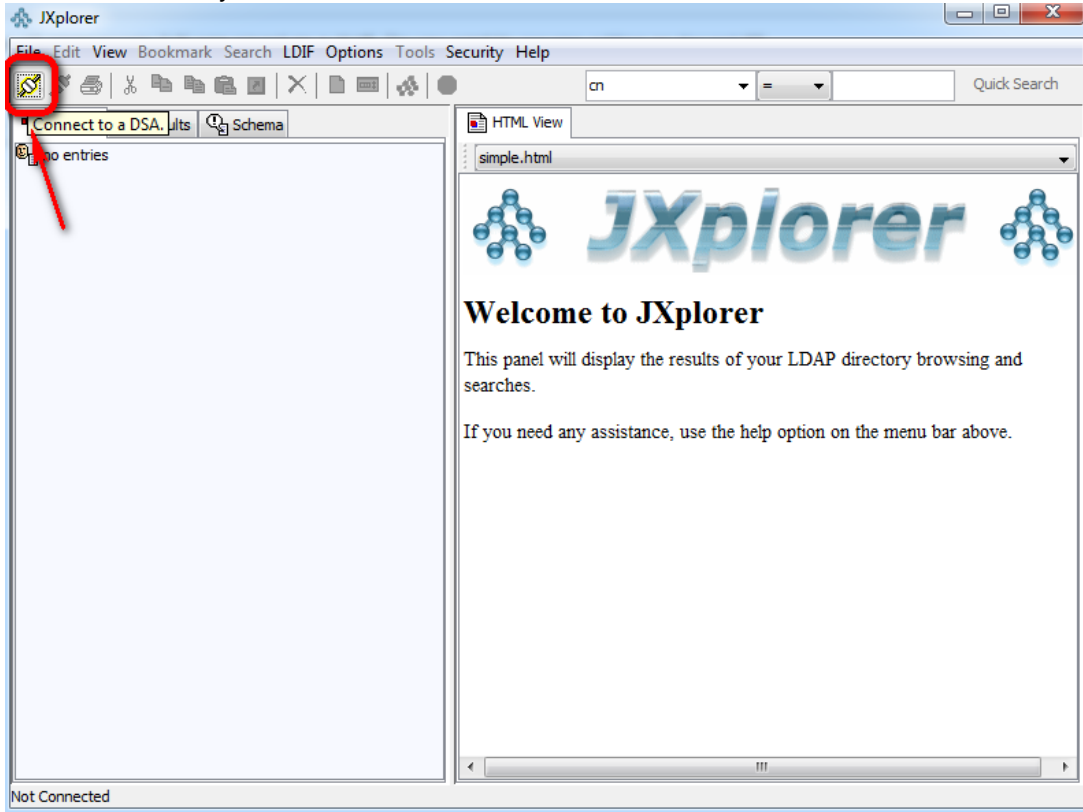
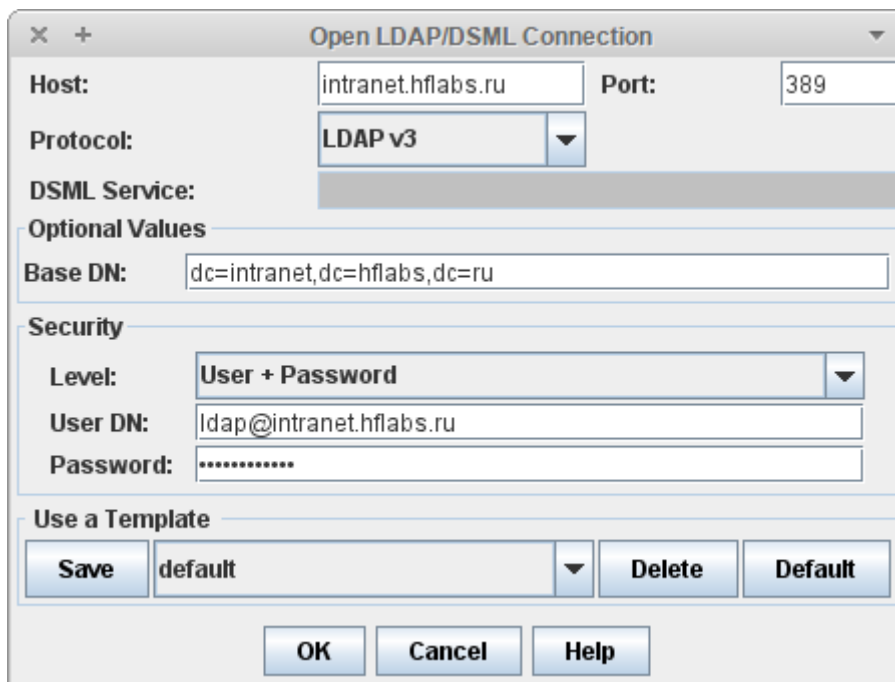


9.5.5 Как проверить учетную запись LDAP?

1. Открыть раздел [Параметры LDAP](#) на вкладке [Конфигурация](#) или получить данные из БД:

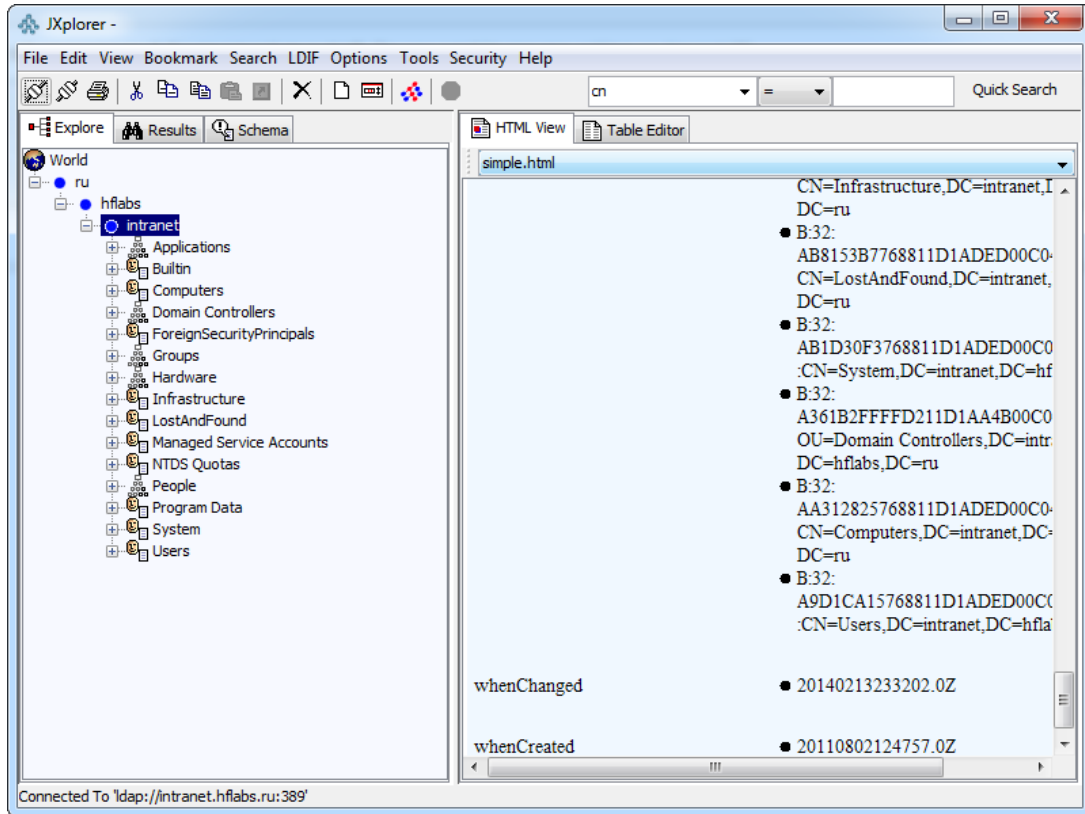
```
select * from config_property where source = 'ldap.properties';
```

2. Скачать и запустить программу [JXplorer](#).

