

Единый Клиент  
Руководство администратора

# СОДЕРЖАНИЕ

<b>РУКОВОДСТВО АДМИНИСТРАТОРА</b> .....	<b>4</b>
Требования к администратору CDI .....	5
Обучение для администраторов .....	6
Архитектура системы .....	8
Отказоустойчивое решение (горячий резерв) .....	10
Вход в систему (администратор) .....	12
Задачи обработки данных .....	13
Планировщик задач .....	23
Взаимодействие с JMS-очередями .....	27
Импорт данных .....	34
Идентификация и слияние дубликатов .....	57
Повторная обработка .....	75
Системные задачи .....	156
Экспорт данных .....	179
Диагностика .....	184
Триггеры .....	210
Возобновляемые задачи .....	217
Непрерываемые задачи .....	218
Проверка по черным спискам .....	219
Необратимое удаление данных .....	226
Закрытие контрагентов .....	228
Порционные задачи .....	229
Проверка по ЕГРЮЛ .....	231
Массовые атрибуты .....	236
Миграция .....	239
Миграция структуры хранения исторических данных .....	250
Сбор статистики .....	254
Фоновые задачи .....	265
Задача массовой обработки конфликтов .....	268
Полное необратимое удаление ВСЕХ данных выбранного типа .....	270
Общие параметры задач .....	271
Замена материализованного представления UNITED_VIEW простым .....	275
Начальное наполнение таблицы exp_party_history .....	276
Админ-панель 3.0 (beta) .....	277
Закрытие неактуальных записей ЧС .....	291
Актуализация ИНН из ФНС .....	292
Настройка системы .....	304
Мэппинг ролей LDAP .....	306
Настройки задач во внешней директории .....	309
Параметры JMS .....	310
Параметры LDAP .....	314
Параметры SOAP и REST сервисов .....	316
Параметры задач .....	319
Параметры задачи дедупликации .....	323
Параметры запуска JBOSS .....	324
Параметры импорта из буферной области .....	325
Параметры импорта справочников .....	326
Параметры модуля мониторинга .....	327
Параметры отправки email .....	328
Параметры подключения к Фактору .....	333
Параметры поискового индекса .....	335
Параметры проверки в ФНС .....	340
Параметры сервисов .....	341

Параметры синхронизации нод	348
Параметры экспорта данных	349
Сессии пользователей	350
Работа со справочниками	351
Журналирование и аудит	354
Журналы «Единого клиента»	355
Журналы Фактора	387
Уровни протоколирования	392
Аудит	393
Установка обновлений	396
Обновление версии Java Runtime Environment	401
Обновление модуля «Единый клиент»	402
Обновление модуля «Фактор»	403
Автоматическая миграция на новую версию	404
Инструкция по переходу с JBoss 4 на JBoss 7 для администраторов	408
Инструкция по переходу с JBoss 7 на WildFly 8 для администраторов	417
Инструкция по переносу дампа БД	424
Автоматическое создание схемы CDI	427
Инструкция по переходу с Wildfly 8 на WildFly 10 для администраторов	429
Инструкция по переходу на openJDK 11 и WildFly 16 для администраторов	437
Инструкция по обновлению модулей «ЕК» и «Фактор» для Linux	510
Инструкция по обновлению модулей «ЕК» и «Фактор» для Windows	512
Инструкция по обновлению модулей «ЕК» и «Фактор» для Windows на двух нодах	514
Инструкция по обновлению модулей «ЕК» и «Фактор» для Linux на двух нодах	518
Инструкция по обновлению модулей «ЕК» и «Фактор» для Linux (23.19+)	521
Инструкция по обновлению модулей «ЕК» и «Фактор» для Linux на двух нодах (23.19+)	523
FAQ	526
Основное (FAQ)	528
Менеджер Данных (FAQ)	536
АРМ Администратора (FAQ)	540
Дополнительно (FAQ)	543
Инструменты для проверки (FAQ)	547
Запуск задач / триггеров	593
FAQ для сопровождающих	600
Фактор (FAQ)	601
Интеграция (faq)	604
Обновление справочников (FAQ)	606
Как добавить или изменить новый тип атрибута	608
Задать новый тип согласия через базу данных	609
Создать новый тип атрибута	610
Обновление справочников	612
Загрузка и обновление ЕГРЮЛ	613
Обновление справочников Подсказок	616
Обновление справочников Фактора	633
Шпаргалка по настройке апдейтера	663
Беспарольная авторизация между серверами	668
Инструкции	671
Подсказки. Подключение почтовых оповещений (Linux)	672
Инструкции по сбору и проверке информации в рамках подготовки к переходу на Spring Boot	674

# РУКОВОДСТВО АДМИНИСТРАТОРА

# ТРЕБОВАНИЯ К АДМИНИСТРАТОРУ CDI

## *Требования*

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

## *Обязанности*

- Базовая настройка 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.
- Передача запросов пользователей в поддержку Единый Клиент и отслеживание их выполнения.

## *Доступы*

Администратору CDI необходимы права администратора на сервере приложений.

# ОБУЧЕНИЕ ДЛЯ АДМИНИСТРАТОРОВ

## ***Что входит?***

Для дальнейшего комфортного сопровождения и эксплуатации «Единого клиента», мы готовы провести специальное обучение, на котором расскажем подробнее про «Единый клиент», как он работает и какие задачи решает:

- Про архитектуру;
- Доступные для интеграции способы отправки и получения данных;
- Сценарии проверки и обработки данных внутри «Единого клиента».

А также про эксплуатацию «Единого клиента»:

- Поставку релизов и инструкции по обновлению;
- Доступные журналы логирования;
- АРМ Администратора и особенности запуска задач в нем;
- Вспомогательные утилиты для анализа проблем и типовые проблемы инфраструктуры.

**Обратите внимание:** Обучение адаптируется под каждую группу обучающихся отдельно, поэтому набор тем может меняться в зависимости от слушателей и ваших пожеланий. Если у вас есть какие-то вопросы / темы, которые вы хотите также обязательно затронуть на обучении, то напишите нам об этом — мы учтем ваши пожелания.

## ***Сколько на обучение нужно времени?***

Типовое обучение обычно занимает два дня по 4 часа.

Если есть потребность добавить в обучение практику, то это обсуждается и планируется время отдельно.

Если есть только меньший слот времени, тоже обсуждаемо и реально, но не желательно, так как это возможно только за счет сокращения программы обучения.

## ***Формат обучения?***

1. Zoom.

Плюс такого обучения: сохраняется запись обучения, которую можно пересматривать после или передавать новым коллегам.

2. Оффлайн ( у нас в офисе на м. Парк Культуры или на вашей площадке).

Для обучения нужна будет аудитория с проектором и возможностью подключения нашего ноутбука к нему, и в идеале дополнительно маркерная доска.

Плюс такого обучения: обычно идет более живое общение и обсуждение вопросов, проще разобрать на месте какие-то реальные кейсы и вместе порисовать что-то на доске.

## ***Как запросить такое обучение?***

Нужно написать на группу рассылки или аккаунту/аналитику/инженеру поддержки со стороны ХФ Лабс, что у вас есть запрос на обучение.

Рассказать в письме:

1. Сколько будет обучающихся и чем они занимаются?
2. Есть ли пожелания по темам / вопросам, которые необходимо включить в обучение?
3. Сколько времени готовы выделить на обучение? Базовый формат: 2 дня по 4 часа.
4. Желаемый формат обучения? Zoom или оффлайн?

Обычно на подготовку обучения требуется около 3 недель. Учитывайте это при планировании обучения, чтобы мы успели подготовить его и провести в ожидаемые вами сроки.

## ***После обучения:***

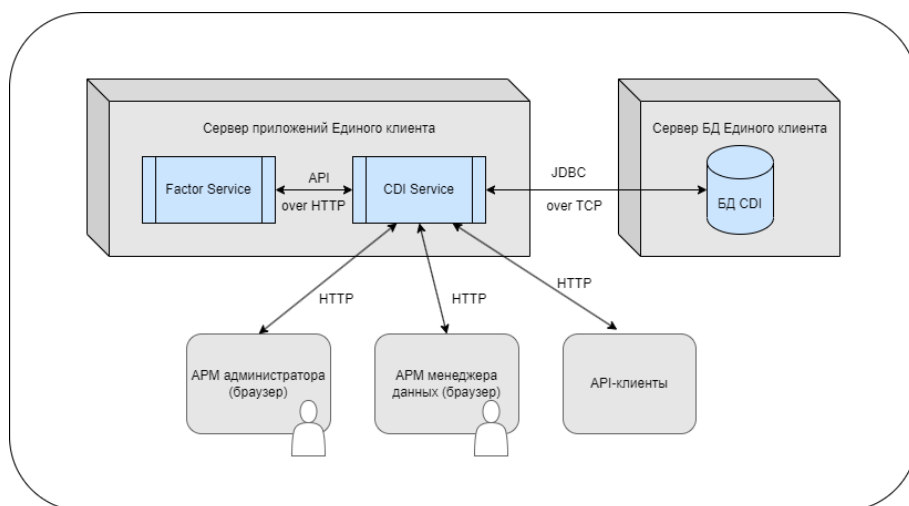
Обучающиеся получают сертификаты о пройденном обучении (если у вас есть такой запрос) .

Материалы лекции в виде pdf-файла, который можно использовать как шпаргалку для дальнейшего быстрого поиска нужной информации у нас в Confluence.

# АРХИТЕКТУРА СИСТЕМЫ

## Типовая схема развёртывания

На рисунке ниже представлена типовая схема развёртывания системы «Единый клиент»:



В разделе описана схема развёртывания одного экземпляра системы. Также возможна реализация отказоустойчивого решения за счет использования горячего резерва.

## Компоненты типовой схемы развёртывания

Компонент схемы	Назначение
APM Администратора	Автоматизированное рабочее место администратора системы. Реализовано как тонкий клиент, доступный через браузер.
APM 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.
CDI Service	Приложение «Единый клиент». Обеспечивает выполнение функций системы «Единый клиент».
Factor Service	Приложение «Фактор» Обеспечивает выполнение функций системы «Фактор».
БД CDI	База данных «Единый клиент». Реализуется на базе СУБД Oracle или PostgreSQL.



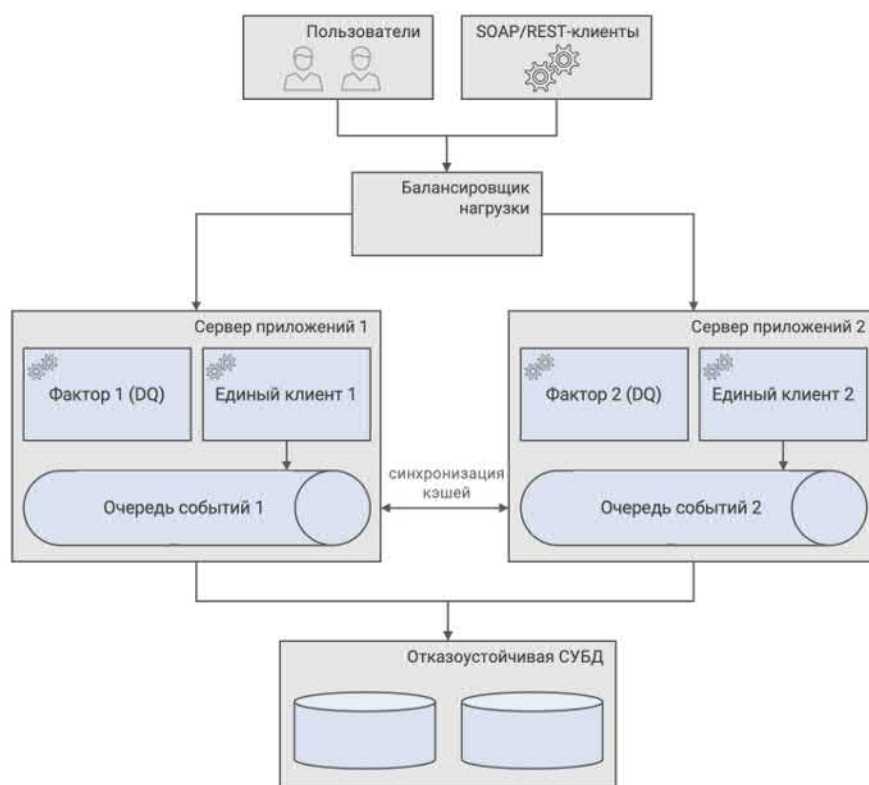
## Взаимосвязи типовой схемы развертывания

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.

# 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:



## 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 работает с БД.
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 БД.
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. Настроить отказоустойчивую конфигурацию СУБД (например, standby, Oracle RAC).
5. Настроить **мониторинг отдельных компонентов схемы**, и выделить необходимые человеческие ресурсы для оперативного реагирования на отказы.

### ***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.

# ВХОД В СИСТЕМУ (АДМИНИСТРАТОР)

Страница входа: [http://СЕРВЕР\\_ЕК:8080/cdi/](http://СЕРВЕР_ЕК: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

Войти

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*. Это дает право работать без прерывания сессии пользователя на месяц без перелога. Сессию можно прервать раньше, если выполнить *logout* (т.е. нажать *Выйти*).

Если галочка *Запомнить меня* не стоит, то прерывание сессии пользователя по умолчанию происходит через 10 мин при неактивности.

Чтобы изменить умолчательное время таймаута при непоставленной галочке *Запомнить меня*, необходимо изменить параметр `web.session.timeoutInSeconds` в блоке *Сессии пользователей* в панели администратора и сделать рестарт *Единого клиента*, чтобы новое значение параметра применилось.

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

# ЗАДАЧИ ОБРАБОТКИ ДАННЫХ

- Группы задач

Импорт данных

- Особенности
- Результат выполнения

Идентификация и слияние дубликатов

Проверка по черным спискам

- Алгоритм сброса статуса проверки по ЧС

Массовые атрибуты

Миграция

Повторная обработка контрагентов

Подготовка списков для повторной обработки

Системные задачи

Работа с задачами

Параметры, общие для всех типов задач

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

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

## 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
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	disableStagingRecords	Деактивировать исходные записи
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	enableStagingRecords	Активировать исходные записи
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	physicalBufferPC	Загрузка инкремента ФЛ
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	transform	Трансформация данных
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	transformWithCustomConfig	Трансформация данных с указанными параметрами

Типовые задачи, которые есть всегда:

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

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

### 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
----------	-----	----------	-----------------------

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

## 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".

- Задача загрузки контрагентов из буферной таблицы БД
- Задача загрузки контрагентов из CSV-файла
- Задача загрузки контрагентов из каталога
- Задача загрузки контрагентов по SFTP
- Задача загрузки договоров из буферной таблицы БД
- Задача загрузки связей договоров из буферной таблицы БД
- Деактивировать исходные записи
- Активировать исходные записи
- Задача загрузки связей
- Трансформация данных
- Задача загрузки справочников ЕГРЮЛ / ЕГРИП
- Общие параметры для задач загрузки данных
- Задача циклической загрузки данных из БД
- Задача загрузки справочника ЕФРСБ
- Загрузка инкремента из универсального буфера

## 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;
- **findGuaranteedDuplicatesAndMerge** — поиск гарантированных дубликатов и слияние;
- **egrCheck** — полная проверка по ЕГРЮЛ;
- **egrCheckActualize** — Проверка по ЕГРЮЛ контрагентов, требующих актуализации.

## ***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. Задача перестраивает онлайн хэши справочников ЧС, обновление осуществляется для всех справочников — даже для тех, которые не были изменены
- **grabBlacklistActualDate** — обновление дат актуальности ЧС
- **listCommonBufferLoad** — загрузка списков из буферных таблиц (с релиза 21.21, подключается отдельно)
- **listFilesImport** — загрузка данных из файла (с релиза 21.21, подключается отдельно)

## ***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).

## ***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

## ***Повторная обработка контрагентов***

- Актуализация атрибутов на основе данных staging
- Актуализация статуса невалидности документов
- Задача актуализации адресов по ФИАС
- Задача восстановления атрибутов по списку NIDов
- Задача восстановления КА на версию по списку NID
- Задача выливания исходных из состава массивных золотых карточек
- Задача заполнения технических полей
- Задача инкрементальной актуализации адресов по ФИАС
- Задача обработки массивных карточек контрагентов
- Задача очистки таблиц параметров для массовой переобработки данных
- Задача переноса открытых связей закрытых транзитивных карточек на активные золотые
- Задача переобработки UNDEF-телефонов
- Задача переобработки СНИЛС
- Задача переобработки ИНН ФЛ
- Задача переоткрытия связей
- Задача переочистки вручную подтверждённых ФИО физлиц
- Задача переочистки маркеров ФНС ФИО
- Задача переочистки реквизитов
- Задача перерасчета hidden-флагов атрибутов по списку контрагентов
- Задача переслияния реквизитов контрагентов
- Задача пересчета скоринговых значений атрибутов
- Задача поиска золотых карточек для переслияния
- Задача поиска массивных карточек контрагентов
- Задача полного переслияния контрагентов
- Закрытие актуальных (результатирующих и всех исходных) атрибутов по списку
- Закрытие исходных атрибутов по списку с переслиянием результирующих
- Заполнение полной ОПФ
- Заполнение ФИО ИП
- Исправьте базу данных в случае неконсистентного состояния
- Переобработка email
- Переобработка телефонов
- Переобработка телефонов. Новый вариант
- Переоткрытие закрытых по ошибке контрагентов с сохранением истории
- Переочистка email
- Переочистка гранулярных кодов качества ФИО

- Переочистка кодов качества дат рождения
- Переочистка кодов качества КПП
- Переочистка кодов качества паспортов
- Подготовка списков для повторной обработки
- Полная переочистка ФИО
- Полное разлияние контрагентов по списку результирующих контрагентов
- Разлияние контрагентов по списку MergedId
- Таблицы с параметрами задач массовой переобработки
- Задача переобработки ИНН ЮЛ

Таблицы с параметрами задач массовой переобработки.

Группа задач, предназначенных для повторной обработки определенного подмножества контрагентов.

Задача	Описание	Действия перед выполнением задачи	Действия после выполнения задачи
1 <b>hiddenFlagRefineByHids</b>	Задача перерасчета hidden-флагов атрибутов по списку контрагентов		
2 <b>reMergeFullByPartyHids</b>	Задача полного переслияния контрагентов		
3 <b>reMergeRequisiteByPartyHids</b>	Задача переслияния реквизитов контрагентов		
4 <b>unMergeByMergedIds</b>	Разлияние контрагентов по списку MergedId		
5 <b>fullUnmergeByResultHids</b>	Полное разлияние контрагентов по списку результирующих контрагентов		
6 <b>attributeRecoverByHids</b>	Задача восстановления атрибутов по списку НИДов		
7 <b>addressActualization</b>	Задача актуализации адресов по ФИАС		
8 <b>mergedPartiesCheckAndRemerge</b>	Исправьте базу данных в случае неконсистентного состояния		
9 <b>partyReopenByHids</b>	Переоткрытие закрытых по ошибке контрагентов с сохранением истории		
10 <b>attributeActualizeByStaging</b>	Актуализация атрибутов на основе данных staging		
11 <b>attributeCloseActualByHids</b>	Закрытие актуальных (результатирующих и всех исходных) атрибутов по списку		
12 <b>cleanTaskParameterTables</b>	Задача очистки таблиц параметров для массовой переобработки данных		

13	<b>fillTaskPartyHidsActual</b>	Подготовка списков для повторной обработки		
	<b>fillTaskPartyHidsOriginal</b>			
	<b>fillTaskPartyHidsResulted</b>			
	<b>fillAddressActualizationTaskPartyHids</b>			
	<b>fillTaskPartyHidsBadAdresses</b>			
	<b>fillTaskPartyHidsOriginalWithClosed</b>			
14	<b>passportInvalidStatusReCleaner</b>	Актуализация статуса невалидности документов		
15	<b>attributeCloseSourceByHids</b>	Закрытие исходных атрибутов по списку с переслиянием результирующих		
16	<b>phoneExtraFieldsReCleaner</b>	Переобработка телефонов		
17	<b>phoneExtraFieldsActualization.physical</b>	Переобработка телефонов. Новый вариант		
	<b>phoneExtraFieldsActualization.legal</b>			
18	<b>partyRecoverByHids</b>	Задача восстановления КА на версию по списку HID		
19	<b>reCleanMockRequisiteTask</b>	Задача переочистки реквизитов		
20	<b>phoneUndefReCleaner</b>	Задача переобработки UNDEF-телефонов		
21	<b>refreshScoresTask</b>	Задача пересчета скоринговых значений атрибутов		
22	<b>physicalFioQcReCleaner</b>	Переочистка гранулярных кодов качества ФИО	Заполнение таблицы актуальными КА	Удаление таблицы по завершении задачи
23	<b>passportQCReCleaner</b>	Переочистка кодов качества паспортов		
24	<b>physicalBirthdateQcReCleaner</b>	Переочистка кодов качества дат рождения	Заполнение таблицы актуальными КА	Удаление таблицы по завершении задачи
25	<b>relationReopenByIds</b>	Задача переоткрытия связей		
26	<b>legalIpReCleaner</b>	Заполнение ФИО ИП		
27	<b>physicalFioConfirmedManuallyReCleaner</b>	Задача переочистки вручную подтверждённых ФИО		

## Подготовка списков для повторной обработки

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

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

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);
- **fillTaskPartyHidsOriginalWithClosed** — все оригинальные контрагенты (**merged\_status** 0 или 1) и закрытые контрагенты (**end\_date != null**);
- **cleanTaskParameterTables** — очистка таблиц с параметрами задач массовой переобработки.

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

✔ Следующие задачи сначала очищают таблицу, с которой работают, и только после этого начинают наполнение данными. Поэтому предварительно выполнять задачу **cleanTaskParameterTables** не нужно.

Для наполнения таблицы **TASK\_ACT\_PARTY\_HIDS** предназначены отдельные задачи:

- **fillAddressActualizationTaskPartyHids** (отображается в АРМ Администратора только, если подключена **Задача актуализации адресов по ФИАС**) — контрагенты, адреса которых должны быть актуализированы по КЛАДР (**mergedStatus**: 0,2);
- **fillTaskPartyHidsBadAddresses** — контрагенты с адресами, отмеченными на ручную обработку (**mergedStatus** 0 или 2), задача не выбирает контрагентов, если все его адреса, отмеченные на ручную обработку, скрыты.

Зада **fillIncrementalAddressActualizationPartyHids** заполняет таблицу **TASK\_ADDRESS\_ACT\_HIDS** (отображается в АРМ Администратора только, если подключена Задача актуализации адресов по ФИАС) — контрагенты, адреса которых нуждаются в актуализации по КЛАДР;

## Системные задачи

- **changeJdbcUser** — изменение пользователя безопасности

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

- **deleteOutdatedConflicts** — удаление старых обработанных конфликтов
- **duplicateGroupActualize** — актуализация групп дубликатов
- **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
- **findHistoricalConflicts** — поиск конфликтных изменений

Группа задач, предназначенных для экспорта данных из Единого клиента:

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

Экспорт данных		
Действия	Название	Описание
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	fullExportRelations	Полный экспорт эталонных связей
<a href="#">Детали</a>   <a href="#">Редактировать</a>   <a href="#">Выполнить</a>	incrementalExportRelations	Инкрементальный экспорт эталонных связей

## Работа с задачами

Администратор имеет возможность:

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

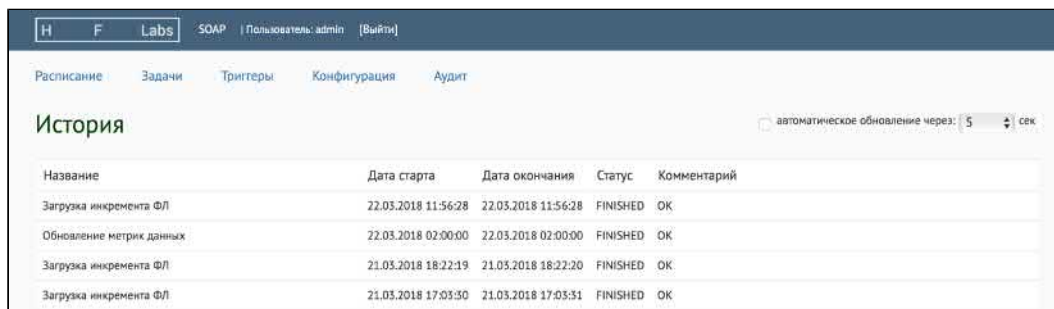
Выполнение любой задачи также может быть инициировано автоматически по заданному расписанию (триггеру).

### Параметры, общие для всех типов задач

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

# 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**.



Название	Дата старта	Дата окончания	Статус	Комментарий
Загрузка инкремента ФЛ	22.05.2018 11:56:28	22.05.2018 11:56:28	FINISHED	OK
Обновление метрик данных	22.05.2018 02:00:00	22.05.2018 02:00:00	FINISHED	OK
Загрузка инкремента ФЛ	21.05.2018 18:22:19	21.05.2018 18:22:20	FINISHED	OK
Загрузка инкремента ФЛ	21.05.2018 17:03:30	21.05.2018 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.

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

# 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						<input type="checkbox"/> автоматическое обновление через: 5 сек
Группа	Название	Статус	Описание	Комментарий	Действия	
system	searchEngineOptimize	PENDING	Оптимизация всех поисковых индексов		<a href="#">Завершить</a>   <a href="#">Детали</a>	

- *Убить* - жесткая остановка. Система принудительно завершает все процессы задачи. Несохраненный 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						<input type="checkbox"/> автоматическое обновление через: 5 сек
Группа	Название	Статус	Описание	Комментарий	Действия	
dedup	findDuplicatesAndMergeAndRebuildGroups	RUNNING	Поиск дубликатов, слияние и перестроение групп дубликатов		<a href="#">Убить</a>   <a href="#">Детали</a>	

# ВЗАИМОДЕЙСТВИЕ С JMS-ОЧЕРЕДЯМИ

## *Взаимодействие с JMS-очередями*

Группа задач, предназначенных для взаимодействия с JMS-очередями:

- Задача очистки хранилища JMS-сообщений
- Задача переобработки сохраненных сообщений из входной очереди
- Задача переотправки сообщений в выходную очередь
- Отправка новых JMS сообщений
- Повторная отправка ошибочных JMS сообщений за последние 24 часа

# ЗАДАЧА ОЧИСТКИ ХРАНИЛИЩА JMS-СООБЩЕНИЙ

CleanEventsPersistence

## 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).

## Конфигурация задачи

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

# ЗАДАЧА ПЕРЕОБРАБОТКИ СОХРАНЕННЫХ СООБЩЕНИЙ ИЗ ВХОДНОЙ ОЧЕРЕДИ

ReprocessIncomingEvents

## 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.

## Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
minCreate Date	Дата в формате <b>dd.mm.yyyy hh:mi:ss</b>	Нижняя граница (включительно) даты создания сообщений, которые будут переобработаны	
maxCreate Date	Дата в формате <b>dd.mm.yyyy hh:mi:ss</b>	Верхняя граница (включительно) даты создания сообщений, которые будут переобработаны	

# ЗАДАЧА ПЕРЕОТПРАВКИ СООБЩЕНИЙ В ВЫХОДНУЮ ОЧЕРЕДЬ

RepublishOutgoingEvents

## 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.

## Конфигурация задачи

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

# ОТПРАВКА НОВЫХ JMS СООБЩЕНИЙ

## publishPendingEvents

### Описание задачи

Отправляет в выходную очередь сохраненные JMS-сообщения в статусе PENDING.

Сообщения, сортируются по дате создания, и пишутся в один поток в очередь пакетами размером в **packSize** штук. После отправки каждого пакета делается пауза в **sleepMilliseconds** мс.

### Схема использования

Используется для управления потоком JMS-сообщений в выходную очередь, когда отключена синхронная отправка.

Без включения задачи сообщения не отправляются в очередь, но сохраняются для последующей отправки.

1) Задача устанавливается на триггер, запускается раз в определенный промежуток времени (например, 15 минут), начинает обрабатывать сохраненные события.

2) Если при попытке отправки возникают ошибки (недоступность шины) – при накоплении определенного числа ошибок задача падает.

Ошибочные сообщения также сохраняются для переотправки (для этого уже есть специальная задача).

3) Далее задача опять запустится по триггеру, если его не отключить. Триггер можно отключить до тех пор, пока доступность не будет восстановлена.

### Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>packSize</b>	Размер пакета сообщений для записи в очередь	1
<b>sleepMilliseconds</b>	Время ожидания (мс) между отправками пакетов сообщений	50
<b>maxSend</b>	Максимальное количество отправленных сообщений за один запуск задачи (включая ошибки), по умолчанию "-1" (без ограничения)	-1
<b>maxErrors</b>	Максимальное количество ошибок, после которого задача останавливается	100
<b>senderName</b>	Название отправителя. Если не указано, то отправляются события для всех отправителей.	

---

<b>stopWhenNoMoreMessages</b>	Остановить задачу, если больше не осталось сообщений к отправке. Если false, то задача работает постоянно, ожидая новых сообщений.	true
-------------------------------	--	------

---

\*Если параметр **stopWhenNoMoreMessages=true**, то задача отправляет только те сообщения в статусе PENDING, которые уже были в таблице JMS\_EVENTS на момент ее запуска. То есть события, которые появились в таблице во время работы задачи, будут обработаны только при следующем запуске.

## ***Конфликты параметров***

Параметр stopWhenNoMoreMessages и maxSend являются конфликтными.

Если параметр stopWhenNoMoreMessages = false, то maxSend должен быть равен -1.

В противном случае будет ошибка — **stopWhenNoMoreMessages=false can't be used with maxSend > 0**

## ***Обработка ошибок и логирование***

В итоговой статистике выводится:

---

**FINISHED task "; processed N entities; 0 with errors; took X seconds or X.X ms; speed XX.0 records /sec**

---



# ПОВТОРНАЯ ОТПРАВКА ОШИБОЧНЫХ JMS СООБЩЕНИЙ ЗА ПОСЛЕДНИЕ 24 ЧАСА

RepublishOutgoingEventsForTheLastDay

## 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.

## Конфигурация задачи

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

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

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

### 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
<b>notifyListeners</b>	Логическое значение	Оповещение поискового индекса КА о трансформации пачек. Если оповещение отключено, после завершения задачи нужно перестроить индекс.	<b>true</b>

## *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".

- Задача загрузки контрагентов из буферной таблицы БД
- Задача загрузки контрагентов из CSV-файла
- Задача загрузки контрагентов из каталога
- Задача загрузки контрагентов по SFTP
- Задача загрузки договоров из буферной таблицы БД
- Задача загрузки связей договоров из буферной таблицы БД
- Деактивировать исходные записи
- Активировать исходные записи
- Задача загрузки связей
- Трансформация данных
- Задача загрузки справочников ЕГРЮЛ / ЕГРИП
- Общие параметры для задач загрузки данных
- Задача циклической загрузки данных из БД
- Задача загрузки справочника ЕФРСБ
- Загрузка инкремента из универсального буфера

# ОБЩИЕ ПАРАМЕТРЫ ДЛЯ ЗАДАЧ ЗАГРУЗКИ ДАННЫХ

## *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.

## *Параметры задач загрузки данных, применимые только для импорта данных в Staging*

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

# ЗАДАЧА ЗАГРУЗКИ КОНТРАГЕНТОВ ИЗ БУФЕРНОЙ ТАБЛИЦЫ БД

✔ Порционная

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

## 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);

```

# ЗАДАЧА ЗАГРУЗКИ ДОГОВОРОВ ИЗ БУФЕРНОЙ ТАБЛИЦЫ БД

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

## 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);
```

# ЗАДАЧА ЗАГРУЗКИ СВЯЗЕЙ ДОГОВОРОВ ИЗ БУФЕРНОЙ ТАБЛИЦЫ БД

## 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
<b>essenceType</b>	Тип контрагента	Тип контрагентов, для которых происходит загрузка договоров
<b>bufferDataReadaBlockSize</b>	Целое	Размер пакета записей для выборки из буферной таблицы
<b>massiveChanging</b>	Логическое значение	Оповещение Фактора изменениями, по умолчанию включено (значение <b>false</b> ). Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>
<b>useRelationCache</b>	Логическое значение	Если задать <b>true</b> , 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)

## 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);

```

# ЗАДАЧА ЗАГРУЗКИ СВЯЗЕЙ

# 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: <b>C:\\upload\\client_data.csv</b>
sourceDataEncoding	Строка	Кодировка файла. Примеры: <b>windows-1251, UTF-8.</b>
sourceDataSeparator	Символ	Разделитель полей файла. Примеры: <b>;</b>   <b>■</b>
sourceDataReadBlockSize	Целое	Размер пакета записей для выборки из файла
rawIdIndex	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 <a href="#">SYSTEM</a> Единого клиента)
essenceType	Тип контрагента	Тип загружаемых контрагентов

# 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 `rawIdIndex`).

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


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

# ЗАДАЧА ЗАГРУЗКИ СПРАВОЧНИКОВ ЕГРЮЛ / ЕГРИП

 EgrDictionaryImport

## 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
<b>sourceFileName</b>	Путь к директории со справочником ЕГР	<b>/path/to/egrul</b>
<b>sourceDataEncoding</b>	Кодировка справочника	<b>UTF-8</b>
<b>bufferDataReadBlockSize</b>	Размер пакета записей для выборки	<b>1000</b>
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>
<b>essenceType</b>	Тип загружаемых контрагентов	<b>EGR</b>
<b>cipher</b>	Загрузка зашифрованного справочника, по умолчанию включена (значение <b>true</b> )	<b>true</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями, по умолчанию включено (значение <b>false</b> ).  Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>
<b>closeAbsentBranches</b>	Закрывать отсутствующие в инкременте филиалы и агентства (по которым нет связей и ручных правок).  По умолчанию <b>true</b> .	<b>true</b>
<b>sourceSystemsToLoad</b>	Список систем, записи из которых загружаются по умолчанию	<b>FTS_LG, FTS_IN, RAFP, MLTST_EGRN</b>

# ТРАНСФОРМАЦИЯ ДАННЫХ

transform

## Описание задачи


Задача нужна для принудительной обработки записей из staging-области.

## Конфигурация задачи

Параметр	Тип	Описание	Значение по умолчанию
<b>terminateOnErrors</b>	Логическое значение	Прервать выполнение при возникновении ошибок.	<b>false</b>
<b>massiveChangging</b>	Логическое значение	Оповещение Фактора изменениями. Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>
<b>notifyListeners</b>	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>	<b>true</b>
<b>iteratorRefreshPeriodMin</b>	Числовое значение	Время обновления итератора по staging  <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставлять в запускаемой задаче</i>	<b>120</b>

# ТРАНСФОРМАЦИЯ ДАННЫХ С УКАЗАННЫМИ ПАРАМЕТРАМИ

transformWithCustomConfig

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

## Описание задачи

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

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>forceUpdateRequisiteNames</b>	Имена реквизитов (через запятую кемел-кейсом), которые нужно безусловно обновлять, остальные реквизиты в штатном порядке.  Пример: <ul style="list-style-type: none"><li>• <b>NONE</b> — нет принудительного обновления;</li><li>• <b>ALL</b> — принудительно обновляем все реквизиты;</li><li>• <b>surname, birthdate, actualityDate</b> — только фио и др и дату актуальности обновляем принудительно.</li></ul>	<b>NONE</b>
<b>forceUpdateAttributeTypes</b>	Имена атрибутов (через запятую), которые нужно безусловно обновлять, остальные атрибуты в штатном порядке.  Пример: <ul style="list-style-type: none"><li>• <b>NONE</b> — нет принудительного обновления;</li><li>• <b>ALL</b> — принудительно обновляем все атрибуты;</li><li>• <b>PHONE, EMAIL</b> — только телефоны и email обновляем принудительно.</li></ul>	<b>NONE</b>
<b>forceUpdate</b>	Безусловное обновления контрагента, без учета настроенных правил	<b>true</b>
<b>requisiteNamesToUpdate</b>	Список имен реквизитов (через запятую кемел-кейсом), которые должны обновляться (по умолчанию обновляются все)  Пример: <ul style="list-style-type: none"><li>• openAccountsFlag</li></ul>	
<b>groupByResultParty</b>	Использовать провайдер записей стейджинга с группировкой по исходным КА одного результирующего	<b>false</b>



<b>terminateOnErrors</b>	Прервать выполнение при возникновении ошибок	<b>false</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями.  Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>
<b>transformPoolSize</b>	Размер пула потоков для трансформации	<b>16</b>
<b>poolSize</b>	Размер пула потоков для стандартизации	<b>кол-во доступных серверов Фактора</b>
<b>pageSize</b>	Число записей в пачке, отдаваемой одному потоку обработки на Факторе	<b>1000</b>
<b>maxStoredErrors</b>	Максимальное кол-во ошибок для хранения в памяти	<b>1000</b>
<b>cleanMapping</b>	Мэппинг для очистки, если надо использовать отдельный (неумолчательный) мэппинг	
<b>customOrigin</b>	Перекрытие стандартного поведения, для каких источников запускать задачу трансформации	
<b>closeMissedAttributes</b>	Флаг необходимости закрывать отсутствующие атрибуты	
<b>cacheSize</b>	Размер кеша страниц	<b>1</b>
<b>iteratorRefreshPeriodMin</b>	Время обновления итератора по staging  <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставить в запускаемой задаче</i>	<b>120</b>
<b>notifyListeners</b>	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 выставить в запускаемой задаче</i>	<b>true</b>

# АКТИВИРОВАТЬ ИСХОДНЫЕ ЗАПИСИ

enableStagingRecords

## Описание задачи

Задача нужна для того, чтобы снова запустить обработку записей, проигнорированных ранее.

## Конфигурация задачи

Параметр	Описание	Пример
<b>stagingHids</b>	Список <b>hid_staging</b> 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	

Сохранить Отмена

Для получения списка клиентов, игнорируемых ранее, необходимо выполнить запрос

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

## Технические детали

Задача устанавливает заданному в параметрах списку контрагентов **STAGING\_STATUS = 0**.

При этом обрабатываются только контрагенты, имеющие **STAGING\_STATUS = 5**, остальные игнорируются

# ДЕАКТИВИРОВАТЬ ИСХОДНЫЕ ЗАПИСИ

disableStagingRecords

- Описание задачи
- Особенности работы с задачей disableStagingRecords при циклической загрузке инкремента.
- Конфигурация задачи
- Технические детали

## 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.

## Конфигурация задачи

Параметр	Описание	Пример
<b>stagingHids</b>	Список <b>hid_staging</b> контрагентов, которые необходимо игнорировать.	1, 4, 100

Значения указываются через запятую

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';
```

## ***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;
```

# ЗАДАЧА ЦИКЛИЧЕСКОЙ ЗАГРУЗКИ ДАННЫХ ИЗ БД

✔ Возобновляемая

✔ Конфликтующая

CyclicIncrementLoad

## Описание задачи

Фактически эта задача являет собой циклическое исполнение следующей последовательности:


1. Обработка хранимой процедуры **GET\_CDI\_BUFFER\_PREPARE**, которая выбирает инкремент для загрузки
2. **physicalUploadTaskPerformer**
3. **legalUploadTaskPerformer**
4. **relationUploadTaskPerformer**

⚠ ВАЖНО ⚠ Пункты 2-4 могут меняться, дополняться, конфигурируется в заказчике отдельно.

⚠ ВАЖНО ⚠ `load_id` инкремента должен быть строго больше нуля. В противном случае задача останавливается, ни один инкремент не загружается.

Параметр	Тип	Описание
<b>terminateOnErrors</b>	Логическое значение	Флаг, указывающий что задачу нужно остановить при возникновении ошибок при обработке данных.
<b>maxIncrements Count</b>	Целое	Максимальное количество инкрементов, загружаемых за одно выполнение задачи.
<b>executionTimeout</b>	Целое	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.

# ЗАДАЧА ЗАГРУЗКИ СПРАВОЧНИКА ЕФРСБ

 EfrsbDictionaryImport

 Возобновляемая

 Конфликтующая

## 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
<b>sourceFileName</b>	Путь к директории со справочником	<b>/path/to /efrsb</b>
<b>sourceDataEncoding</b>	Кодировка справочника	<b>UTF-8</b>
<b>bufferDataReadBlockSize</b>	Размер пакета записей для выборки	<b>1000</b>
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>
<b>cipher</b>	Загрузка зашифрованного справочника, по умолчанию включена (значение <b>true</b> )	<b>true</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями, по умолчанию включено (значение <b>false</b> ).  Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>

# ЗАГРУЗКА ИНКРЕМЕНТА ИЗ УНИВЕРСАЛЬНОГО БУФЕРА

UniversalBufferImport

⚠ Важно! Загрузка из универсального буфера не является равнозначной полноценной пакетной загрузке, ее следует использовать в исключительных случаях для точечного обновления.

Важно! Задача техническая и не предназначена для постоянного использования! Подключается как временное решение проблем интеграции/исправления ошибок!

## Описание

Задача вычитывает и применяет задания из таблицы CDI\_EASY\_BUFFER. Если задание не удалось применить, об этом пишется в лог.

Успешно обработанные записи удаляются из буфера (ошибочные остаются).

## Алгоритм работы

1. Исходя из значения параметра задачи **partyType** выбирается изменяемая сущность.
2. По уникальному ключу **external\_id+source\_system\_cd** выбирается изменяемая карточка.
3. На основании значения поля **attribute\_type** определяется тип изменяемых данных:
  - если поле пустое, загружается реквизит;
  - если поле содержит корректный тип атрибута, загружается атрибут данного типа, определяемый по значению поля **attribute\_external\_id**.
4. В выбранной карточке в поле реквизита\атрибута, указанное в **field\_name** таблицы CDI\_BUFFER\_MIGRATION, записывается значение из поля **field\_value**.

## Особенности

Можно обновлять только реквизиты, которые не парсятся из rawSource и состоят из одного поля (автор не считается).

Если указать неподходящее поле, обновление не будет применено. При этом задача не прервется.

Нельзя обновлять стандартизируемые поля.

## Параметры задачи

Название	Описание	Значение по умолчанию
partyType	Тип контрагента	PHYSICAL
forceUpdate	Принудительное обновление в обход правил обновления	true

## Обработка ошибок и логирование

В случае некорректных данных в буферной таблице задача запишет в лог ошибку:

```
ERROR EasyBufferImportTask - Can not process buffer record ID=501: Can not update composite requisite PhysicalParty.inn  
ERROR EasyBufferImportTask - Can not process buffer record ID=505: Can not find PHYSICAL with uid SBL:1025  
ERROR EasyBufferImportTask - Can not process buffer record ID=504: Can not find PHONE with rawId 'PHONE.1026'  
ERROR EasyBufferImportTask - Can not process buffer record ID=506: Can not find attribute type DOCUMENT in party PhysicalParty  
ERROR EasyBufferImportTask - Can not process buffer record ID=507: Can not convert value '1 апреля 2015 года' to class java.util.Date for AbstractPassport.expiryDate  
(...)
```

В итоговой статистике выводится:

```
INFO TaskLauncherTemplate - Task 'UniversalBufferImport' finished with status FINISHED in 242 ms; — easyBufferImportTaskPerformer: processed 6 entities (party updates: 1, processed: 6); 0 with errors; took 1 seconds or 155 ms; speed 6.0 records/sec;
```




# ИДЕНТИФИКАЦИЯ И СЛИЯНИЕ ДУБЛИКАТОВ

## *Идентификация и слияние дубликатов*

Группа задач, предназначенных для идентификации и слияния дубликатов:

- **findDuplicatess** — поиск дубликатов по всей базе контрагентов;
- **merge** — слияние гарантированных и подтвержденных дубликатов;
- **findDuplicatessAndMerge** — поиск дубликатов по всей базе контрагентов с последующим слиянием гарантированных дубликатов;
- **findDuplicatessAndMergeAndRebuildGroups** — поиск дубликатов по всей базе контрагентов, слияние гарантированных и перестроение индекса негарантированных групп дублей;
- **incrementDuplicatessAndMerge** — 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;
- **findGuaranteedDuplicatessAndMerge** — поиск гарантированных дубликатов и слияние;
- **egrCheck** — полная проверка по ЕГРЮЛ;
- **egrCheckActualize** — Проверка по ЕГРЮЛ контрагентов, требующих актуализации.

# DUPLICATEGROUPACTUALIZE. АКТУАЛИЗАЦИЯ ГРУПП ДУБЛИКАТОВ

 Задача прерываемая, конфликтная, возобновляемая

## Актуализация групп дубликатов — задача `duplicateGroupActualize`

Выполнить задачу **`duplicateGroupActualize`** в АРМ Администратора.

В режиме горячего резерва задачу достаточно запустить на одной из нод, при этом актуализация групп дубликатов будет выполнена на обеих нодах.

### **Ограничения:**

Задача актуализации пропускает изменения в больших группах (ограничение по умолчанию не более 200 карточек в группе).

Поэтому большие группы дубликатов могут отображаться в интерфейсе в неактуальном состоянии или не быть отражены вовсе как на страницах просмотра групп дубликатов, так и на карточки клиента в блоке "похожих".

Для перестроения всех групп дубликатов без ограничений, необходимо выполнить триггер **`full RefreshDuplicateGroupTrigger`**.

# FULLREFRESHDUPLICATEGROUPTRIGGER. ТРИГГЕР ДЛЯ ПОЛНОГО ПЕРЕСТРОЕНИЯ ГРУПП ДУБЛИКАТОВ

**fullRefreshDuplicateGroupTrigger** — предназначен для полной разметки групп дубликатов и перестроения индекса.


Триггер содержит невозобновляемые, конфликтные задачи. Его нельзя запускать в параллель с чем-то еще.

В режиме горячего резерва запускать на одной ноде.

Включает в себя следующие задачи:

Название задачи	Описание
<code>duplicateGroupFullMarkup</code>	Полная разметка групп дубликатов
<code>duplicateGroupIndexRebuild</code>	Перестроение индекса групп дубликатов

# DUPLICATEGROUPFULLMARKUP. ПОЛНАЯ РАЗМЕТКА ГРУПП ДУБЛИКАТОВ

 Невозобновляемая, конфликтная.

## Полная разметка групп дубликатов — duplicateGroupFullMarkup.

Задача просматривает все найденные пары дубликатов и маркирует их специальным образом. Это помогает в дальнейшем использовать разные механизмы обработки групп дубликатов в зависимости от их размера, а также обрабатывать все изменения по парам дубликатов из одной группы одновременно и оптимальнее по производительности.

### Особенности выполнения

Задача выполняет лишь разметку групп дубликатов, перестроение индекса групп дубликатов в рамках данной задачи не выполняется, поэтому рекомендуем выполнять задачу комплексно в составе триггера [fullRefreshDuplicateGroupTrigger](#).

Запускать вне времени выполнения других задач.

В режиме горячего резерва достаточно выполнить только на одной ноде.

### Логирование процесса выполнения задачи


Задача в процессе выполнения пишет в лог сколько записей обработала (логируется каждый миллион):

Сохранение данных в результирующие таблицы пачками с промежуточными коммитами. В одной пачке pageSize групп. Также логируется, сколько сохранили групп.

```
2020-05-17 16:11:04,478 [TaskManagerExecutor-0] INFO DuplicateGroupFullMarkupTask - Found 909775 'LEGAL' duplicate groups in 1.288 min
2020-05-17 16:11:04,478 [TaskManagerExecutor-0] INFO DuplicateGroupFullMarkupTask - Let's save groups to DB...
2020-05-17 16:11:05,146 [DuplicateGroupStoreTaskExecutor-15] INFO DuplicateGroupStoreTask - Processed 100 000 essences
2020-05-17 16:11:05,681 [DuplicateGroupStoreTaskExecutor-4] INFO DuplicateGroupStoreTask - Processed 200 000 essences
2020-05-17 16:11:06,237 [DuplicateGroupStoreTaskExecutor-11] INFO DuplicateGroupStoreTask - Processed 300 000 essences
2020-05-17 16:11:06,684 [DuplicateGroupStoreTaskExecutor-4] INFO DuplicateGroupStoreTask - Processed 400 000 essences
2020-05-17 16:11:07,179 [DuplicateGroupStoreTaskExecutor-2] INFO DuplicateGroupStoreTask - Processed 500B 000 essences
2020-05-17 16:11:07,666 [DuplicateGroupStoreTaskExecutor-14] INFO DuplicateGroupStoreTask - Processed 600B 000 essences
2020-05-17 16:11:08,136 [DuplicateGroupStoreTaskExecutor-4] INFO DuplicateGroupStoreTask - Processed 700 000 essences
```

**2020-05-17 16:11:08,532 [DuplicateGroupStoreTaskExecutor-3]**  
**INFO DuplicateGroupStoreTask - Processed 800 000 essences...**

# DUPLICATEGROUPINDEXREBUILD. ПЕРЕСТРОЕНИЕ ИНДЕКСА ГРУПП ДУБЛИКАТОВ

 Задача прерываемая, конфликтная, возобновляемая

## Перестроение индекса групп дубликатов — задача `duplicateGroupIndexRebuild`

Для полного перестроения индекса групп дубликатов необходимо в АРМ выполнить задачу `duplicateGroupIndexRebuild`.

### *Особенности выполнения*

Индекс перестраивается с учетом выполненной маркировки групп дубликатов (`duplicateGroupFullMarkup`).

Если группа дубликатов не была ранее промаркирована, то индекс по такой группе может быть не перестроен, поэтому рекомендуем выполнять задачу комплексно в составе триггера [fullRefreshDuplicateGroupTrigger](#).

Если все-таки задача запускается отдельно в ручную, тогда в [режиме горячего](#) резерва ее нужно будет выполняться последовательно на каждой из нод.

# ЗАКРЫТИЕ ДУБЛИРУЮЩИХ ДУБЛИКАТОВ

closeDuplicatedDuplicates

 Возобновляемая, неконфликтная

## Описание задачи

Задача ищет неслитые открытые дубликаты ЮЛ, совпадающие по хидам и правилам.

Если совпадающая пара дубликатов найдена, то закрытие производится по следующему алгоритму:

1. Выбирается наиболее приоритетный дубликат по статусу пары дубликатов
2. Выбирается дубликат без классификатора (EGR\_CHECK)
3. Менее приоритетный дубликат закрывается

## Приоритет статуса пары дубликатов

Наивысший приоритет 10

Код	Литера	Описание	Приоритет
0	UNKNOWN	Дубликаты не проверены, решение не принято	40
1	ACCEPT	Дубликаты подтверждены	10
2	REJECT	Дубликаты отклонены	20
3	POSTPONE	Решение отложено	30
4	CANCEL	Разлит	15
5	AUTO_REJECT	Отклонен автоматически (с релиза 19.24)	25
6	CONFLICT	Конфликт слияния (с релиза 20.23)	26

## Конфигурация задачи

Конфигурировать задачу не надо.

Она выгружает все дубликаты ЮЛ. Каждый раз выгружает всю базу.

# ИНКРЕМЕНТАЛЬНЫЙ ПОИСК ДУБЛИКАТОВ

incrementDuplicates

 Невозобновляемая, конфликтная, прерываемая

## Описание задачи

Задача инициирует инкрементальный поиск дубликатов на Факторе.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которого будет запущена задача (указывать несколько нельзя).  Если не задан, запускается для всех типов.	
<b>keepDeDuplicationFiles</b>	Флаг, указывающий на необходимость сохранить (в рабочую папку dedup) временные файлы: <ul style="list-style-type: none"><li>out_PHYSICAL — найденные инкрементальные дубликаты;</li><li>ids_PHYSICAL — хиды проверенных карточек.</li></ul>	false



# КРОСС-СВЕРКА ФЛ-ЮЛ

findCrossRelationsPhysicalAndLegal

## **Описание задачи**

Задача ищет дубликаты между ФЛ и ИП по набору правил.

Если совпадение найдено — между клиентами проставляется связь.

Если раньше совпадение находилось, а теперь не находится, связь закрывается.

## **Конфигурация задачи**

Конфигурировать задачу не надо.

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

# ПЕРЕСТРОЕНИЕ ХЭШЕЙ КРОСС-СВЕРКИ ФЛ-ЮЛ

```
rebuildCrossRelationsPhysicalAndLegalHashes
```

## ***Описание задачи***

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

## ***Конфигурация задачи***

Конфигурировать задачу не надо.

# КРОСС-СВЕРКА ФЛ И ПРОСПЕКТОВ ФЛ

findCrossRelationsPhysicalAndProspects

 Непрерываемая

## ***Описание задачи***

Задача ищет дубликаты между ФЛ и проспектами по набору правил.

Если совпадение найдено — между клиентами проставляется связь.

Если раньше совпадение находилось, а теперь не находится, связь закрывается.

## ***Конфигурация задачи***

Конфигурировать задачу не надо.

Она выгружает все физические лица и проспекты . Каждый раз выгружает всю базу.

# ПЕРЕСТРОЕНИЕ ХЭШЕЙ ПОЛНОГО ПОИСКА ДУБЛИКАТОВ

`rebuildFullDedupHashes`

## 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.

## Конфигурация задачи

Настраиваемые параметры:

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

# ПОИСК ГАРАНТИРОВАННЫХ ДУБЛИКАТОВ И СЛИЯНИЕ

## *Конфигурация задачи*

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента	PHYSICAL_PARTY

# ОЧИСТКА КЭША ИНКРЕМЕНТАЛЬНОГО ПОИСКА ДУБЛИКАТОВ

`clearIncDeDuplicationCache`

## Общее описание

Запускает инкрементальный поиск дубликатов в Факторе с параметром **appendIncrementOnly = true**, игнорируя сам поиск дубликатов и объединяя инкрементальный и полный срез данных.

## Конфигурация задачи

Параметр	По умолчанию	Описание
<b>clearFullCaches</b>	<b>false</b>	Количество дней, которые конфликты хранятся в базе
<b>executionTime out</b>	<b>0</b>	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.
<b>terminateOnErrors</b>	<b>false</b>	Прекратить выполнение при возникновении ошибки?

# ОЧИСТКА ХЭШЕЙ ПОИСКА ДУБЛИКАТОВ

`truncateDeDuplicationData`

## *Общее описание*

Очищает данные для дедупликации. При кластерном запуске выполняется на двух нодах.

## *Конфигурация задачи*

Параметр	По умолчанию	Описание
<code>clearFullCaches</code>	<code>false</code>	Нужно ли помимо инкрементальных удалить полные данные дедупликации

# СЛИЯНИЕ ДУБЛИКАТОВ ПО ВНЕШНЕМУ СПИСКУ

## Общее описание

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

## Механизм

Задача вызывает хранимую функцию `get_external_duplicates` с параметром `partyType` (**P** **HYSICAL** / **I** **NDIVIDUAL** / **L** **E** **G** **A** **L**).

Функция возвращает курсор на выборку в следующем формате:

Поле	Описание
<code>hid_2</code>	НID второй PARTY
<code>hid_1</code>	НID первой PARTY
<code>match_scope</code>	коэффициент схожести для создаваемых дубликатов
<code>rule</code>	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.

## Примечание

Для выполнения данной задачи в БД должны быть корректно добавлены исполняемые процедуры.

Если задачу выполняете в первый раз, то перед запуском обратитесь в HFLabs.



## 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
<b>hid_2</b>	НID второй PARTY
<b>hid_1</b>	НID первой PARTY
<b>match_scope</b>	коэффициент схожести для создаваемых дубликатов
<b>rule</b>	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.

# ВАЛИДАЦИЯ ДАННЫХ ДЕДАП СЕРВИСА

validateDeDuplicationData

## Общее описание

Задача производит сверку данных из дедапа и ЕК для всех типов КА.

Если есть расхождения, то задача падает, а результат пишется в **{cdi.root.folder}/dedup/validate\_PHYSICAL(LEGAL,etc)**.

Файлы с дампами:

- **dump\_cdi**
- **dump\_factor**

Файлы с расхождениями:

- **diff\_data** — разные данные по КА;
- **only\_cdi\_ids** — айдишники есть в ЕК, но нет в Факторе;
- **only\_factor\_ids** — айдишники есть в Факторе, но нет в ЕК.

## Конфигурация задачи

Параметр	По умолчанию	Описание
<b>keepDumpFiles</b>	<b>false</b>	Оставлять ли файлы с дампом данных.
<b>partyType</b>		По умолчанию проверяются все типы КА, уходящие на дедап с дефолтным профилем DEFAULT_PROFILE.
<b>excludeAttributes</b>		Атрибуты через запятую, которые надо исключить из сравнения записей. Например, чтобы выкинуть все договоры, указываем маленькими буквами так, как он указан в профиле (agreement).

# ПОВТОРНАЯ ОБРАБОТКА

## *Повторная обработка контрагентов*

- Актуализация атрибутов на основе данных staging
- Актуализация статуса невалидности документов
- Задача актуализации адресов по ФИАС
- Задача восстановления атрибутов по списку НIDов
- Задача восстановления КА на версию по списку НID
- Задача выливания исходных из состава массивных золотых карточек
- Задача заполнения технических полей
- Задача инкрементальной актуализации адресов по ФИАС
- Задача обработки массивных карточек контрагентов
- Задача очистки таблиц параметров для массовой переобработки данных
- Задача переноса открытых связей закрытых транзитивных карточек на активные золотые
- Задача переобработки UNDEF-телефонов
- Задача переобработки СНИЛС
- Задача переобработки ИНН ФЛ
- Задача переоткрытия связей
- Задача переочистки вручную подтверждённых ФИО физлиц
- Задача переочистки маркеров ФНС ФИО
- Задача переочистки реквизитов
- Задача перерасчета hidden-флагов атрибутов по списку контрагентов
- Задача переслияния реквизитов контрагентов
- Задача пересчета скоринговых значений атрибутов
- Задача поиска золотых карточек для переслияния
- Задача поиска массивных карточек контрагентов
- Задача полного переслияния контрагентов
- Закрытие актуальных (результатирующих и всех исходных) атрибутов по списку
- Закрытие исходных атрибутов по списку с переслиянием результирующих
- Заполнение полной ОПФ
- Заполнение ФИО ИП
- Исправьте базу данных в случае неконсистентного состояния
- Переобработка email
- Переобработка телефонов
- Переобработка телефонов. Новый вариант
- Переоткрытие закрытых по ошибке контрагентов с сохранением истории
- Переочистка email
- Переочистка гранулярных кодов качества ФИО
- Переочистка кодов качества дат рождения

- Переочистка кодов качества КПП
- Переочистка кодов качества паспортов
- Подготовка списков для повторной обработки
- Полная переочистка ФИО
- Полное разлияние контрагентов по списку результирующих контрагентов
- Разлияние контрагентов по списку MergedId
- Таблицы с параметрами задач массовой переобработки
- Задача переобработки ИНН ЮЛ

Таблицы с параметрами задач массовой переобработки.

Группа задач, предназначенных для повторной обработки определенного подмножества контрагентов.

Задача	Описание	Действия перед выполнением задачи	Действия после выполнения задачи
1 <b>hiddenFlagRefineByHids</b>	Задача перерасчета hidden-флагов атрибутов по списку контрагентов		
2 <b>reMergeFullByPartyHids</b>	Задача полного переслияния контрагентов		
3 <b>reMergeRequisiteByPartyHids</b>	Задача переслияния реквизитов контрагентов		
4 <b>unMergeByMergedIds</b>	Разлияние контрагентов по списку MergedId		
5 <b>fullUnmergeByResultHids</b>	Полное разлияние контрагентов по списку результирующих контрагентов		
6 <b>attributeRecoverByHids</b>	Задача восстановления атрибутов по списку HIDs		
7 <b>addressActualization</b>	Задача актуализации адресов по ФИАС		
8 <b>mergedPartiesCheckAndRemerge</b>	Исправьте базу данных в случае неконсистентного состояния		
9 <b>partyReopenByHids</b>	Переоткрытие закрытых по ошибке контрагентов с сохранением истории		
10 <b>attributeActualizeByStaging</b>	Актуализация атрибутов на основе данных staging		
11 <b>attributeCloseActualByHids</b>	Закрытие актуальных (результирующих и всех исходных) атрибутов по списку		
12 <b>cleanTaskParameterTables</b>	Задача очистки таблиц параметров для массовой переобработки данных		
13 <b>fillTaskPartyHidsActual</b>	Подготовка списков для повторной обработки		
<b>fillTaskPartyHidsOriginal</b>			

	<b>fillTaskPartyHidsResulted</b>			
	<b>fillAddressActualizationTaskPartyHids</b>			
	<b>fillTaskPartyHidsBadAdresses</b>			
	<b>fillTaskPartyHidsOriginalWithClosed</b>			
14	<b>passportInvalidStatusReCleaner</b>	Актуализация статуса невалидности документов		
15	<b>attributeCloseSourceByHids</b>	Закрытие исходных атрибутов по списку с переслиянием результирующих		
16	<b>phoneExtraFieldsReCleaner</b>	Переобработка телефонов		
17	<b>phoneExtraFieldsActualization.physical</b>	Переобработка телефонов. Новый вариант		
	<b>phoneExtraFieldsActualization.legal</b>			
18	<b>partyRecoverByHids</b>	Задача восстановления КА на версию по списку НID		
19	<b>reCleanMockRequisiteTask</b>	Задача переочистки реквизитов		
20	<b>phoneUndefReCleaner</b>	Задача переобработки UNDEF-телефонов		
21	<b>refreshScoresTask</b>	Задача пересчета скоринговых значений атрибутов		
22	<b>physicalFioQcReCleaner</b>	Переочистка гранулярных кодов качества ФИО	Заполнение таблицы актуальными КА	Удаление таблицы по завершении задачи
23	<b>passportQCReCleaner</b>	Переочистка кодов качества паспортов		
24	<b>physicalBirthdateQcReCleaner</b>	Переочистка кодов качества дат рождения	Заполнение таблицы актуальными КА	Удаление таблицы по завершении задачи
25	<b>relationReopenByIds</b>	Задача переоткрытия связей		
26	<b>legallpReCleaner</b>	Заполнение ФИО ИП		
27	<b>physicalFioConfirmedManuallyReCleaner</b>	Задача переочистки вручную подтверждённых ФИО		

# АКТУАЛИЗАЦИЯ АТТРИБУТОВ НА ОСНОВЕ ДАННЫХ STAGING

attributeActualizeByStaging

## 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 запакованного XML-документа в поле **plain\_xml**.

## 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**.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется актуализация атрибутов	<b>PHYSICAL</b>
<b>attributeType</b>	Тип атрибута	<b>ADDRESS</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями. Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>

## Инструкция по переобработке адресов, отмеченных на ручную обработку

Для переобработки адресов зайдите в АРМ Администратора и для каждого типа контрагента выполните шаги:

## 1. Заполнить список к переобработке HIDами контрагентов

- Откройте настройки задачи **fillTaskPartyHidsBadAddresses**.
- Укажите **тип контрагента** в параметре **procedureParams**.
- Запустите задачу.

## 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.

# ИНСТРУКЦИЯ ПО ПЕРЕОБРАБОТКЕ АДРЕСОВ, ОТМЕЧЕННЫХ НА РУЧНУЮ ОБРАБОТКУ

Для переобработки адресов зайдите в АРМ Администратора и для каждого типа контрагента выполните шаги:

## 1. Заполнить список к переобработке NIDами контрагентов

- Откройте настройки задачи **fillTaskPartyHidsBadAddresses**.
- Укажите **тип контрагента** в параметре **procedureParams**.
- Запустите задачу.

## 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.



# АКТУАЛИЗАЦИЯ СТАТУСА НЕВАЛИДНОСТИ ДОКУМЕНТОВ

passportInvalidStatusReCleaner

✔ Возобновляемая

## 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**).

## 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.

## 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.

## **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.

## **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.

## ***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.

## ***Конфигурация задачи***

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

# ЗАДАЧА АКТУАЛИЗАЦИИ АДРЕСОВ ПО ФИАС

addressActualization

 Возобновляемая

## Описание задачи

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

## 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.

## Частичная актуализация

Актуализации подвергаются индексы (почтовый и по ФИАС), группа полей Страна-Улица, Федеральный округ, коды КЛАДР и ОКАТО, ОКСМ, ОКТМО, Часовой пояс. 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 ФИАС адреса соответствует ФИАСу 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 ФИАС), Фактор возвращает предыдущие результаты распознавания без изменений.

## 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 ФИАС адреса соответствует ФИАСу в ФАКТОРе, Фактор возвращает предыдущие результаты распознавания без изменений.

## 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.

Задача **addressActualization** не конфликтует с задачами переобработки **markPopularPhysicalAttributes**, **markPopularLegalAttributes**, **passportInvalidStatusReCleaner** и выполняется в фоне.

## Конфигурация задачи

Настраиваемые параметры:

Имя	Описание	Значение по умолчанию
<b>executionTimeOut</b>	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.	<b>0</b>
<b>partyType</b>	Тип контрагента	<b>PHYSICAL</b>
<b>pageSize</b>	Размер одной страницы данных (увеличивает время блокировок БД)	<b>0</b>
<b>poolSize</b>	Количество потоков, которые будут обрабатывать страницы с данными	<b>0</b>
<b>maxStoredErrors</b>	Количество ошибок, которое можно накапливать в памяти	<b>0</b>
<b>iteratorRefreshPeriodMin</b>	Как часто (в минутах) переоткрывать итератор по данным, чтобы избежать переполнения UNDO	<b>0</b>
<b>terminateOnError</b>	Параметр, указывающий что задачу нужно остановить при возникновении N	<b>false</b>

<b>ors</b>	ошибок	
<b>massiveChanging</b>	Признак массового изменения - сигнал к отключению оповещения инкрементального дедапа и, возможно, др. слушателей	<b>false</b>
<b>cacheSize</b>	Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память)	<b>0</b>
<b>notifyListeners</b>	Флаг необходимости оповещать слушателей	<b>false</b>

# ЗАДАЧА ВОССТАНОВЛЕНИЯ АТТРИБУТОВ ПО СПИСКУ HИDОВ

attributeRecoverByHids

## 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.

## Особенности задачи

Если атрибут был переслит за откатываемый промежуток времени, то задача упадет с ошибкой.

Изменений в связях атрибутов не происходит.

## Запуск 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 ...
```

# ЗАДАЧА ВОССТАНОВЛЕНИЯ КА НА ВЕРСИЮ ПО СПИСКУ НID

 partyRecoverByHids

 Возобновляемая

## 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 НID-ов, которые были не закрыты в указанный промежуток времени.

## 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.

## 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**

## **Конфигурация задачи**

Настраиваемые параметры:


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

Формат даты — 10.06.2018 00:00:03.



# ЗАДАЧА ВЫЛИВАНИЯ ИСХОДНЫХ ИЗ СОСТАВА МАССИВНЫХ ЗОЛОТЫХ КАРТОЧЕК

## massivePartyUnMergeTask

 Конфликтная, прерываемая

Предназначена для выливания обработки массивных контрагентов.

Выливает из результирующих карточек, указанных в таблице MASSIVE\_PARTY\_REGISTRY, исходные, чьи ID\_MERGED перечислены в MASSIVE\_PARTY\_UNMERGE\_IDS.

### MASSIVE\_PARTY\_UNMERGE\_IDS — реестр массивных КА для разливания

Имя	Тип	Описание
id_merged	bigint	Идентификатор id_merged
hid_result_party	bigint	Уникальный идентификатор результирующего контрагента
party_type	varchar(50)	Тип контрагента

Задача добавлена в ежедневный и еженедельный триггеры. Данные в таблицы попадают при следующих действиях над массивными карточками:

- разливание (при стратегии разрешения конфликтов UNMERGE\_AND\_APPLY)

 Добавление связей на исходные не обновляет индекс для массивных золотых КА.

# ЗАДАЧА ЗАПОЛНЕНИЯ ТЕХНИЧЕСКИХ ПОЛЕЙ

**extraUtilityFieldsFill**

✔ Возобновляемая, конфликтная

## Описание задачи

Задача предоставляет исходные идентификаторы, коды систем-источников, даты создания для карточек и unified-идентификаторы для ФЛ и ЮЛ, которые уже загружены в БД.

## Запуск задачи

Просто запустить задачу **extraUtilityFieldsFill**.

Дополнительные шаги по наполнению вспомогательных таблиц идентификаторами КА не нужны.

Задача сама создаст таблицу **tmp\$util\_fields\_party\_hids**, наполнит ее хидами, переобработает записи и, если не было ошибок при переобработке, удалит таблицу.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>poolSize</b>	Максимальное количество потоков	4
<b>partyType</b>	Тип контрагента	PHYSICAL
<b>fillFieldsForMerged</b>	Заполнять поля для влитых карточек	true
<b>fillSourceIdentifiers</b>	Заполнять поля <b>source_identifier</b> , <b>source_system</b> , <b>created_date</b>	true
<b>fillUnifiedId</b>	Заполнять поле <b>unified_id</b>	true

## Логирование

Задача стандартно логирует обработку каждых N записей (по умолчанию 100 000):

```
INFO SetExtraUtilityFieldsTask - Processed 100 000 essences
INFO SetExtraUtilityFieldsTask - Processed 200 000 essences
INFO SetExtraUtilityFieldsTask - Processed 300 000 essences
```

В итоговой статистике выводится:

**INFO TaskLauncherTemplate - Task 'extraUtilityFieldsFill' finished with status FINISHED  
2.59 sec; — setExtraUtilityFieldsTaskPerformer: processed 30 entities; 0 with errors; took  
2 seconds or 2.39 sec; speed 15.0 records/sec;**

# ЗАДАЧА ИНКРЕМЕНТАЛЬНОЙ АКТУАЛИЗАЦИИ АДРЕСОВ ПО ФИАС

addressIncrementalActualization

✓ Возобновляемая

## Описание задачи

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

## Запуск задачи

Выполнить последовательно задачи:

1. **fillIncrementalAddressActualizationPartyHids** — инициализировать список контрагентов для актуализации адресов
2. **addressIncrementalActualization**

## Алгоритм работы задачи

На актуализацию выбираются все контрагенты для актуализации адресов которых есть основания. Адреса результатов слияния актуализируются за счёт обновления исходных адресов в соответствии с правилами обновления.

Ниже укрупненно приведена предварительная последовательность проверок для принятия Фактором решения по адресу. Пункты выполняются последовательно, пока не принято решение:

3.1 Поиск по `fias_code` – если Фактор вернул пустоту, т.е. в ФИАС этот адрес был скорее всего закрыт, то такой адрес отправляется на актуализацию, иначе выполнить проверку по п. 3.2.

3.2 Проверка `address_all_changes` - адреса с улицей по умолчанию всегда отправляются на актуализацию (`address_all_changes` содержит `STREET_DEFAULT`), иначе выполнить проверку по п. 3.3;

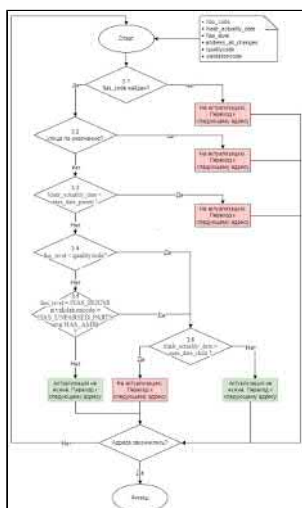
3.3 Проверка `kladr_actuality_date` – если дата в поле `kladr_actuality_date` меньше максимальной даты обновления «по родителям», которую вернул Фактор для `fias_code`, такой адрес отправляется на актуализацию, иначе перейти к проверке по п. 3.4. Т.е. если изменения в ФИАС по адресу свежее, чем версия адреса у КА то можно предположить, что в адресе могут быть изменения.

3.4 Проверка соответствия `fias_level` значению в `qualitycode` – если уровень соответствия адреса ФИАСу ниже чем код полноты результирующего адреса, то перейти к проверке по п. 3.6, иначе к проверке по п. 3.5. Таблица соответствий кода полноты уровню адреса в ФИАС:

qualitycode	Соответствующий fias_level
UNDEF_01	FIAS_COUNTRY
UNDEF_02	FIAS_SUBJECT
UNDEF_03	FIAS_DISTRICT, FIAS_CITY
UNDEF_04	FIAS_SETTLEMENT, FIAS_STREET, FIAS_PLANNING_STRUCTURE
UNDEF_05	FIAS_HOUSE
UNDEF_06	UNKNOWN_ADDRESS, EMPTY
UNDEF_07	FOREIGN_ADDRESS
GOOD	FIAS_HOUSE, FIAS_FLAT
ON_DEMAND	FIAS_CITY, FIAS_CITY_AREA
POSTAL_BOX	FIAS_SETTLEMENT, FIAS_STREET, FIAS_PLANNING_STRUCTURE

3.5 Проверка соответствия fias\_level значению в validationcode – если fias\_level равен FIAS\_HOUSE и при этом validationcode = NOT\_VALIDATED\_HAS\_UNPARSED\_PARTS или NOT\_VALIDATED\_HAS\_AMBI, то перейти к проверке по п. 3.6, иначе принимаем решение, что актуализация не требуется. Таким образом мы проверяем хорошие адреса, разобранные до дома, но у которых может быть неразобранный хвост из строений, корпусов и квартир.

3.6 Проверка kladr\_actuality\_date – если дата в поле kladr\_actuality\_date меньше максимальной даты обновления «по детям первого уровня», которую вернул Фактор для fias\_code, то принимается решение о необходимости актуализации адреса, иначе актуализация не требуется.



## Особенности задачи

В отличие от других задач повторной обработки использует отдельную таблицу для хранения идентификаторов контрагентов (**TASK\_ADDRESS\_ACT\_HIDS**), которые должны быть обработаны.