3. Нажать на кнопку *Connect to DSA*4. Заполнить форму (см. [Пример заполнения формы](#)).The image shows a dialog box titled 'Open LDAP/DSML Connection'. It contains several fields and buttons. The 'Host' field is filled with 'intranet.hflabs.ru' and the 'Port' field with '389'. The 'Protocol' dropdown menu is set to 'LDAP v3'. The 'DSML Service' field is empty. Under the 'Optional Values' section, the 'Base DN' field is filled with 'dc=intranet,dc=hflabs,dc=ru'. In the 'Security' section, the 'Level' dropdown is set to 'User + Password', the 'User DN' field is filled with 'ldap@intranet.hflabs.ru', and the 'Password' field is filled with a series of dots. At the bottom, there is a 'Use a Template' section with a 'Save' button highlighted in blue, followed by a dropdown menu showing 'default', a 'Delete' button, and a 'Default' button. At the very bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

5. Нажать ОК

6. Если запустилось такое окно, значит, учетная запись работает:



9.5.5.1 Пример заполнения формы Результат запроса из базы:

NAME	VALUE	SOURCE
ldap.account	Ivanov	ldap.properties
ldap.domain	intranet.mycompany.ru	ldap.properties
ldap.password	password	ldap.properties
ldap.searchBase	dc=intranet,dc=hflabs,dc=ru	ldap.properties
ldap.url	ldap://mycompany.ru:389	ldap.properties

Заполнение формы:

1. host и port взять из параметра ldap.url – "mycompany.ru", 389.
2. Base DN – значение параметра ldap.searchBase – "dc=intranet,dc=hflabs,dc=ru"
3. Level – выбрать из списка пункт "User + Password"
4. User DN – объединить через @ значения параметров ldap.account и ldap.domain, – "Ivanov@intranet.mycompany.ru"
5. Password – значение параметра ldap.password

9.5.6 Как посмотреть JMS, уходящие в Tivco

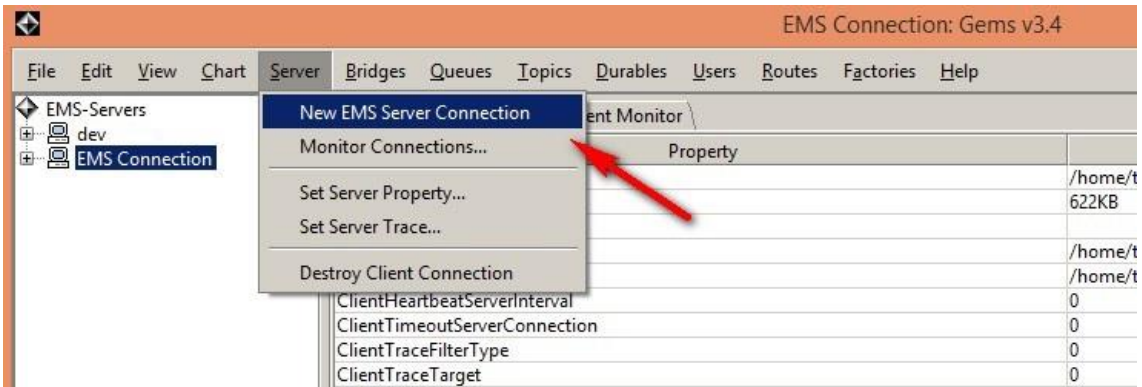
9.5.6.1 Подготовка

Установить утилиту Gems: распаковать из [архива](#), подложить в папку lib джарники для вашей версии Tivco.

9.5.6.2 Проверка сообщений

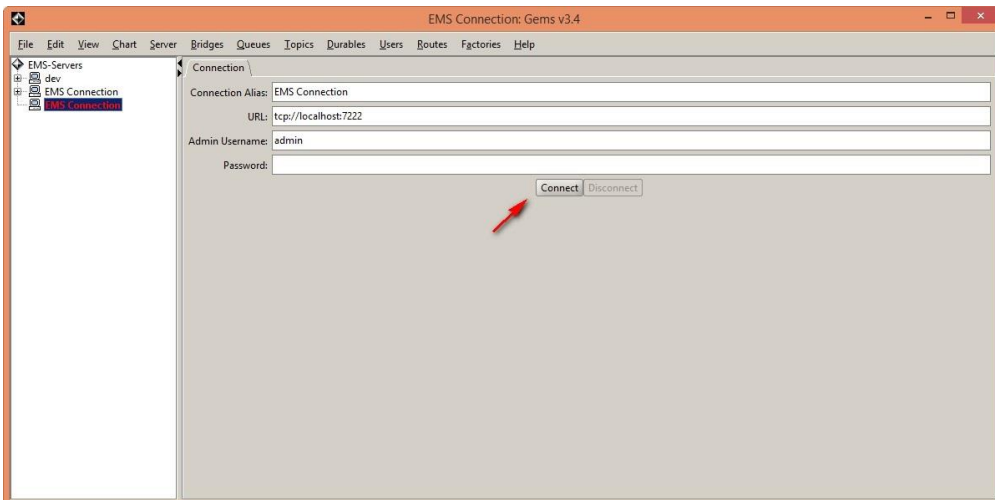
1. Запустить Gems – rungems.bat

2. Server – New EMS Server Connection

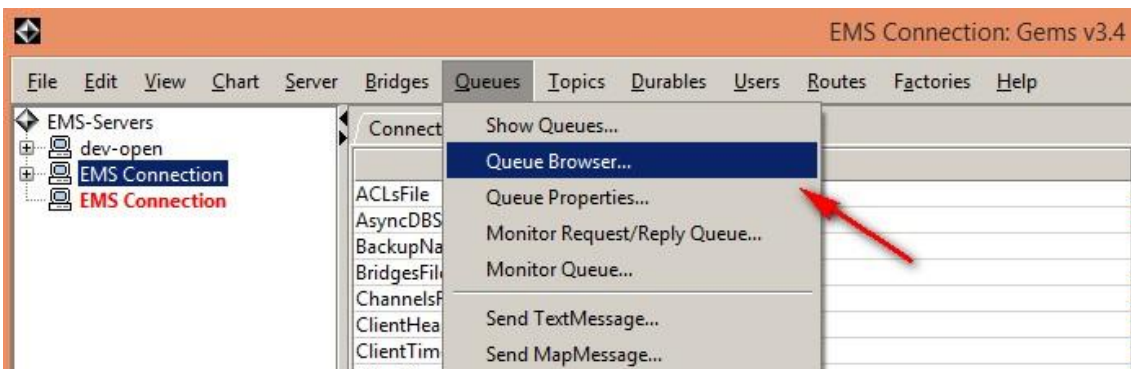


3. Указать настройки подключения, нажать Connect:

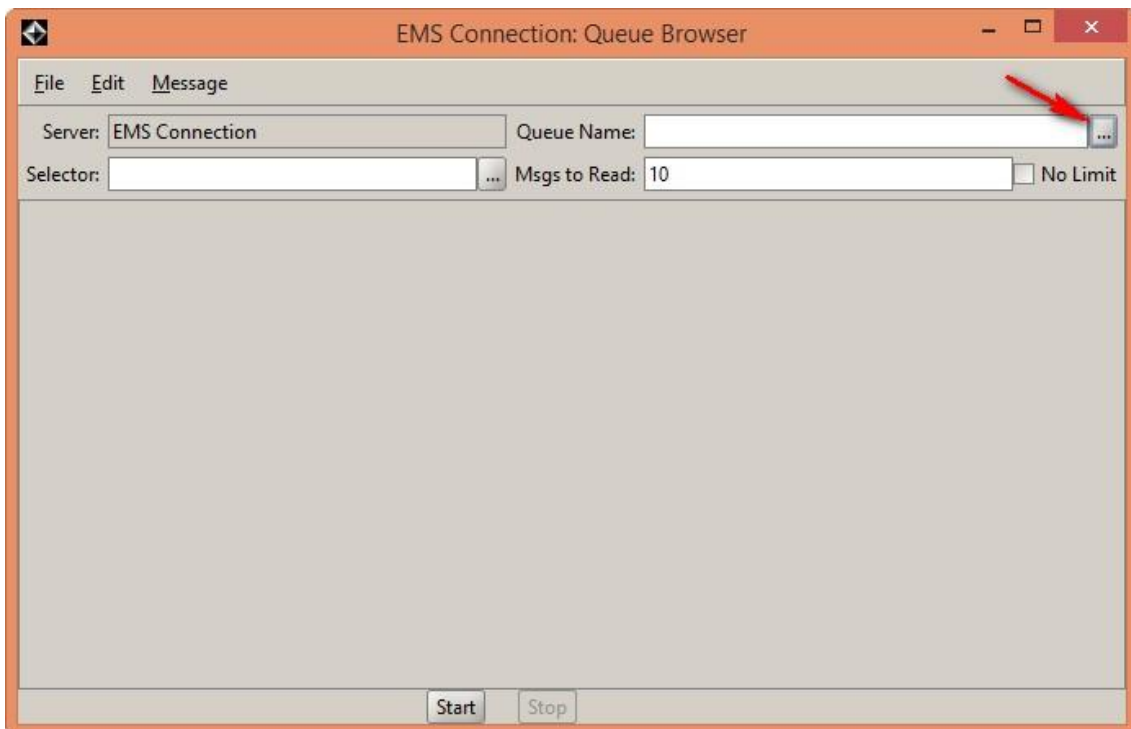
- Alias – имя, которое будет отображаться слева в меню
- URL – URL подключения к Tivco
- Username\password – данные для подключения



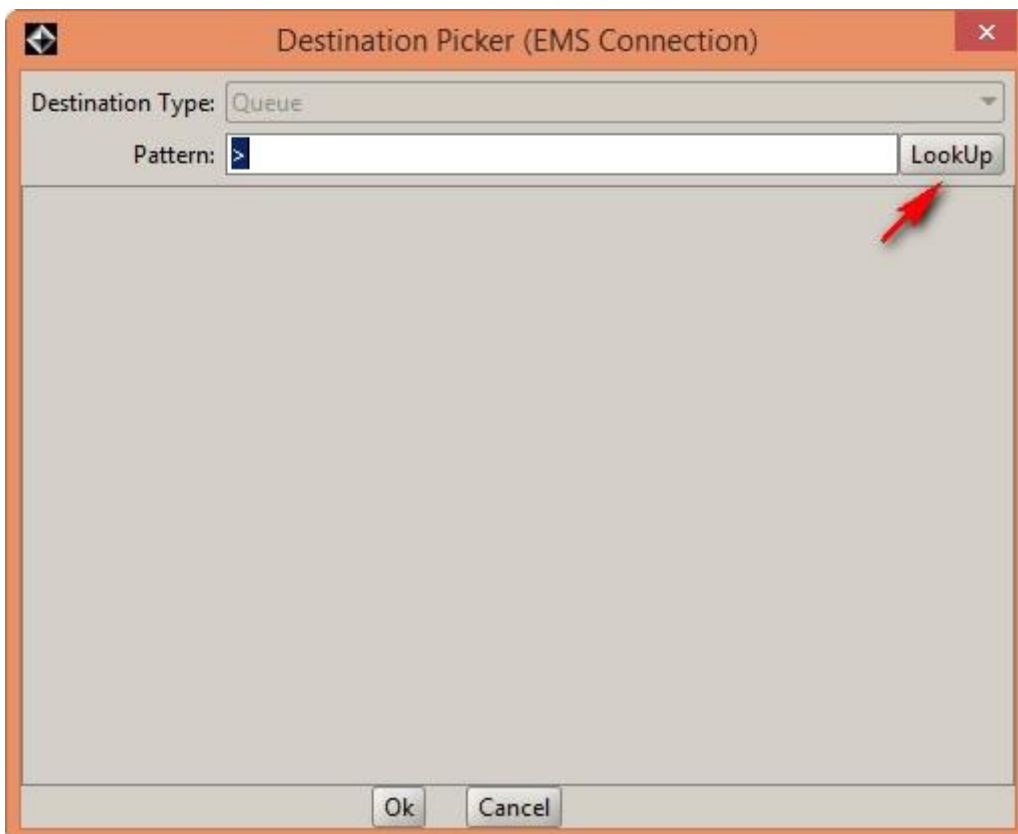
4. Выделить соединение – Queues – Queue Browser



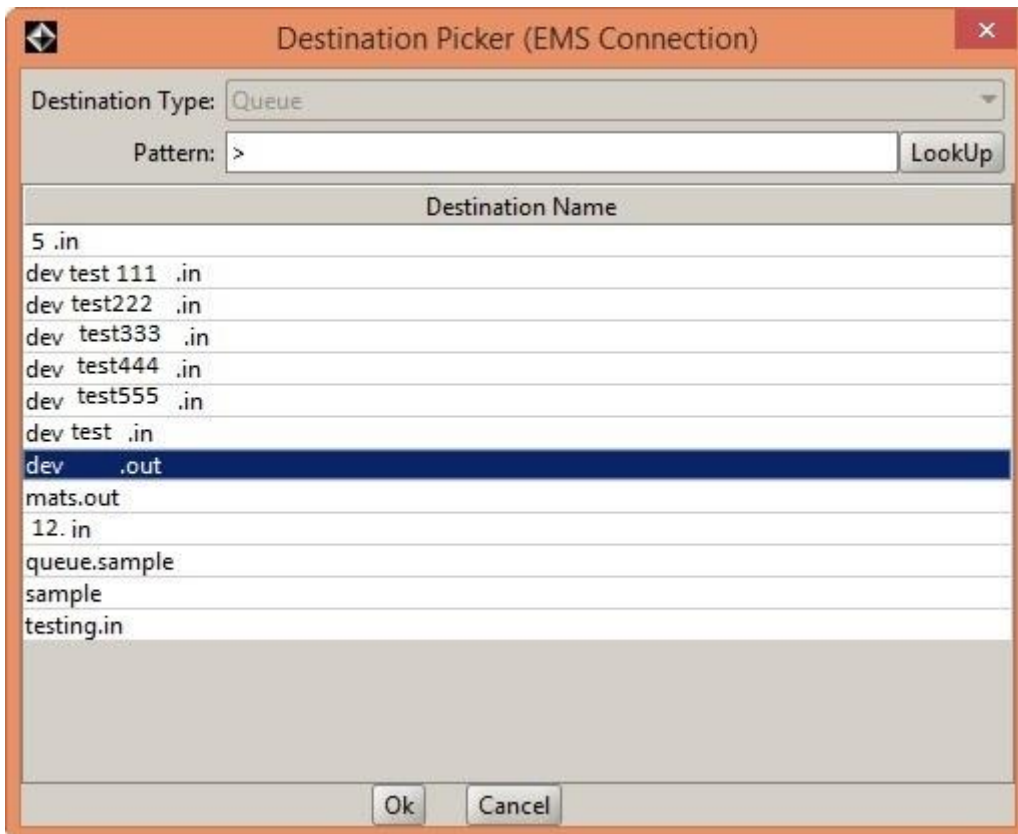
5. Нажать «...» около Queue Name



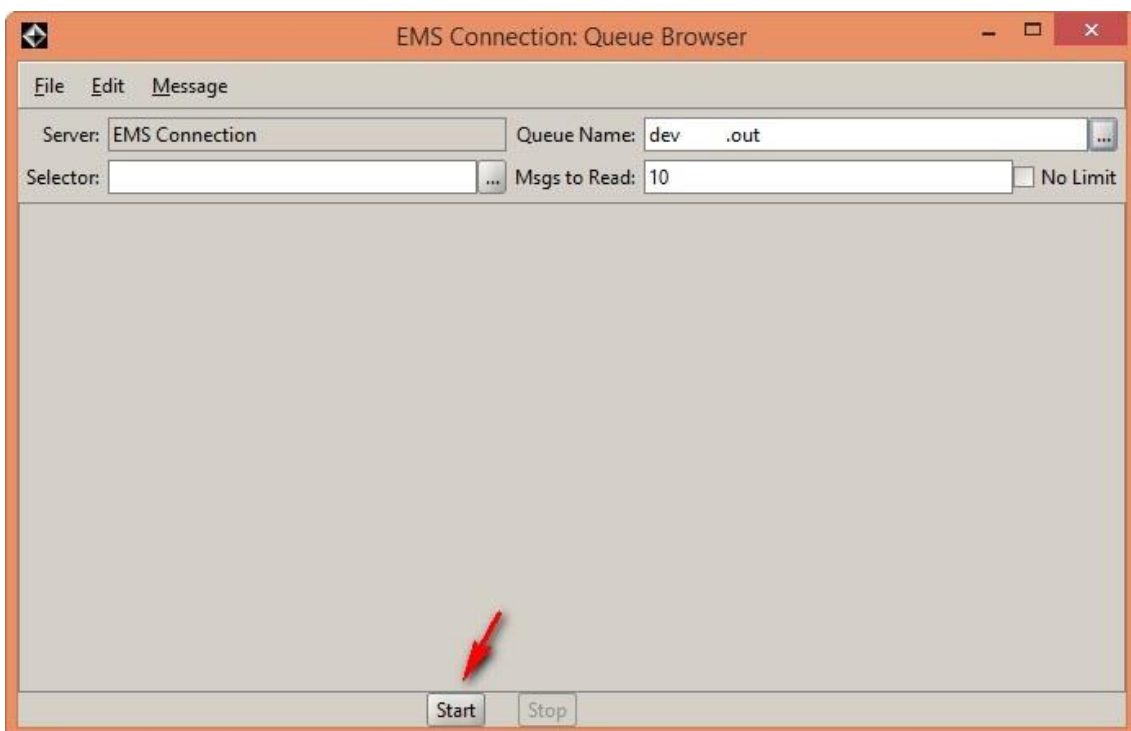
6. LookUp – система покажет все доступные очереди



7. Выбрать очередь, щелкнуть дважды. Выбираем исходящую очередь:

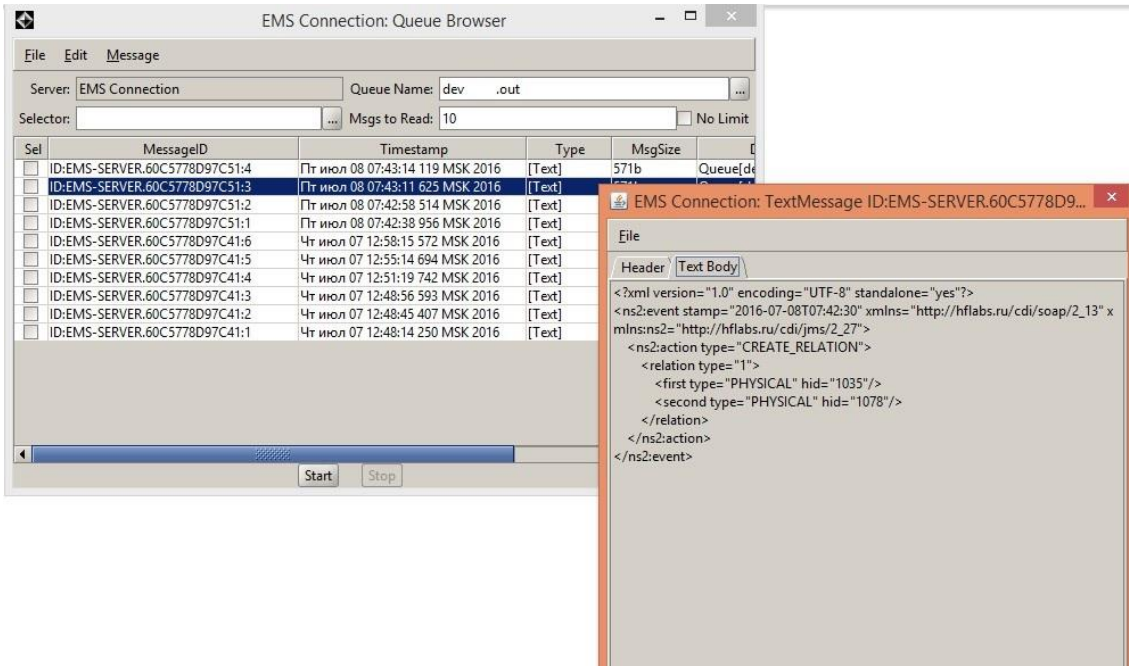


8. Нажать Start



9. Выбрать любое сообщение из списка. Если список пуст, сгенерить сообщение вручную, например, отправив запрос [save](#)

На вкладке Text Body будет текст JMS-события



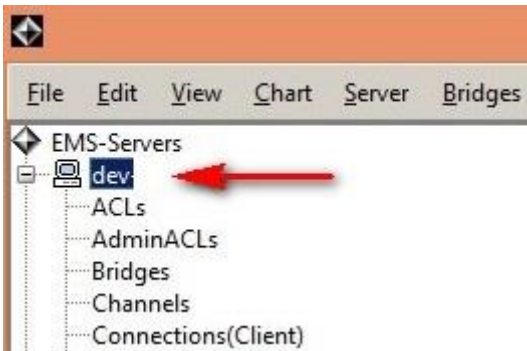
9.5.7 Как послать в Tibco входящее JMS-сообщение

9.5.7.1 Подготовка

Установить утилиту Gems: распаковать из [архива](#), подложить в папку lib джарники для вашей версии Tibco.

9.5.7.2 Отправка входящего JMS-сообщения со стандартными заголовками

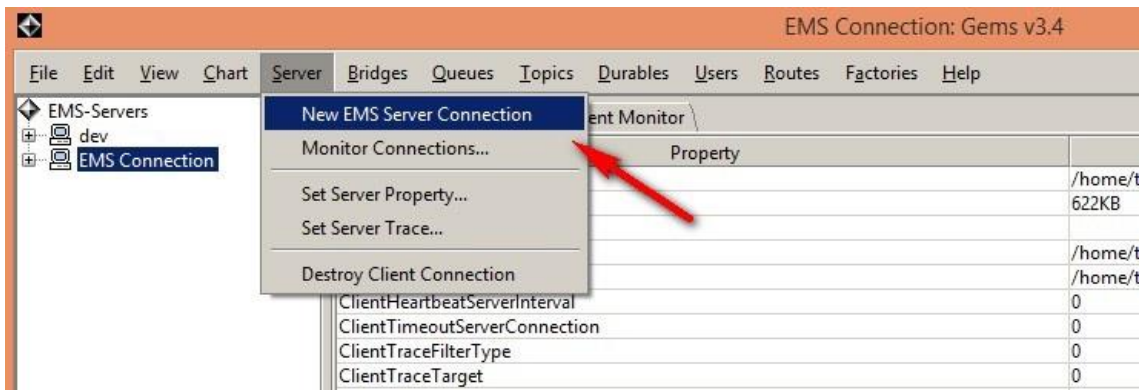
1. Запустить Gems – `rungems.bat`
2. Если уже есть EMS Server Connection – выделить его



Если соединения еще нет, создаем:

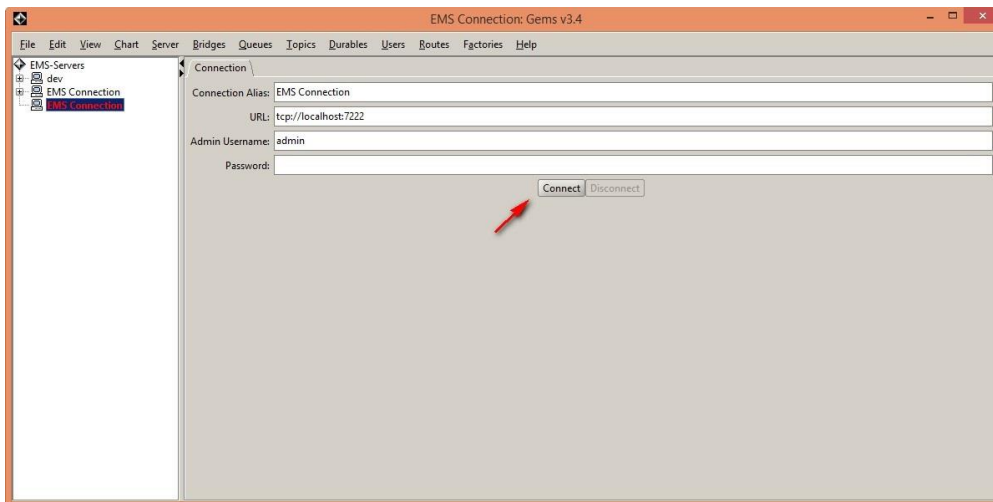
Как создать

Server – New EMS Server Connection

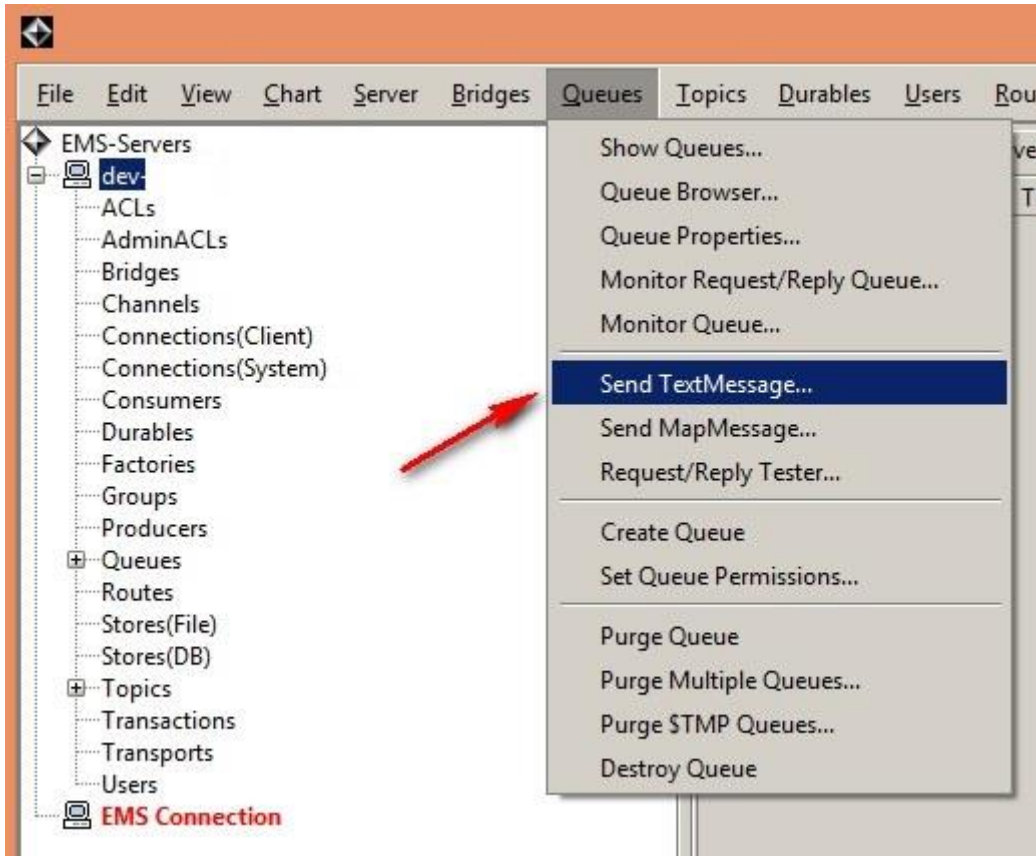


Указать настройки подключения, нажать Connect:

- Alias – имя, которое будет отображаться слева в меню
- URL – URL подключения к Tibco
- Username\password – данные для подключения



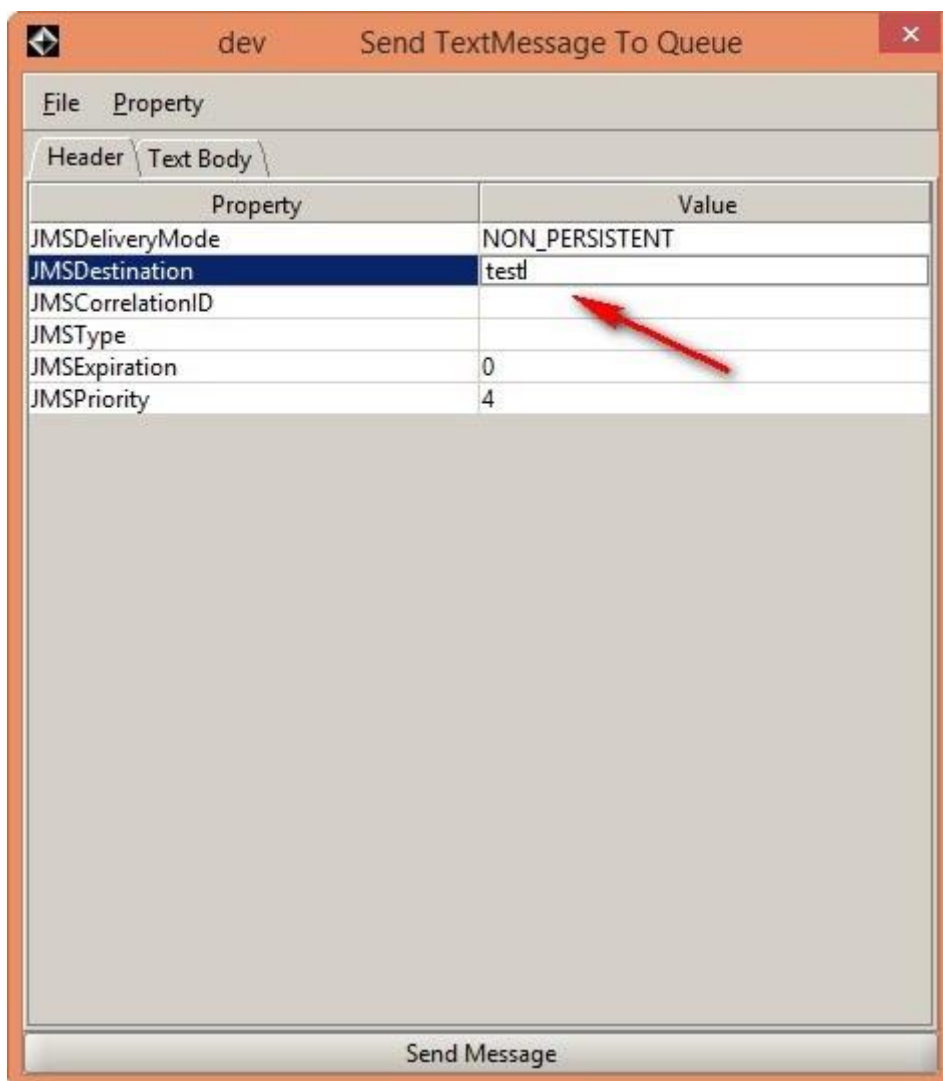
3. Queues – Send TextMessage



4. Указать параметр JMSDestination – место назначения.

Для EK в админке настроить то же место назначения:

- Конфигурация – Параметры JMS – `jms.soap.CFT.destination.name` (это для системы CFT, для других аналогично).
- Пример названия – `dev-open.CFT.in` (занимать уже существующее нельзя, а то разные машины будут воевать за эту очередь)



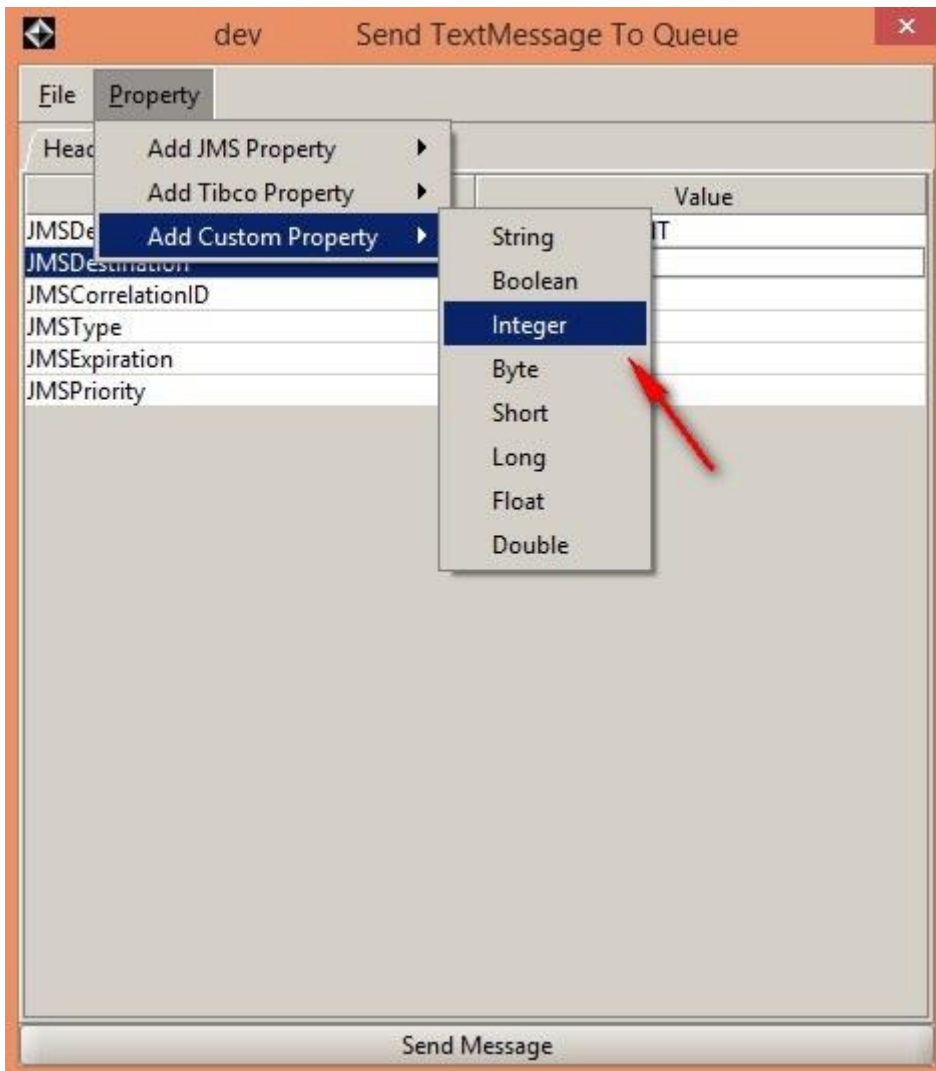
5. Все, стандартные заголовки есть, переключаемся на вкладку Text Body и отправляем запрос. Запрос обычный, без особых тегов, можно полностью скопировать из Soap Ui.