Для инициализации списка контрагентов на обработку необходимо выполнить задачу `fillIncrementalAddressActualizationPartyHids`.


## Конфигурация задачи

Настраиваемые параметры:

Имя	Описание	Значение по умолчанию
<b>executionTimeout</b>	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.	<b>0</b>
<b>partyType</b>	Тип контрагента	<b>PHYSICAL</b>
<b>pageSize</b>	Размер одной страницы данных (увеличивает время блокировок БД)	<b>0</b>
<b>poolSize</b>	Количество потоков, которые будут обрабатывать страницы с данными	<b>0</b>
<b>maxStoredErrors</b>	Количество ошибок, которое можно накапливать в памяти	<b>0</b>
<b>iteratorRefreshPeriodMin</b>	Как часто (в минутах) переоткрывать итератор по данным, чтобы избежать переполнения UNDO	<b>0</b>
<b>terminateOnErrors</b>	Параметр, указывающий что задачу нужно остановить при возникновении N ошибок	<b>false</b>
<b>massiveChanging</b>	Признак массового изменения - сигнал к отключению оповещения инкрементального дедапа и, возможно, др. слушателей	<b>false</b>
<b>cacheSize</b>	Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память)	<b>0</b>
<b>notifyListeners</b>	Флаг необходимости оповещать слушателей	<b>false</b>
<b>reportEach</b>	Количество записей в пачке для логирования	<b>-1 (по умолчанию не логируем)</b>

# ЗАДАЧА ОБРАБОТКИ МАССИВНЫХ КАРТОЧЕК КОНТРАГЕНТОВ

## massivePartyReMergeTask

 конфликтная, прерываемая

Предназначена для обработки массивных контрагентов.

Пересливает все карточки полностью (реквизиты и атрибуты) по хидам из таблицы massive\_party\_remerge\_hids.

### MASSIVE\_PARTY\_REMERGE\_HIDS — реестр массивных КА для переслияния

Имя	Тип	Описание
hid_party	bigint	Исторический идентификатор контрагента
party_type	varchar(50)	Тип контрагента

Задача обработки массивных карточек добавлена в ежедневный и еженедельный триггеры.

Данные в таблицу попадают при следующих действиях над массивными карточками:

- разлияние (при стратегии разрешения конфликтов UNMERGE\_AND\_APPLY)
- обновление исходной
- закрытие исходной

 Добавление связей на исходные не обновляет индекс для массивных золотых КА.

# ЗАДАЧА ОЧИСТКИ ТАБЛИЦ ПАРАМЕТРОВ ДЛЯ МАССОВОЙ ПЕРЕОБРАБОТКИ ДАННЫХ

cleanTaskParameterTables

## *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](#) (кроме **task\_remerge\_hids** и **task\_engine\_actualize\_hids**).



# ЗАДАЧА ПЕРЕНОСА ОТКРЫТЫХ СВЯЗЕЙ ЗАКРЫТЫХ ТРАНЗИТИВНЫХ КАРТОЧЕК НА АКТИВНЫЕ ЗОЛОТЫЕ

## **moveTransitionalPartyRelations**

✔ Возобновляемая, неконфликтная

### *Особенности задачи*

Задача перебирает все транзитивные золотые, определяет наличие на них открытых связей и переносит связи на актуальную золотую.

### *Запуск задачи*

Просто запустить задачу **moveTransitionalPartyRelations**, она сама создаст временную таблицу **tmp\$move\_trans\_relations\_hids**, наполнит ее идентификаторами КА каждого типа поочередно и удалит после обработки всех записей.

### *Параметры задачи*

Параметр	Описание	Значение по умолчанию
<b>skipJmsEventsPublish</b>	Не отправлять события в обратный поток	true

# ЗАДАЧА ПЕРЕОБРАБОТКИ UNDEF-ТЕЛЕФОНОВ

phoneUndefReCleaner

 Возобновляемая

## Описание задачи

Задача переобрабатывает все UNDEF-телефоны с целью простановки кода качества OUT\_OF\_RANGE («Не найден в Росвязи») и маркеров причин, почему телефон плохой (если подключены).

При этом телефоны могут распознаться и как GOOD и FOREIGN и поменять профиль.

## Запуск задачи

Запустить задачу **phoneUndefReCleaner**, предварительно указав в параметре **partyType** нужный тип контрагента. Задача сама создаст временную таблицу, наполнит ее идентификаторами актуальных контрагентов, переобработает записи и, если не было ошибок при переобработке, удалит таблицу.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка телефонов	PHYSICAL

# ЗАДАЧА ПЕРЕОБРАБОТКИ СНИЛС

**partySnilsReCleaner**

✔ Возобновляемая, конфликтующая

## Описание задачи

Задача переобработывает СНИЛС и его код качества по списку КА из таблицы **task\_party\_hids**.

1. Если таблица **task\_party\_hids** пустая, то она автоматически наполнится хидами контрагентов с `merged_status` 0 и 2 при запуске задачи. Либо можно предварительно наполнить её вручную скриптом.
2. При переобработке создается новая версия контрагента.
3. Задача умеет переобработывать также СНИЛС Проспектов ФЛ и списочников. Для этого достаточно указать параметр **partyType=PHYSICAL\_PROSPECT** или **PHYSICAL\_LIST** соответственно.

## Запуск задачи

Для переобработки СНИЛС у всех физлиц без выборки:

1. Убедиться, что таблица `TASK_PARTY_HIDS` пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

2. Запустить задачу **partySnilsReCleaner**.

Для выборочной переобработки СНИЛС:

1. Убедиться, что таблица `TASK_PARTY_HIDS` пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

2. Наполнить таблицу **task\_party\_hids** хидами контрагентов для переобработки:

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

-- Нужно переобработать набор клиентов. Работает для оригинальных и золотых,
поэтому ставим условие merged_status in (0, 2)
insert into task_party_hids
select hid_party from physical_party where merged_status in (0, 2) and ... ;
```

3. Запустить задачу **partySnilsReCleaner**.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>PHYSICAL</b>
	Возможные значения:	
	<ul style="list-style-type: none"><li>• <b>PHYSICAL_PROSPECT</b></li><li>• <b>PHYSICAL_LIST</b></li></ul>	

# ЗАДАЧА ПЕРЕОБРАБОТКИ ИНН ФЛ

**partyPhysicalInnReCleaner**

✓ Возобновляемая, конфликтующая

## Описание задачи

Задача переобработывает ИНН и его код качества по списку КА из таблицы **task\_party\_hids**.

1. Если таблица **task\_party\_hids** пустая, то она автоматически наполнится хидами контрагентов с `merged_status` 0 и 2 при запуске задачи. Либо можно предварительно наполнить её вручную скриптом.
2. При переобработке создается новая версия контрагента.
3. Задача умеет переобработывать также ИНН Проспектов ФЛ и списочников ФЛ. Для этого достаточно указать параметр **partyType=PHYSICAL\_PROSPECT** или **PHYSICAL\_LIST** соответственно.

## Запуск задачи

Для переобработки ИНН у всех физлиц без выборки:

1. Убедиться, что таблица `TASK_PARTY_HIDS` пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

2. Запустить задачу **partySnilsReCleaner**.

Для выборочной переобработки ИНН:

1. Убедиться, что таблица `TASK_PARTY_HIDS` пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

2. Наполнить таблицу **task\_party\_hids** хидами контрагентов для переобработки:

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

-- Нужно переобработать набор клиентов. Работает для оригинальных и золотых,
поэтому ставим условие merged_status in (0, 2)
insert into task_party_hids
select hid_party from physical_party where merged_status in (0, 2) and ... ;
```

3. Запустить задачу **partyPhysicalInnReCleaner**.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>PHYSICAL</b>
	Возможные значения:	
	<ul style="list-style-type: none"><li>• <b>PHYSICAL_PROSPECT</b></li><li>• <b>PHYSICAL_LIST</b></li><li>• <b>PHYSICAL</b></li></ul>	

# ЗАДАЧА ПЕРЕОТКРЫТИЯ СВЯЗЕЙ

relationReopenByIds

✓ Возобновляемая

## Описание задачи

Задача переоткрывает ошибочно закрытые связи по списку с ID\_RELATION из таблицы **task\_relation\_ids**.

Обработка записей из списка происходит по следующей логике:

- если связь с таким ID открыта, то она пропускается;
- если связи с таким ID нет, она тоже пропускается;
- если связь с таким ID есть, создается новая связь с такими же данными, но новой startDate и пустой endDate и с новым ID\_RELATION.

Помимо прочего задача проверяет, что обе стороны связи (хиды пати) являются актуальными КА (не закрыты и существуют), в противном случае связи не переоткрываются.

Количество созданных и пропущенных связей фиксируется в cdi.log, например:

```
2019-08-05 18:49:30,504 [TaskManagerExecutor-4] INFO TaskPerformerStatsInterceptor -  
FINISHED task 'relationReopenByIds [reopenRelationsTaskPerformer]'; processed 4 entities  
(create: 2, skip: 2); 0 with errors; took 1 seconds or 78 ms; speed 4.0 records/sec;
```

## Запуск задачи

1. Перед запуском задачи необходимо убедиться, что в таблице **relation\_task\_ids** нет записей.

При необходимости очистить данную таблицу:

```
truncate table task_relation_ids;
```

2. Заполнить таблицу **relation\_task\_ids** идентификаторами связей, которые необходимо переоткрыть.

Пример ручного наполнения таблицы для переоткрытия связей:

```
insert into task_relation_ids  
select id_relation from relation  
where ID_RELATION_TYPE = 150 and enddate = to_date('2019-07-26 00:00:00','YYYY-MM-DD  
HH24:MI:SS');  
  
commit;
```

3. Запустить задачу **relationReopenByIds**.

## ***Конфигурация задачи***


Настраиваемые параметры:

<b>Параметр</b>	<b>Описание</b>	<b>Значение по умолчанию</b>
<b>poolSize</b>	Максимальное количество потоков	<b>1</b>



# ЗАДАЧА ПЕРЕОЧИСТКИ ВРУЧНУЮ ПОДТВЕРЖДЁННЫХ ФИО ФИЗЛИЦ

physicalFioConfirmedManuallyReCleaner

 Конфликтующая

## *Описание задачи*

Задача используется для переочистки ФИО физлиц от лишних пробелов, которые были добавлены дата-стюардами и подтверждены вручную.

## *Особенности задачи*

Задача берет на переочистку компоненты ФИО — **SURNAME, NAME, PATRONYMIC** которые были подтверждены вручную, т. е. `fullmane_qc = CONFIRMED_MANUALLY`. Очистка проводится без отправки в Фактор. Пробелы режутся внутри задачи. Новые версии КА не создаются.

## *Запуск задачи*

Запустить задачу **physicalFioConfirmedManuallyReCleaner**.

В режиме горячего резерва запускать только на одной ноде.

# ЗАДАЧА ПЕРЕОЧИСТКИ МАРКЕРОВ ФНС ФИО

**physicalFioFnsStatusReCleaner**

✔ Возобновляемая, конфликтная

## Описание задачи

Задача проверяет указанные ФИО физлиц по требованиям ФНС. Новая версия КА при этом не создается.

Работает по таблице TASK\_PARTY\_HIDS.

## Запуск задачи

1. Наполнить таблицу TASK\_PARTY\_HIDS хидами актуальных контрагентов, ФИО которых нужно проверить на требования ФНС.
2. Выполнить задачу **physicalFioFnsStatusReCleaner**.

В режиме горячего резерва запускать только на одной ноде.

# ЗАДАЧА ПЕРЕОЧИСТКИ РЕКВИЗИТОВ

reCleanMockRequisiteTask

✔ Возобновляемая

## Описание задачи

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

## Особенности задачи

Работает через провайдер для получения всех оригинальных контрагентов.

Переочищенные реквизиты обновляются принудительно.

## Конфигурация задачи

Настраиваемые параметры:

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

## 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
```

# ЗАДАЧА ПЕРЕРАСЧЕТА HIDDEN-ФЛАГОВ АТТРИБУТОВ ПО СПИСКУ КОНТРАГЕНТОВ

hiddenFlagRefineByHids

✓ Возобновляемая

## 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 оригинальным карточкам.

## 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 наполнения таблицы hid-ами оригинальных актуальных карточек:

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

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется перерасчета hidden-флагов атрибутов	<b>PHYSICAL</b>
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>

<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями. Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>

Группа reprocess  
Название hiddenFlagRefineByHids

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	8	
pageSize	1000	
partyType	PHYSICAL	

Сохранить Отмена

# ЗАДАЧА ПЕРЕСЛИЯНИЯ РЕКВИЗИТОВ КОНТРАГЕНТОВ

reMergeRequisiteByPartyHids

✓ Возобновляемая

## 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.

## Запуск задачи

Просто запустить задачу **reMergeRequisiteByPartyHids**, она сама наполнит таблицу **task\_remerge\_hids** хидами золотых карточек, пересольет их и очистит таблицу.

Если нужно переслать только определенные карточки, а не всю базу, добавить список данных идентификаторов в таблицу **task\_remerge\_hids** вручную.

Пример ручного наполнения таблицы:

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

## Конфигурация задачи

Настраиваемые параметры:

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

Если отключать оповещение, то потом потребуется перестроение индексов и хешей Фактора.

<b>skipJmsEventsPublish</b>	Флаг отключения публикации событий ОП	<b>false</b>
<b>executionTimeout</b>	Максимальное время выполнения задачи в минутах. При превышении задача останавливается	0
<b>sendEmail</b>	Признак необходимости отправлять письмо	<b>false</b>
<b>sendTo</b>	Список получателей письма через запятую	
<b>terminateOnErrors</b>	Прекратить выполнение при возникновении ошибки?	<b>true</b>
<b>includeClosed</b>	Включать закрытые записи	<b>false</b>
<b>hidsTableName</b>	Имя таблицы Hid'ов для переобработки	LEGAL
<b>mergedStatus</b>	Статусы слияния	RESULT
<b>cacheSize</b>	Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память)	1
<b>iteratorRefreshPeriodMin</b>	Как часто (в минутах) переоткрывать итератор по данным, чтобы избежать переполнения UNDO	60
<b>forcePartyUpdate</b>	Обновить КА, даже если нет значительных изменений	<b>false</b>
<b>skipDedupNotify</b>	Не оповещать кэш дубликатов об изменениях	<b>false</b>
<b>fillTaskPartyHids</b>	Признак заполнения таблицы Hid'ов для переобработки - если false, то таблица не наполняется	<b>false</b>
<b>silentMode</b>	Режим тихой работы (динамический подбор размера пула задач в зависимости от нагрузки на сервер)	<b>false</b>
<b>maxStoredErrors</b>	Количество ошибок, которое можно накапливать в памяти	1000
<b>maxOriginalPartiesCount</b>	Максимальное количество оригинальных контрагентов в пачке для переслияния	0
<b>groupByResultParty</b>	Группировать по исходным КА одного результирующего	
<b>skipSearchNotify</b>	Не оповещать поисковый индекс об изменениях	<b>false</b>

Группа reprocess  
Название reMergeRequisiteByPartyHids

Параметры задачи

Название параметра	Значение	Обязательность
<input type="text" value="poolSize"/>	<input type="text" value="8"/>	
<input type="text" value="pageSize"/>	<input type="text" value="1000"/>	
<input type="text" value="partyType"/>	<input type="text" value="PHYSICAL"/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	

# ЗАДАЧА ПЕРЕСЧЕТА СКОРИНГОВЫХ ЗНАЧЕНИЙ АТТРИБУТОВ

**refreshScoresTask**

## *Запуск задачи*

1. Выполнить задачу **fillTaskPartyHidsResulted** для наполнения таблицы **task\_party\_hids** идентификаторами КА для переобработки.
2. Запустить задачу **refreshScoresTask** для пересчета скоринга.

## *Логирование*

Задача стандартно логирует обработку каждых N записей (по умолчанию 100 000):

```
INFO RefreshScoresTask - Processed 100 000 essences  
INFO RefreshScoresTask - Processed 200 000 essences  
WARN RefreshScoresTask - Skipped 2 non-RESULT parties  
INFO RefreshScoresTask - Processed 300 000 essences  
INFO TaskPerformerStatsInterceptor - FINISHED task 'refreshScoresTask  
[refreshScoresTaskPerformer]'; processed 300 000 entities; 0 with errors; took 13 seconds  
or 723 ms; speed 204.0 records/sec;
```



# ЗАДАЧА ПОИСКА ЗОЛОТЫХ КАРТОЧЕК ДЛЯ ПЕРЕСЛИЯНИЯ

```
findPartiesToRemerge
```

✔ Возобновляемая, неконфликтная

## Описание задачи

Это многопоточковая задача, которая ищет золотые карточки с неслитыми идентичными атрибутами для переслияния.

Задача наполняет найденными хидами золотых карточек таблицу **task\_remerge\_hids**.

## Алгоритм работы

На момент запуска задача проверяет, что таблица **task\_remerge\_hids** пустая.

Если таблица не пустая, то выводит ошибку:

```
Table task_remerge_hids must be empty
```

Индекс карточки итерируется по золотым карточкам.

Для поиска карточек с определенным partyType необходимо задать этот тип параметра. При пустом параметре задача будет искать карточки по всем возможным partyType.

Алгоритм внесения хидов в **task\_remerge\_hids**:

1. в золотых карточках контрагентов задача пробует слить атрибуты;
2. если слияние удалось, значит предыдущие слияние контрагентов прошло некорректно, и задача вносит хид такой карточки в таблицу **task\_remerge\_hids** (само слияние задача не выполняет, только пробует такую возможность);
3. количество нашедшихся хидов записывается в счетчик. Если значение счётчика по окончании задачи равно 0, значит переслияние карточек не требуется.

Задачу можно запускать с пустым параметром partyType. В таком случае поиск будет идти по всем сущностям.

После завершения задача укажет, какое количество контрагентов на переслияние нашлось. Строчка вида **(to remerge: XXX)**.

Пример:

```

— findPartiesToRemergeTaskPerformer:
PHYSICAL.findPartiesToRemergeTaskPerformer: processed 24327721 entities (to remerge: 868); 0 with errors; took 2215 seconds or 36.92 min; speed 10983.169 records/sec;
LEGAL.findPartiesToRemergeTaskPerformer: processed 291659 entities (to remerge: 30); 0 with errors; took 41 seconds or 41.16 sec; speed 7113.6343 records/sec;
BLACKLIST.findPartiesToRemergeTaskPerformer: processed 0 entities; 0 with errors; took 1 seconds or 47 ms;
EGR.findPartiesToRemergeTaskPerformer: processed 0 entities; 0 with errors; took 1 seconds or 534 ms;
AGREEMENT.findPartiesToRemergeTaskPerformer: processed 0 entities; 0 with errors; took 1 seconds or 85 ms;

```


## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переслияние	
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>
<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>executionTimeout</b>	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.	<b>0</b>
<b>cacheSize</b>	Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память)	<b>0</b>
<b>iteratorRefreshPeriodMin</b>	Как часто (в минутах) переоткрывать итератор по данным, чтобы избежать переполнения UNDO	<b>0</b>
<b>terminateOnErrors</b>	Параметр, указывающий что задачу нужно остановить при возникновении N ошибок	<b>false</b>
<b>maxStoredErrors</b>	Количество ошибок, которое можно накапливать в памяти	<b>0</b>
<b>silentMode</b>	Выполнение задачи в фоновом режиме	<b>true</b>

# ЗАДАЧА ПОИСКА МАССИВНЫХ КАРТОЧЕК КОНТРАГЕНТОВ

## findMassiveParties

 неконфликтная, непрерываемая

Предназначена для поиска массивных контрагентов.

Внутри задачи есть параметр **services.party.massive.sourceCountThreshold** (по умолчанию равен 1000 КА).

Этот параметр настраивается в [админке](#), требует рестарта ЕК.

### Алгоритм работы задачи

Выполняется селекты по `merged_party`.

Если в контрагенте исходных больше, чем **services.party.massive.sourceCountThreshold**, то `hid_party` сохраняется в таблицу **MASSIVE\_PARTY\_REGISTRY**.

При попадании хида в таблицу **MASSIVE\_PARTY\_REGISTRY**, он не лочится при загрузке инкремента и при обновлении из SOAP, а обрабатывается отдельным процессом.

Контрагентов можно добавить вручную, указав тип. Записи с флагом **TYPE = true** игнорируются при повторных запусках задачи.

Если при отправке работе SaM был обнаружен дубль с хидом из **MASSIVE\_PARTY\_REGISTRY**, то автоматического слияния для гарантированных правил не происходит, а в SaM возвращается исх. созданная карточка.

### Таблица **MASSIVE\_PARTY\_REGISTRY** — реестр массивных КА для отложенной обработки

Имя	Тип	Описание
<code>hid_party</code>	<code>bigint</code>	Исторический идентификатор контрагента
<code>party_type</code>	<code>varchar(50)</code>	Тип контрагента
<code>created</code>	<code>timestamp</code>	Дата слияния
<code>type</code>	<code>boolean</code>	Признак ручного добавления записи
<code>reason</code>	<code>varchar(250)</code>	Причина добавления

Обработка массивных карточек делается задачей **massivePartyReMergeTask**.

# ЗАДАЧА ПОЛНОГО ПЕРЕСЛИЯНИЯ КОНТРАГЕНТОВ

reMergeFullByPartyHids

✓ Возобновляемая

! Конфликтующая

## 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.

Задачу **reMergeFullByPartyHids** необходимо запускать для каждого **partyType** отдельно.

Задача пересливает карточки из таблицы **task\_remerge\_hids** только по указанному **partyType**. Остальные карточки остаются в таблице. Поэтому можно перезапускать задачу для следующих сущностей без повторного запуска задачи **findPartiesToRemerge**.

## Запуск задачи

Просто запустить задачу **reMergeFullByPartyHids**, она сама наполнит таблицу **task\_remerge\_hids** хидами золотых карточек, пересольет их и очистит таблицу.

Если нужно переслить только определенные карточки, а не всю базу, добавить список данных идентификаторов в таблицу **task\_remerge\_hids** вручную.

Пример ручного наполнения таблицы для полного переслияния объединенных контрагентов:

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

commit;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переслияние	<b>PHYSICAL</b>
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>
<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями. Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>
<b>notifyListeners</b>	Флаг необходимости оповещать слушателей  Если отключать оповещение, то потом потребуется перестроение индексов и хешей Фактора.	<b>true</b>

Группа reprocess:  
Название reMergeFullByPartyHids

Параметры задачи

Название параметра	Значение	Обязательность
poolSize	8	
pageSize	1000	
partyType	PHYSICAL	

# ЗАКРЫТИЕ АКТУАЛЬНЫХ (РЕЗУЛЬТИРУЮЩИХ И ВСЕХ ИСХОДНЫХ) АТТРИБУТОВ ПО СПИСКУ

attributeCloseActualByHids

## Описание задачи

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

## 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 **attributeCloseActualByHids**.

## Особенности задачи

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

При закрытии атрибута маркер **manual\_status=2** не проставляется, как например при ручном закрытии.

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.

# ЗАКРЫТИЕ ИСХОДНЫХ АТТРИБУТОВ ПО СПИСКУ С ПЕРЕСЛИЯНИЕМ РЕЗУЛЬТИРУЮЩИХ

attributeCloseSourceByHids

## Описание задачи

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

## 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** (пример для атрибута адрес):

```
INSERT INTO task_attribute_hids (attribute_type, hid_attribute, party_type, hid_party)
SELECT 'ADDRESS', a.hid_address, a.party_type, a.hid_party FROM address a join
physical_party pp on pp.hid_party = a.hid_party AND a.version = 0 AND pp.version = 0
AND pp.enddate is null AND pp.merged_status in (0,1);
```

4. Запустить задачу **attributeCloseSourceByHids**.

## Особенности задачи

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

Игнорирует атрибуты, которые:

- закрыты;
- не найдены по идентификатору.

Проигнорированные атрибуты задача пишет в лог-файл.

Если атрибут находится в закрытом контрагенте или относится к «золотым» клиентам, задача упадет с ошибкой

# ЗАПОЛНЕНИЕ ПОЛНОЙ ОПФ

**legalOpfFullCleaner**

✔ Возобновляемая, неконфликтная

## Особенности задачи

Задача обновляет только одно поле **opf\_full** без создания версии и в обход правил обновления. На вход принимает краткую форму ОПФ, на выход выдает полную.

## Запуск задачи

Просто запустить задачу **legalOpfFullCleaner**, она сама создаст временную таблицу **tmp\$legal\_opf\_full\_party\_hids**, наполнит ее идентификаторами ЮЛ и удалит после обработки всех записей.

## Параметры задачи

Параметр	Описание	Значение по умолчанию
<b>skipJmsEventsPublish</b>	Не отправлять события в обратный поток	true



# ЗАПОЛНЕНИЕ ФИО ИП

**legallpReCleaner**

## Особенности задачи

1. Задача работает по списку **task\_party\_hids**.
2. Обрабатываются все актуальные исходные записи ЮЛ, в которых не заполнены все компоненты ФИО ИП.
3. Компоненты ФИО ИП заполняются для всех записей, у которых заполнено стандартизированное наименование ЮЛ и выполняется хотя бы одно из следующих условий:
  - a. Заполнено стандартизированное значение ОПФ и оно соответствует ОПФ индивидуальных предпринимателей (ИП, Адвокаты, Главы КФХ и т.д.).
  - b. Заполнено стандартизированное значение ИНН и оно соответствует ИНН ИП (12 знаков).

## Запуск задачи

Перед запуском:

1. Выполнить задачу **cleanTaskParameterTables** для очистки таблицы **task\_party\_hids**;
2. Заполнить таблицу **task\_party\_hids**, выполнив один из следующих запросов:
  - Запрос для баз данных с одиночными таблицами:

```
insert into TASK_PARTY_HIDS(HID_PARTY, PARTY_TYPE)
select HID_PARTY, 'LEGAL'
from LEGAL_PARTY
where VERSION = 0 and MERGED_STATUS in (0, 1) and ENDDATE is null and
(ip_surname is NULL and ip_name is NULL and ip_patronymic is NULL)
```

- Запрос для баз данных с историческими таблицами:

```
insert into TASK_PARTY_HIDS(HID_PARTY, PARTY_TYPE)
select HID_PARTY, 'LEGAL'
from LEGAL_PARTY
where MERGED_STATUS in (0, 1) and ENDDATE is null and (ip_surname is NULL and
ip_name is NULL and ip_patronymic is NULL)
```

После этого выполнить задачу **legallpReCleaner**.

# ИСПРАВЬТЕ БАЗУ ДАННЫХ В СЛУЧАЕ НЕКОНСИСТЕНТНОГО СОСТОЯНИЯ

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

# ПЕРЕОБРАБОТКА EMAIL

**emailNotValidReCleaner**

✔ Возобновляемая, неконфликтная

## Описание задачи

В релизе 20.10 «Единый клиент» обзавёлся маркерами некорректности email. Задача создана для того, чтобы переобработать существующий объём базы емейлов с КК **NOT\_VALID**:

- проставить им причины некорректности email в виде маркеров
- добавить список изменений, проведённых с email в процессе стандартизации

## Особенности задачи

1. Задача очищает только емейлы с кодом качества **NOT\_VALID**.
2. Сама создаёт себе таблицу **tmp\$rcIn\_email\_not\_valid\_hids** и заполняет актуальными хидами при запуске.
3. Если у email уже проставлены все возможные коды, то такая запись не закрывается и новая запись не создаётся.

## Запуск задачи

Нужно запустить задачу **emailNotValidReCleaner**, дополнительных параметров указывать не надо.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которого выполняется переобработка	<b>PHYSICAL</b>
<b>unsafe.forceExecution</b>		<b>true</b>

# ПЕРЕОБРАБОТКА ТЕЛЕФОНОВ

phoneExtraFieldsReCleaner

✓ Возобновляемая

! Конфликтующая

## 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 всей базы данных.

## 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 для оригинальных и золотых,
поэтому ставим условие merged_status in (0, 2)
insert into task_party_hids
select hid_party from physical_party where merged_status in (0, 2) and ... ;
```

4. Задать для задачи **phoneExtraFieldsReCleaner** параметр **hidsTableName** = **task\_party\_hids**.

5. Запустить переобработку.

## Переобработка всей базы

Если хотите переобработать всю базу, просто запустите задачу **phoneExtraFieldsReCleaner**: она сама создаст временную таблицу, заполнит ее хидами актуальных контрагентов, переобработает их телефоны и очистит таблицу, если процесс прошел без ошибок.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>PHYSICAL</b>
<b>massiveChanging</b>	Оповещение Фактора инкрементальными изменениями. При массовой переобработке нужно задать <b>true</b> , а потом запустить полный поиск дубликатов	<b>false</b>
<b>hidsTableName</b>	Название таблицы для заполнения идентификаторами КА, по умолчанию пусто	

## Особенности задачи

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

Игнорирует идентификаторы записей (без ошибки в лог), которые:

- относятся к влитым 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.

# ПЕРЕОБРАБОТКА ТЕЛЕФОНОВ. НОВЫЙ ВАРИАНТ

phoneExtraFieldsActualization.physical

phoneExtraFieldsActualization.legal

 Возобновляемая

## Описание задачи

Эта задача призвана заменить долгую и тяжёлую `phoneExtraFieldsReCleaner`. Механизм работы задачи тот же, она 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.

Однако вместо `task_party_hids` (которую по своему выбору можно наполнять теми или иными наборами хидов, а по умолчанию обрабатывается вся база) данная задача использует аналогичную таблицу `task_phone_party_hids`, которая заполняется автоматически в рамках дедупликации хидами тех контрагентов, у которых:

1. Есть мобильный телефон (`numberProfile = MOBILE`)
2. У этого телефона ранее менялся оператор, а теперь было изменено одно из доп. полей
3. Вместо п.2 оператор был изменён после добавления телефона в ЕК (в таком случае изменение доп. полей роли не играет)

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<code>partyType</code>	Тип контрагента, для которых выполняется переслияние	<b>PHYSICAL</b>

## 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.



# ПЕРЕОТКРЫТИЕ ЗАКРЫТЫХ ПО ОШИБКЕ КОНТРАГЕНТОВ С СОХРАНЕНИЕМ ИСТОРИИ

partyReopenByHids

✔ Возобновляемая

## 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.

## 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;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется восстановление	<b>PHYSICAL</b>
<b>reopenRelations</b>	Переоткрыть связи контрагента?	false
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>

<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями. Если задать <b>true</b> , то Фактор не будет вызываться с инкрементальными <b>populate</b>	<b>false</b>

Группа reprocess  
 Название partyReopenByHids

Параметры задачи

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

Сохранить Отмена

# ПЕРЕОЧИСТКА EMAIL

**emailAllReCleaner**

✔ Возобновляемая, неконфликтная

## Описание задачи

В релизе 20.22 «Единый клиент» стал заполнять новые поля с маркерами email, emailForDedup и распознавать типы доменов. Задача создана для того, чтобы переобработать существующий объём базы емейлов:

- добавить список изменений, проведённых с email в процессе стандартизации
- проставить тип домена
- заполнить поле emailForDedup, используемое для поиска дубликатов и на странице сравнения дубликатов показать emailForDedup, если заполнен

## Особенности задачи

emailForDedup используется для подсчёта популярности, если заполнено.

## Запуск задачи

Нужно запустить задачу **emailAllReCleaner**, дополнительных параметров указывать не надо.

В режиме горячего резерва запускать только на одной ноде.

При запуске с умолчательными параметрами создаст таблицу **tmp\$rcIn\_email\_invalid\_hids** и заполнит актуальными хидами при запуске. Далее обработка будет выполняться по этой таблице.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которого выполняется переобработка	<b>PHYSICAL</b>

## Логирование

Задача логирует переочистку каждые N записей (по умолчанию **100 000**).

---

**INFO AttributeReCleanRefineTask - Processed 10900 'PHYSICAL' parties. Iterated 100000 parties.**

---

В итоговой статистике выводится:

---

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

---

# ПЕРЕОЧИСТКА ГРАНУЛЯРНЫХ КОДОВ КАЧЕСТВА ФИО

**physicalFioQcReCleaner**

✔ Возобновляемая

✔ Неконфликтующая

По итогам работы задачи происходит принудительное **inplace** обновление (без создания новой версии) гранулярных кодов качества, маркеров ФИО и основного кода качества ФИО, остальные поля ФИО остаются без изменений.

Статус ФИО и факт ручного подтверждения не влияют на заполнение гранулярных кодов качества ФИО задачей.

Факт ручного подтверждения учитывается при переработке основного кода качества ФИО: если изначальный основной КК ФИО «Подтвержден контролером», задача его не меняет.

## Запуск задачи

Нужно запустить задачу **physicalFioQcReCleaner**, указав параметр **fillTaskPartyHids = true** и название **hidsTableName**, например **tmp\$fio\_qc\_party\_hids**. Имя таблицы не должно быть длиннее 27 символов.

Задача сама создаст временную таблицу, заполнит ее данными и удалит по окончании работы.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>poolSize</b>	Максимальное количество потоков	<b>4</b>
<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>fillTaskPartyHids</b>	Нужно ли предварительно заполнить таблицу hid-ами контрагентов. Первичное заполнение не выполняется, если в таблице что-то есть.	<b>false</b>
<b>hidsTableName</b>	Название таблицы для заполнения	<b>task_party_hids</b>

## *Логирование*

Задача логирует переочистку каждые N записей (по умолчанию **100 000**).

---

**INFO RequisiteReCleanTask - Processed 100 000 'PHYSICAL' parties. Iterate 100 000 parties.**

---

В итоговой статистике выводится:

---

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

---

# ПЕРЕОЧИСТКА КОДОВ КАЧЕСТВА ДАТ РОЖДЕНИЯ

physicalBirthdateQcReCleaner

✔ Возобновляемая, неконфликтная

## Описание задачи

В релизе 19.12 «Единый клиент» обзавелся новыми кодами качества дат рождения. Задача создана для того, чтобы переобработать существующий объем базы дат рождения:

- проставить им более подробные коды качества.
- заполнить поле **birthdate\_for\_dedup**, которое используется для дедупликации.

## Особенности задачи

1. Задача работает по списку hid-ов из таблицы, указанной в параметре **hidsTableName**, либо **task\_party\_hids**, если параметр не задан; обновление атрибутов принудительное.
2. Следующие даты рождения не отправляются на переочистку:
  - a. Подтвержденные контролером.
  - b. Имеющие один из кодов качества:

Код	Описание
CURRENT_DATE	Текущая дата
FUTURE_DATE	Дата из будущего

3. После успешного выполнения таблица, указанная в параметре, будет удалена.

## Запуск задачи

Запустить задачу **physicalBirthdateQcReCleaner** без предварительных шагов. Задача сама создаст временную таблицу, наполнит ее идентификаторами актуальных контрагентов, переобработает записи и, если не было ошибок при переобработке, удалит таблицу.

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
----------	----------	-----------------------

<b>partyType</b>	Тип контрагента, для которого выполняется переобработка	<b>PHYSICAL</b>
<b>fillTaskPartyHids</b>	Автоматическое заполнение временной таблицы хидами контрагентов	<b>true</b>



# ПЕРЕОЧИСТКА КОДОВ КАЧЕСТВА КПП

legalKppReCleaner

✓ Возобновляемая

! Конфликтующая

## Описание задачи

Задача переочищает КПП по таблице TASK\_PARTY\_HIDS

## Особенности задачи

1. Задача работает по списку КА из таблицы TASK\_PARTY\_HIDS, которую необходимо предварительно заполнить хидами на переобработку.
2. Задача переобрабатывает КПП и код качества КПП без создания новой версии ЮЛ.
3. Задача умеет переобрабатывать КПП списочников из ЧС 2.0 (LEGAL\_LIST\_PARTY). Для этого достаточно в параметрах задачи указать partyType=LEGAL\_LIST.