9.5.7.3 Нестандартные заголовки

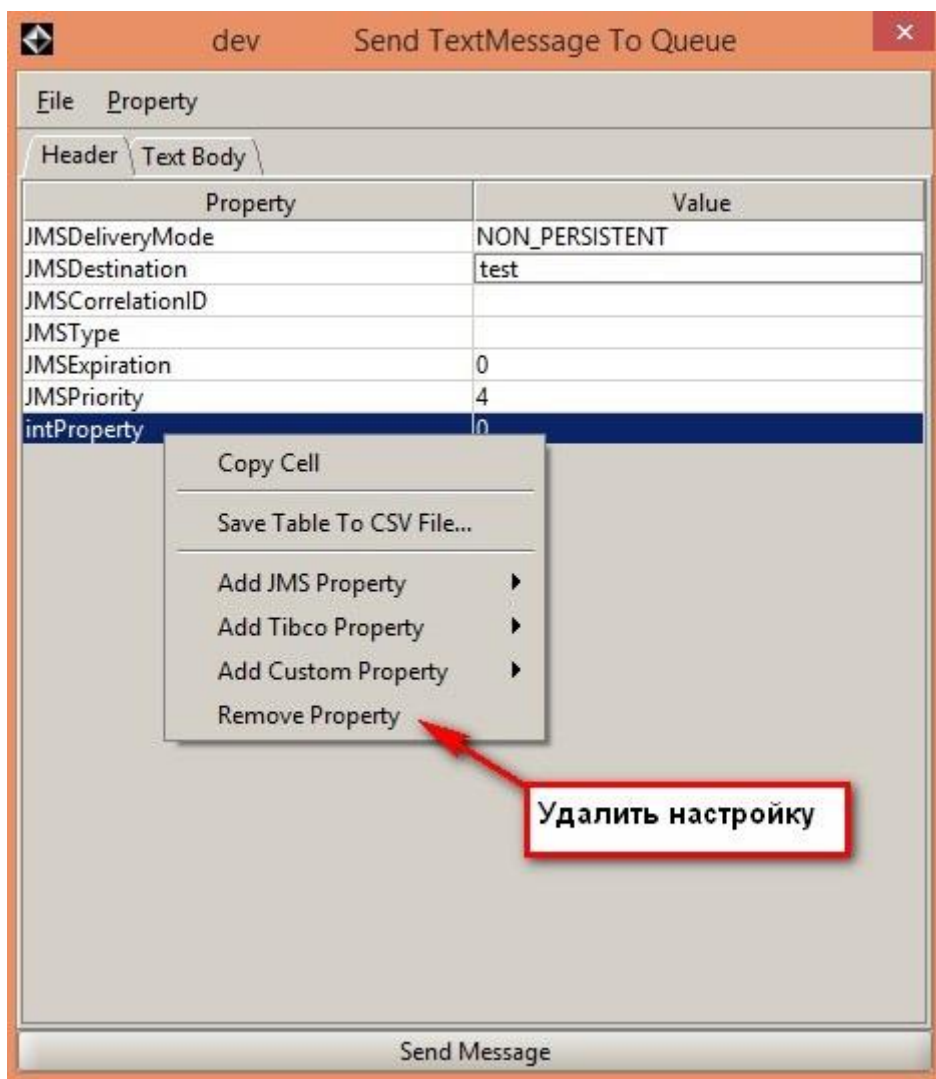
Если у нас не один читатель очереди, а 10, их надо как-то разделять. Для этого во входящее сообщение добавляется новый параметр в заголовке. Допустим, это будет ID с типом Integer.

Как проверить, что можно послать в приложение сообщение с ID в заголовке?

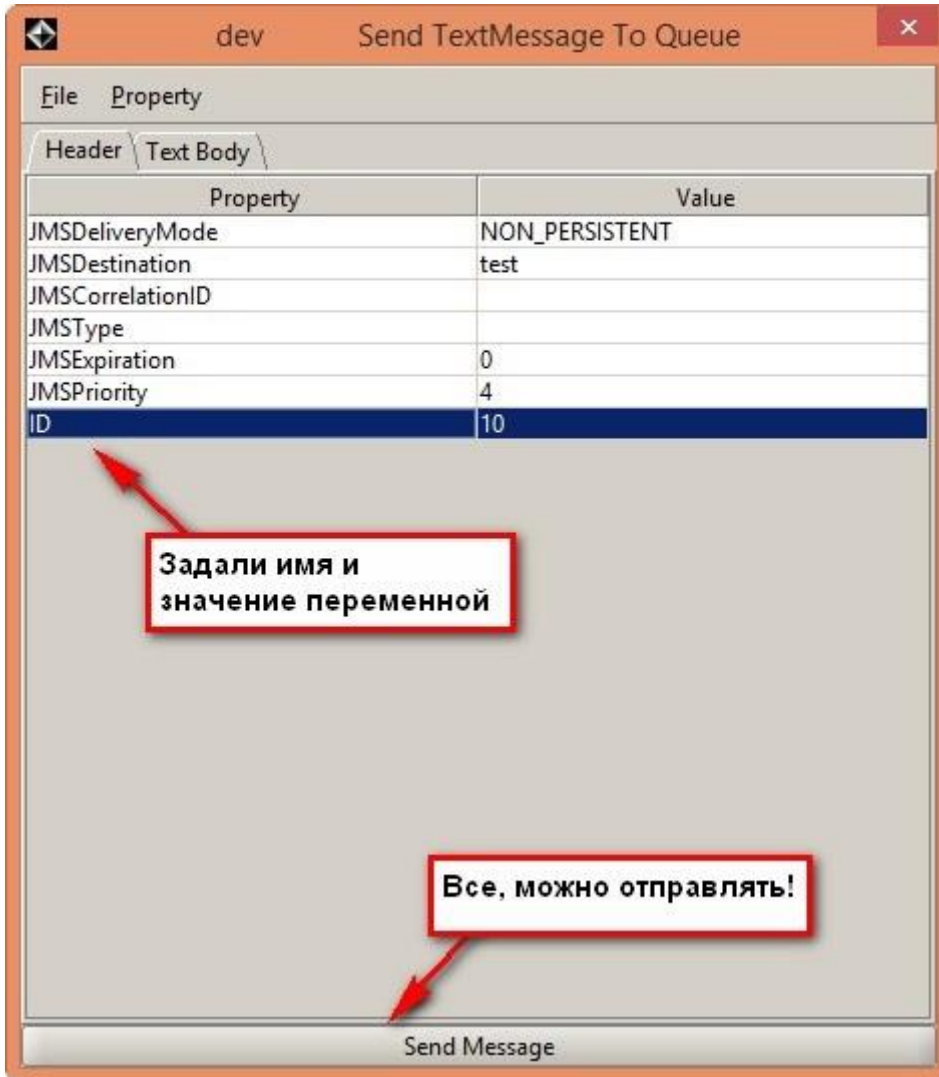
1. Queues – Send TextMessage
2. Property – Add Custom Property – <выбираем тип>



Настройка добавлена! При желании ее можно удалить по правой кнопке мыши. Стандартные заголовки удалять нельзя



3. Указываем имя и значение переменной – Send Message



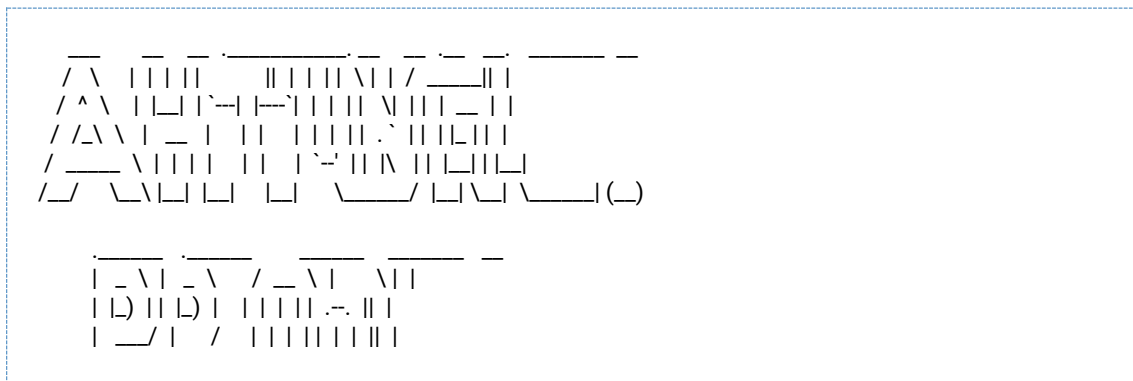
Все, наше сообщение отправлено с новым заголовком!

9.5.8 Как настроить приветствие на Linux

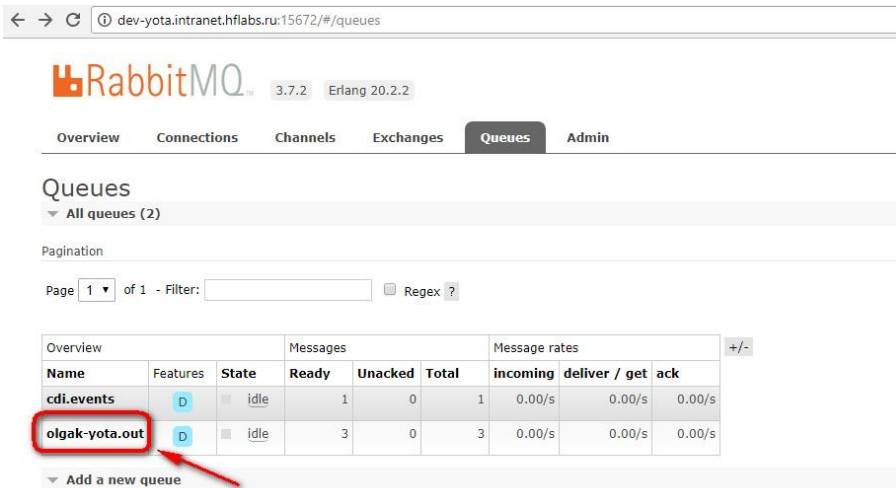
Если TEST и PROD находятся оба на linux-серверах, можно перепутать — пойти тестировать, а попасть на продакшен. Чтобы не тронуть ничего лишнего, настройте себе на PROD отдельное приветствие в командной строке.

9.5.8.1 Настройка

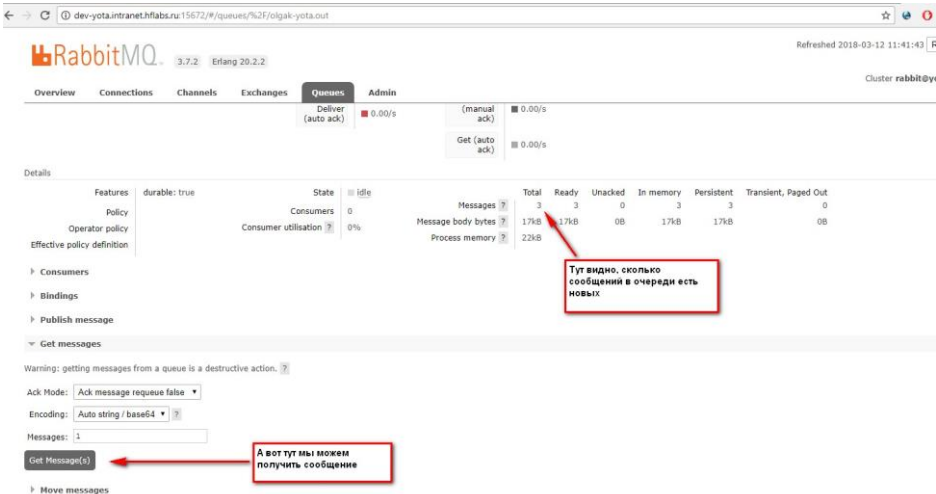
Указать приветствие в файле /etc/motd



2. Выбрать ту очередь, что настроили в destination.name:



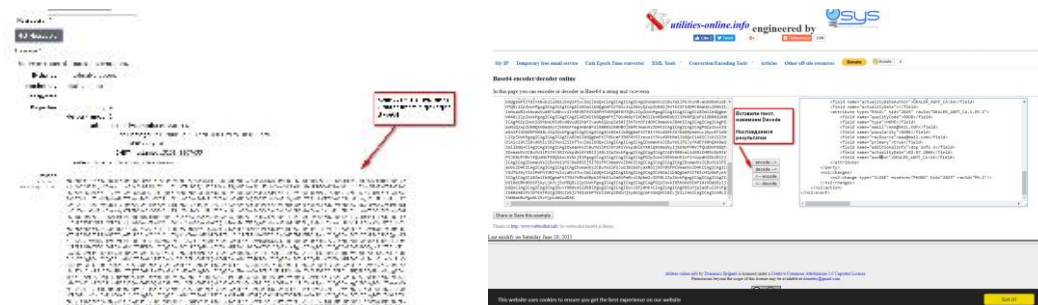
3. Нажать «Get Message» – получите сообщения или ошибку «очередь пуста». Количество сообщений видно сверху.



Как только вы получили сообщение, оно пропадает из очереди → это нормально. Это в тибке сообщения остаются, а тут если получил, то второй раз не посмотришь из интерфейса.

4. Скопировать сообщение (оно в кодировке Base64) и декодировать его. Сервисы декодирования:

- <http://www.utilities-online.info/base64/#.WqZNgGrFK70> → более удобный
- <https://www.base64decode.org/>



9.6 Запуск задач / триггеров

В: Уточните, пожалуйста. Задачу `passportInvalidFlagActualization` безопасно в рабочее время запускать?

О: Запускать в рабочее время безопасно, идти по ветке «Обработка всей базы».

Перед проверкой рекомендую [обновить](#) перечень недействительных документов на серверах, если этот процесс ещё не автоматизирован

9.6.1 Фазы миграции на `_HIST` таблицы

В: Как проходит миграция? Зачем она нужна? Что означают все эти фазы?

9.6.1.1 Зачем нужна миграция

Переезд на исторические таблицы нужен для переноса части БД на медленный диск. Все таблицы, в которых хранились версии будут разделены на две:

PHYSICAL_PARTY → PHYSICAL_PARTY + PHYSICAL_PARTY_HIST

В PHYSICAL_PARTY останутся только актуальные клиенты (`version = 0`), а все старые версии (изменения имени, даты рождения, даты актуальности и другие) будут храниться отдельно.

Это поможет:

1. Ускорить задачи:
 - a. Перестроение индекса — не нужно будет делать отсев прошлых версий в селекте
 - b. Перестроение мат. view — чем меньше данных в таблице, тем быстрее работает select
2. Купить под актуальные данные в БД SSD-диск, а старые хранить на обычном. SSD стоит дорого, покупать его под большую базу невыгодно.

Ускорение будет заметно только на больших базах, где уже накопилась история: записей с `version ≠ 0` сильно больше, чем актуальных (`version = 0`).

9.6.1.2 Как проходит миграция

Миграция проходит в 4 фазы, чтобы избежать длительного даунтайма сервиса.

9.6.1.2.1 Фаза 1. Создание таблиц `_ACT`

STAGING

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING_ACT

ID_STAGING	HID_STAGING	VERSION

Создаем пустой клон версионной таблицы с префиксом `_ACT` с такими же полями и размерностью.

Например, была таблица:

STAGING

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

Рядом создается пустая таблица:

STAGING_ACT

ID_STAGING	HID_STAGING	VERSION

Операция повторяется для каждой версионной таблицы.

Фаза проходит быстро и автоматически. Сотрудники ХФЛабс добавляют скрипты автомиграции в поставку, вручную ничего делать не нужно.

9.6.1.2.2 Фаза 2. Заполнение таблиц _ACT

STAGING


ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING_ACT

ID_STAGING	HID_STAGING	VERSION
1026	1024	0



В ней мы заполняем _ACT таблицы нулевыми версиями. То есть происходит копирование актуальных записей (version = 0) из версионной таблицы.

 Важно иметь в виду, что удаления не происходит, поэтому на данном этапе объем БД увеличивается.

После фазы 2 в нашем примере будут таблицы:

STAGING (не изменилась)

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING_ACT (скопировали сюда актуальную версию)

ID_STAGING	HID_STAGING	VERSION
1026	1024	0

Пока это подготовительный этап, система все еще живет на старых таблицах.

Наполнение _АСТ таблиц происходит двумя путями:

- самостоятельно — любое изменение / создание клиента инициирует его перенос в актуальные таблицы;
- принудительно — если запустить задачу [migratePhase1](#), система начнет заполнять таблицы сама.

То есть можно просто ждать, пока система сама фоном все отмигрирует, постепенно обновляя данные, но этот процесс будет идти очень долго. Или форсировать миграцию и запускать по ночам или на выходные migratePhase1. Система при этом продолжает работать как раньше, в привычных таблицах никаких изменений не видно.

9.6.1.2.3 Фаза 3. Переезд на новые таблицы

STAGING_HIST
~~STAGING~~

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING
~~STAGING_ACT~~

ID_STAGING	HID_STAGING	VERSION
1026	1024	0

Самый сложный этап — переезд на новые таблицы. На этой фазе придется гасить сервер, причем обе ноды в режиме горячего резерва.

На этом этапе:

- все текущие таблицы переименовываются в _HIST. STAGING → STAGING_HIST;
- все актуальные таблицы переименовываются, оттуда удаляется префикс _ACT. STAGING_ACT → STAGING;
- все view пересоздаются, чтобы они умели работать с новой структурой БД.

Команда ХФЛабс выдает скрипт ручной миграции. Нужно остановить сервис и запустить его.

9.6.1.2.4 Фаза 4. Удаление дублированных данных

Эта фаза необратимо удаляет данные. Поэтому сначала надо убедиться, что все прошло хорошо:

- неделю пожить в фазе 3
- перед запуском задачи обязательно снять бекап с базы

STAGING_HIST

ID_STAGING	HID_STAGING	VERSION
1024	1024	1
1025	1024	2
1026	1024	0

STAGING

ID_STAGING	HID_STAGING	VERSION
1026	1024	0

Теперь, когда все работает, можно удалить дублирование данных. Администратор запускает задачу [migratePhase3](#), в результате:

Система удаляет все нулевые версии из _HIST таблиц. После окончания переезда получаем структуру:

- STAGING – актуальная версия;
- STAGING_HIST – исторический объем.