## Запуск задачи

1. Убедиться, что таблица TASK\_PARTY\_HIDS пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

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 ЮЛ
insert into task_party_hids(hid_party, party_type) values (1, 'LEGAL');

-- Нужно переобработать набор клиентов. Работает для оригинальных и золотых,
поэтому ставим условие merged_status in (0, 2)
insert into task_party_hids
select hid_party from legal_party where merged_status in (0, 2) and ... ;
```

4. Запустить задачу legalKppReCleaner.

В режиме горячего резерва запускать только на одной ноде.

# ПЕРЕОЧИСТКА КОДОВ КАЧЕСТВА ПАСПОРТОВ

passportQCReCleaner

✔ Возобновляемая, неконфликтная

## Описание задачи

Использовать следует в случае, когда нужно переочистить существующий объем ДУЛ.

## Особенности задачи

Задача работает по списку **task\_party\_hids**, обновление атрибутов принудительное.

Документы подтвержденные контролером на переочистку не отправляются.

## Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу **cleanTaskParameterTables** для очистки таблицы **task\_party\_hids**;
2. Выполнить задачу **fillTaskPartyHidsActual**, указав в параметре **partyType = PHYSICAL**;

После этого запустить задачу **passportQCReCleaner**.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>PHYSICAL</b>
<b>massiveChange</b>	Оповещение Фактора изменениями, по умолчанию выключено (значение <b>true</b> ).	true

# ПЕРЕОЧИСТКА КОДОВ КАЧЕСТВА EDITED ПАСПОРТОВ

## *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 **passportQCReCleaner.edited**

## *Конфигурация задачи*

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>PHYSICAL</b>

# ПЕРЕОЧИСТКА КОДОВ КАЧЕСТВА EMPTY\_DATE ПАСПОРТОВ

## *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 **passportQCReCleaner.emptyDate**

## *Конфигурация задачи*

Настраиваемые параметры:

---

**partyType** Тип контрагента, для которых выполняется переобработка **PHYSICAL**

---

# ПЕРЕОЧИСТКА КОДОВ КАЧЕСТВА NOT\_SURE ПАСПОРТОВ

## *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** и идентификаторами исходных клиентов;

После этого можно выполнять задачу `passportQCR Cleaner.notSure`

## *Конфигурация задачи*

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>PHYSICAL</b>

# ПОДГОТОВКА СПИСКОВ ДЛЯ ПОВТОРНОЙ ОБРАБОТКИ

## Подготовка списков для повторной обработки

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

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

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);
- **fillTaskPartyHidsOriginalWithClosed** — все оригинальные контрагенты (**merged\_status** 0 или 1) и закрытые контрагенты (**end\_date != null**);
- **cleanTaskParameterTables** — очистка таблиц с параметрами задач массовой переобработки.

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



Следующие задачи сначала очищают таблицу, с которой работают, и только после этого начинают наполнение данными. Поэтому предварительно выполнять задачу **cleanTaskParameterTables** не нужно.

Для наполнения таблицы **TASK\_ACT\_PARTY\_HIDS** предназначены отдельные задачи:

- **fillAddressActualizationTaskPartyHids** (отображается в АРМ Администратора только, если подключена **Задача актуализации адресов по ФИАС**) — контрагенты, адреса которых должны быть актуализированы по КЛАДР (**mergedStatus: 0,2**);

- **fillTaskPartyHidsBadAddresses** — контрагенты с адресами, отмеченными на ручную обработку (**mergedStatus** 0 или 2), задача не выбирает контрагентов, если все его адреса, отмеченные на ручную обработку, скрыты.

Зада **fillIncrementalAddressActualizationPartyHids** заполняет таблицу **TASK\_ADDRESS\_ACT\_HIDS** (отображается в АРМ Администратора только, если подключена **Задача актуализации адресов по ФИАС**) — контрагенты, адреса которых нуждаются в актуализации по КЛАДР;



# ПОЛНАЯ ПЕРЕОЧИСТКА ФИО

**physicalFioFullReCleaner**

✔ Возобновляемая

✔ Неконфликтующая

Задача переобрабатывает ФИО и обновляет поля с созданием новой версии.

Обновляются поля:

- SURNAME
- NAME
- PATRONYMIC
- GENDER
- SURNAME\_QC
- FIRSTNAME\_QC
- PATRONYMIC\_QC
- GENDER\_QC
- FULL\_NAME\_EDITED\_CHANGES
- FULL\_NAME\_NOT\_SURE\_REASONS
- FULL\_NAME\_QC

Задача оставляет без изменений ФИО, пол и основной КК ФИО (FULL\_NAME\_QC), если они подтверждены оператором. Гранулярные коды качества ФИО и пола, маркеры ФИО могут обновиться, если изменились при переобработке.

## *Запуск задачи*

### Переобработка всех актуальных ФИО

Запустите задачу с параметрами:

Параметр	Описание	Значение
<b>fillTaskPartyHids</b>	Нужно ли предварительно заполнить таблицу hid-ами контрагентов.	<b>true</b>
<b>hidsTableName</b>	Название таблицы для заполнения	<b>tmp\$fio_reclean_party_hids</b>

Задача сама создаст таблицу `tmp$fioreclean_party_hids`, отберет хиды актуальных КА и наполнит ими таблицу, а затем переобработает ФИО. Имя таблицы в **hidsTableName** может быть любым, но не длиннее 30 символов.

## Переобработка заданных ФИО

Наполните заранее таблицу хидами КА, ФИО которых необходимо переобработать. Можно использовать, например, таблицу `TASK_ACT_PARTY_HIDS`.

Запустите задачу с параметрами:

Параметр	Описание	Значение
<b>fillTaskPartyHids</b>	Нужно ли предварительно заполнить таблицу hid-ами контрагентов.	<b>false</b>
<b>hidsTableName</b>	Название таблицы с хидами для переобработки	<b>TASK_ACT_PARTY_HIDS</b>

Задача переобработает только ФИО указанных КА.

В режиме горячего резерва запускать только на одной ноде.

## Логирование

Задача логирует переочистку каждые N записей (по умолчанию **100 000**).

```
INFO RequisiteReCleanTask - Processed 100 000 'PHYSICAL' parties. Iterate 100 000 parties.
```

В итоговой статистике выводится:

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

# ПОЛНОЕ РАЗЛИЯНИЕ КОНТРАГЕНТОВ ПО СПИСКУ РЕЗУЛЬТИРУЮЩИХ КОНТРАГЕНТОВ

fullUnmergeByResultHids

✓ Возобновляемая

## 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 разливании)

## 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;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется разливание	<b>PHYSICAL</b>
<b>unmergeDescription</b>	Комментарий к задаче (например, причина, по которой выполняется разливание), сохраняется в истории	
<b>unmergePackLimit</b>	Лимит на количество слитых записей в одной обрабатываемой пачке	<b>1000</b>
<b>poolSize</b>	Максимальное количество потоков	<b>4</b>
<b>pageSize</b>	Размер пачки	<b>50</b>

---

**massiveChanging** Оповещение Фактора изменениями. **false**

Если задать **true**, то Фактор не будет вызываться с инкрементальными **populate**

---

**preventFurther Merging** 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	

# РАЗЛИЯНИЕ КОНТРАГЕНТОВ ПО СПИСКУ MERGEDID

unMergeByMergedIds

✓ Возобновляемая

## Описание задачи

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

## 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 ...
)
;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется разлияние	<b>PHYSICAL</b>
<b>unmergeDescription</b>	Комментарий к задаче (например, причина, по которой выполняется разлияние)	
<b>poolSize</b>	Максимальное количество потоков	<b>1</b>
<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>massiveChanging</b>	Оповещение Фактора изменениями.	<b>false</b>

---

Если задать **true**, то Фактор не будет вызываться с инкрементальными **populate**

---

**preventFurtherMerging** Запретить разлитым контрагентам сливаться в дальнейшем (проставить **false** им Confirmation = 4)

---

Группа reprocess.  
Название unMergeByMergedIds

Параметры задачи

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

## ***Выполнение задачи***

Задача запускается из АРМ Администратора.

# ТАБЛИЦЫ С ПАРАМЕТРАМИ ЗАДАЧ МАССОВОЙ ПЕРЕОБРАБОТКИ

- Описание таблиц

Таблица task\_party\_hids

Таблица task\_merged\_ids

Таблица task\_act\_party\_hids

Таблица task\_attribute\_hids

Таблица delete\_party\_hids

Таблица close\_party\_hids

Таблица task\_doubtful\_reclean\_hids

Таблица task\_phone\_party\_hids

Таблица task\_remerge\_hids

Таблица task\_engine\_actualize\_hids

Таблица task\_recalculate\_risk\_hids

## Описание таблиц

Данные таблицы используются для хранения параметров контрагентов, которых необходимо переобработать.

### Таблица task\_party\_hids

№	Поле	Описание
1	HID_PARTY	Уникальный идентификатор контрагента в ЕК
2	PARTY_TYPE	Тип контрагента (литера)
3	HID_RESULT_PARTY	Уникальный идентификатор результирующего контрагента (для более эффективной группировки при обработке)

### 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.

### 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

### *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

### *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)

### *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)

### *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)



### **Таблица *task\_phone\_party\_hids***

Таблица, используемая задачей `phoneExtraFieldsActualization` для актуализации дополнительных полей телефонов.

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

### **Таблица *task\_remerge\_hids***

Таблица, используемая задачами `reMergeFullByPartyHids` и `reMergeRequisiteByPartyHids` для хранения списка идентификаторов контрагентов, которые нужно переслать.

Может быть заполнена задачей поиска золотых карточек для переслания.

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

### **Таблица *task\_engine\_actualize\_hids***

Используется задачей `searchEngineActualizeByParties` для актуализации поискового индекса по списку контрагентов.

Заполняется задачей `searchEngineValidateByIds` при выявлении КА с расхождениями между БД и индексом.

1	HID_PARTY	Уникальный идентификатор контрагента в ЕК
2	PARTY_TYPE	Тип контрагента (литера)

### **Таблица *task\_recalculate\_risk\_hids***

Используется задачей `riskLevelActualize` для расчета уровня риска по списку контрагентов.

№	Поле	Описание
1	HID_PARTY	Уникальный идентификатор контрагента в ЕК

# ЗАДАЧА ПЕРЕОБРАБОТКИ ИНН ЮЛ

**partyLegalInnReCleaner**

✓ Возобновляемая

! Конфликтующая

## Описание задачи

Задача переобрабатывает ИНН и его код качества по списку КА из таблицы **task\_party\_hids**.

1. Если таблица **task\_party\_hids** пустая, то она автоматически наполнится хидами контрагентов с `merged_status` 0 и 2 при запуске задачи. Либо можно предварительно наполнить её вручную скриптом.
2. При переобработке создается новая версия контрагента.
3. Задача умеет переобрабатывать также ИНН списочников ЮЛ. Для этого достаточно указать параметр **partyType=LEGAL\_LIST**.

## Запуск задачи

Для переобработки ИНН у всех юрлиц без выборки:

1. Убедиться, что таблица `TASK_PARTY_HIDS` пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

2. Запустить задачу **partyLegalInnReCleaner**.

Для выборочной переобработки ИНН:

1. Убедиться, что таблица `TASK_PARTY_HIDS` пустая. Если нет, очистить её

```
truncate table task_party_hids;
```

2. Наполнить таблицу **task\_party\_hids** хидами контрагентов для переобработки:

```
-- Хотим переобработать конкретного ЮЛ
insert into task_party_hids(hid_party, party_type) values (1, 'LEGAL');

-- Нужно переобработать набор клиентов. Работает для оригинальных и золотых,
поэтому ставим условие merged_status in (0, 2)
insert into task_party_hids(hid_party, party_type)
select hid_party, 'LEGAL' from legal_party where merged_status in (0, 2) and ... ;
```

### 3. Запустить задачу **partyLegalInnReCleaner**.

В режиме горячего резерва запускать только на одной ноде.

## *Конфигурация задачи*

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется переобработка	<b>LEGAL</b>
	Возможные значения:	
	<ul style="list-style-type: none"><li>• <b>LEGAL_LIST</b></li><li>• <b>LEGAL</b></li></ul>	

# СИСТЕМНЫЕ ЗАДАЧИ

## *Системные задачи*

- **changeJdbcUser** — изменение пользователя безопасности

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

- **deleteOutdatedConflicts** — удаление старых обработанных конфликтов
- **duplicateGroupActualize** — актуализация групп дубликатов
- **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
- **findHistoricalConflicts** — поиск конфликтных изменений

# ALLENGINESREBUILDBYNAME. ПЕРЕСТРОЕНИЕ УКАЗАННЫХ ПОИСКОВЫХ ИНДЕКСОВ

## Общее описание

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

## 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

## 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** — договоры;
- **EGR** — ЕГРЮЛ;
- **PHYSICAL\_LIST** — списки ФЛ;
- **LEGAL\_LIST** — списки ЮЛ;
- **ALL** — все индексы.

# ALLENGINESREBUILDINVALID. ПЕРЕСТРОЕНИЕ ВСЕХ НЕ ПРОШЕДШИХ ВАЛИДАЦИЮ ПОИСКОВЫХ ИНДЕКСОВ

## *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**.

# ВАЛИДАЦИЯ ПОИСКОВЫХ ИНДЕКСОВ ПО ИДЕНТИФИКАТОРАМ

searchEngineValidateByIds

✔ Возобновляемая

❗ Конфликтующая

## Общее описание

Задача валидации поисковых индексов по идентификаторам.

## Алгоритм работы

- Идентификаторы из базы данных и индекса собираются в сортированные файлы на диске.
- Определяется разница — что есть в БД и нет в индексе, и наоборот.
- Отобранные идентификаторы с несоответствиями между БД и индексом фильтруются с учетом возможных временных эффектов:
  - идентификаторы, которые есть в индексе, но нет в БД: в базе ищутся сущности с такими идентификаторами и отбрасываются те, которые создавались или обновлялись после начала сверки (минус 10 мин. на подвисшие транзакции);
  - идентификаторы, которые есть в БД, но нет в индексе: из них отбираются те, которые последний раз обновлялись до начала сверки и при этом являются актуальными на текущий момент. Если такие нашлись, то они «кривые».
- Все «кривые» идентификаторы записываются в таблицу **engine\_validation\_error** — в колонку **id\_essence** (отрицательные id у графа — это ребра **merged**, положительные — связи).
- Также хиды «кривых» КА из индексов записываются в таблицу **task\_engine\_actualize\_hids**, по которой потом работает задача актуализации поискового индекса **searchEngineActualizeByParties**.

В случае наличия расхождений данных между БД и индексом задача завершается со статусом **ERROR**

## Таблица ENGINE\_VALIDATION\_ERROR для записи найденных расхождений

COLUMN_ID	COLUMN_NAME	Описание
1	ID_ERROR	Идентификатор записи

2	ENGINE_NAME	Индекс
3	ID_ESSENCE	Идентификатор КА с расхождением
4	ERROR_TYPE	Тип несоответствия: <ul style="list-style-type: none"> <li>• NOT_FOUND_IN_ENGINE</li> <li>• NOT_FOUND_IN_DB</li> </ul>
5	CREATED	Дата создания
6	NODE_ID	Идентификатор ноды

## Конфигурация задачи

Имя	Тип	Описание	Обязательное	Значение по умолчанию
executionTime out	int	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.		0
cacheSize	int	Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память)		0
terminateOnErrors	boolean	Прекратить выполнение при возникновении ошибки?		false
poolSize	int	Количество потоков, которые будут обрабатывать страницы с данными		0
silentMode	boolean	Режим тихой работы (динамический подбор размера пула задач в зависимости от нагрузки на сервер)		false
pageSize	int	Размер одной страницы данных (увеличивает время блокировок БД)		0
maxStoredErrors	int	Количество ошибок, которое можно накапливать в памяти		0
iteratorRefreshPeriodMin	int	Как часто (в минутах) переоткрывать итератор по данным, чтобы избежать переполнения UNDO		0



# АКТУАЛИЗАЦИЯ ПОИСКОВОГО ИНДЕКСА ПО СПИСКУ КАРТОЧЕК

searchEngineActualizeByParties

✔ Возобновляемая

## Общее описание

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

## Запуск задачи



1. Очистить таблицу **task\_engine\_actualize\_hids**, выполнив запрос:

```
truncate table task_engine_actualize_hids;
```

2. Добавить идентификаторы карточек **hid** для актуализации в таблицу **task\_engine\_actualize\_hids**;
3. Сконфигурировать и запустить задачу (см. ниже).

Пример:

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

  Таблицу **task\_engine\_actualize\_hids** можно наполнить, запустив задачу валидации поисковых индексов по идентификаторам **searchEngineValidateById**.

## Конфигурация задачи

В задаче конфигурируется только параметр **partyType** — тип сущности (по умолчанию **PHYSICAL**).

Группа system  
Название searchEngineActualizeByParties

Параметры задачи

Название параметра	Значение	Обязательность
partyType	PHYSICAL	

Сохранить Отмена

# ВЫПОЛНЕНИЕ ВСПОМОГАТЕЛЬНОЙ ХРАНИМОЙ ПРОЦЕДУРЫ

## *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.

## *Конфигурация задачи*

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

# УДАЛЕНИЕ СТАРЫХ ОБРАБОТАННЫХ КОНФЛИКТОВ

`deleteOutdatedConflicts`

## *Общее описание*

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

## *Конфигурация задачи*

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

# СОЗДАНИЕ И РЕДАКТИРОВАНИЕ СИСТЕМНЫХ УЧЕТНЫХ ЗАПИСЕЙ

changeJdbcUser

✔ Системная

! Функционал использования БД пользователей не доступен по умолчанию, поэтому задача в АРМ Администратора может отсутствовать.

Для подключения напишите нам в поддержку

## Описание задачи

Задача добавляет, изменяет и удаляет БД пользователей.

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

## Конфигурация задачи

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

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.

! Начиная с релиза 23.7 и выше, необходимо исключить префикс \$s0 из значения сгенерированного хеша при последующем создании/изменении пароля пользователя.

Было:

```
passwordHash = $s0$e0801$zx5wPRhdtSTbIFuAzEqOvg==$c60b4IgC+4GpARAE4adV0Xv/HPeOecvIJQiMBUsLD7Y=
```

Стало:

```
passwordHash = $e0801$zx5wPRhdtSTbIFuAzEqOvg==$c60b4IgC+4GpARAE4adV0Xv/HPeOecvIJQiMBUsLD7Y=
```

## Примеры

### Создать нового пользователя

Параметр	Значение
<b>username</b>	<b>user1</b>
<b>passwordHash</b>	<b>\$e0801\$6ZeK6idgRjPL3Za0sezDFA==\$ZpdgXt8jWpU02EuZ9btIhHPiB9A1S2eWOUHqhDx27Co=</b>
<b>roles</b>	<b>ROLE_ADMINISTRATOR, ROLE_PERFORMER</b>
<b>delete</b>	<b>false</b>

### Изменить пароль

Параметр	Значение
<b>username</b>	<b>user1</b>
<b>passwordHash</b>	<b>\$e0801\$SKg6OZwd84heZhCMn7nvcA==\$MixhOphiPBsGIWGy0sWyLhIVsga082LCL0GEVhRNKWM=</b>
<b>delete</b>	<b>false</b>

### Изменить роли

Параметр	Значение
<b>username</b>	<b>user1</b>
<b>roles</b>	<b>ROLE_OPERATOR</b>
<b>delete</b>	<b>false</b>

### Удалить пользователя

Параметр	Значение
<b>username</b>	<b>user1</b>
<b>delete</b>	<b>true</b>

# ШИФРОВАНИЕ ПАРОЛЯ ЧЕРЕЗ УТИЛИТУ SCRYPTPASS

✔ Утилита **scryptpass** используется для шифрования пароля по алгоритму SCrypt.

## Установка

❗ Должна быть определена переменная окружения `JAVA_HOME`.

Распаковать архив **scryptpass.zip**.

## Способ 1. (рекомендованный) Запуск шифрования паролей из консоли

В консоли ввести:

```
{path}\scryptpass> run password1 password2 password3
```

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

```
$s0$e0801$jyKC7AAbZsu8XBcuRHPlpw==$60d4+3vRC05ZOa5I6x0bw+EK3FIJzfET+wCgU33  
LCZc=  
$s0$e0801$fqDFHtqwp7MUEFKkZJCUUw==$VBNVjgjqy2QdzZYWUwfGPtwywlbeB197M0Xykr  
71Z30=  
$s0$e0801$9IqwdWeg2R5exDQui5cHDQ==$cSrlW0H30cmRNf0jeSDExser43b4vc1RZblqAqB6  
wxw=
```

## Способ 2. Запуск шифрования паролей указанных в файле

❗ **Обратите внимание**

Перед шифрованием убедитесь в корректности кодировки файла и отсутствии скрытых символов в файле с паролем.

Если после создания пароля данным образом авторизация не проходит из-за ошибочного пароля рекомендуем воспользоваться шифрованием из консоли.

Добавить в файл **scryptpass\data.txt** пароли для шифрования.

Запустить **scryptpass/run.bat**.

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

## Пример:

На вход:

```
scryptpass\data.txt
```

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

На выходе:

```
crypted.txt
# Файл в кодировке UTF-8 (без BOM)
# Комментарии копируются как есть
# Можно задавать несколько колонок, разделитель табуляция. Кодируется первая
  колонка. Ниже строки для примера.
$s0$e0801$naKBJ92C00i11ERiNIXNfQ==$DgjGHcoeGJYA6b6qoR
/PdYTnkWSVFHwfziKcWgIVo+0=
$s0$e0801$iehrFvlwQ9FNiO574WwkbQ==$OSfM7rTReZjlgfM6zK3UUo7e1igsSzMnKqRXj14C
QHg=      неcodируемое_значение_1      неcodируемое_значение_2
```



# SEARCHENGINEOPTIMIZE. ОПТИМИЗАЦИЯ ПОИСКОВЫХ ИНДЕКСОВ

## Общее описание

Задача **searchEngineOptimize** предназначена для оптимизации поисковых индексов:

- граф связей — **GRAPH**;
- конфликты обновления — **CONFLICT**;
- контрагенты — **PHYSICAL** (и др настроенные типы контрагентов);
- группы — **MANUAL\_DUPLICATE\_GROUP**.

## Особенности

Если индекс пустой или количество сегментов меньше указанного в параметрах задачи **segmentsCount**, то индекс остается без изменений.

По логу понять этот момент нельзя, задача все равно напишет, что индекс был оптимизирован (держится люсиновский метод и он не говорит пустой индекс или нет, и что именно с ним сделано).

## Конфигурация задачи

Параметр	По умолчанию	Описание
<b>segmentsCount</b>	<b>3</b>	Количество сегментов индекса, до которых нужно сливать индекс
<b>poolSize</b>	<b>2</b>	Максимальное количество потоков
<b>engines</b>		Список индексов, которые будут оптимизированы (по умолчанию все).
<b>skipEngines</b>		Список индексов, которые будут пропущены (по умолчанию никого не спускает).

## Логирование

Пример:

```
INFO TaskLauncherTemplate - Task 'searchEngineOptimize' STARTED by performer
INFO SearchEngineTemplate - Optimize search engine 'CONFLICT' with 3 max segments...
INFO SearchEngineTemplate - Search engine 'CONFLICT' optimized in 3.407 ms
INFO SearchEngineTemplate - Optimize search engine 'GRAPH' with 3 max segments...
INFO SearchEngineTemplate - Optimize search engine 'PHYSICAL' with 3 max segments...
INFO SearchEngineTemplate - Search engine 'PHYSICAL' optimized in 2.258 ms
INFO SearchEngineTemplate - Search engine 'GRAPH' optimized in 2.999 ms
INFO SearchEngineTemplate - Optimize search engine 'BLACKLIST' with 3 max segments...
INFO SearchEngineTemplate - Optimize search engine 'MANUAL_DUPLICATE_GROUP' with
```

**3 max segments...**

**INFO SearchEngineTemplate - Search engine 'MANUAL\_DUPLICATE\_GROUP' optimized in 2.073 ms**

**INFO SearchEngineTemplate - Search engine 'BLACKLIST' optimized in 2.685 ms**

**INFO TaskLauncherTemplate - Task 'searchEngineOptimize' finished with status FINISHED in 356 ms**

# GRAPHENGINEFULLREBUILD ПЕРЕСТРОЕНИЕ ГРАФА СВЯЗЕЙ

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

Не поддерживается продолжение перестроения после остановки, т.е. задача невозобновляемая.

## *Конфигурация задачи*

Параметр	Тип	Описание	Примеры значений
strategy	Строка	Стратегия перестроения, блокирующая, не блокирующая	BLOCKING, CONCURRENT
poolSize	Число	Количество потоков, по умолчанию берет из engine.rebuild.poolSize	

# ENGINESFULLREBUILD. ПОЛНОЕ ПЕРЕСТРОЕНИЕ ПОИСКОВОГО ИНДЕКСА

## Общее описание

Задача **enginesFullRebuild** предназначена для полного перестроения поисковых индексов контрагентов (ФЛ, ЮЛ, договоры, и т.п.).

⚠ Не перестраивает граф связей! Рекомендуется запускать после перестроения графа связей.

Для бесшовной работы приложения новый слепок поискового индекса создается рядом с текущим, поэтому для корректного выполнения задачи должно быть доступно большое количество свободной памяти (для ориентира: не менее 40% свободной памяти в дисковом пространстве где развернут CDI). После корректного выполнения задачи старый слепок индексов будет удален.

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 **allEnginesRebuild ByName**, чтобы не тратить время на перестроение ненужных индексов.

При работе в параллель с другими задачами скорость перестроения может заметно падать.

## Конфигурация задачи

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

# FINDHISTORICALCONFLICTS. ПОИСК КОНФЛИКТНЫХ ИЗМЕНЕНИЙ

findHistoricalConflicts

✔ Возобновляемая

✔ Неконфликтная

## Описание задачи

Поиск конфликтных изменений по оригинальным и золотым карточкам в актуальной базе (отдельно среди физлиц и юрлиц)

## 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 с нужным party\_type.

После этого можно выполнять задачу **findHistoricalConflicts**.

Если ранее запускали задачу инициализации прошлых версий **pastValuesInit**, перед запуском **findHistoricalConflicts** нужно очистить таблицу **CONFLICTED\_PARTY\_HIDS**.

В режиме горячего резерва запускать только на одной ноде.

## Алгоритм

1. Для актуальных клиентов из **TASK\_PARTY\_HIDS** просматривается их история по изменению ФИО, ИНН, ОГРН.  
Для каждой актуальной версии реквизита проверяется, не является ли его значение конфликтным по отношению к предыдущему значению и к золотой карточке (для влитых КА).
2. Найденные конфликты записываются в **CONFLICTED\_PARTY\_HIDS**.

3. Для найденных конфликтов между исходной карточкой и результирующей дополнительно сохраняется *id\_merged* в таблицу **TASK\_MERGED\_IDS**, по которой работает задача **unMergeByMergedIds** — разлияние контрагентов по списку MergedId.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется поиск конфликтов	<b>PHYSICAL</b>

## Логирование

После выполнения задачи в лог пишется статистика:

- *conflicted* — сколько всего нашли конфликтов (= сколько записей попало в **conflicted\_part\_y\_hids** за этот запуск задачи)
- *to\_unmerge* — сколько к разлиянию (= сколько записей попало в **task\_merged\_ids** за запуск)
- *total\_original* — сколько всего оригинальных неслитых карточек (*merged\_status=0*) с конфликтом среди своих обновлений (версий)
- *total\_merged* — сколько всего влитых карточек (*merged\_status=1*) с конфликтом среди своих обновлений (версий)
- *total\_resulted* — сколько уникальных золотых карточек (*merged\_status=2*), которые конфликтуют с какой-либо из своих исходных по актуальным данным

Пример лога:

```
— findHistoricalConflictsTaskPerformer: processed 6756104 entities (conflicted: 948, to_unmerge: 95, total_original: 709, total_merged: 157, total_resulted: 60); 0 with errors; took 618 seconds or 10.31 min; speed 10932.207 records/sec;
```

Из лога: *total\_original*: 709, *total\_merged*: 157 - это карточки, у которых конфликты внутри своих версий, а *total\_resulted* - это результирующие, у которых есть конфликт по данным с актуальными версиями исходных.

В *conflicted* попадают только исходные (это кол-во исходных карточек, которые имеют конфликт внутри своих версий или с рез-м по данным (или все вместе)). Под *total\_resulted* может быть по несколько исходных на каждого. Такие учитываются в *to\_unmerge*.

Сумма *to\_unmerge*: 95, *total\_original*: 709, *total\_merged*: 157 должна быть не меньше *conflicted*.

# ЗАДАЧА ОБНОВЛЕНИЯ КЭШЕЙ

`clearCache`

## *Общее описание*

Задача поможет немедленно увидеть изменения, если например:

- добавили новое значение в таблицу REFERENCE
- увеличили длину поля в какой-то таблице
- добавили новый тип связи
- добавили новую систему в таблицу SYSTEM и т. д.

## *Запуск задачи*

1. Внести изменения в базу данных.
2. Запустить задачу `clearCache`.

Задачу достаточно запустить на любой ноде, она сама пробежит по всем остальным.

# СНЯТИЕ БЛОКИРОВКИ С SQL ЗАПРОСА

✔ Неконфликтующая

## Общее описание

Задача `releaseFrozenSQL` предназначена для снятия блокировки зависшего SQL запроса.

Хэш запроса можно получить из `cdi.log`, в сообщении вида:

```
INFO IterableSupport - Sql with MD5 [47907d7fee83f5ea14a003b3e795e4e4] is activated
```

Также если запрос в статусе `waiting` находится больше часа, то пишем ошибку уровня `ERROR` в `cdi.log` и `cdi-full.log` соответственно:

```
ERROR IterableSupport - Sql with MD5 [47907d7fee83f5ea14a003b3e795e4e4] is running more than 3600 seconds
```

## Конфигурация задачи

Параметр	Тип	Описание
<code>sqlMD5</code>	Строка	Хэш запроса

После успешного снятия блоикровки в логе будет сообщение вида:

```
INFO IterableSupport - Sql with MD5 [47907d7fee83f5ea14a003b3e795e4e4] is released
```



# FASTALLENGINESREBUILDBYNAME. БЫСТРОЕ ПЕРЕСТРОЕНИЕ ПОИСКОВЫХ ИНДЕКСОВ

## Общее описание

Задача **fastAllEnginesRebuildByName** предназначена для быстрого перестроения поисковых индексов контрагентов (ФЛ, ЮЛ, договоры, и т. п.).

⚠ Перестраивает только указанные поисковые индексы контрагентов!

## Конфигурация задачи

Параметр	Тип	Описание	Примеры значений	По умолчанию
<b>enginesToRebuild</b>	Строка	Названия индексов, которые будут перестроены (через запятую). Перестроение возможно только для индексов контрагентов.	PHYSICAL, LEGAL	PHYSICAL
<b>strategy</b>	Строка	Быстрая переиндексация с блокировкой базы — FAST_CONCURRENT_WITH_LOCK	FAST_CONCURRENT	FAST_CONCURRENT_WITH_LOCK
		Быстрая переиндексация без блокировки базы — FAST_CONCURRENT	FAST_CONCURRENT_WITH_LOCK	

## Допустимые значения параметра *enginesToRebuild*

- **PHYSICAL** — физические лица
- **MANUAL\_DUPLICATE\_GROUP** — группы для ручной проверки дубликатов
- **INDIVIDUAL** — индивидуальные предприниматели
- **LEGAL** — юридические лица
- **AGREEMENT** — договоры
- **EGR** — ЕГРЮЛ

# ЗАДАЧА ВЫВОДА В ЛОГ АКТИВНЫХ SQL-ЗАПРОСОВ

 loggingRunningSQL

 Неконфликтующая

## Общее описание

Задача *loggingRunningSQL* показывает активные блокирующие SQL-запросы.

Хэш запроса можно получить из *cdi.log*, в сообщении вида:

```
INFO LoggingRunningSQLTask - -----  
INFO LoggingRunningSQLTask - Active sql request count: 1  
INFO LoggingRunningSQLTask - Active blocked sql request :  
[781c4b01b52b61794028318b56273f18] since 21.07.2023 15:47:29  
INFO LoggingRunningSQLTask - -----
```

## Особенности

Дополнительно задача сообщает дату старта SQL-запроса.

Если же в момент запуска задачи *loggingRunningSQL* активных запросов нет, то в *cdi.log* запишется сообщение вида:

```
INFO LoggingRunningSQLTask - -----  
INFO LoggingRunningSQLTask - Active sql request count: 0  
INFO LoggingRunningSQLTask - -----
```

## Конфигурация задачи

У задачи нет обязательных параметров.

# 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

# 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.

## 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».

## 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.

\* При параметре **updateEndDateForDeleted = true** связи не удаляются, а закрываются.

## Конфигурация задачи

С релиза 20.18 у задачи появился настраиваемый параметр:

Название параметра	Тип	Описание	Значение по умолчанию
<b>updateEndDateForDeleted</b>	true / false	Параметр определяет, удалять ли запись о закрытой связи или проставлять ей дату закрытия.	<b>false</b>

---

Если задать **true**, запись не удалится из таблицы, а закроется.

---

## Логирование

В логах cdi, cdi-full пишется INFO:

1. Старт задачи
2. Какое количество записей будет обрабатывать
3. Прогресс работы задачи каждые 100 тысяч записей
4. Финиш с результатом

```
2022-06-08 00:40:42,223 [task:scheduler#555bc10-1] INFO TaskLauncherTemplate - Task
'incrementalExportRelations' STARTED by HFLabs-CDI@unknown
2022-06-08 00:40:42,234 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Export date interval: [07.06.2022 15:51:00, 08.06.2022 00:40:42]
2022-06-08 00:40:43,946 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
The size of all changed relations: 601403
2022-06-08 00:40:43,946 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Remove relations by merged essences
2022-06-08 00:40:51,294 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Removed 12881 relations
2022-06-08 00:40:51,294 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Actualize relations by merged/unmerged essences
2022-06-08 00:45:14,060 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Created 33886 and removed 28042 relations
2022-06-08 00:45:14,060 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Actualize created/closed relations
2022-06-08 00:45:47,666 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Processed 100 450 essences (create: 59456, delete:
2022-06-08 00:52:21,503 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Iterated 100418 relation
2022-06-08 01:02:50,820 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Processed 200 011 essences (create: 81640, delete: 118371)
2022-06-08 01:04:07,091 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Iterated 200418 relation
2022-06-08 01:04:31,679 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Processed 300 094 essences (create: 176815, delete: 123279)
2022-06-08 01:10:15,097 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Iterated 300418 relation
2022-06-08 01:12:37,193 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Processed 400 508 essences (create: 242136, delete: 158372)
2022-06-08 01:12:50,418 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Iterated 400418 relation
2022-06-08 01:13:13,904 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Processed 500 082 essences (create: 341608, delete: 158474)
2022-06-08 01:13:27,403 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Iterated 500418 relation
2022-06-08 01:13:52,855 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Processed 600 584 essences (create: 441758, delete: 158826)
2022-06-08 01:14:03,036 [TaskManagerExecutor-2] INFO IncrementalExportRelationTask -
Iterated 600418 relation
2022-06-08 01:14:03,561 [TaskManagerExecutor-2] INFO DBPropertiesService - Update
property 'incrementalExportRelations.lastDate' from value '07.06.2022 15:51:00' to
'08.06.2022 00:40:42'
2022-06-08 01:14:03,563 [AsyncApplicationEventListenerExecutor-70] INFO
propertiesChange - Изменено свойство: incrementalExportRelations.lastDate (07.06.2022
```

**15:51:00 -> 08.06.2022 00:40:42)**  
**2022-06-08 01:14:03,564 [TaskManagerExecutor-2] INFO TaskPerformerStatsInterceptor - FINISHED task 'incrementalExportRelations [incrementalExportRelationTaskPerformer]'; processed 634393 entities (create: 475283, delete: 159110); 0 with errors; took 2001 seconds or 33.36 min; speed 317.038 records/sec;**  
**2022-06-08 01:14:03,565 [TaskManagerExecutor-2] INFO TaskLauncherTemplate - Task 'incrementalExportRelations' finished with status FINISHED in 33.36 min; — incrementalExportRelationTaskPerformer: processed 634393 entities (create: 475283, delete: 159110); 0 with errors; took 2001 seconds or 33.36 min; speed 317.038 records/sec;**

# 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**.

## 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. Если связь ранее подтверждалась Фактором, а сейчас нет, то такая связь выгрузится закрытой.

## 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**

# ДИАГНОСТИКА

## Когда использовать

По просьбе сотрудника техподдержки.

## 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 75Мб 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**

# СБОР ЛОГОВ

## Общее описание

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

## 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
<b>sendEmail</b>	Признак необходимости отправки диагностики по почте	<b>true</b>
<b>params</b>	Диапазон дат, задается в формате <b>гггг-мм-дд</b> . По умолчанию даты не заданы, тогда будут собраны только текущие логи.	<b>lowerBound=; upperBound=</b>
<b>runInCluster</b>	Для сбора в режиме горячего резерва	<b>true</b>
<b>sendTo</b>	Если нужно временно перекрыть настройку «кому писать письмо» из конфигурации	
<b>archivePassword</b>	Опциональный параметр для защиты паролем архивов диагностики, отправленных через почтовый сервер. Удаляется из параметров задачи после её выполнения	

## 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

# СБОР ДИАГНОСТИКИ ПО КОНТРАГЕНТУ

## Общее описание

Задача **diagnosticParty** предназначена для сбора информации по конкретному контрагенту или по группе контрагентов. С релиза 20.23 можно собирать в диагностике информацию по связанным контрагентам и потенциальным дубликатам.

**i** Параметры **includeClosedRelations** и **duplicateStatus** появились и работают с релиза 21.3

В релизе 21.3 отключили поддержку параметра **params**. Теперь HID КА задается в параметре **hid**, уровень сбора диагностики в параметре **level**.

## Конфигурация задачи

<b>Задача: diagnosticParty</b>	
Группа: Диагностика	
<b>Название параметра</b>	<b>Значение</b>
profiles	sql.party,mapdb-state
hid	1024
sendEmail	true
sendTo	email@hflabs.ru

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию	Пример
<b>sendEmail</b>	Признак необходимости отправки диагностики по почте	<b>true</b>	
<b>sendTo</b>	Список получателей письма с диагностикой через запятую		
<b>hid</b>	HID конкретного контрагента или группы контрагентов, указанных через ","	<b>1024</b>	
<b>level</b>	Уровень сбора диагностики, можно указывать несколько уровней через ","		<b>BASIC, INCLUDE_RELATED</b>
<b>includeS</b>	Признак необходимости собирать данные из поискового индекса	<b>false</b>	

---

**earchIndex**

---

**archivePassword** Опциональный параметр для защиты паролем архивов диагностики, отправленных через почтовый сервер. Удаляется из параметров задачи после её выполнения

---

## Уровни сбора диагностики

- **BASIC** – в диагностику собираются только запрошенные КА (по умолчанию всегда подразумевается этот уровень, его можно не указывать);
- **INCLUDE\_RELATED** – в диагностику включаются связанные с заданными КА сущности;

По умолчанию собирается информация по всем связанным КА и по КА, которые были связаны когда-то, но сейчас связь закрыта. Если в диагностике не нужна информация по КА из закрытых связей, то добавьте в задачу параметр

**includeClosedRelations = false**

- **INCLUDE\_POTENTIAL\_DUPLICATES** – в диагностику включаются потенциальные дубликаты для заданных КА.

По умолчанию собирается информация только по потенциальным дубликатам — это пары дублей в статусе UNKNOWN, ACCEPT, POSTPONE. Но есть возможность собирать данные и по остальным статусам, например, REJECT или CONFLICT.

Параметр **duplicateStatus** принимает на вход статусы пары дублей, можно задавать как один статус, например, `duplicateStatus = CONFLICT`, так и несколько статусов через запятую — `duplicateStatus = CONFLICT, UNKNOWN, CANCEL`. В этом случае в диагностику включаются данные с указанными статусами дублей.

## Данные из поискового индекса

Данные собираются по всем типам контрагентов, из графа связей и из индекса групп дубликатов. Содержимое индексов десериализуется и помещается в файлы вида **% имя\_search\_engine%\_index.json**.

Примеры:

- PHYSICAL\_index.json
- GRAPH\_MERGED\_index.json
- MANUAL\_DUPLICATE\_GROUP\_index.json
- CONFLICT\_index.json

Файл не создается для данного **searchEngine**, если с его помощью ничего не найдено.

# СБОР ДИАГНОСТИКИ С УКАЗАННЫМ ПРОФИЛЕМ

## Общее описание

Задача **diagnosticProfiles** собирает диагностику по профилю.

## Конфигурация задачи

Название параметра	Значение	Обязательность
runInCluster	true	
profiles	config,versions	

Сохранить Отмена

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
runInCluster	Сбор диагностики в режиме горячего резерва	<b>true</b>
profiles	Список профилей, указанных через ","	<b>config,db.connection,logs,search,sql.counts,sql.db,sql.metadata,versions</b>
sendEmail	Признак необходимости отправить письмо	<b>false</b>
sendTo	Email, на который требуется отправляться архив диагностики. Список получателей письма через запятую	
archivePassword	Опциональный параметр для защиты паролем архивов диагностики, отправленных через почтовый сервер. Удаляется из параметров задачи после её выполнения	

## Несколько профилей

Задача **diagnosticProfiles** позволяет собирать в один архив диагностику сразу по нескольким профилям и параметрам.

## Пример

Параметр	Описание	Значение по умолчанию
<code>profiles</code>	Список профилей, указанных через ","	<b>config,db.connection,logs,search,sql.counts,sql.db,sql.metadata,versions</b>
<b>params</b>	Список необходимых параметров для разных профилей. Если не указать необходимый параметр, например, <code>hid_party</code> для профиля <code>sql.db</code> , то профиль не соберется.	<b>lowerBound=2020-01-10;hid=1024;task=findDuplicates</b>

В результате выполнения задачи на почту придет один архив со всей необходимой диагностикой.

# ПРОВЕРКА БД НА НЕПРОТИВОРЕЧИВОСТЬ

## 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.

## Конфигурация задачи

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

Группа diagnostic  
Название runDBCheckers

Параметры задачи

Название параметра	Значение	Обязательность
procedureParams	1,2,10,5	

Сохранить Отмена

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

Группа diagnostic  
Название runDBCheckers

Параметры задачи

Название параметра	Значение	Обязательность
procedureParams	1,2,10,5	

Сохранить Отмена

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

# ПРОВЕРКА БД НА НЕПРОТИВОРЕЧИВОСТЬ (ТЕХПОДДЕРЖКА, СЧЕСКЕР)

## 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 fix_ma.zip (См. CDI-2348)
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	<b>PHONE_0</b>	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.



## 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
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**.

# ПРОФИЛИ ДИАГНОСТИКИ

Задачи диагностики предоставляют возможность сбора следующей информации:

Профиль	Параметры	Какую информацию собирает
<b>logs</b>	<b>lowerBound</b> <b>upperBound</b>	Журналы ЕК и Фактора в диапазоне дат [ <b>lowerBound</b> , <b>upperBound</b> ]. Оба параметра опциональные, задаются в формате <b>гггг-мм-дд</b> .
<b>logs-full</b>	<b>lowerBound</b> <b>upperBound</b>	Полные журналы ЕК и Фактора в диапазоне дат [ <b>lowerBound</b> , <b>upperBound</b> ]. Оба параметра опциональные, задаются в формате <b>гггг-мм-дд</b> .
<b>sql.party</b>	<b>hid</b>	Информация по конкретному контрагенту или по группе контрагентов, указанных через ";"
<b>sql.party.old</b>	<b>hid</b>	Устаревший профиль сбора диагностики по КА. Работает аналогично <b>sql.party</b> , но использует другой механизм.
<b>mapdb-state</b>	<b>hid</b>	Состояние mapdb-индекса по контрагенту. Обычно собирается в паре с <b>sql.party</b>
<b>graph-state</b>	<b>hid</b>	Состояние графа связей по контрагенту.
<b>cdi.jvm</b>		Информация о JVM Единого клиента: дампы потоков, параметры обработки HTTP-запросов, использование ресурсов ОС
<b>config</b>		Конфигурационные файлы
<b>db.connection</b>		Информация о соединении с БД
<b>factor.jvm</b>		Информация о JVM Фактора: дампы потоков, параметры обработки HTTP-запросов, использование ресурсов ОС.
<b>search</b>		Статистика по поисковому индексу
<b>sql.common</b>		Системные таблицы
<b>sql.counts</b>		Количество контрагентов
<b>sql.db</b>		Физические характеристики БД (нужны права <b>GRANT SELECT_CATALOG_ROLE TO &lt;user&gt;;</b> )
<b>sql.duplicate</b>		Общая статистика по дубликатам
<b>sql.duplicate.physical</b>		Статистика по дубликатам (физические лица)
<b>sql.metadata</b>		Метаданные БД
<b>sql.quality.physical</b>		Статистика по качеству данных (физические лица)
<b>sql.quality.phone</b>		Статистика по качеству данных (телефоны)
<b>sql.data.statistics.previous.month</b>		Статистика по качеству данных за прошлый месяц (общая)
<b>versions</b>		Версии ЕК, Фактора и справочников Фактора

<b>sql.checker</b>		Проверка целостности и согласованности бизнес-объектов системы
<b>sql.stat.task</b>	<b>task=task_key</b>	Информация обо всех запусках указанной задачи. <b>task_key</b> - ключ задачи как в админке, например, <b>findDuplicates</b>
<b>sql.buffer</b>		Сбор диагностики с таблицей <b>cdi_buffer_increments</b> . Выгружается вся таблица.
<b>user.env</b>		Информации о браузере
<b>user.event</b>	<b>lowerBound</b> <b>upperBound</b>	Информации о событиях в веб-интерфейсе в диапазоне дат [ <b>lowerBound</b> , <b>upperBound</b> ]. Оба параметра опциональные, задаются в формате <b>гггг-мм-дд</b> .
<b>metadata</b>		Метаинформация о модели данных для автоматической генерации документации

# ПРОВЕРКА СЛИТЫХ КОНТРАГЕНТОВ НА НЕПРОТИВОРЕЧИВОСТЬ

## *Общее описание*

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

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

## *Конфигурация задачи*

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

## *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 менеджеров данных), возможны ложно-положительные результаты.

# ПОИСК НЕКОНСИСТЕНТНОСТИ "КОНТРАГЕНТ ЗАКРЫТ, СВЯЗЬ — НЕТ"

## *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.

## *Конфигурация задачи*

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

# НАЙТИ И ИСПРАВИТЬ ПРОБЛЕМЫ В ЗОЛОТЫХ КАРТОЧКАХ

## *Общее описание*

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

## *Конфигурация задачи*

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

# СБОР ДИАГНОСТИКИ В РЕЖИМЕ ГОРЯЧЕГО РЕЗЕРВА

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

## Значения параметра `runInCluster` :

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

## Значения параметра `runInCluster` в задачах диагностики по умолчанию:

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

# СБОР МЕТРИК

## 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 деперсонализованная (не сохраняем никакие данные о клиентах)

## Конфигурация задачи

Группа diagnostic  
Название diagnosticMetrics

Параметры задачи

Название параметра	Значение	Обязательность
profiles	user.event,user.env	
params	lowerBound=,upperBound=	

Сохранить Отмена

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>profiles</b>	Профиль диагностики	<b>user.event, user.env.</b>
<b>params</b>	Диапазон дат, задается в формате гггг-мм-дд. По умолчанию даты не заданы.	<b>lowerBound=; upperBound=</b>
<b>attachLogs</b>	Нужно ли собирать вместе с метриками логи	<b>false</b>
<b>sendEmail</b>	<b>true / false</b> Признак необходимости отправить письмо	<b>false</b>
<b>sendTo</b>	Email, на который требуется отправляться архив диагностики. Список получателей письма через запятую	



---

**archiveP  
assword** Опциональный параметр для защиты паролем архивов диагностики, отправленных  
через почтовый сервер. Удаляется из параметров задачи после её выполнения

---

## *Пример*

---

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

---

# СБОР ДИАГНОСТИКИ ПО ЗАДАЧЕ

## 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.

## Конфигурация задачи

### Создать/Редактировать задачу

Группа diagnostic  
Название diagnosticTaskStat

Параметры задачи

Название параметра	Значение
profiles	sql.stat.task
params	task=findDuplicat

Настраиваемые параметры:

Параметр	Значение	Описание	Значение по умолчанию
<b>params</b>	task=<название задачи>	Название задачи, по которой нужна диагностика. Берется из общего списка задач, колонка «Название». Можно указать только одну задачу	<b>task=findDuplicat es</b>  ?  Соберет диагностику по задаче <b>findDuplicat es</b>
<b>sendEmail</b>	true / false	true / false	<b>false</b>
<b>sendTo</b>		Признак необходимости отправить письмо	
		Email, на который требуется отправляться архив диагностики. Список получателей письма через запятую	

---

**archiveP  
assword**

Опциональный параметр для защиты паролем архивов диагностики, отправленных через почтовый сервер. Удаляется из параметров задачи после её выполнения

---

# СБОР ЛОГОВ ЗА СЕГОДНЯ

⚠ Задача недоступна для ручного редактирования и запуска

## Общее описание

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

## 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
<b>sendEmail</b>	Признак необходимости отправки диагностики по почте	<b>true</b>
<b>runInCluster</b>	Для сбора в режиме горячего резерва	<b>true</b>
profiles	Профиль диагностики	logs, sql.common
<b>sendTo</b>	Если нужно временно перекрыть настройку «кому писать письмо» из конфигурации	
outputFileDirectory	Определяет каталог, в который будет сохранен файл с диагностикой.	

## Результат

В результате выполнения задачи будет указан путь до архива с собранной диагностикой логов.

Если задача выполнялась с опцией отправки диагностики по почте, то данный архив также будет продублирован на указанные для получения диагностики адреса электронной почты.

# СБОР МЕТРИК ЗА ПРЕДЫДУЩИЙ МЕСЯЦ

## 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 деперсонализованная (не сохраняем никакие данные о клиентах)

## Конфигурация задачи

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

Сохранить Отмена

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>profiles</b>	Профиль диагностики	<b>user.event.previous.month, user.env</b>
<b>attachLogs</b>	Нужно ли собирать вместе с метриками логи	<b>false</b>
<b>sendEmail</b>	<b>true / false</b> Признак необходимости отправить письмо	<b>false</b>
<b>sendTo</b>	Email, на который требуется отправляться архив диагностики. Список получателей письма через запятую	
<b>archivePassword</b>	Оptionальный параметр для защиты паролем архивов диагностики, отправленных через почтовый сервер. Удаляется из параметров задачи после её выполнения	

# ИМПОРТ ДАТАСЕТА С КОНТРАГЕНТАМИ (XLS)

## importDataSetTask

⚠ Страница под рестриктом для hfl-conf-worker, заказчикам нельзя такое видеть.

Задача только для тестирования.

## Общее описание

Загружает в БД данные из xls-файла, затирая те, что были (не все, а только те таблицы, что будут загружены).

## Особенности

Показывается в разделе диагностики, если в **standalone.conf/standalone.conf.bat** указать секретный ключ:

### Windows: standalone.conf.bat

```
set "JAVA_OPTS=%JAVA_OPTS% -Dcdi.task.showImportDataSetTask=true"
```

### Linux: standalone.conf

```
JAVA_OPTS="$JAVA_OPTS -Dcdi.task.showTestTasks=true"
```

⚠ По завершении задачи не забыть перестроить индексы.

⚠ Возможна проблема с **sequences**. Если у вас в какой-то момент полезут ошибки вроде "**Some records were updated concurrently**", то скорее всего у вас конфликт ID-шников.

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>dataSetFile</b>	Путь к .xls файлу с данными.	\path\to\xls\file.xls
<b>schemaName</b>	Имя схемы, в которую нужно положить данные. Важно для PostgreSQL	
<b>append</b>	Не удалять существующие данные, а добавить к ним (возможны конфликты по ID)	<b>false</b>
	<ul style="list-style-type: none"><li>• <b>false</b> — транкейт базы перед заполнением, операция <b>CLEAN_INSERT</b></li><li>• <b>true</b> — заполнение базы без предварительного транкейта, операция <b>INSERT</b></li></ul>	



# ЗАПУСК ПРОФИЛИРОВКИ

**runProfiling**

## Общее описание

Задача предназначена для запуска профилировки — сбора диагностической информации о работе приложения.

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
reportPrefix	Префикс для файлов диагностики	<b>cdi-profiling-manual</b>
lengthInMinutes	Длительность сбора профилировки	<b>15</b>

## Дополнительная информация

Задача запускает сбор профилировки и складывает результат в файлы вида **<reportPrefix>-<date>.jfr** в директорию с журналами «Единого клиента».

Файл профилировки можно скачать по прямой ссылке:

**<host>:<port>/cdi/ui/api/profiling/<постфикс>**

Формат постфикса - "auto-2022-05-01T16-17", где auto\manual — тип профилировки, 2022-05-01T16-17 - дата начала

Весь постфикс можно взять из результата выполнения задачи — "profiling report saved to ... standalone/log/cdi-profiling-auto-2022-05-01T16-17.jfr"

⚠ Требуется права администратора!

Для ручной очистки файлов профилировки можно выполнить задачу **profilingCleanup**, которая удалит все файлы профилировок старше n дней, указанных в конфигурации в параметре **task.profiling.lifetime.days**.

Если файлы профилировки имеют отличный от "cdi-profiling-manual" префикс, то его надо задать в задаче параметром reportPrefix.

Повторный запуск задачи пропускается, в случае если задача по первому запуску еще не отработала. В подобной ситуации логе можно увидеть:

```
[TaskManagerExecutor-1] INFO TaskWorker - Cannot start new recording: current recording is running.  
[TaskManagerExecutor-1] INFO TaskLauncherTemplate - Task 'runProfiling' finished with status SKIPPED in 84 ms
```





# ТРИГГЕРЫ

- Триггер
  - Пример
- Список триггеров
- Работа с триггерами
  - Запуск триггеров
  - Создание триггеров
  - Редактирование триггеров
  - Удаление триггеров
  - Настройка времени выполнения задачи в триггере

Действия	Название	Дата следующего запуска	Стоп выражение
Редактировать   Удалить   Выполнить	afterDatabaseBackupTrigger		
Редактировать   Удалить   Выполнить   Выключить	deleteOutdatedConflictsTrigger	24.05.2018 21:00:00	0 0 21 ? * SAT
Редактировать   Удалить   Выполнить   Включить	detectPopularAttributes		
Редактировать   Удалить   Выполнить	rebuildAllIndexesTrigger		
Редактировать   Удалить   Выполнить   Выключить	refreshDataStatisticTrigger	23.05.2018 2:00:00	0 0 2 * * ?
Редактировать   Удалить   Выполнить   Выключить	searchEngineOptimizeTrigger	24.05.2018 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.

## 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.

## 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, сперва, 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 (см. [Настройка времени выполнения задачи в триггере](#))
- *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...


## 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
- *Сtop-выражение* — стоп-выражение, по которому настроен автоматический запуск триггера

## 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.

## 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	Срок выражение
Редактировать   Удалить   Выполнить	afterDatabaseBackupTrigger		
Редактировать   Удалить   Выполнить   Выключить	deleteOutdatedConflictsTrigger	24.05.2018 21:00:00	0 0 21 ? * SAT
Редактировать   Удалить   Выполнить   Включить	detectPopularAttributes		
Редактировать   Удалить   Выполнить	rebuildAllIndexesTrigger		
Редактировать   Удалить   Выполнить   Выключить	refreshDataStatisticTrigger	23.05.2018 2:00:00	0 0 2 * * ?
Редактировать   Удалить   Выполнить   Выключить	searchEngineOptimizeTrigger	24.05.2018 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	Срок выражение
Редактировать   Удалить   Выполнить	afterDatabaseBackupTrigger		
Редактировать   Удалить   Выполнить   Выключить	deleteOutdatedConflictsTrigger	24.05.2018 21:00:00	0 0 21 ? * SAT
Редактировать   Удалить   Выполнить   Включить	detectPopularAttributes		
Редактировать   Удалить   Выполнить	rebuildAllIndexesTrigger		
Редактировать   Удалить   Выполнить   Выключить	refreshDataStatisticTrigger	23.05.2018 2:00:00	0 0 2 * * ?
Редактировать   Удалить   Выполнить   Выключить	searchEngineOptimizeTrigger	24.05.2018 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	Срок выражение
Редактировать   Удалить   Выполнить	afterDatabaseBackupTrigger		
Редактировать   Удалить   Выполнить   Выключить	deleteOutdatedConflictsTrigger	24.05.2018 21:00:00	0 0 21 ? * SAT
Редактировать   Удалить   Выполнить   Включить	detectPopularAttributes		
Редактировать   Удалить   Выполнить	rebuildAllIndexesTrigger		
Редактировать   Удалить   Выполнить   Выключить	refreshDataStatisticTrigger	23.05.2018 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.

## Создание триггеров

**i** 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	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	23.05.2018 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
- *Сноп выражение* — сноп-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

### Сноп триггер

Название:

Сноп 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

Сохранить Отмена

После задания параметров, нажимаем кнопку *Сохранить*.

### Сноп триггер

Название:

Сноп выражение:

Продолжать при ошибках:

Продолжать при конфликтах:

Выберите список задач к выполнению:

- rebuildBlackListCheckHashes
- Дедупликация и слияние
  - findDuplicates
  - findDuplicatesAndMerge
  - findDuplicatesAndMergeAndRebuildGroups
  - importExternalDuplicateWithMerge
  - incrementDuplicates
  - incrementDuplicatesAndMerge
- Диагностика
  - diagnosticLogs
  - diagnosticMessages

Сохранить Отмена

Новый триггер появится в списке.

Список триггеров

Действия	Название	Дата следующего запуска	Слов 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

## 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.

## Удаление триггеров

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

## Настройка времени выполнения задачи в триггере

Создать триггер и настроить слов-выражение, по которому будет выполняться автоматический запуск триггера.

Добавить в *Список задач к выполнению* задачу, для которой нужно настроить время выполнения. Задача должна иметь признак *Возобновляемая*.

Перейти на страницу *Задачи* и кнопку *Редактировать* около нужной задачи.

Действия	Название	Описание
Детали   Редактировать   Выполнить	addressActivation	Актуализация адресов
Детали   Редактировать   Выполнить	archivePartyClose	Закрытие устаревших картонки
Детали   Редактировать   Выполнить	archivePartyCloseRepeat	Отбор устаревших картонки для закрытия

В пустой строке в поле *Название параметра* ввести *executionTimeout*, в поле *Значение* ввести максимальное время выполнения задачи в минутах (при превышении задача останавливается).

Создать/Редактировать задачу

Группа: test@sova  
 Название: archivePartyClose

Параметры задачи

Название параметра	Значение	Обязательность
executionTimeout	1	

Сохранить Отмена

Нажать кнопку *Сохранить*.

# СПИСОК ТРИГГЕРОВ

По умолчанию в системе подключены триггеры:

Название	Список задач	Описание	Расписание работы
<b>afterDatabaseBackupTrigger</b>	<ol style="list-style-type: none"><li>1. Полное перестроение графа связей (раздел <i>Системные</i>).</li><li>2. Полное перестроение поискового индекса (раздел <i>Системные</i>).</li><li>3. Перестроение хэшей полного поиска дубликатов (раздел <i>Дедупликация и слияние</i>).</li><li>4. Перестроение хэшей полной проверки ЧС (раздел <i>Проверка по черным спискам</i>).</li><li>5. Актуализация групп дубликатов (раздел <i>Системные</i>).</li><li>6. Полное перестроение индекса конфликтов обновления (раздел <i>Системные</i>).</li><li>7. Перестроение хэшей полной проверки по ЕГРЮЛ (раздел <i>Дедупликация и слияние</i>).</li></ol>	Обновление информации в поисковых индексах после поднятия БД из бэкапа. Запускается вручную.	
<b>deleteOutdatedConflictsTrigger</b>	Удаление неактуальных конфликтов	Удаление обработанных конфликтов старше, чем 30 дней назад	Каждую субботу в 21:00
<b>detectPopularAttributesTrigger</b>	<ol style="list-style-type: none"><li>1. Найти массовые атрибуты (раздел <i>Массовые атрибуты</i>).</li><li>2. Проставить популярность по найденным значениям (ФЛ) (раздел <i>Массовые атрибуты</i>).</li><li>3. Проставить популярность по найденным значениям (другие типы контрагентов).</li></ol>	Выявление и проставление популярности массовым атрибутам.	Каждое первое воскресенье месяца в 12:00
<b>diagnosticLogsForTodayTrigger</b>	Отправить логи за сегодня на указанную в настройках почту.		Каждый день в 23:55 и в 7:55
<b>diagnosticMetricsForPreviousMonthTrigger</b>	Отправить веб-метрики и общую статистику по качеству данных за предыдущий месяц на указанную в настройках почту.		Каждое первое число месяца в 02:40
<b>rebuildAllIndexesTrigger</b>	<ol style="list-style-type: none"><li>1. Полное перестроение графа связей (раздел <i>Системные</i>).</li><li>2. Полное перестроение поискового индекса (раздел <i>Системные</i>).</li></ol>	Перестроение поисковых индексов. Запускается вручную.	

---

3. Актуализация групп дубликатов  
(раздел *Системные*).

---

<b>searchEngineOptimizeTrigger</b>	Оптимизация поисковых индексов	Оптимизирует поисковый индекс. Запускается автоматически раз в неделю.	Каждую субботу в 22:00
<b>refreshDataStatisticalTrigger</b>	Сбор метрик по задаче <b>refreshDataStatistical</b>		Каждый день в 02:00

---



# 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).

## 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.

# 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.

# ПРОВЕРКА ПО ЧЕРНЫМ СПИСКАМ

## *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. Задача перестраивает онлайн хэши справочников ЧС, обновление осуществляется для всех справочников — даже для тех, которые не были изменены
- **grabBlacklistActualDate** — обновление дат актуальности ЧС
- **listCommonBufferLoad** — загрузка списков из буферных таблиц (с релиза 21.21, подключается отдельно)
- **listFilesImport** — загрузка данных из файла (с релиза 21.21, подключается отдельно)

# ЗАГРУЗКА ДАННЫХ ИЗ ФАЙЛА В БУФЕРНЫЕ ТАБЛИЦЫ

 **listFilesImport**

 Возобновляемая

 Конфликтующая

## Общее описание

Загружает данные из файла ЧС в буферные таблицы: один файл — один инкремент. Директория с файлом указывается в параметре **sourceFileName**.

Для загрузки задачей пока доступен только справочник террористов в новом формате — **terrorist\*\*\*.xml**.

Задача составная: вторым этапом запускается загрузка из буферных таблиц.

## Алгоритм

Задача являет собой циклическое исполнение следующей последовательности:

1. Загрузка данных из файла в буферные таблицы — **listTerroristFilesImportTaskPerform**.
2. Запуск циклической загрузки инкремента списков **cyclicListBufferLoadTask**.


## Конфигурация задачи

У задачи только один обязательный параметр: **sourceFileName** — путь к директории файла со справочником.

# ИМПОРТ СПИСКОВ ИЗ ОБЩИХ БУФЕРНЫХ ТАБЛИЦ

 **listCommonBufferLoad**

 Возобновляемая, циклическая

 Конфликтующая

## *Общее описание*

Задача загружает поочередно инкременты по таблице **list\_buffer\_load**.

## *Алгоритм*

Задача являет собой циклическое исполнение следующей последовательности:

1. Обработка хранимой процедуры **get\_buffer\_list\_next\_load**, которая выбирает инкремент для загрузки.
2. Запуск циклической загрузки инкремента списков **cyclicListBufferLoadTask**:
  - для каждого инкремента (одна запись в таблице **list\_buffer\_load**) вызывает задачи загрузки данных ФЛ и ЮЛ
    - **listBufferPhysicalUploadTaskPerformer**
    - **listBufferLegalUploadTaskPerformer**
  - помечает загруженный инкремент как SUCCESS/ERROR (на ошибках не останавливаемся, идем дальше)
    - **mark\_buffer\_list\_load\_success / mark\_buffer\_list\_load\_error**

## *Конфигурация задачи*

У задачи нет обязательных параметров.

# ОБНОВЛЕНИЕ ДАТ АКТУАЛЬНОСТИ ЧС

 grabBlacklistActualDate

 Непрерываемая

## *Описание задачи*

Задача запрашивает у Фактора список и даты актуальности всех справочников ЧС, по которым на текущий момент ведется проверка и сохраняет их в таблицу [BLACKLIST\\_DATE](#).

## *Алгоритм задачи*

ЕК через веб-сервис Фактора **getBuildInfo** получает информацию по справочникам ЧС (вида "**BlackList** / ") и их датам актуальности, затем сохраняет данные в таблицу [BLACKLIST\\_DATE](#).

Справочники, по которым не пришла информация от Фактора, из таблицы удаляются.

# ПЕРЕСЧЕТ УРОВНЯ РИСКА

**riskLevelActualize**

Возобновляемая

⚠ Задача актуальна только для модуля ЧС 2.0

## Общее описание

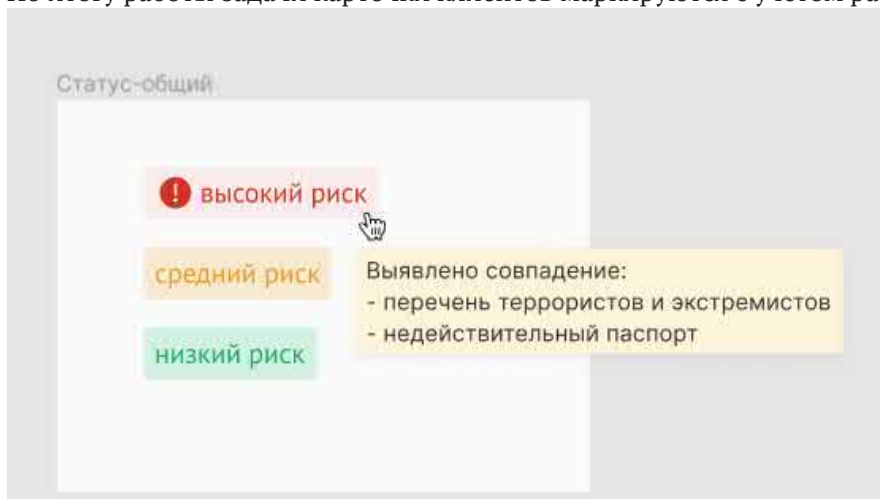
Задача рассчитывает уровень риска (`aml_risk_level`) контрагентов с учетом правил, настроенных конфигурациях, в параметрах сервисов.


При расчете уровня риска учитывают факты вхождения клиента в различные перечни. Каждая организация вправе по своему усмотрению определять факторы, которые влияют на уровень риска и в какой степени.



По завершению работы обработанным контрагентам могут быть присвоены следующие уровни риска в зависимости от заданных параметров:

- Высокий уровень риска;
- Средний уровень риска;
- Низкий уровень риска;
- Неизвестный уровень риска.

По итогу работы задачи карточки клиентов маркируются с учетом рассчитанного уровня риска:




**Абдуллов Равиль Хусяинович** ! высокий риск

атрибутика   история   исходные   связи   проверки    нет похожих 

---

| ФИО • SPH  
 Абдуллов • Равиль • Хусяинович

| Дата рождения • SPH  
 03.06.1959

| ИНН • SPH  
 131601523633

## Запуск задачи

1. Выяснить идентификаторы и типы контрагентов (**hid\_party**), для которых необходимо рассчитать уровень риска (по умолчанию **PARTY\_TYPE = PHYSICAL**).
2. Добавить список данных идентификаторов в таблицу **task\_recalculate\_risk\_hids** вручную.
3. Запустить задачу.

Пример ручного наполнения таблицы:

```
insert into task_recalculate_risk_hids(hid_party, party_type) values (1024, 'PHYSICAL');
commit;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется разрешение конфликта	<b>PHYSICAL</b>
<b>poolSize</b>	Максимальное количество потоков	максимальное
<b>pageSize</b>	Размер пачки	1000
<b>sendEmail</b>	Признак необходимости отправки диагностики по почте	<b>true</b>
<b>sendTo</b>	Если нужно временно перекрыть настройку «кому писать письмо» из конфигурации	
<b>terminateOnErrors</b>	Прекратить выполнение при возникновении ошибки?	<b>false</b>



## *Логирование*

Пример:

---

**INFO TaskPerformerStatsInterceptor - FINISHED task 'riskLevelActualizeTaskPerformer'; processed 3 entities (MISSED: 3); 0 with errors; took 1 seconds or 24 ms;**

---

В финальном логировании выводится число всех обработанных КА, а также количество КА для каждого типа уровня риска.

# НЕОБРАТИМОЕ УДАЛЕНИЕ ДАННЫХ

## 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!

## 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. Удаление дубликатов выполняется следующим образом: задача ищет актуальные дубликаты, содержащие HID-ы удаляемых контрагентов и удаляет все версии этих дубликатов.
5. Задача удаляет все связи, которые на одном из концов содержат идентификатор удаляемой сущности. Таким образом после работы задачи могут остаться одинокие ранее связанные с КА сущности (MSISDN, договоры, машины и т. п.).

## 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

## 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:

- НID-ы открытых клиентов. При обнаружении НID'ы записываются в лог, удаляются из **delete\_party\_hids**.
- НID-ы «золотых» клиентов. При обнаружении НID'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.

# ЗАКРЫТИЕ КОНТРАГЕНТОВ

PartyClose

## Запуск задачи

1. Выяснить идентификаторы и типы контрагентов (**hid\_party**), которые необходимо закрыть.
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 (1024, 'PHYSICAL');  
commit;
```

## 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:

- ИД-ы закрытых, несуществующих, разлитых контрагентов — удаляются из **close\_party\_hids**, в лог не записываются.
- ИД-ы «золотых» слитых 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.

# 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.

## 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

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

## Работа задачи

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

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

## Параметры

Параметры импорта из буферной области

Название параметра	Значение
importedToday	10.02.2016:3
oldMaxId	20915

Сохранить

### *Подключение порционности*

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

# 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.

# ПОЛНАЯ ПРОВЕРКА ЮРЛИЦ 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.



# ПРОВЕРКА ПО ЕГРЮЛ КОНТРАГЕНТОВ, ТРЕБУЮЩИХ АКТУАЛИЗАЦИИ

# АКТУАЛИЗАЦИЯ ЮЛ – КАРТОЧЕК-КОПИЙ ЕГРЮЛ

**egrMirrorPartiesActualize**

✔ Возобновляемая, конфликтная

## Описание задачи

Во время проверки по ЕГРЮЛ для каждой карточки с типом EGR (к которой был найден дубликат среди юрлиц) создается зеркало — новая карточка с типом **LEGAL**, являющаяся копией ЕГРЮЛ: от обычных отличается системой источником **EGRM** и полем **EGR\_MIRROR = 1**.

Со временем в карточку из ЕГРЮЛ могут быть добавлены новые поля или атрибуты. Данные поля будут автоматически попадать как в новые карточки-копии, так и в уже существующие. В случае если произошли изменения в модели ЮЛ и появились новые поля, которые могут быть пополнены из существующих карточек EGR без обновления справочника ЕГРЮЛ, необходимо выполнить эту задачу.

## 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 выбрать для обновления все актуальные карточки ЕГРЮЛ, для которых есть карточка-копия в ЮЛ запросом:

```
insert into task_party_hids (hid_party, party_type)
select hid_party, 'EGR' from egr_party ep
join relation r on (
  r.id_relation_type = 90
  and r.first_related_hid = ep.hid_party
  and r.enddate is null)
where
  ep.version = 0 and ep.enddate is null;
commit;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>poolSize</b>	Максимальное количество потоков	<b>8</b>
<b>pageSize</b>	Размер пачки	<b>1000</b>
<b>executionTimeout</b>	Максимальное время выполнения задачи в минутах. При превышении задача останавливается.	

# МАССОВЫЕ АТТРИБУТЫ

## *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).

# ПОИСК МАССОВЫХ РЕКВИЗИТОВ И АТТРИБУТОВ

findPopularValues

 Непрерываемая

## *Описание задачи*

ЕК проходит весь поисковый индекс и вычисляет часто встречающиеся реквизиты и атрибуты. Все найденные хиды контрагентов записываются в таблицу **task\_doubtful\_reclean\_hids** с указанием типа реквизита/атрибута на переобработку.

# ПРОСТАВЛЕНИЕ ПОПУЛЯРНОСТИ ПО НАЙДЕННЫМ ЗНАЧЕНИЯМ (ФЛ И ЮЛ)

markPopularPhysicalAttributes

markPopularLegalAttributes

 Возобновляемая

## Описание задачи

ЕК забирает контрагентов из таблицы **task\_doubtful\_reclean\_hids** и отправляет в Фактор для проставления кода популярности.

# МИГРАЦИЯ

- Автопрогон скриптов выдачи грантов пользователям
- Актуализация реестров
- Задача переобработки места рождения
- Заполнение даты последнего изменения карточек
- Инициализация прошлых версий атрибутов
- Миграция нового поля
- Переименование исходной системы

# 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.



# АКТУАЛИЗАЦИЯ РЕЕСТРОВ

## Добавление нового реестра

 Количество записей в одном реестре не должно превышать 100 000.

Загрузка реестра большего размера может привести, при работе с фильтром по реестрам на списке групп дубликатов, к ошибке : **Ваш запрос находит слишком много записей. Уточните условие запроса.**

Первым делом добавить название реестра и тип контрагентов в таблицу **EXTERNAL\_LIST**:

```
-- id_list (идентификатор реестра) генерится автоматом из сиквенса (или триггером),  
заполнять его не нужно
```

```
insert into external_list (list_name, party_type)  
values ('Перечень ВИП-клиентов', 'PHYSICAL');  
commit;
```

Получить сгенеренный идентификатор реестра:

```
-- взять id_list из верхней строчки, попутно проверив что название соответствует
```

```
select * from external_list order by id_list desc;
```

Затем загрузить идентификаторы клиентов (**hid\_party**) в таблицу **EXTERNAL\_LIST\_HIDS**, не забыв указать правильный **LIST\_ID** нужного реестра:

```
insert into external_list_hids (id_list, hid_party)  
values (10, 1024);  
commit;
```

Теперь можно запустить задачу **actualizeExternalListIds**.

В режиме горячего резерва запускать только на одной ноде.

## Удаление реестра

 **Обратите внимание**

Удалить часть записей реестра невозможно, только реестр целиком. Поэтому если необходимо обновить какой-либо реестр необходимо удалить текущий реестр и прогрузить список как новый реестр с таким же наименованием, но новым ИД (**id\_list**).