9.6.2 Что делать, если инкремент упал с ошибкой

9.6.2.1 Вариант 1 – исправление проблемы

1. Посмотреть в админке или в логах текст ошибки.
2. Найти «виноватых» и устранить ошибку.
3. Сделать перевыгрузку данных

И все хорошо, инкремент прогружается, oldMaxId меняется. Собственно, он потому и не меняется при ошибках – чтобы была возможность исправить инкремент и снова запустить задачу.

9.6.2.2 Вариант 2 – пока поправить не можем

Бывает такое, что устранить ошибку быстро нельзя. Тогда нужно [деактивировать исходные записи](#) → инкремент прогрузится успешно, а записи всегда можно будет найти.

1. Отредактировать задачу [disableStagingRecords](#), указав ей список *stagingHids*, которые надо деактивировать. *stagingHid* можно найти в логах, там перечислены все проблемные случаи.
2. Запустить задачу.
3. Запустить задачу загрузки инкремента → она пройдет успешно, не пытаясь трогать «плохие записи», счетчик oldMaxId обновится.
4. После исправления проблемы [активировать записи](#) и запустить задачу *transform*.

Но резюме такое, что ошибки в системе не надо оставлять, с ними надо разбираться

9.7 FAQ для сопровождающих

1. Нужно добавить задачу / триггер прямо в war-файл хотфиксом, как?

WAR-ник → WEB-INF → classes

9.8 Фактор (FAQ)

9.8.1 Есть ли doBatchClean для телефонов?

Q: Можно ли вызвать [CleanService.phone](#) через doBatchClean? Как?

A: Можно, как и любой другой меппинг. Пример:

Запрос

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cle="http://clean.soap.cleaner.hflabs.ru">
  <soapenv:Header/>
  <soapenv:Body>
    <cle:doBatchCleanRequest>
      <!--1 or more repetitions-->
      <data>
        <dataFields>id 1</dataFields>
        <dataFields>домашний</dataFields>
        <dataFields>HOME</dataFields>
        <dataFields>74952235443</dataFields>
      </data>
      <data>
        <dataFields>id 1</dataFields>
        <dataFields>мобильный</dataFields>
        <dataFields>OTHER</dataFields>
        <dataFields>79262235443</dataFields>
      </data>
      <!--Mapping-->
      <mapping>clean-phone</mapping>
    </cle:doBatchCleanRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Ответ

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:doBatchCleanResponse xmlns:ns2="http://clean.soap.cleaner.hflabs.ru">
      <data>
        <dataFields>id 1</dataFields>
        <dataFields>домашний</dataFields>
        <dataFields>HOME</dataFields>
        <dataFields>74952235443</dataFields>
        <dataFields>7</dataFields>
        <dataFields>495</dataFields>
        <dataFields>2235443</dataFields>
      <dataFields/>
    </ns2:doBatchCleanResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```

<dataFields>GOOD</dataFields>
<dataFields>PROBABLY_MOBILE</dataFields>
<dataFields>ПАО "Мобильные ТелеСистемы"</dataFields>
<dataFields>Москва</dataFields>
<dataFields/>
<dataFields>РОССИЯ</dataFields>
<dataFields>МОСКВА</dataFields>
<dataFields/>
<dataFields>МОСКВА</dataFields>
<dataFields/>
<dataFields>UTC+3</dataFields>
<dataFields>MSK+0</dataFields>
</data>
<data>
  <dataFields>id 1</dataFields>
  <dataFields>мобильный</dataFields>
  <dataFields>OTHER</dataFields>
  <dataFields>79262235443</dataFields>
  <dataFields>7</dataFields>
  <dataFields>926</dataFields>
  <dataFields>2235443</dataFields>
  <dataFields/>
  <dataFields/>
  <dataFields>GOOD</dataFields>
  <dataFields>MOBILE</dataFields>
  <dataFields>ПАО "МегаФон"</dataFields>
  <dataFields>Москва и Московская область</dataFields>
  <dataFields/>
  <dataFields>РОССИЯ</dataFields>
  <dataFields>Москва и Московская область</dataFields>
  <dataFields/>
  <dataFields/>
  <dataFields/>
  <dataFields>UTC+3</dataFields>
  <dataFields>MSK+0</dataFields>
</data>
</ns2:doBatchCleanResponse>
</soapenv:Body>
</soapenv:Envelope>

```

9.9 Интеграция (faq)

9.9.1 Как передать двух клиентов с одной связью

Вопрос: как передать в онлайне двух новых клиентов, если между ними есть взаимосвязь?

- Создаешь А со связью А-Б — падает на том, что Б еще не существует.
- Создаешь Б со связью Б-А — ошибки нет

Передавать А без связи неправильно, ведь выгружается объект целиком.

Ответ: Чтобы ошибки не было исходно, можно передать обоих клиентов сразу через запрос [saveAndMerge](#), тогда ошибки не будет. Просто при выгрузке из своей системы, если видим, что есть связь, передаем сразу обоих клиентов

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="http://hflabs.ru/cdi/soap/2_13">
  <soapenv:Header/>
  <soapenv:Body>
    <saveAndMerge>

      <party type="PHYSICAL" sourceSystem="SOZ" rawId="1">
        <field name="fullNameRawSource">Первый Олег Михайлович</field>
        <relation type="4">
          <second type="PHYSICAL" sourceSystem="SOZ" rawId="2"/>
        </relation>
      </party>

      <party type="PHYSICAL" sourceSystem="SOZ" rawId="2">
        <field name="fullNameRawSource">Второй Олег Михайлович</field>
        <relation type="4">
          <first type="PHYSICAL" sourceSystem="SOZ" rawId="1"/>
        </relation>
      </party>

    </saveAndMerge>
  </soapenv:Body>
</soapenv:Envelope>
```

9.10 Обновление справочников (FAQ)

9.10.1 Я обновил перечень ПНП на Linux, но он не подхватывается

Q: Обновил перечень ПНП по [инструкции](#). Подложил справочник в директорию `/home/invalid_doc`, создал `.ready`-файл, но ничего не происходит. Почему?

A: Основные причины:

- Не указана переменная `FACTOR_PASSPORT_INVALID_PATH` в `FACTOR_HOME/bin/standalone.conf`.
- Справочник подкладывался под пользователем `root`. Тогда `factor` просто не будет иметь доступ к нему. Дайте права пользователю Фактора. Пример команд:

Code Block 12 Linux

```
-- закинули справочник под root
chown -R factor:factor /home/invalid_doc
su factor
cd /home/invalid_doc
touch list_of_expired_passports.csv.bz2.ready
```

10 Как добавить или изменить новый тип атрибута

10.1 Создать новый тип атрибута

Для создания нового типа необходимо добавить новый тип в ЕК и Фактор.

10.1.1 Добавить новый тип в ЕК

Выполнить процедуру `create_reference_record`. После чего рестартануть сервис **или** подождать час, чтобы ЕК подхватил изменения.

Создадим новый тип телефона, например, «Телефон поручителя»:

```
BEGIN
reference_pkg.create_reference_record(
  p_ref_type => 'PHONE_TYPE',      -- Тип классификатора
  p_code     => 'GUARANTOR',       -- Литера, будет отображаться в SOAP
  p_ordinal  => 10,                -- Числовой код, будет храниться в БД
  p_label    => 'Телефон поручителя' -- Краткое описание, будет отображаться в АРМ
  Менеджера данных
);
END;
```

В процедуре задать [тип классификатора](#) — тип которые хотим пополнить. Задать числовой код, по которому можно будет искать значение в базе. Значение не должно пересекаться с уже существующими. Если новый код добавляется для заказчика, а не в core, тогда использовать нумерацию более 50.

Проверить, какие коды есть для телефонов можно:

```
select * from reference_view
where ref_type = 'PHONE_TYPE';
-- интересуется колонка ordinal
-- значения: от 0 до 7
```

10.1.2 Добавить новый тип в Фактор

Добавить в исходники Фактора: `/factor/yota/resources/contactTypes.txt` новое значение.

После чего собрать обновленную версию Фактора и установить на стенд.

10.1.3 Результат

Теперь телефон можно создать через SOAP, указав значение `p_code`

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="http://hflabs.ru/cdi/soap/2_13">
  <soapenv:Header/>
  <soapenv:Body>
    <save>
      <party type="PHYSICAL" sourceSystem="AL" rawId="14400">
```

```
<field name="fullNameRawSource">Гарантийная Алина Григорьевна</field>
<attribute type="PHONE">
  <field name="type">GUARANTOR</field>
  <field name="rawSource">89266655444</field>
</attribute>
</party>
</save>
</soapenv:Body>
</soapenv:Envelope>
```

В АРМ Менеджера данных будет отображаться значение p_label