1. Из таблицы **EXTERNAL\_LIST** удалить запись с устаревшим реестром. Записи, соответствующие данному реестру, будут автоматически удалены из таблицы **EXTERNAL\_LIST\_HIDS**.
2. Запустить задачу **actualizeExternalListIds**.

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента	

При указании **partyType**, задача будет актуализировать данные только по одному типу КА, а не по всем.

## Логирование

Задача стандартно логирует обработку каждых N записей (по умолчанию 1 000):

```
INFO ActualizeExternalListIdsTask - Processed 1 000 essences
```

В итоговой статистике выводится количество загруженных записей:

```
INFO TaskPerformerStatsInterceptor - FINISHED task 'actualizeExternalListIds [actualizeExternalListIdsTaskPerformer]'; processed 23 000 entities; 0 with errors; took 2.37 min;
```

## Отображение данных

❗ С релиза 21.9 не нужно запускать задачу `duplicateGroupIndexRebuild`! Индекс групп дубликатов обновится автоматически.

Чтобы дубликаты отобразились после загрузки нового реестра, необходимо запустить задачу `duplicateGroupIndexRebuild` на обеих нодах.

# ЗАДАЧА ПЕРЕОБРАБОТКИ МЕСТА РОЖДЕНИЯ

**birthPlacelnitTask**

✔ Возобновляемая

❗ Конфликтующая

## Описание задачи

Задача стандартизирует реквизит «Место рождения».

## Запуск задачи

Перед запуском задачи необходимо:

- Выполнить задачу **cleanTaskParameterTables** для очистки таблицы **task\_party\_hids**.
- Выполнить задачу **fillTaskPartyHidsActual**, указав параметр **partyType** равным PHYSICAL.

После этого запустить задачу **birthPlacelnitTask**.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента	PHYSICAL

# ЗАПОЛНЕНИЕ ДАТЫ ПОСЛЕДНЕГО ИЗМЕНЕНИЯ КАРТОЧЕК

**fillLastChangeTimestamp**

✔ Возобновляемая, не конфликтующая

## Описание задачи

Задача первичного заполнения даты последнего изменения карточки `LAST_CHANGE_TIMESTAMP` (формат `unix timestamp` в мс) в таблице `PARTY_CHANGE_META`.

## Запуск задачи

Запустить задачу **fillLastChangeTimestamp** без предварительных шагов. Задача сама создаст временную таблицу, наполнит ее хидами контрагентов, выполнит процедуру заполнения временных меток, удалит таблицу и повторит цикл для каждого типа КА.

В режиме горячего резерва запускать только на одной ноде.

⚠ После задачи необходимо перестроить поисковый индекс на каждой ноде.

## Конфигурация задачи

У задачи нет настраиваемых параметров.

# ИНИЦИАЛИЗАЦИЯ ПРОШЛЫХ ВЕРСИЙ АТТРИБУТОВ

## Запуск задачи

Задача работает по списку клиентов, что делает её 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**.

В режиме горячего резерва запускать только на одной ноде.

## Конфигурация задачи

Настраиваемые параметры:

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

## После выполнения задачи

### Перестроение поискового индекса физических лиц

1. Отредактировать задачу **allEnginesRebuildByName** в АРМ Администратора, установить параметр **enginesToRebuild = PHYSICAL**
2. Запустить задачу **allEnginesRebuildByName**

В режиме горячего резерва выполнить на каждой ноде.

# МИГРАЦИЯ НОВОГО ПОЛЯ

fooMigrateRequisiteTask

## 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

## 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*)

## 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-soap	

Сохранить  Отмена

# ПЕРЕИМЕНОВАНИЕ ИСХОДНОЙ СИСТЕМЫ

renameSourceSystem

✓ Возобновляемая

! Конфликтующая

## Описание задачи

Задача обработает все таблицы системы «Единый клиент», подменяя в них старый код исходной системы на новый.

## Алгоритм работы

Задача отбирает все карточки (вкл закрытые), удовлетворяющие условиям:

- непустой STAGING\_HID
- MERGED\_STATUS in (0,1)
- MERGE\_AUTHOR = OLD\_SYSTEM:%

Сохраняет их хиды в tmp\$rename\_system\_hids. При желании можно писать в др таблицу tmp\$some\_table(hid\_party), указав параметры:

- fillTaskPartyHids = false
- hidsTableName = tmp\$some\_table

После обработки исходных задача выходит на золотых и подменяет систему также и в них.

Список хидов исходных изменённых исходных карточек записывается в rename\_system\_log (hid\_party, party\_type, old\_system, new\_system, raw\_id, created), диагностика собирает ее

## Запуск задачи

Запустить задачу renameSourceSystem.

В режиме горячего резерва запускать только на одной ноде.

## После запуска задачи

Перестроить индекса групп дубликатов задачей duplicateGroupIndexRebuild



## Конфигурация задачи

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента	PHYSICAL
<b>oldSystemCode</b>	Старый код исходной системы.  Система должна быть закрыта (непустой SYSTEM.CANCELED в БД)	
<b>newSystemCode</b>	Новый код исходной системы.  Система НЕ должна быть закрыта	
<b>skipSearchNotify</b>	Не оповещать поисковый индекс об изменениях	false
<b>updateRelationAttributes</b>	Обновлять атрибуты связей	false

## Логирование

Задача стандартно логирует обработку каждых N записей (по умолчанию 100 000):

```
INFO RenameSourceSystemTask - Processed 100 000 essences (migrate: 100 000)
```

В итоговой статистике выводится количество загруженных записей:

```
INFO TaskPerformerStatsInterceptor - FINISHED task 'renameSourceSystem [renameSourceSystemTaskPerformer]'; processed 2 entities (migrate: 2); 0 with errors; took 1 seconds or 829 ms; speed 2.0 records/sec;
```

Если был обнаружен неуникальный id (`oldSystemCode:id = newSystemCode:id`), то карточка не мигрируется и в лог идет запись:

```
WARN: Skip migration because same rawIds found for new system AL: [1]
```

# МИГРАЦИЯ СТРУКТУРЫ ХРАНЕНИЯ ИСТОРИЧЕСКИХ ДАННЫХ

- Форсирование миграции структуры исторических данных (Фаза 1)
- Форсирование миграции структуры исторических данных (Фаза 3)

# ФОРСИРОВАНИЕ МИГРАЦИИ СТРУКТУРЫ ИСТОРИЧЕСКИХ ДАННЫХ (ФАЗА 1)

## 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
<b>tableNameToMigrate</b>	<ul style="list-style-type: none"><li>• Если задан — будет мигрировать только указанную таблицу</li><li>• Если не задан — все, что ещё не отмигрировано</li></ul> <p>:</p> <pre>select * from migration_phase;</pre>	
<b>logPeriod</b>	Параметр, определяющий частоту логирования прогресса. Будет логироваться каждая N пачка	100
<b>pageSize</b>	Размер пачки	1000
<b>poolSize</b>	Количество одновременных потоков	16

# ФОРСИРОВАНИЕ МИГРАЦИИ СТРУКТУРЫ ИСТОРИЧЕСКИХ ДАННЫХ (ФАЗА 3)

migratePhase3

✔ Возобновляемая

## 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.

## Запуск задачи

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

При необходимости настроить триггер запуска по ночам и указать параметр `executionTimeout`.

## 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**:

## Было до миграции

### STAGING\_HIST

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

## 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.

## 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
<b>tableNameToMigrate</b>	<ul style="list-style-type: none"><li>Если задан — будет мигрировать только указанную таблицу</li><li>Если не задан — все, что ещё не отмигрировано</li></ul> <p>:</p> <pre>select * from migration_phase;</pre>	
<b>logPeriod</b>	Параметр, определяющий частоту логирования прогресса. Будет логироваться каждая N пачка	100
<b>pageSize</b>	Размер пачки	1000
<b>poolSize</b>	Количество одновременных потоков	16

# СБОР СТАТИСТИКИ

## *Конфигурация задачи*

Для стандартного запуска задачу конфигурировать не нужно.

При специальном запросе поддержки, можно настраивать следующие параметры:

Параметр	Описание	Значение по умолчанию
<b>poolSize</b>	Максимальное количество потоков	1

## *Особенности задачи*

Задача не конфликтующая.

# НАСТРОЙКА СБОРА СТАТИСТИКИ

Со временем функционал сбора статистики был расширен, и были добавлены новые возможности.

1. Создание alias-ов в запросе.
2. Добавление итераторов по перечислимым полям.
3. Создание sql-alias-ов.

В Core добавлен bean для разбора запроса, который обращается к другим бинам:

```
<bean id="dataStatisticRequestParser" class="ru.hflabs.cdi.task.performer.statistics.parser.ParserChainDataStatisticRequestParser">
  <constructor-arg name="parsersChain">
    <list>
      <ref bean="aliasDataStatisticRequestParser"/> <!-- извлекает alias-ы из запроса -->
      <ref bean="iterDataStatisticRequestParser"/> <!-- извлекает перечислимые поля -->
      <ref bean="allSystemAliasDataStatisticRequestParser"/> <!-- ищет итераторы по
системам из Core -->
      <ref bean="customSqlQueryAliasDataStatisticRequestParser"/> <!-- ищет кастомные
sql-alias-ы -->
    </list>
  </constructor-arg>
</bean>
```

## allSystemAliasDataStatisticRequestParser

```
<bean id="allSystemAliasDataStatisticRequestParser" parent="
sqlQueryAliasDataStatisticRequestParserTemplate">
  <constructor-arg name="sqlAliases">
    <map>
      <entry key="%allSystemsCrossIter%" value="select ss1.name || '*' || ss2.name from
system ss1, system ss2 where ss1.name &lt;= ss2.name"/>
      <entry key="%allSystemsIter%" value="select name from system"/>
    </map>
  </constructor-arg>
</bean>
```

## Пример 1

Чтобы добавить constant alias, нужно добавить в bean constantDataMetricAliases.custom ключевое значение, например:

```
<util:map id="constantDataMetricAliases.custom">
  <entry key="mainSystems" value="(AL* or BT* or GM*)"/>
</util:map>
```

## Пример 2

Для перебора всех возможных значений определенного поля, для него нужно указать %iter%, как в примере ниже для перечисления всех типов документов ФЛ.

```
? Passport.qualifierType = (%iter%)
```

## Пример 3

Для добавления сложного запроса в метрики можно добавить кастомный sql-alias заказчику и использовать его в запросе.

Пример.

1. В sqlDataMetricAliases.custom Заказчика добавить:

```
<entry key="%notGuaranteedRules%" value="select distinct rule from duplicate where match_scope &lt;= 96"/>
```

\*P.S. '&lt;=' это '<='. Будут отобраны все правила из таблицы duplicate с коэффициентом <= 96.

2. data\_metric .

```
? ManualDuplicateGroup.partyType = MOCK and Duplicate.rule=%notGuaranteedRules%
```

## Пример 4

Итераторы по типам контрагентов и атрибутов

```
? ManualDuplicateGroup.partyType = %allPartyTypes%
```

Аналогично можно использовать итератор по типам атрибутов **%allAttributeTypes%**.



# DATA\_METRIC

## 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.  Всегда указывается только одно значение, несколько значений указывать нельзя.  Может принимать следующие значения: <ul style="list-style-type: none"><li>▪ Тип контрагента. Например, <b>PHYSICAL</b> или <b>LEGAL</b>;</li><li>▪ Идентификатор поиска по группам дубликатов. Значение <b>MANUAL_DUPLICATE_GROUP</b></li></ul>
QUERY	VARCHAR2 (1500)	No	null	4	Поисковый запрос

## Индексы

INDEX_NAME	UNIQUENESS	COLUMNS
PK_DATA_METRIC	UNIQUE	ID_METRIC

# DATA\_STATISTIC

## 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

## Индексы

INDEX_NAME	UNIQUENESS	COLUMNS
UK_DATA_STATISTIC	-	ID_METRIC, COUNT_TS

# ДОБАВЛЕНИЕ НОВЫХ МЕТРИК КАЧЕСТВА ДАННЫХ

- Инструкция по добавлению новых метрик
- Особенности формирования поисковых запросов для сбора метрик данных
  - Сбор метрик по всем значениям справочника
  - Особенности использования:
    - Пример 1. Количество ФЛ каждого пола
- SQL-метрики
  - Особенности использования:
- Примеры метрик
  - Для контрагентов
    - Пример 2. Добавление метрики для физического лица
    - Пример 3. Добавление метрики для юридического лица
  - Для групп дубликатов
    - Пример 4. Добавление метрики для группы дубликатов
  - SQL-метрики
    - Пример 5. Добавление sql-метрики, возвращающей одно значение
    - Пример 6. Добавление sql-метрики, возвращающей два значения

## Инструкция по добавлению новых метрик

Создание новых метрик осуществляется путем добавления записей в таблицу `DATA_METRIC`.

Для этого необходимо:

1. Подготовить **поисковый запрос**, который будет использовать для сбора метрик;
2. Добавить новую метрику в таблицу `DATA_METRIC` запросом:

```
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)
VALUES
(..., '...', '...', '...');
COMMIT;
```

**Важно!** При добавлении новых метрик необходимо использовать уникальное значение `ID_METRIC` в диапазоне от 1 до 1 000 000.

3. Если необходимо, чтобы добавленные метрики были доступны для новых контуров или при чистой установке системы, направьте, пожалуйста, нам в поддержку уведомление о добавленных метриках. Они будут поддержаны в коде «Единого клиента».

При следующем выполнении задачи сбора метрик, статистика будет собрана также и вновь добавленной метрике.

## Особенности формирования поисковых запросов для сбора метрик данных

### Сбор метрик по всем значениям справочника

Иногда необходимо собрать метрики в разрезе значений того или иного справочника. Справочники имеют свойство расширяться и изменяться со временем. Чтобы метрики оставались актуальными и их не приходилось дорабатывать при каждом изменении справочника, в поисковом запросе можно указывать специальное значение (**%iter%**).

При сборе статистики по запросу, содержащему (%iter%):

- Для каждого возможного значения справочника будет сформирован и выполнен отдельный подзапрос: равный запросу, указанному в query, но с подстановкой справочного значения вместо (%iter%).
- Результат выполнения подзапроса будет зафиксирован в таблице **DATA\_STATISTIC**. Значение справочника, для которого был сформирован подзапрос, будет сохранено в поле Param.

### Особенности использования:

- В каждом запросе можно указать только один итератор. При наличии более одного итератора сбор метрик по данному запросу выполняться не будет, в логе будет зафиксировано предупреждение.
- Если итератор задан для поля, которое не является справочным (например, фамилия), то сбор статистики для данного запроса не происходит, в логе фиксируется предупреждение.

### Пример 1. Количество ФЛ каждого пола

Вместо трех запросов:

```
? PhysicalParty.gender = MALE;  
? PhysicalParty.gender = FEMALE;  
? PhysicalParty.gender = UNKNOWN;
```

правильно указать один:

```
? PhysicalParty.gender = (%iter%); --Вместо отдельных значений справочника задаем значение %iter%
```

Пример запроса на добавление метрики с итератором по справочнику:

```
-- 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 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

## SQL-метрики

С релиза 21.18 появилась возможность считать метрики SQL запросами по базе. SQL-метрики создаются как и обычные метрики, путем добавления записи в таблицу **DATA\_METRIC**, в поле **QUERY** необходимо указать SQL запрос.

### Особенности использования:

- Запрос должен вернуть одно поле, тогда результат будет зафиксирован в поле **COUNT** в таблице **DATA\_STATISTIC**.
- Запрос должен вернуть два поля — первый результат будет записан в поле **PARAM**, второй в поле **COUNT**.
- Запрос должен вернуть 3 поля и больше, то сбор метрик по данному запросу выполняться не будет, в логе будет зафиксировано предупреждение — **WARN RefreshDataStatistics Task - Too much columns (N) for metric #1 with param 'null'. Skipped.**
- Запрос ничего не должен вернуть, в логе будет зафиксировано предупреждение — **WARN RefreshDataStatisticsTask - Sql statistic is empty for metric #3.**

**i** Результат запроса вида **select \* from table** будет зафиксирован в таблице **DATA\_STATISTIC**, если количество возвращаемых столбцов не больше 2-х.

Если количество возвращаемых столбцов будет больше 2-х, в логе будет стандартное предупреждение — **WARN RefreshDataStatisticsTask Too much columns (N) for metric #1 with param 'null'. Skipped.**

## Примеры метрик

### *Для контрагентов*

Название индекса, по которому выполняется поиск для контрагентов совпадает с его типом. Например, PHYSICAL.

#### *Пример 2. Добавление метрики для физического лица*

```
-- 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)');  
COMMIT;
```

#### *Пример 3. Добавление метрики для юридического лица*

```
-- 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');  
COMMIT;
```

### *Для групп дубликатов*

Для сбора статистики по группам дубликатов необходимо использовать тип индекса **MANUAL\_DUPLICATE\_GROUP**.

**Важно!** в запросе необходимо указать тип контрагентов для поиска. Например, **ManualDuplicateGroup.partyType = PHYSICAL**.

#### *Пример 4. Добавление метрики для группы дубликатов*

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

## *SQL-метрики*

### *Пример 5. Добавление sql-метрики, возвращающей одно значение*

```
-- Посчитать количество конфликтов обновления  
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)  
VALUES (4, 'Количество конфликтов слияния', 'PHYSICAL', 'select count (*) from conflict');  
COMMIT;
```

### *Пример 6. Добавление sql-метрики, возвращающей два значения*

```
-- Посчитать количество клиентов каждого пола  
INSERT INTO DATA_METRIC(ID_METRIC, DESCRIPTION, PARTY_TYPE, QUERY)  
VALUES (5, 'Количество клиентов', 'PHYSICAL', 'select gender, count (*) from  
physical_party group by gender');  
COMMIT;
```

# ИСПОЛЬЗОВАНИЕ ВСТРОЕННОЙ ВОЗМОЖНОСТЬ ЛЮСИН ПО ИТЕРАЦИИ ПО ВСЕМ ВОЗМОЖНЫМ ВАРИАНТАМ ПОЛЯ

 **CDI-6277** - Научиться использовать в метриках синтаксис \$allOf[Address.city] [ЗАКРЫТ](#)

Мы сейчас имеет уже несколько итераторов в метриках, но они покрывают не все задачи, т.к. либо считаются по базе, либо по енум полям.

Люсин имеет встроенную возможность итерации по всем возможным вариантам поля.

Синтаксис запроса: \$allOf[<наименование поля>]

## Например, итератор по типам телефона:

**? Phone.type = \$allOf[Phone.type]**

Каждое значение итератора пишем в param, а значение метрики — в статистику.

ID_METRIC	COUNT_TS	PARAM	COUNT
1	2011-10-01 10:00:00.0	home	2
1	2011-10-01 10:00:00.0	fax	5
1	2011-10-01 10:00:00.0	mobile	1



# ФОНОВЫЕ ЗАДАЧИ

Задачи, помеченные признаком *Фоновая*, выполняются за счет оставшихся свободных ресурсов приложения после перераспределения ресурсов между онлайн-поток и штатными задачами.

Если свободных ресурсов для фоновых задач нет, а фоновая задача еще не завершила свою работу, то в панели администратора продолжает отображаться прогресс-бар выполнения фоновой задачи.

## **Алгоритм мониторинга и перераспределения нагрузки между онлайн-поток и фоновыми задачами**

Для мониторинга онлайн-потока используется снятие статистики по 90-процентили (время обработки, в которое укладывается 90% успешных вызовов, в мс.). Для горячего резерва статистика снимается с той ноды, на которой запускается фоновая задача.

### **Основной алгоритм**

Для перераспределения потоков нагрузки между онлайн-поток и фоновыми задачами используется следующий алгоритм:

1. При старте приложения производится замер общего количества ядер на сервере приложений.
2. Запускается фоновая задача. Количество потоков для работы фоновой задачи при старте = 0.
3. Если в конфигурации фоновых задач не указано ограничение по числу ядер для использования фоновыми задачами, то устанавливается ограничение —  $2/3$  от общего количества ядер на сервере приложения, на котором будет запускаться фоновая задача.
4. Инструмент «тихой загрузки» мониторит нагрузку на онлайн-поток, получает данные по 90-процентили заданных в конфигурации методов с заданным минимальным количеством вызовов за интервал времени (по умолчанию 5 минут).
5. Если показания 90-процентили не выходят за заданные показатели, то инструмент «тихой загрузки» добавляет 4 потока для работы фоновой задачи. Далее переходим на шаг 6 мониторинга.
  - a. Если показания 90-процентили выходят за заданные показатели, то в панели администратора отображается прогресс-бар запущенной фоновой задачи, но количество потоков для работы по-прежнему =0. Далее повторяется шаг 4 мониторинга.
  - b. Если показания 90-процентили за последние 5 временных интервалов незначительны, то инструмент «тихой загрузки» наращивает потоки до верхнего лимита, заданного в п.3. Далее см.детальное описание данной ветки алгоритма ниже.
  - c. Если задача использовала все выделенные ресурсы, показания 90% находятся в норме, то продолжаем обрабатывать записи с использованием всех доступных ресурсов с регулярным мониторингом п.4. При отсутствии записей завершаем задачу.

- d. Если данные для работы фоновой задачи закончились или отсутствуют (например, все записи уже обработаны), то задача завершается.
6. Мониторинг нагрузки на онлайн продолжается каждые 5 минут. Если показания 90-процентили не выходят за заданные показатели, то инструмент «тихой загрузки» наращивает число потоков для работы фоновой задачи каждые 5 минут. Дельта для наращивания числа потоков задается в конфигурации задаче — **poolSize**. По умолчанию — это 4 потока.
- a. Если показания 90-процентили выходят за заданные показатели, то в панели администратора отображается прогресс-бар запущенной фоновой задачи, но количество потоков для работы сбрасывается на последнюю наращенную дельту (по умолчанию 4 потока). Далее повторяется шаг 4 мониторинга.
  - b. Если показания 90-процентили за последние 5 временных интервалов незначительны, то инструмент «тихой загрузки» наращивает потоки до верхнего лимита, заданного в п. 3. Далее см. детальное описание данной ветки алгоритма ниже.
  - c. Если задача использовала все выделенные ресурсы, показания 90% находятся в норме, то продолжаем обрабатывать записи с использованием всех доступных ресурсов с регулярным мониторингом п.4. При отсутствии записей завершаем задачу.
  - d. Если данные для работы фоновой задачи закончились или отсутствуют (например, все записи уже обработаны), то задача завершается.
7. Если в процессе работы фоновой задачи запускаются штатные задачи (отдельно или в составе триггера) и они выполняются более минуты (подстраховка от мелких быстрых задач), то количество потоков для работы фоновой задачи сбрасывается до 0, но она отображается в панели администратора.

## Алгоритм при отсутствующем или незначительном онлайн-потоке

При отсутствии онлайн-потока для большей производительности можно наращивать число потоков резко.

Если статистика по показаниям 90-процентили за все последние 5 интервалов незначительна, т. е. онлайн-поток отсутствует или нет превышения по количеству запросов ни у одного метода, то используется следующий алгоритм:

1. Число потоков, выделяемых для фоновой задачи, резко наращивается до максимального лимита (п.3 основного алгоритма).
2. Если показания 90-процентили находятся в норме, то продолжаем обрабатывать записи максимальным количеством потоков, пока записи не закончатся.
3. Если показания 90-процентили начинают превышать норму, то уменьшаем число потоков на дельту (по умолчанию 4 потока). Далее переходим к п.6 основного алгоритма.

## Режим Горячего резерва

В режиме Горячего резерва также идет мониторинг очереди между нодами.

// Пример сообщения из лога, о забитой очереди:

```
INFO JmsQueueSizeMonitor - 'cdi.nodeSync' queue size is 196495 (consumers=12)
```

---

Если размер очереди превышает настроенный предел (параметр **task.silent.nodeSync.queueSizeLimit**, по умолчанию 1000), то число потоков понижается на дельту.

# ЗАДАЧА МАССОВОЙ ОБРАБОТКИ КОНФЛИКТОВ

massConflictResolution

✓ Возобновляемая

! Для корректного выполнения задачи УЗ, от имени которой запускается задача, нужна роль **ROLE\_CONFLICT\_MANAGER**

## Описание задачи

Задачу следует использовать для автоматического разрешения/отклонения конфликтов.

## Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу **cleanTaskParameterTables** для очистки таблицы **task\_party\_hids**.
2. Выяснить идентификаторы **исходных** (merged\_status in (0, 1)) контрагентов (**hid\_party**), для которых необходимо обработать конфликты.
3. Добавить список данных идентификаторов в таблицу **task\_party\_hids**.

Пример ручного наполнения таблицы:

```
-- Отбор всех ФЛ с необработанными конфликтами.  
insert into task_party_hids  
select c.hid_party, 'PHYSICAL', m.hid_final_scion from conflict c  
join physical_party pp ON c.hid_party = pp.hid_party  
-- Дополнительное ускорение с группировкой по золотой карточке КА  
left join merged m on c.hid_party = m.hid_party and m.canceled is null and m.transitional = 0  
where pp.version = 0 and enddate is null and c.status = 0;  
commit;
```

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
<b>partyType</b>	Тип контрагента, для которых выполняется разрешение конфликта	<b>PHYSICAL</b>

<b>poolSize</b>	Максимальное количество потоков	4
resolveStatus	Статус конфликта после обработки	REJECTED

- REJECTED — Отклонен.
- UPDATED\_AFTER\_UNMERGE — Применен (обновлен) после предварительного разлияния.

**Создать/Редактировать задачу**

Группа: reprocess  
 Название: massConflictResolution

**Параметры задачи**

Название параметра	Значение
poolSize	4
resolveStatus	REJECTED
partyType	PHYSICAL

# ПОЛНОЕ НЕОБРАТИМОЕ УДАЛЕНИЕ ВСЕХ ДАННЫХ ВЫБРАННОГО ТИПА

PartyTruncate

✔ Возобновляемая, неконфликтующая

## Общее описание

Задача **необратимо** удаляет **ВСЕ** данные указанного типа контрагента из ЕК.

⚠ Данные удаляются полностью, восстановить их будет нельзя!

## Запуск задачи

1. Запустить задачу, указав в параметре partyType нужный тип КА.
2. Перестроить индексы (поисковый, граф связей, конфликты, дубли), в [режиме горячего резерва](#) на каждой ноде.

## Конфигурация задачи

Настраиваемые параметры:

Параметр	Описание	Значение по умолчанию
partyType	Тип контрагента, данные которого нужно удалить из базы	BLACKLIST
poolSize	Максимальное количество потоков	максимальное
pageSize	Размер пачки	1000

## Логирование

Пример:

```
INFO PartyTruncateTask - Processed 100 000 essences (BLACKLIST: 100000)
...
INFO TaskPerformerStatsInterceptor - FINISHED task 'PartyTruncate
[partyTruncateTaskPerformer]'; processed 2 entities (BLACKLIST: 2); 0 with errors; took 1
seconds or 363 ms; speed 2.0 records/sec;
```

В финальном логировании выводится число всех удаленных КА (включая исторических).

## ОБЩИЕ ПАРАМЕТРЫ ЗАДАЧ

- Секретный параметр `unsafe.forceExecution`
- Параметр `jmsEventFlag`
- Параметр `sendEmail`

# СЕКРЕТНЫЙ ПАРАМЕТР UNSAFE. FORCEEXECUTION

! Страница под рестриктом

Параметр позволяет запустить задачу в параллель, несмотря ни на что. На свой страх и риск.

Есть много комбинаций задач, которые не работают в параллель.

Примеры:

- нельзя на двух нодах стресс-стенда запустить перестроение индекса
- в МТС на первой ноде инкремент стопится, когда на второй ноде идет перестроение графа связей

Теперь в любую задачу можно вписать секретный параметр:

Параметр	Значение
<b>unsafe.forceExecution</b>	<b>true</b>

И она запустится, даже если сейчас работают какие-то другие задачи.

В доку о задачах параметр не выгружается.

! **Поставил параметр, запустил задачу — немедленно сходи и удали параметр!**



# ПАРАМЕТР JMSEVENTFLAG

❗ Страница под рестриктом.

Не все задачи работают с параметром `jmsEventFlag` (но при необходимости его можно быстро добавить).

Сейчас параметр работает с задачами:

- загрузки данных
- пересливания/разливания
- закрытия
- актуализации адресов

Параметр позволяет разметить загружаемые из буферной области определенной задачей данные каким-либо флагом в обратном потоке JMS.

Для разметки необходимо до запуска задачи в ее конфигурации проставить параметр **`jmsEventFlag`**.

События, формируемые в результате работы такой задачи, помечаются в таблице `JMS_EVENT` (поле `FLAG`) указанным в параметре флагом.

## *Пример*

Размечаем в обратном потоке флагом массовости `MASSIVE`:

Параметр	Значение
<b><code>jmsEventFlag</code></b>	<b><code>MASSIVE</code></b>

# ПАРАМЕТР SENDEMAIL

Параметр позволяет отправлять уведомление на email о завершении задачи. Добавить можно в любую задачу.

Параметр	Описание
<b>sendEmail</b>	<b>true / false</b> Включение/выключение отправки уведомления на email
<b>sendTo</b>	Email, на который требуется отправлять уведомление Если <b>sendEmail=true</b> , а почта в <b>sendTo</b> не указана, то письмо будет направлено на эл.почту, указанную в Параметрах отправки email в <b>mail.supportRequest.to</b>

По завершению отправляется письмо, что задача завершена, а также детали в теле письма. Статус завершения задачи (FINISHED, CANCELED, ERROR, SKIPPED, CONFLICTED) не влияет на отправку уведомления.

Для задач диагностики отдельный email о завершении работы не отправляется, приходит только сама диагностика.

## Пример текста письма по умолчанию

**Задача: Поиск дубликатов и слияние**  
**Ключ: findDuplicatessAndMerge**  
**Статус: CANCELED**  
**Нода: 192.168.0.139**  
**Старт: 11.10.2022 16:17:22**  
**Финиш: 11.10.2022 16:17:27**  
**Время работы: 5.06 sec**  
**Пользователь: user**  
**Триггер: нет**

— **deDuplicationWithHouseholdSearchTaskPerformer:**  
**PHYSICAL.importDuplicateTask: processed 0 entities; 0 with errors; took 2 seconds or 2.47 sec;**  
**PHYSICAL.importHouseholdTask: processed 0 entities; 0 with errors; took 1 seconds or 26 ms;**  
**LEGAL.deDuplicationWithHouseholdSearchTaskPerformer: processed 0 entities; 0 with errors; took 2 seconds or 2.23 sec;**

В cdi.log также пишется информация об успешной отправке уведомления:

## Запись в логе для всех задач, кроме задач диагностики

**2022-10-11 16:12:14,608 [TaskManagerExecutor-0] INFO OnTaskFinishEmailNotifier - Task 'findDuplicatessAndMerge' status email sent to email@email.ru**

## Запись в логе для задач диагностики

**2022-10-11 16:54:31,298 [TaskManagerExecutor-5] INFO DiagnosticCollectListener - Diagnostic sent to email@email.ru**

# ЗАМЕНА МАТЕРИАЛИЗОВАННОГО ПРЕДСТАВЛЕНИЯ UNITED\_VIEW ПРОСТЫМ

Задача **migrateUnitedView** заменяет материализованное представление UNITED\_VIEW на обычное представление.

Нужна для замены представлений с целью уменьшения нагрузки на систему управления



Задачу можно запустить только один раз. Задача не перепрогоняемая.

Особенности задачи

1. Запускать задачу только при наличии в базе данных таблицы `exp_party_history`.
2. Запускать только один раз.

# НАЧАЛЬНОЕ НАПОЛНЕНИЕ ТАБЛИЦЫ EXP\_PARTY\_HISTORY

## Общее описание

Задача **exportPartyHistoryInitTask** обрабатывает исторический объем данных по заданным типам контрагентов и наполняет таблицу **EXP\_PARTY\_HISTORY**.

Таблица с историей исходных записей актуализируется в онлайн прямо в момент создания /закрытия/слияния/разливания карточек.

Задача создана для заполнения истории существующего объема данных.



Возобновляемая

Неконфликтующая

## Запуск задачи

Поставить параметр `poolSize =4` (значение по умолчанию).

В режиме горячего резерва запускать только на одной ноде.

## Особенности выполнения

Перед запуском никакие таблицы очищать не нужно. Если таблица **EXP\_PARTY\_HISTORY** уже чем-то заполнена, задача перезагрузит имеющиеся в ней данные, заполнив таблицу актуальными на данный момент.

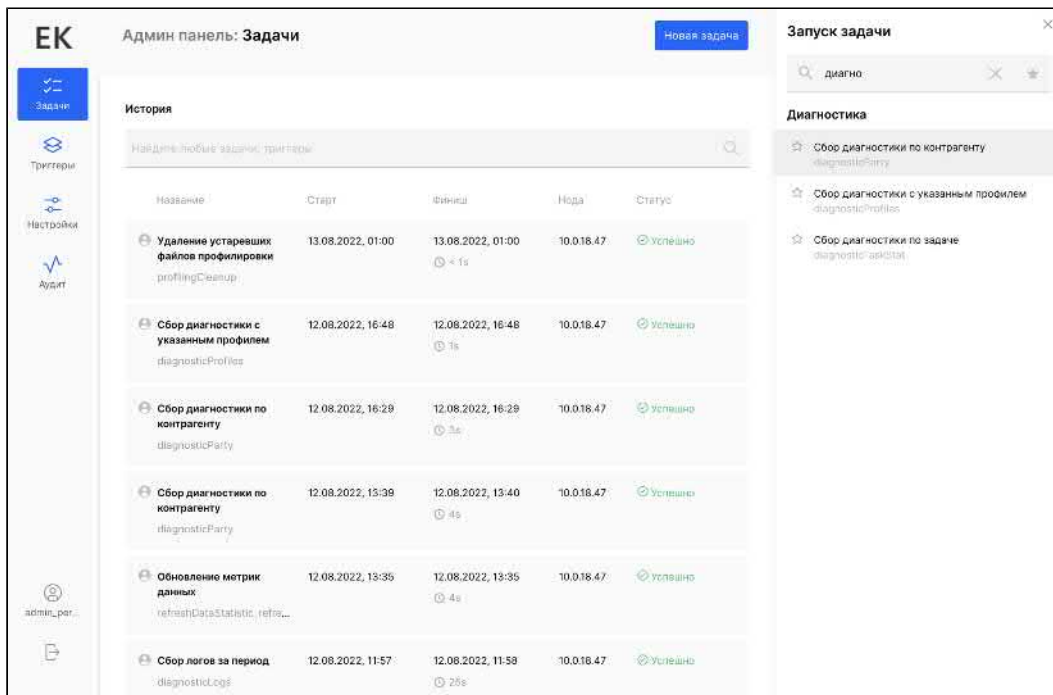
## Конфигурация задачи

Имя	Тип	Описание	Обязательное	Значение по умолчанию
poolSize	int	Количество потоков, которые будут обрабатывать страницы с данными	нет	4

## АДМИН-ПАНЕЛЬ 3.0 (БЕТА)

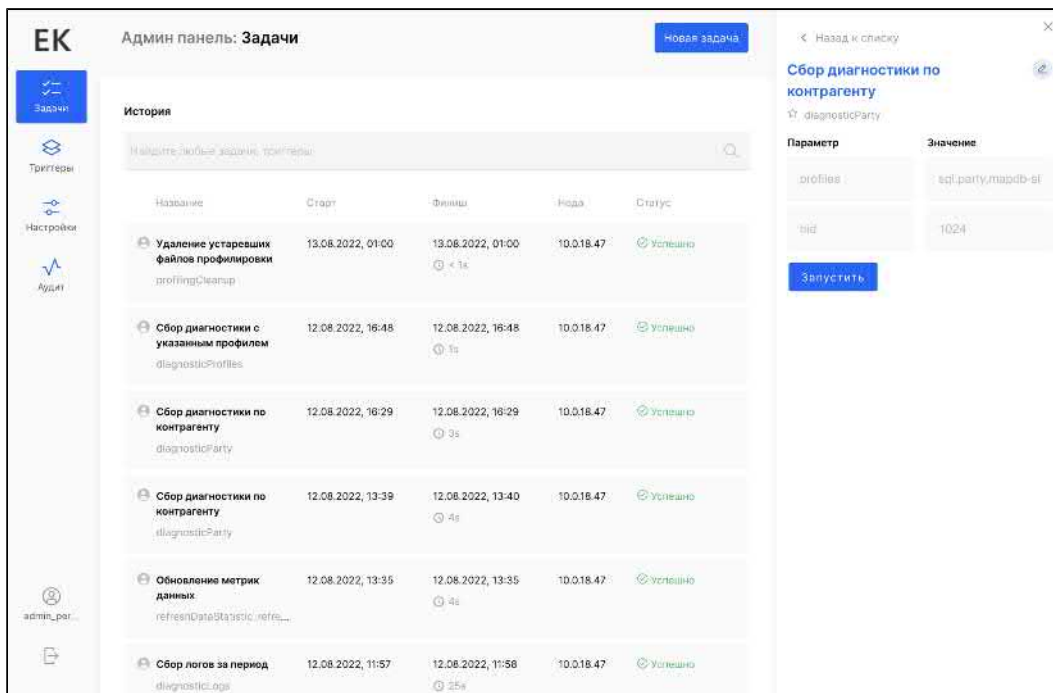
- Запуск новой задачи
- Просмотр истории выполнения задач
- Таблицы новой админки
- Триггеры. Редактирование и запуск
- Управление горячим резервом

# ЗАПУСК НОВОЙ ЗАДАЧИ



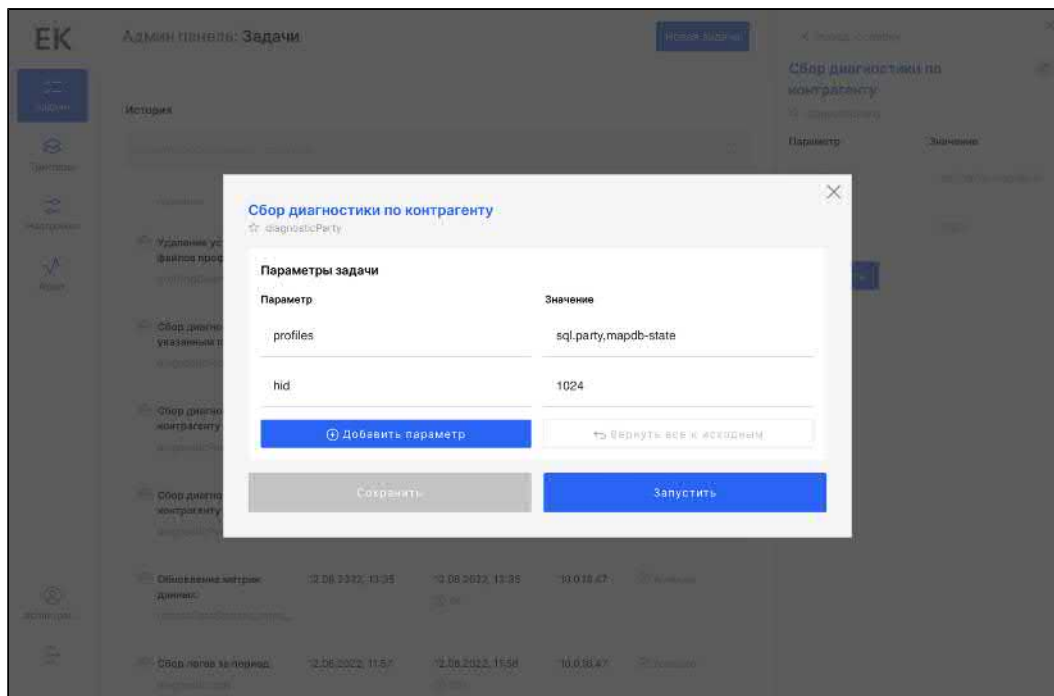
Для запуска новой задачи необходимо нажать на кнопку «Новая задача»

Далее в списке выбрать необходимую задачу. Пользователю доступен поиск, а также сохранение задач в избранное.

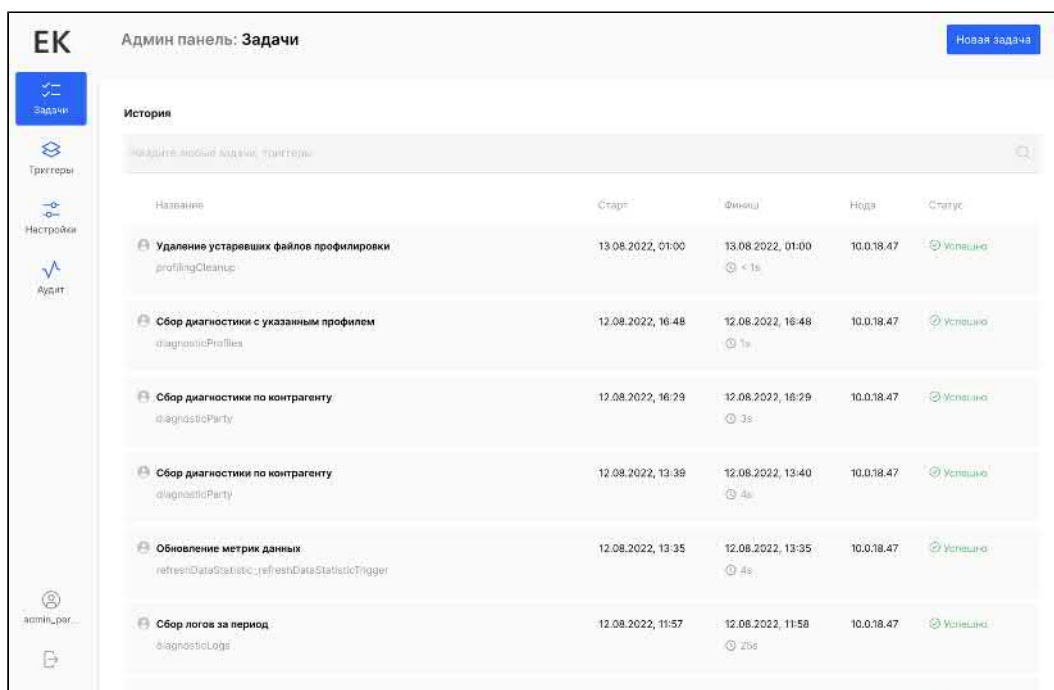


После выбора конкретной задачи есть возможность сразу её запустить с указанными параметрами.

При необходимости отредактировать параметры, необходимо перейти к изменению задачи.

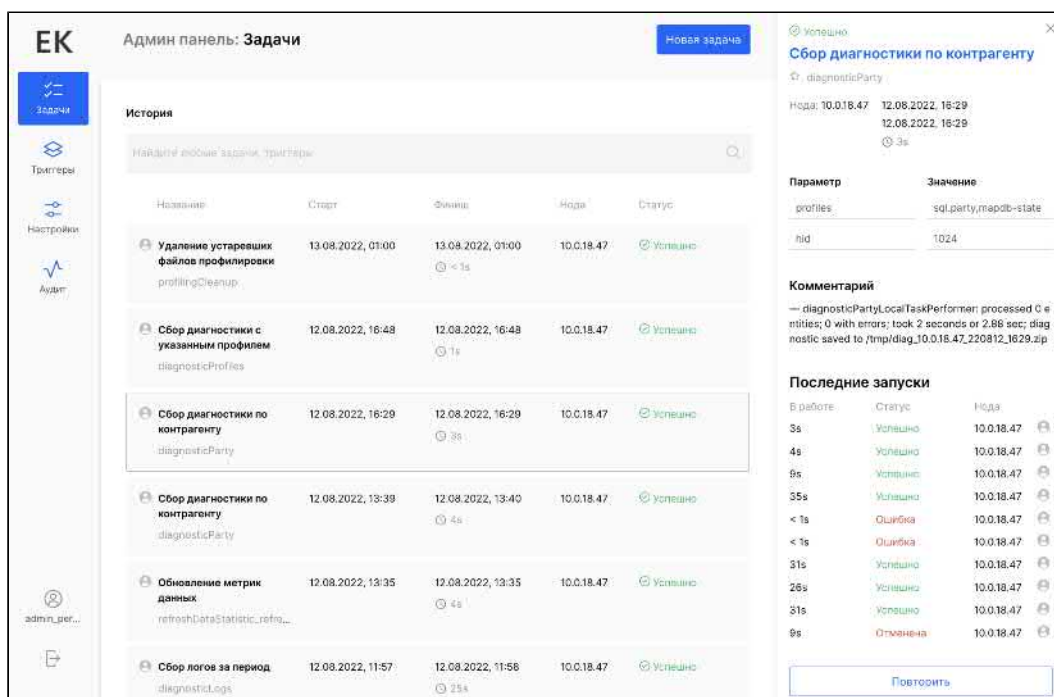


# ПРОСМОТР ИСТОРИИ ВЫПОЛНЕНИЯ ЗАДАЧ



На главной странице админ-панели отображается история выполнения задач.

Каждую запись в истории можно выбрать, чтобы посмотреть подробности выполнения задачи.



На экране подробностей отображается:



- время выполнения задачи;
- параметры выполнения задачи;
- статус завершения задачи и комментарий;
- история последних запусков задач.

Есть возможность перезапустить задачу сразу из этого экрана.

Важно. При повторении запуска задачи из этого экрана задача запустится именно с указанными параметрами.

# ТАБЛИЦЫ НОВОЙ АДМИНКИ

## *task\_record*

Содержит все задачи. Если задача есть в триггере, то для нее создается отдельная строка, где к имени задачи дописывается суффикс "\_имяТриггера".

Колонка	Описание
task_key	Финальное имя задачи (с суффиксом, если в триггере)
instance_key	Оригинальное имя задачи
performer_name	Имя перформера задачи
task_group	Группа, в которой находится задача
task_description	Описание задачи
node_id	Нода, на которой выполняется задача
local_run	Локально ли работает задача
cluster_run	Запускается ли задача в кластере
cluster_run_task_key	Задача, которая дергается на другой ноде при кластерном запуске
trigger_task	Входит ли задача в триггер

## *trigger\_record*

Содержит все триггеры, их параметры и флаг кастомного триггера.

Колонка	Описание
trigger_key	Имя триггера
continue_on_error	Продолжать при наличии ошибок
continue_on_skip	Продолжать при пропусках задачи
continue_on_conflict	Продолжать при наличии конфликтов
optimize_on_nodes	<b>Пока не используется</b>
cron_expression	CRON-выражение
active	Включен ли триггер
trigger_description	Описание триггера
custom_trigger	Кастомный ли это триггер (т.е. был создан пользователем)

## ***task\_to\_trigger\_record***

Содержит связи задач и триггеров, а также порядок задач.

<b>Колонка</b>	<b>Описание</b>
trigger_key	Имя триггера
task_key	Содержит имя задачи из таблицы task_record
order_number	Порядковый номер задачи в триггере, начиная с 0

## ***trigger\_execution***

Содержит историю выполнения триггеров.

<b>Колонка</b>	<b>Описание</b>
id_execution	ID в виде хеша
node_id	Нода, на которой запускался триггер
trigger_key	Имя триггера
status	Статус выполнения триггера
registration_date	Дата и время, когда триггер был вызван
start_date	Дата и время, когда триггер стартовал
end_date	Дата и время, когда триггер закончил работу
total_execution_time	Суммарное время выполнения триггера в миллисекундах

# ТРИГГЕРЫ. РЕДАКТИРОВАНИЕ И ЗАПУСК

- Список триггеров
- Редактирование триггера
  1. Редактирование параметров запуска триггера
  2. Редактирование списка задач в триггере.
  3. Редактирование параметров задач внутри триггера

## Список триггеров

В разделе триггеров доступен полный список триггеров. Триггеры бывают:

- **Активные** — имеющие запланированную дату запуска. Такие триггеры запускаются автоматически по указанному расписанию.
- **Периодические, но неактивные** — триггеры, которые имеют расписание, но временно выключены. Такие триггеры не имеют следующей даты запуска и не будут запускаться автоматически до ручного запуска триггера или перевода его в «активные».
- **Не периодические** — не имеют расписания запуска. Такие триггеры предусмотрены только для разовых точечных запусков вручную. Такой триггер нельзя активировать и поставить на расписание.

Admin панель: Триггеры

Создать триггер

Имя	Запланировано	Расписание	Активность
afterDatabaseBackupTrigger 8 задач		нет	ⓘ ⏪ ⏩ ⏸
dedup 2 задачи		нет	ⓘ ⏪ ⏩ ⏸
detectPopularAttributesTrigger 3 задачи	04.09.2022, 12:00	В 12:00, с 1 по 7 число месяца, и в воскресенье	ⓘ ⏪ ⏩ ⏸
diagnosticLogsForTodayTrigger 1 задача		В 07:55 и 23:55	ⓘ ⏪ ⏩ ⏸
diagnosticMetricsForPreviousMonthTrigge 1 задача	01.09.2022, 02:40	В 02:40, в 1 число месяца	ⓘ ⏪ ⏩ ⏸
fullRefreshDuplicateGroupTrigger 2 задачи		нет	ⓘ ⏪ ⏩ ⏸
generateAndLoadDataForPresentation 1 задача		нет	ⓘ ⏪ ⏩ ⏸
rebuildAllIndexesTrigger 4 задачи		нет	ⓘ ⏪ ⏩ ⏸
refreshDataStatisticTrigger 1 задача	13.08.2022, 02:00	В 02:00	ⓘ ⏪ ⏩ ⏸

**dedup**

Расписание не задано

Следующий запуск: Не известно

Продолжить при пропуске задачи

**Задачи в триггере**

01. findDuplicates

02. duplicateGroupActualize

## Редактирование триггера

Любой триггер можно отредактировать, выбрать соответствующую опцию в строке с нужным триггером.

При изменении триггеров доступно:

### 1. Редактирование параметров запуска триггера

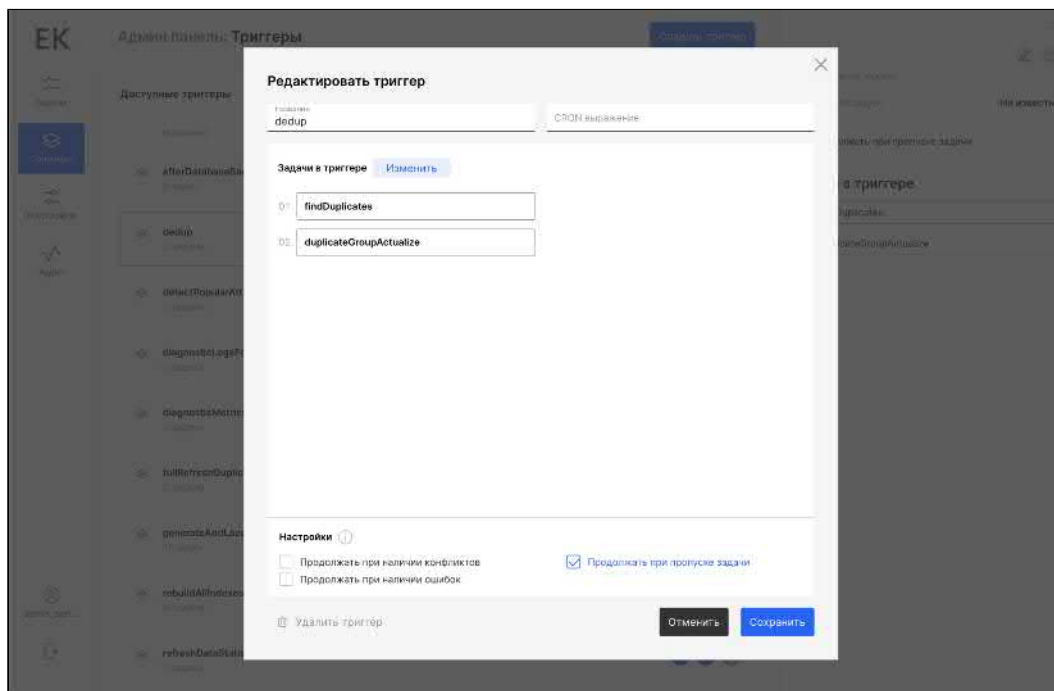
Для триггера уже после его создания можно неограниченное количество раз изменять его название, расписание запуска, настройки запуска.

**⚠️ Обратите внимание:** При добавлении CRON выражения триггер автоматически включится в расписание и будет запланирован для следующего запуска.

### 2. Редактирование списка задач в триггере.

При отображении триггера отображается его текущий состав.

Для перехода к изменению набора задач необходимо выбрать опцию «Изменить».



В открывшемся окне можно:

- добавить новые задачи в триггер;
- в правой части окошка можно перетаскивать задачи по списку, изменяя их порядок;
- удалять задачи из триггера.

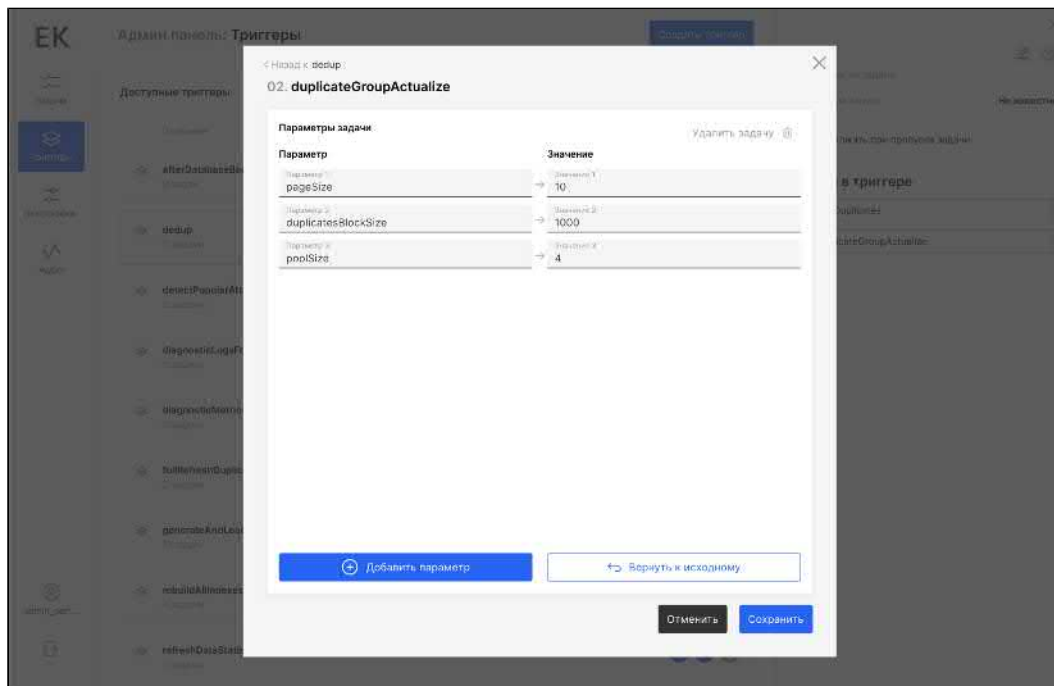
После внесения изменений в набор задач нужно выбрать опцию «Сохранить».

### 3. Редактирование параметров задач внутри триггера

Каждую задачу внутри триггера можно отредактировать — изменить параметры её запуска.

Для этого достаточно просто кликнуть по нужной задаче в окне редактирования триггера.

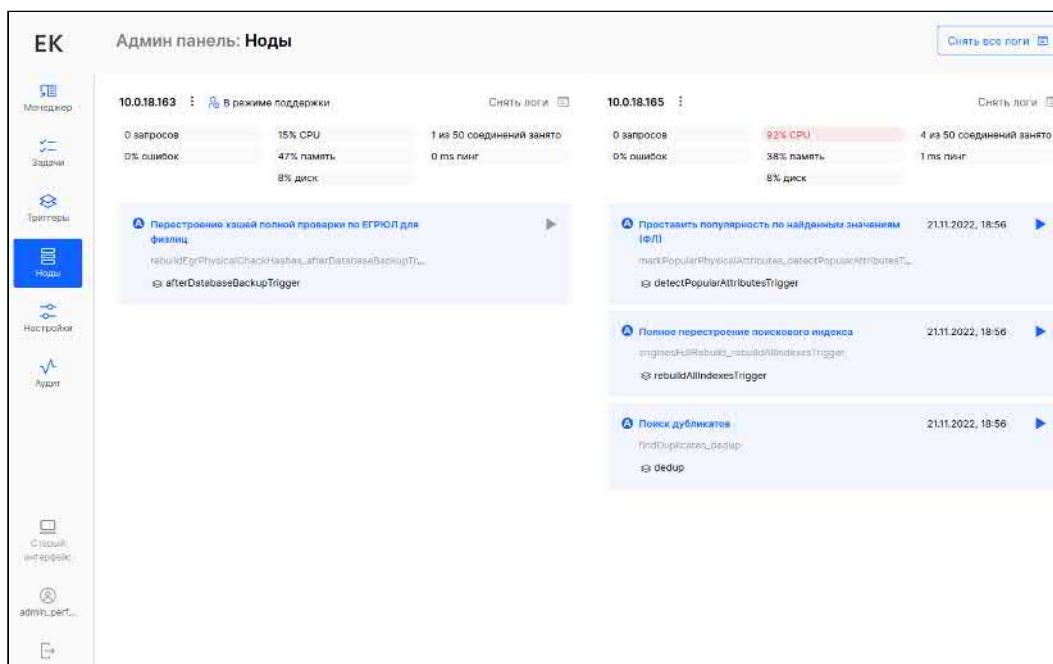
**⚠** Обратите внимание: изменение параметров задачи внутри триггера не влияет на аналогичные задачи внутри других триггеров и задачи без триггеров. А также наоборот.



После завершения редактирования необходимо сохранить изменения.

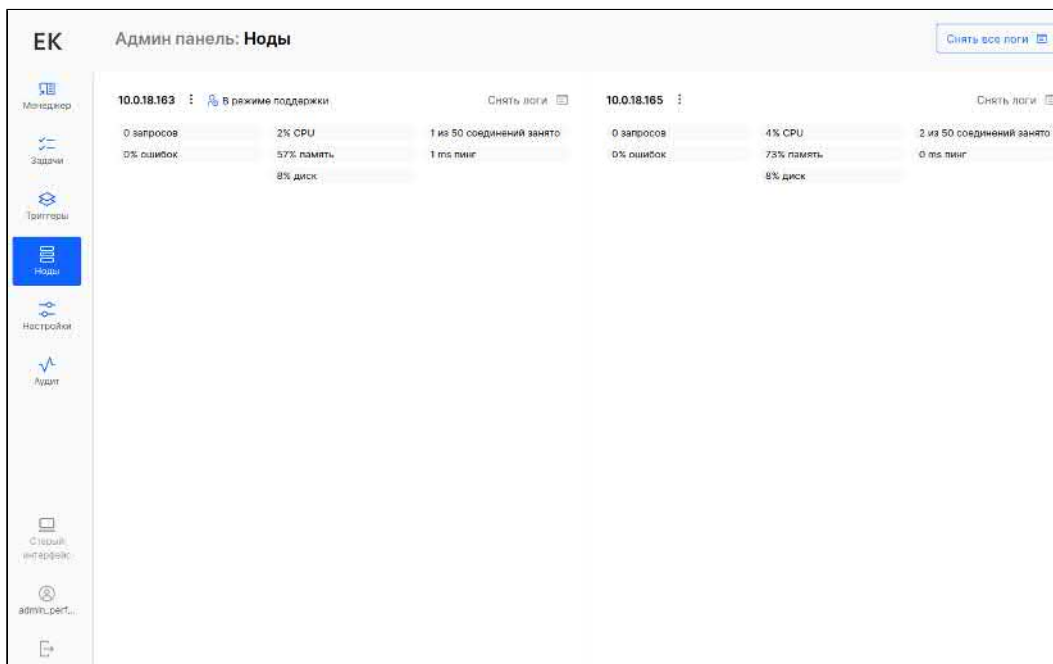
# УПРАВЛЕНИЕ ГОРЯЧИМ РЕЗЕРВОМ

В отдельном меню «Ноды» видно все активные ноды, запущенные на них задачи и показатель «здоровья». Экран позволяет в том числе выводить ноды в режим поддержки для проведения обновления.

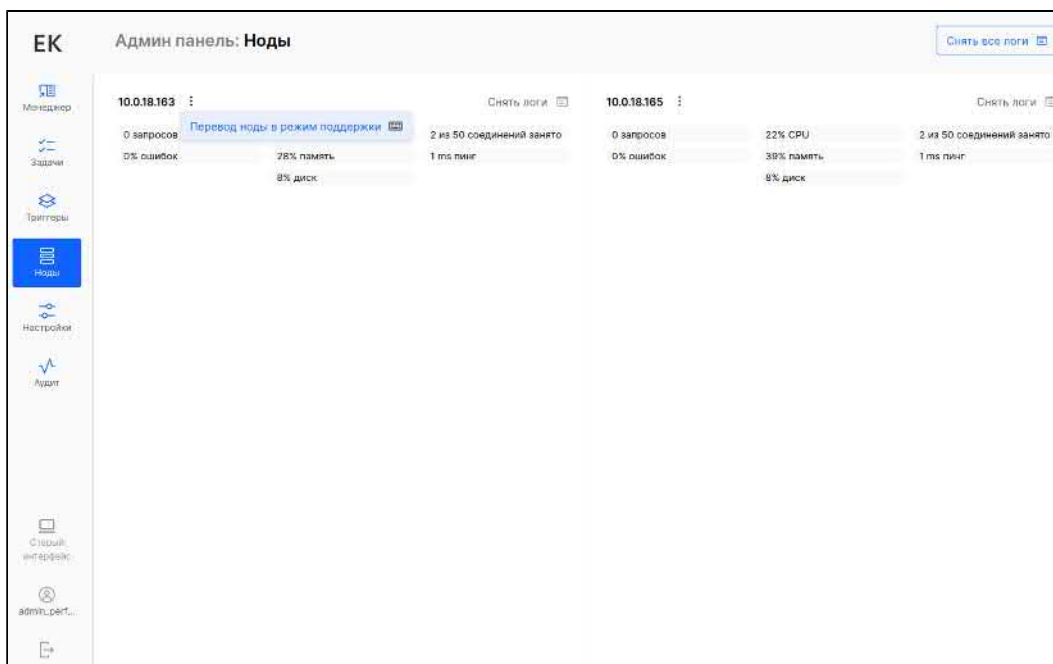


На экране «Ноды» отображаются показатели «здоровья»:

- количество запросов за 5 минут
- процент ошибочных запросов за 5 минут
- текущая нагрузка CPU
- процент занятой оперативной памяти
- процент занятой памяти на диске
- количество занятых соединений к БД из пула
- пинг до БД

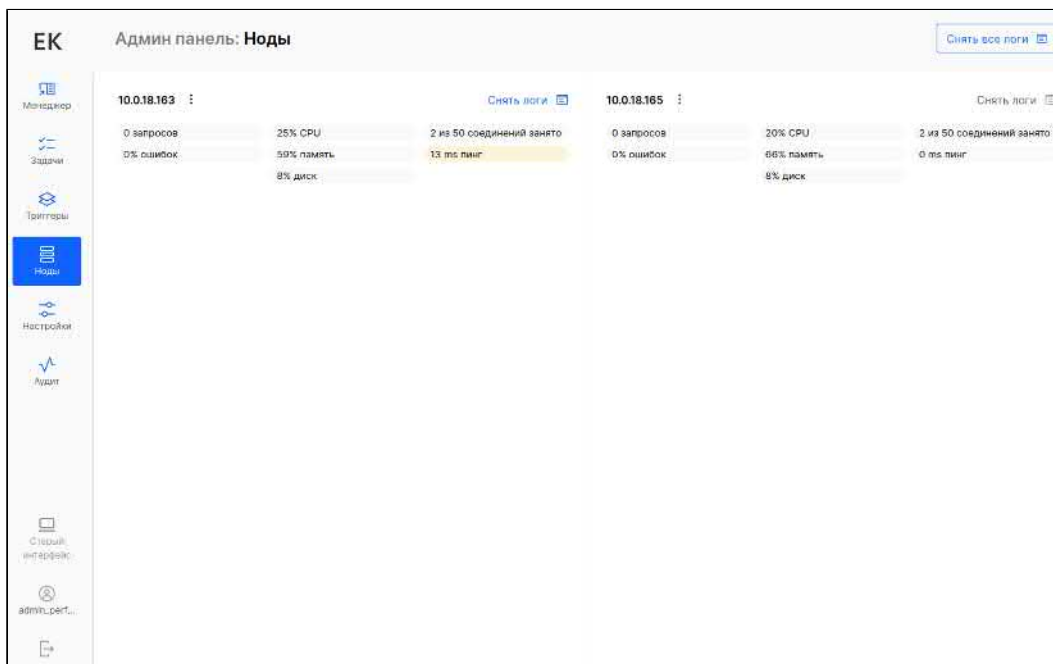


Чтобы перевести или вывести ноду из режима поддержки надо нажать на три точки у IP нужной ноды и выбрать пункт меню. Из интерфейса каждой ноды можно работать с обеими нодами «Единого клиента».



По кнопке «Снять логи» на емейл можно отправить логи нужной ноды. По умолчанию логи отправляются в поддержку «Единого клиента» [log.catcher@hflabs.ru](mailto:log.catcher@hflabs.ru). Дополнительный адрес электронной почты можно предварительно указать в настройках «Параметры отправки email» [mail.supportRequest.to](mailto:mail.supportRequest.to) через запятую.





Можно отправить сразу все логи с обеих нод одной кнопкой — «Снять все логи». Работает так же, как для одной ноды: по умолчанию письмо отправляется только в поддержку «Единого клиента» [log.catcher@hflabs.ru](mailto:log.catcher@hflabs.ru). Дополнительный адрес можно указать в настройках.



На каждой ноде отображаются запущенные задачи. Каждую запись можно выбрать, чтобы посмотреть подробности выполнения задачи. При необходимости задачу можно «Убить» или «Повторить».

**ЕК**
Админ панель: **Ноды**
Снять все логи

Менеджер

Задачи

Триггеры

**Ноды**

Настройки

Аудит

Старый интерфейс

admin\_perf...

**10.0.18.163** Снять логи

0 запросов    2% CPU    2 из 50 соедин...

0% ошибок    37% память    1 ms пинг

8% диск

**10.0.18.165** Снять логи

0 запросов    14% CPU    4 из 50 соедин...

0% ошибок    56% память    1 ms пинг

8% диск

**Загрузка инкремента договоров**    21.11.2022, 19:31

AgreementAutoInc\_gen...

generateAndLoadD...

**Успешно**

**Генерация инкремента договоров со связями**

agreementIncrementGenerateTask\_generateAndLoadDataForPresentation

Нода: 10.0.18.165    21.11.2022, 19:31

21.11.2022, 19:31

Параметр	Значение
incrementSize	100000
percents	100,0,0

**Комментарий**

— agreementIncrementGenerateTaskPerformer: process load 100000 entities: 0 with errors; took 5 seconds or 5.39 sec; speed 20000.0 records/sec;

**Входит в триггеры**

generateAndLoadDataForPresentation

**Последние запуски**

В работе	Статус	Нода
5s	Успешно	10.0.18.165
11s	Успешно	10.0.18.165
7s	Успешно	10.0.18.163
6s	Успешно	10.0.18.165
5s	Успешно	10.0.18.165
5s	Успешно	10.0.18.163
5s	Успешно	10.0.18.163
6s	Успешно	10.0.18.163
8s	Успешно	10.0.18.163
6s	Успешно	10.0.18.165

Повторить

# ЗАКРЫТИЕ НЕАКТУАЛЬНЫХ ЗАПИСЕЙ ЧС

 listPartyCloseUnactual

 Возобновляемая

 Конфликтующая

 Задача актуальна только для модуля ЧС 2.0

## ***Общее описание***

Задача закрывает неактуальные карточки фигурантов списков.

## ***Особенности***


При импорте справочников все карточки фигурантов списков в базе, которых нет в загружаемом списке, должны быть закрыты. Однако задача импорта реализует это путём сверки нового справочника с последним загруженным справочником. Т.к. были выявлены случаи сохранения неактуальных фигурантов списков, то задача импорта справочников проигнорирует подобные записи. Для их закрытия необходимо разово запустить данную задачу.

Список отказников ЦБ данная задача игнорирует.

## ***Конфигурация задачи***

У задачи нет обязательных параметров.

# АКТУАЛИЗАЦИЯ ИНН ИЗ ФНС

 Доступны только заказчикам с подключенным модулем актуализация ИНН из ФНС.

Группа задач, предназначенных для актуализация ИНН из ФНС для физлиц:

Название	Описание	Когда использовать?
<b>fnsInnActualizeForSourceParties</b>	Актуализация ИНН у исходных КА (mergedStatus: 0,1) - исторический массив	Для обработки исходных одиночных и влитых незакрытых карточек (MERGED_STATUS in (0,1)), созданных <b>до</b> подключения модуля ИНН.  Такие карточки будут иметь статус взаимодействия с ФНС «Неизвестно» (UNKNOWN).
<b>fnsInnActualizeForResultParties</b>	Актуализация ИНН у результирующих КА (mergedStatus: 2) - исторический массив	Для обработки результирующих незакрытых карточек (MERGED_STATUS=2), созданных <b>до</b> подключения модуля ИНН.  Такие карточки будут иметь статус взаимодействия с ФНС «Неизвестно» (UNKNOWN).
<b>incrementalFnsInnActualizeForSourceParties</b>	Актуализация ИНН у исходных КА (mergedStatus: 0,1) - инкремент	Для обработки исходных одиночных и влитых незакрытых карточек (MERGED_STATUS in (0,1)), созданных/обновленных <b>после</b> подключения модуля ИНН.  Такие карточки будут иметь статус «На проверку» (TO_CHECK) или «Пере проверка» (RECHECK).
<b>incrementalFnsInnActualizeForResultParties</b>	Актуализация ИНН у результирующих КА (mergedStatus: 2) - инкремент	Для обработки результирующих незакрытых карточек (MERGED_STATUS=2), созданных <b>после</b> подключения модуля ИНН.  Такие карточки будут иметь статус взаимодействия с ФНС «На проверку» (TO_CHECK) или «Пере проверка» (RECHECK).

# АКТУАЛИЗАЦИЯ ИНН У ИСХОДНЫХ КА (MERGEDSTATUS: 0,1) - ИСТОРИЧЕСКИЙ МАССИВ

fnsInnActualizeForSourceParties

✔ Возобновляемая, неконфликтующая

## Описание задачи

Задача актуализирует ИНН на актуальных исходных (merged\_status in (0,1)) карточках, удовлетворяющих условиям:

- не висит конфликт обновления
- fnsDataCheckStatus = UNKNOWN (или null)

Работает по таблице **tmp\$source\_party\_inn\_act\_hids**:

- если таблица отсутствует, задача ее создает;
- если таблица пустая, сначала выполняется ее наполнение согласно правилам отбора, отбор идет по всей базе, без приоритизации;
- по завершении работы таблица удаляется.

## Алгоритм обработки исходных карточек

### Отбор пакетов

Из таблицы отбирается пакет карточек и блокируется.

Если карточки являются влитыми (merged\_status = 1), то все исходные, относящиеся к одной результирующей, попадают в один пакет.

### Алгоритм обработки пачки

Пакет обрабатывается следующим образом:

Этапы	Описание
Расчет хеша	Выполняется расчет хеша (поле FNS_DATA_HASH) по данным карточки — ФИО, ДР и ДУЛ.  Если данных не хватает, то хеш становится пустым и выставляется статус NO_DATA, далее обработка этой записи не идет.

---

Получение ИНН по хешу	Хеши всех, кто не получил NO_DATA, прогоняются <a href="#">через кэш</a> . Те, кто не был там найден, группируются и отправляются <a href="#">батчевым запросом в ФНС</a> , результаты сохраняются в кэш.
Обработка результата	Если получен ИНН, он <a href="#">сохраняется в карточку</a> , статус карточки меняется на VALIDATED. Если получена ошибка, статус карточки меняется на NOT_VALIDATED.

---

Далее переходим к следующему пакету.

Для результирующих карточек (`merged_status = 2`) изменения хеша и статуса не происходит.

# АКТУАЛИЗАЦИЯ ИНН У РЕЗУЛЬТИРУЮЩИХ КА (MERGEDSTATUS: 2) - ИСТОРИЧЕСКИЙ МАССИВ

fnsInnActualizeForResultParties

✔ Возобновляемая, неконфликтующая

## Описание задачи

Задача актуализирует ИНН на результирующих (`merged_status = 2`) карточках, удовлетворяющих условиям:

- не висит конфликт обновления,
- `fnsDataCheckStatus = UNKNOWN` (или `null`),
- у всех ее исходных `fnsDataCheckStatus != UNKNOWN/TO_CHECK/RECHECK`.

Работает по таблице `tmp$result_party_inn_act_hids`:

- если таблица отсутствует, задача ее создает;
- если таблица пустая, сначала выполняется ее наполнение согласно правилам отбора, отбор идет по всей базе, без приоритизации;
- по завершении работы таблица удаляется.

## Алгоритм обработки результирующих карточек

### Отбор пакетов

Из таблицы отбирается пакет карточек и блокируется.

### Алгоритм обработки пачки

Пакет обрабатывается следующим образом:

Этапы	Описание
Расчет хеша	Выполняется <a href="#">расчет хеша</a> (поле <code>FNS_DATA_HASH</code> ) по данным карточки — ФИО, ДР и ДУЛ.  Если данных не хватает, то хеш становится пустым и выставляется статус <code>NO_DATA</code> , далее обработка этой записи не идет.
Получение	Хеши всех, кто не получил <code>NO_DATA</code> , прогоняются <a href="#">через кэш</a> .

---

ИНН по хешу	Те, кто не был там найден, группируются и отправляются <b>батчевым запросом в ФНС</b> , результаты сохраняются в кэш.
-------------	---

---

Обработка результата	Если получен ИНН, он сверяется с ИНН в результирующей карточке: <ul style="list-style-type: none"><li>• если в результирующей заполнен ИНН от ФНС и он совпадает с полученным, то просто меняется статус на VALIDATED</li><li>• в ином случае (если ИНН пустой или отличается), то <b>создается карточка от ФНС</b> и подливается к результирующей, статус меняется на VALIDATED</li></ul>
----------------------	--

---

Если получена ошибка, статус карточки меняется на NOT\_VALIDATED.

---

Далее переходим к следующему пакету.



# АКТУАЛИЗАЦИЯ ИНН У ИСХОДНЫХ КА (MERGEDSTATUS: 0,1) - ИНКРЕМЕНТ

incrementalFnsInnActualizeForSourceParties

✔ Возобновляемая, неконфликтующая

## Описание задачи

Задача актуализирует ИНН на актуальных исходных (merged\_status in (0,1)) карточках, удовлетворяющих условиям:

- не висит конфликт обновления,
- fnsDataCheckStatus in (TO\_CHECK или RECHECK или UNKNOWN или null).

Работает по таблице **FNS\_INC\_CHECK\_SOURCE\_HIDS**.

Задача таблицу не создает и не заполняет, туда попадают карточки, у которых менялись значимые поля (ФИО, ДР, ДУЛ, ИНН) в процессе работы с данными.

## Алгоритм обработки исходных карточек

### Отбор пакетов

Из таблицы отбирается пакет карточек и блокируется.

Если карточки являются влитыми (merged\_status = 1), то все исходные, относящиеся к одной результирующей, попадают в один пакет.

### Алгоритм обработки пачки

Пакет обрабатывается следующим образом:

Этапы	Описание
Расчет хеша	Выполняется <a href="#">расчет хеша</a> (поле FNS_DATA_HASH) по данным карточки — ФИО, ДР и ДУЛ. Если данных не хватает, то хеш становится пустым и выставляется статус NO_DATA, далее обработка этой записи не идет.
Получение ИНН по хешу	Хеши всех, кто не получил NO_DATA, прогоняются <a href="#">через кэш</a> . Те, кто не был там найден, группируются и отправляются <a href="#">батчевым запросом в ФНС</a> , результаты сохраняются в кэш.
Обработка результата	Если получен ИНН, он <a href="#">сохраняется в карточку</a> , статус карточки меняется на VALIDATED.

---

Если получена ошибка, статус карточки меняется на NOT\_VALIDATED.

---

Далее переходим к следующему пакету.

Для результирующих карточек (`merged_status = 2`) изменения хеша и статуса не происходит.

# АКТУАЛИЗАЦИЯ ИНН У РЕЗУЛЬТИРУЮЩИХ КА (MERGEDSTATUS: 2) - ИНКРЕМЕНТ

incrementalFnsInnActualizeForResultParties

✓ Возобновляемая, неконфликтующая

## Описание задачи

Задача актуализирует ИНН на актуальных исходных (merged\_status in (0,1)) карточках, удовлетворяющих условиям:

- не висит конфликт обновления,
- fnsDataCheckStatus in (UNKNOWN, TO\_CHECK, RECHECK),
- у всех ее исходных fnsDataCheckStatus not in (UNKNOWN, TO\_CHECK, RECHECK).

Работает по таблице **FNS\_INC\_CHECK\_RESULT\_HIDS**.

Задача таблицу не создает и не заполняет, туда попадают карточки, у которых менялись значимые поля (ФИО, ДР, ДУЛ, ИНН) в процессе работы с данными.

## Алгоритм обработки результирующих карточек

### Отбор пакетов

Из таблицы отбирается пакет карточек и блокируется.

### Алгоритм обработки пачки

Пакет обрабатывается следующим образом:

Этапы	Описание
Расчет хеша	Выполняется <a href="#">расчет хеша</a> (поле FNS_DATA_HASH) по данным карточки — ФИО, ДР и ДУЛ.  Если данных не хватает, то хеш становится пустым и выставляется статус NO_DATA, далее обработка этой записи не идет.
Получение ИНН по хешу	Хеши всех, кто не получил NO_DATA, прогоняются <a href="#">через кэш</a> .  Те, кто не был там найден, группируются и отправляются <a href="#">батчевым запросом</a> в ФНС, результаты сохраняются в кэш.
Обработка результата	Если получен ИНН, он сверяется с ИНН в результирующей карточке:

- 
- если в результирующей заполнен ИНН от ФНС и он совпадает с полученным, то просто меняется статус на VALIDATED
  - в ином случае (если ИНН пустой или отличается), то создается карточка от ФНС и подливается к результирующей, статус меняется на VALIDATED

Если получена ошибка, статус карточки меняется на NOT\_VALIDATED.

---

Далее переходим к следующему пакету.

## ПОДРОБНОСТИ | FNSCHECK

- Алгоритм обработки исходных карточек
- Алгоритм обработки результирующих карточек

# АЛГОРИТМ ОБРАБОТКИ ИСХОДНЫХ КАРТОЧЕК

## Алгоритм обработки исходных карточек

### Отбор пакетов

Из таблицы отбирается пакет карточек и блокируется.

Если карточки являются влитыми (`merged_status = 1`), то все исходные, относящиеся к одной результирующей, попадают в один пакет.

### Алгоритм обработки пачки

Пакет обрабатывается следующим образом:

Этапы	Описание
Расчет хеша	Выполняется <a href="#">расчет хеша</a> (поле <code>FNS_DATA_HASH</code> ) по данным карточки — ФИО, ДР и ДУЛ. Если данных не хватает, то хеш становится пустым и выставляется статус <code>NO_DATA</code> , далее обработка этой записи не идет.
Получение ИНН по хешу	Хеши всех, кто не получил <code>NO_DATA</code> , прогоняются <a href="#">через кэш</a> . Те, кто не был там найден, группируются и отправляются <a href="#">батчевым запросом в ФНС</a> , результаты сохраняются в кэш.
Обработка результата	Если получен ИНН, он <a href="#">сохраняется в карточку</a> , статус карточки меняется на <code>VALIDATED</code> . Если получена ошибка, статус карточки меняется на <code>NOT_VALIDATED</code> .

Далее переходим к следующему пакету.

Для результирующих карточек (`merged_status = 2`) изменения хеша и статуса не происходит.

# АЛГОРИТМ ОБРАБОТКИ РЕЗУЛЬТИРУЮЩИХ КАРТОЧЕК

## *Алгоритм обработки результирующих карточек*

### *Отбор пакетов*

Из таблицы отбирается пакет карточек и блокируется.

### *Алгоритм обработки пачки*

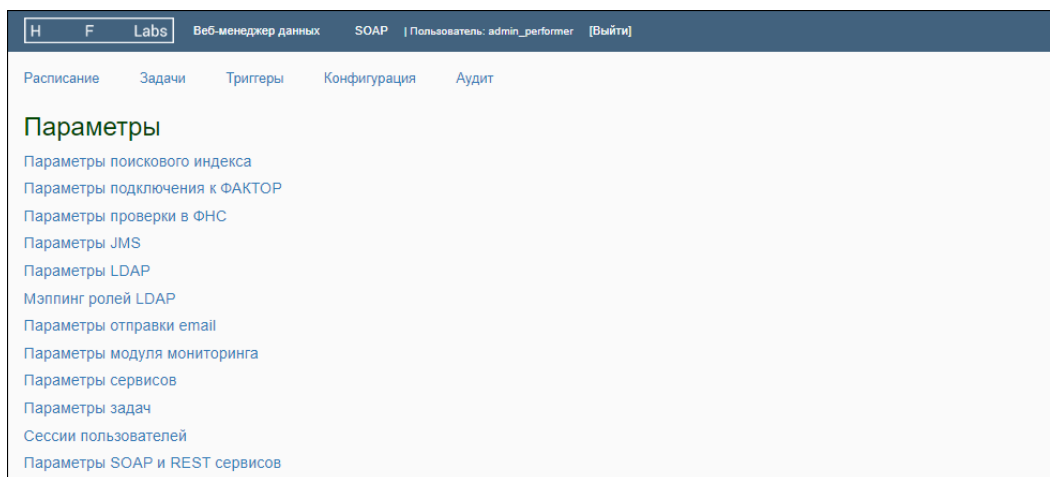
Пакет обрабатывается следующим образом:

Этапы	Описание
Расчет хеша	Выполняется <a href="#">расчет хеша</a> (поле FNS_DATA_HASH) по данным карточки — ФИО, ДР и ДУЛ.  Если данных не хватает, то хеш становится пустым и выставляется статус NO_DATA, далее обработка этой записи не идет.
Получение ИНН по хешу	Хеши всех, кто не получил NO_DATA, прогоняются <a href="#">через кэш</a> .  Те, кто не был там найден, группируются и отправляются <a href="#">батчевым запросом в ФНС</a> , результаты сохраняются в кэш.
Обработка результата	Если получен ИНН, он сверяется с ИНН в результирующей карточке: <ul style="list-style-type: none"><li>• если в результирующей заполнен ИНН от ФНС и он совпадает с полученным, то просто меняется статус на VALIDATED</li><li>• в ином случае (если ИНН пустой или отличается), то создается <a href="#">карточка от ФНС</a> и подливается к результирующей, статус меняется на VALIDATED</li></ul> Если получена ошибка, статус карточки меняется на NOT_VALIDATED.

Далее переходим к следующему пакету.

# НАСТРОЙКА СИСТЕМЫ

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



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

Описание стандартных настраиваемых параметров:

- Мэппинг ролей LDAP
- Настройки задач во внешней директории
- Параметры JMS
- Параметры LDAP
- Параметры SOAP и REST сервисов
- Параметры задач
- Параметры задачи дедупликации
- Параметры запуска JBOSS
- Параметры импорта из буферной области
- Параметры импорта справочников
- Параметры модуля мониторинга
- Параметры отправки email
- Параметры подключения к Фактору
- Параметры поискового индекса
- Параметры проверки в ФНС
- Параметры сервисов
- Параметры синхронизации нод
- Параметры экспорта данных
- Сессии пользователей



Вне интерфейса администратора настраиваются:

- Параметры запуска JBOSS
- Настройки задач во внешней директории

# МЭПШИНГ РОЛЕЙ LDAP



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

Название параметра	Значение
ROLE_ADMINISTRATOR	CDI_LDAP_ADMINISTRATOR
ROLE_CONFLICT_MANAGER	CDI_LDAP_CONFLICT_MANAGER
ROLE_CONSENT_MANAGEMENT_CENTER	CHANGE_LDAP_CONSENT_MANAGEMENT_CENTER
ROLE_CONSENT_MANAGER	CDI_LDAP_CONSENT_MANAGER
ROLE_DUPLICATE_MANAGER	CDI_LDAP_DUPLICATE_MANAGER
ROLE_EXTERNAL_SYSTEM	CDI_LDAP_EXTERNAL_SYSTEM
ROLE_EXTERNAL_SYSTEM_READ	CDI_LDAP_EXTERNAL_SYSTEM_READ
ROLE_EXTERNAL_SYSTEM_WRITE	CDI_LDAP_EXTERNAL_SYSTEM_WRITE
ROLE_GUARD	CDI_LDAP_GUARD
ROLE_OPERATOR	CDI_LDAP_OPERATOR
ROLE_PERFORMER	CDI_LDAP_PERFORMER
ROLE_RELATION_MANAGER	CDI_LDAP_RELATION_MANAGER

Если значение какого-то параметра пустое — приложение запустится, в логе будет запись вида:

```
2019-05-28 12:46:59,946 [ServerService Thread Pool -- 78] INFO  
ConfigurationPropertiesToPropertiesConverter - Value for property  
ROLE_RELATION_MANAGER doesn't set
```

Группа LDAP	Системная роль
<b>ROLE_ADMINISTRATOR</b>	администратор
<b>ROLE_GUARD</b>	офицер информационной безопасности
<b>ROLE_OPERATOR</b>	оператор (read-only)
<b>ROLE_CONSENT_MANAGER</b>	оператор согласий (read-only)
<b>ROLE_PERFORMER</b>	менеджер данных
<b>ROLE_DUPLICATE_MANAGER</b>	менеджер дубликатов
<b>ROLE_EXTERNAL_SYSTEM</b>	исходная система с полным доступом к онлайн-интерфейсам
<b>ROLE_EXTERNAL_SYSTEM_READ</b>	исходная система с доступом только к методам на чтение
<b>ROLE_EXTERNAL_SYSTEM_WRITE</b>	исходная система с доступом только к методам на запись
<b>ROLE_RELATION_MANAGER</b>	верификатор связей
<b>ROLE_CONFLICT_MANAGER</b>	менеджер конфликтов
<b>ROLE_CONSENT_MANAGEMENT_CENTER</b>	менеджер ЦУС
<b>ROLE_JOURNAL_MANAGER</b>	оператор журналов проверки по ЧС

Если значение — **\${ldap.roles...}**, то группа не настроена, такой роли в системе нет.



Учетные записи для интегрируемых систем рекомендуется сохранять как **системные учетные записи**, а не как учетные записи AD, чтобы не было задержек при обработке запросов из за сервера AD.

# СПИСОК МЕТОДОВ, ДОСТУПНЫХ ТОЛЬКО ДЛЯ ЧТЕНИЯ/ЗАПИСИ

## *Методы, доступные для чтения*

1. search
2. fuzzySearch
3. checkBlakList
4. getByHID
5. getByRawID
6. getRelations
7. getSourceAttribute
8. getReference

## *Методы, доступные для записи*

1. save
2. saveAndMerge
3. saveRelations
4. close
5. closeAttribute
6. closeAttributeByRawId

# НАСТРОЙКИ ЗАДАЧ ВО ВНЕШНЕЙ ДИРЕКТОРИИ

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

- Вместо конкретных значений в описании триггеров и задач отображаются плейсхолдеры, например - `${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`

# ПАРАМЕТРЫ JMS

- Общие параметры
- Специфичные для очередей TIBCO параметры
- Специфичные для очередей IBM MQ параметры
- Параметры для передачи по HTTP-протоколу
- Параметры для отправки в KAFKA
- Специфичные для очередей RabbitMQ параметры



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

## Общие параметры

Название параметра	Значение по умолчанию	Описание
<b>jms.connectionFactory.name</b>	-	Название connectionFactory, если используется.
<b>jms.events.destination.isTopic</b>	<b>false</b>	Выходная очередь — topic?
<b>jms.events.destination.name</b>	-	Название выходной очереди.
<b>jms.events.enabled</b>	<b>false</b>	Рассылка сообщений в выходную очередь включена? После изменения значения данного параметра рестарт CDI не требуется.
<b>jms.soap.destination.isTopic</b>	<b>false</b>	Входная очередь — topic? Если false, используется очередь. (Очередь — один получатель, топик — много)
<b>jms.soap.destination.name</b>	-	Название входной очереди.
<b>jms.soap.enabled</b>	<b>false</b>	Получение сообщений из входной очереди включено?
<b>jms.soap.processingNodeId</b>	null	Список идентификаторов нод, на которые должен приходить входящий jms поток Указывать можно через запятую. Если не указано ни одной ноды, то активными считаются все ноды, у которых не перекрыто в standalone.conf настройка <i>-Djms.soap.enabled=false</i> Указывать нужно ip-адреса или идентификаторы нод, если они указаны в standalone.conf через <i>-Dcdi.app.nodeId=node1</i>

		✓ Рестарт CDI не требуется
<b>jms.events.attributes.defaultValues</b>	<b>false</b>	Возвращать в обратный поток согласия по умолчанию?
<b>jms.events.changeDetailsEnabled</b>	<b>false</b>	Возвращать ли результат в детализированном виде
<b>jms.send.slowSendThresholdSec</b>	<b>10</b>	Время в секундах работы с очередью, при превышении которого в лог записывается ошибка следующего вида: <b>ERROR JmsEventPublishHelper - Sending JMS message took 14.575 s! Event: 1040</b>
<b>jms.events.filter.enabled</b>	<b>false</b>	Включить фильтры при формировании событий?  Набор фильтров уникален для каждого заказчика. Изменение применяется без рестарта.
<b>jms.events.maxSize</b>	<b>100</b>	Количество action в событии ОП  ✓ Рестарт CDI не требуется
<b>jms.events.attributes.pastValues</b>	<b>false</b>	Отправка прошлых значений атрибутов ФИО и ДУЛ в JMS сообщениях. Информация о прошлых значениях возвращается в сущности <b>pastAttribute</b> .  ✓ Рестарт CDI не требуется

## Специфичные для очередей TIBCO параметры

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

## Специфичные для очередей IBM MQ параметры

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

## Параметры для передачи по HTTP-протоколу

Название параметра	Значение по умолчанию	Описание
--------------------	-----------------------	----------

<b>jms.events.rest.enabled</b>	<b>false</b>	Флаг для отключения отправителя
<b>jms.events.rest.login</b>		Логин для авторизации
<b>jms.events.rest.password</b>		Пароль для авторизации
<b>jms.events.rest.poolSize</b>	10	Во сколько потоков посылать данные
<b>jms.events.rest.timeout</b>	<b>60000</b>	Максимальное время ожидания ответа от сервиса-получателя
<b>jms.events.rest.url</b>	http://host:8888	Строка для подключения к сервису-получателю

## Параметры для отправки в КАФКА

Название параметра	Значение по умолчанию	Описание
<b>jms.events.kafka.enabled</b>	<b>false</b>	Отправить сообщения в кафку? ✔ При изменении параметра рестарт CDI не требуется
<b>jms.events.kafka.bootstrapServers</b>	<b>localhost:9092</b>	Список серверов кафки
<b>jms.events.kafka.topic</b>	<b>cdi_events</b>	Название топика
<b>jms.events.kafka.timeoutInSec</b>	5	
<b>jms.events.kafka.transactionId</b>	cdi	
<b>jms.events.kafka.maxRequestSize</b>	20971520	Максимальный размер исходящего сообщения
<b>jms.events.kafka.bufferMemory</b>	125829120	Максимальный размер исходящего буфера. Если продюсер не смог отправить сообщения брокеру (брокер отвалился, например), он начинает накапливать батчи сообщений в этом буфере.
<b>jms.events.kafka.batchSize</b>	20971520	Размер батча в байтах. Батчи используются, когда нужно записать несколько сообщений в одну партицию. Заполнился батч для партиции ? отправили сообщение в брокер
<b>jms.events.kafka.maxBlockMs</b>	120000	Когда исходящий буфер заполнится, ровно столько миллисекунд будет ждать продюсер, прежде чем выкинуть эксепшн, если буфер не очистится
	120000	,



---

**jms.events.  
kafka.  
requestTimeout  
Ms**

---

**jms.events.  
kafka.  
deliveryTimeou  
tMs**

---

## Специфичные для очередей RabbitMQ параметры

---

Название параметра	Описание
<b>jms.events.rabbitmq.enabled</b>	Включена ли отправка сообщений в шину.
<b>jms.events.rabbitmq.destination.name</b>	Название выходной очереди. При отправке сообщения значение данного параметра указывается в <b>Routing Key</b> .
<b>jms.events.rabbitmq.destination.exchange</b>	Место назначения. При отправке сообщения значение данного параметра указывается в <b>Exchange</b> . Может использоваться в RabbitMQ для маршрутизации сообщений. По умолчанию: - не заполнено. RabbitMQ пустое значение данного параметра воспринимает как — <b>AMQP default</b> .
<b>jms.events.rabbitmq.host</b>	Хост шины / несколько хостов через запятую. Важно: Все указанные хосты должны быть настроены на порт, указанный в <b>jms.events.rabbitmq.port</b> . Система выбирает хост из указанных и отправляет сообщения на него. Если хост перестает отвечать, сообщения отправляются на другой доступный хост из списка.
<b>jms.events.rabbitmq.port</b>	Порт шины
<b>jms.events.rabbitmq.username</b>	Логин для авторизации.
<b>jms.events.rabbitmq.password</b>	Пароль для авторизации.
<b>jms.events.rabbitmq.connectionTimeout</b>	Максимальное время ожидания (в мс) ответа от JMS. Значение по умолчанию: 5000

---

# ПАРАМЕТРЫ LDAP



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

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.warmDurationMs	100

Сохранить

Название параметра	Значение по умолчанию	Описание
<b>ldap.url</b>	ldap://hostname:port	Hostname и порт сервера LDAP
<b>ldap.account</b>	-	Логин системной учетной записи
<b>ldap.domain</b>	-	Имя домена локальной сети
<b>ldap.groupSearchBase</b>	-	Ветка, в которой хранятся учетные записи для ролей
<b>ldap.maxActive</b>	50	Максимальное количество активных соединений одного вида (чтение/запись)
<b>ldap.maxIdle</b>	10	Максимальное количество бездействующих соединений одного типа (чтение/запись) в пуле
<b>ldap.maxWait</b>	-1	Максимальная длительность (миллисек) ожидания свободного соединения пока не будет выброшено исключение Если равно -1, ждать бесконечно долго, пока не будет свободного соединения

<b>ldap.password</b>	-	Пароль системной учетной записи
<b>ldap.searchBase</b>	-	Ветка, в которой хранятся учетные записи пользователей (поиск ведется в данной ветке и всех дочерних)
<b>ldap.searchFilter</b>	(sAMAccountName={0})	Атрибут, в котором хранится имя пользователя (как он его указывает при входе в систему)
<b>ldap.testOnBorrow</b>	true	Булевый параметр (true/false), определяющий, должен ли объект пройти валидацию сканером (если таковой есть) прежде чем будет взят из пула
<b>ldap.testWhileIdle</b>	true	Булевый параметр (true/false), указывающий, что необходимо проводить валидацию простаивающих потоков пула с интервалом <b>ldap.timeBetweenEvictionRunsMillis</b>
<b>ldap.timeBetweenEvictionRunsMillis</b>	60000	Количество миллисекунд между запусками потока-сканера
<b>ldap.connect.timeout</b>	60000	Таймаут операции соединения с LDAP-сервером в миллисекундах
<b>ldap.read.timeout</b>	60000	Таймаут операции чтения ответа от LDAP-сервера в миллисекундах
<b>ldap.warnDurationMs</b>	100	Пороговое значение длительности запроса к LDAP при превышении которого выводится WARN-запись в журнал <code>cdi-security.log</code>

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.

# ПАРАМЕТРЫ SOAP И REST СЕРВИСОВ

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

 Если не указана возможность применения настроек без рестарта

Название параметра	Значение по умолчанию	Описание
<b>webservice.error.logMode</b>	FULL	Если <b>FULL</b> , то в <b>cdi.log</b> записывается текст ошибочного запроса и вызванная им ошибка. Если <b>SHORT</b> и включена трассировка запросов к веб-сервисам, то в <b>cdi.log</b> записывается ошибка и идентификатор полного сообщения в <b>soap-trace.log</b> . Если <b>SHORT</b> и не включена трассировка запросов к веб-сервисам, то в <b>cdi.log</b> записывается только ошибка.
<b>webservice.stats.periodInSeconds</b>	1800	Периодичность вывода в лог статистики по вызову веб-сервисов (в секундах).
<b>webservice.response.attributes.defaultValues</b>	false	Если <b>true</b> , то согласия по умолчанию возвращаются в веб-сервисах. Для методов <b>search</b> , <b>getByHid</b> и <b>fuzzySearch</b> «Единый клиент» начинает использовать БД для загрузки согласий
<b>webservice.response.attributes.pastValues</b>	false	Если <b>true</b> , то прошлые значения возвращаются в веб-сервисах Исключением является метод <b>search</b> , он не зависит от значения параметра
<b>webservice.concurrent.numberOfRetries</b>	3	Количество попыток повторной обработки контрагента при получении ошибки о параллельной модификации
<b>webservice.concurrent.delayBetweenRetries</b>	1000	Задержка между попытками повторной обработки при получении ошибки о параллельной модификации
<b>webservice.response.relations.maxCountByParty</b>	2000	Ограничение на количество возвращаемых связей в SOAP-методах ( <b>getByHid</b> , <b>getByRawId</b> , <b>getRelations</b> )
<b>webservice.maxPartiesCount</b>	2000	Ограничение на количество возвращаемых контрагентов для методов <b>search</b> и <b>fuzzySearch</b>
<b>webservice.mergeEnabled</b>	true	Разрешено ли слияния и дедубликация в soap методах? Если <b>false</b> — то метод работает как простой <b>save</b>
<b>webservice.logging.filters</b>	<b>search=WARN:1s, CRITICAL:2s;</b>	Настройки перекрывают умолчательное поведение, заданное в <b>webservice.logging.defaultBehaviour</b> . Для каждого вида запросов можно конкретно указать таймаут и тип реакции. Пример: <b>search=WARN:1s,CRITICAL:5s; other=WARN:5s</b> Расшифровывается так:

**getByHID=**  
**WARN:1s,**  
**CRITICAL:**  
**2s;**

- если вызвали "**search**", то  
если он длился больше 5 секунд — **CRITICAL**  
если больше 1 секунды — **WARN**  
иначе умолчательная реакция (обычно **SKIP**)

**getByRawl**  
**D=WARN:**  
**1s,**  
**CRITICAL:**  
**2s;**

- для остальных методов  
если метод длился больше 5 секунд — **WARN**  
иначе умолчательная реакция

**fuzzySearch=**  
**WARN:**  
**5s,**  
**CRITICAL:**  
**10s;**

На данный момент при превышении значения CRITICAL особое поведение не предусмотрено. Однако, если среднее значение выполнения запроса превышает CRITICAL, в лог выводится ошибка

```
[SoapLongopsAnalyzer#Flusher] ERROR  
SoapLongopsAnalyzer - Average processing time of  
'name' is x.xx sec (critical=x sec)
```

**saveAndMerge=**  
**WARN:**  
**N:2s,**  
**CRITICAL:**  
**5s;**

Иногда запросы очень редкие и система может срабатывать на один долгий запрос — это часто некорректно, поэтому используется усреднение за период, указанный в **webservice.stats.periodInSeconds**.

**save=**  
**WARN:**  
**N:2s,**  
**CRITICAL:**  
**5s;**

По умолчанию, если за период не накопилось больше 10 запросов за период, то это не засчитывается. Переопределить условие на количество можно при помощи параметра COUNT.

**saveRelations=**  
**WARN:**  
**1s,**  
**CRITICAL:**  
**2s;**

Пример: **search=WARN:1s,CRITICAL:2s,COUNT:45;**

Расшифровывается так:

**other=**  
**WARN:**  
**5s,**  
**CRITICAL:**  
**15s**

- если вызвали "**search**", то  
если за период было меньше 45 вызовов — ничего не делать,  
если среднее больше 2 секунд — **CRITICAL**  
если среднее больше 1 секунды — **WARN**

С условием  
на  
количество:

 Изменения данных параметров вступает в силу без рестарта ЕК.


**search=**  
**WARN:**  
**1s,**  
**CRITICAL:**  
**2s,COUNT:**  
**45 ;**

**webservice.**  
**logging.sampling.**  
**periodInMs**

100

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.

Чтобы выключить jvmt-трассировку долгих запросов, установите отрицательное значение (например, -1).

 Изменение данного параметра вступает в силу без рестарта ЕК.

<b>webservice. searchRelatedParties.response. maxEntries</b>	20	<p>Ограничение на количество возвращаемых пар связанных записей в методе <b>searchRelatedParties</b>.</p> <p>Возможен вариант без ограничений, для этого установите значение -1.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px;">  Изменение данного параметра вступает в силу без рестарта ЕК. </div>
<b>webservice. saveAndMerge. dedupAndMergeRelatedParties</b>	false	<p>Если установлено в true:</p> <ul style="list-style-type: none"> <li>из указанных в <b>SaveAndMerge</b> связей отбираем КА с сильной связью (strong=true в таблице relation_type БД);</li> <li>группируем КА по типу и отправляем на дедупликацию и слияние.</li> </ul>
<b>webservice. saveAndMerge. dedupAndMergeRelatedPartiesMaximum</b>	50	<p>Действует только при <b>webservice.saveAndMerge.dedupAndMergeRelatedParties=true</b>.</p> <p>Если количество связанных КА одного типа больше указанного значения, то они не будут отправлены на дедупликацию и слияние.</p>
<b>webservice. logging.longops.ui. warnTimeThresholdInMillis</b>	1000	<p>При достижении warn порога начинается сбор информации об использовании CPU во время выполнения запроса веб-интерфейса.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px;">  Изменение данного параметра вступает в силу без рестарта ЕК. </div>
<b>webservice. logging.longops.ui. criticalTimeThresholdInMillis</b>	3000	<p>При достижении critical порога вся информация об использовании CPU во время выполнения запроса веб-интерфейса записывается в лог cdi-ui-longops.log.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px;">  Изменение данного параметра вступает в силу без рестарта ЕК. </div>
<b>webservice. transform. mixAttributeFieldsForPartialUpdate</b>	true	<p>Перемешать текущие поля атрибута, реквизита (место рождения) с новыми полями из <b>PARTIAL</b>-обновления?</p> <ul style="list-style-type: none"> <li>если значение true — на стандартизацию в «Фактор» уходят поля текущего атрибута и присланные в обновлении поля;</li> <li>если значение false — на стандартизацию в «Фактор» уходят только присланные в обновлении поля.</li> </ul>
<b>webservice. useSearchIndexForGetBy</b>	false	<p>Позволяет методам getByHid/getByRawId брать данные из индекса, а не из базы.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; margin-top: 10px;">  Изменение данного параметра вступает в силу без рестарта ЕК. </div>

# ПАРАМЕТРЫ ЗАДАЧ



1. Изменения настроек вступают в силу только после рестарта сервиса Единого клиента!
2. Настройки **фоновых задач** не требуют рестарта сервиса.

Название параметра	Значение
task.deduplication.cacheSize	10
task.deduplication.freeSpaceRatio	1.25
task.deduplication.pageSize	1000
task.deduplication.poolSize	-1
task.finishedTaskDescriptorExpirationPeriod	12h
task.merge.pageSize	500
task.merge.poolSize	-1
task.merge.threshold	5000
task.multiclean.timeout	10
task.merge.maxOriginalPartiesCount	1000
task.sql.connector.timeout	2h
task.sql.db.procedure.timeout	24h
task.transform.poolSize	-1
task.triggers.scheduleOnStartup	true

Сохранить

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

<b>task.finishedTaskDescriptorExpirationPeriod</b>	<b>12h</b>	Время "протухания" дескрипторов задач после их завершения Формат значений: " <b>1 day</b> ", " <b>35 min</b> ", " <b>20h</b> " и т.д.
<b>task.merge.pageSize</b>	<b>500</b>	Размер пакета (контрагентов) при слиянии (за один раз объединяется N групп дубликатов, общее количество контрагентов в которых не превышает pageSize)
<b>task.merge.poolSize</b>	<b>8</b>	Количество потоков слияния Если равно -1, устанавливается равным количеству процессоров (ядер)*2
<b>task.merge.threshold</b>	<b>100</b>	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
<b>task.multiclean.timeout</b>	<b>10</b>	Время ожидания (в минутах) появления свободного сервиса очистки, при превышении которого задача завершается
<b>task.reMerge.maxOriginalPartiesCount</b>	<b>1000</b>	Максимальное число оригинальных контрагентов для страницы переслияния
<b>task.transform.poolSize</b>	<b>16</b>	Количество потоков трансформации. Если равно -1, устанавливается равным количеству процессоров (ядер)*2
<b>task.triggers.scheduleOnStartup</b>	<b>true</b>	Активировать триггеры по расписанию на старте приложения
<b>task.physical.household.relationTypepeld</b>	<b>8</b>	Идентификатор типа связи для задачи поиска "Родственник" физ.лиц (при подключении <a href="#">поиска домохозяйств</a> )
<b>task.physical.household.hhRelationTypepeld</b>	<b>500</b>	Идентификатор типа связи для задачи поиска "Домохозяйств" физ.лиц (при подключении <a href="#">поиска домохозяйств</a> )
<b>task.sql.dbProcedure.timeout</b>	<b>24h</b>	Таймаут для выполнения хранимых процедур и sql-файлов, варианты записи значения см в дочерней странице
<b>task.sql.connector.timeout</b>	<b>2h</b>	Таймаут для коннекторов, которые возвращают данные (допустим, не удалось подключиться к БД — коннектор отваливается), варианты записи значения см в дочерней странице
<b>task.silent.poolSize.delta</b>	<b>4</b>	Дельта наращивания/уменьшения количества потоков для работы фоновых задач. По умолчанию — 4 потока.
<b>task.silent.poolSize.max</b>	<b>-1</b>	Ограничение по числу используемых ядер для работы фоновых задач. По умолчанию, используется 2/3 от всех ядер сервера.
<b>task.silent.soapStats.minRequestCounts</b>	<b>other: 50</b>	Минимальное количество вызовов для мониторинга 90-перцентилей. Если количество запросов меньше указанного, то даже если они будут долгими, этот факт будет проигнорирован.  По умолчанию — 50 вызовов для всех методов.  Можно указывать для каждого метода свое значение (через разделитель ";").
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center; margin: 0;">Пример</p> <p style="text-align: center; margin: 0;"><b>getByHID:40; other:21; customSave:25</b></p> </div>		



		<p>⚠ Указывать только название метода, вариант <b>"PartyWS.search"</b>, дефолтное ограничение <b>"other"</b>, без него загружается.</p>
<b>task.silent.soapStats.percentile90</b>	<b>other: 5s</b>	<p>Показания 90-перентиля для каждого используемого метода. По умолчанию — 5 секунд.</p> <p>Можно указывать для каждого метода свое значение (через разделитель ";").</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; text-align: center;"> <p>Пример</p> <p><b>getByHID:10s; other:5s</b></p> </div> <p>⚠ Указывать только название метода, вариант <b>"PartyWS.search"</b> не будет работать, также <u>обязательно</u> указывать дефолтное ограничение <b>"other"</b>, без него загружается.</p>
<b>task.silent.soapStats.periodSec</b>	<b>300</b>	Частота снятия статистики по онлайн нагрузке. По умолчанию снимается 1 раз в 5 минут.
<b>task.silent.soapStats.seriesLength</b>	<b>5</b>	Количество временных интервалов по которому принимается решение об онлайн нагрузке.
<b>task.silent.nodeSync.queueSizeLimit</b>	<b>1000</b>	Максимальный размер очереди синхронизации между нодами горячего резерва, после превышения которого фоновые задачи начинают сбрасывать количество потоков.
<b>task.profiling.auto.enabled</b>	<b>false</b>	Включение или выключение режима автоматического запуска профилировки
<b>task.profiling.auto.length</b>	<b>15</b>	Длительность работы автоматической профилировки
<b>task.profiling.auto.seriesLength</b>	<b>5</b>	Количество подряд сработавших предупреждений о медленной работе SOAP методов, после которого запускается автоматическая профилировка
<b>task.profiling.auto.soapStats.periodSec</b>	<b>300</b>	Частота снятия статистики по онлайн нагрузке для определения необходимости запуска профилировки
<b>task.profiling.lifetime.days</b>	<b>2</b>	Количество дней, после которых будут удаляться файлы профилировки
<b>task.profiling.cleaner.auto.enabled</b>	<b>true</b>	Включение или выключение автоматического удаления устаревших файлов профилировки

# 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

## 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.

# 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
<b>DeDuplicationTask.PHYSICAL.in.fileSize</b>	Размер IN файла, в который выгружаются данные по контрагентам для отправки в Фактор
<b>DeDuplicationTask.PHYSICAL.out.fileSize</b>	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**).

# ПАРАМЕТРЫ ЗАПУСКА JBOSS

При старте система проверяет наличие следующих параметров запуска для JVM:

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

# 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	21839

[Сохранить](#)

Название параметра	Значение по умолчанию	Описание
<b>oldMaxId</b>	0	Последний (максимальный) идентификатор

# ПАРАМЕТРЫ ИМПОРТА СПРАВОЧНИКОВ

Название параметра	Значение по умолчанию	Описание
<b>dictionary.conf.egr.date</b>		Дата актуальности загруженного справочника ЕГРЮЛ
<b>dictionary.conf.egr.hash</b>		Контрольная сумма загруженного справочника ЕГРЮЛ


# ПАРАМЕТРЫ МОДУЛЯ МОНИТОРИНГА



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

Название параметра	Значение по умолчанию	Описание
<b>monitoring.applicationHealth.badHealthRefreshPeriodInSeconds</b>	600	Как часто (в секундах) логировать состояние приложения, находящегося в критическом состоянии.
<b>monitoring.applicationHealth.badHealthThreshold</b>	0.95	Если отношение занятой памяти к общему её объему больше чем установленное значение, состояние приложения считается критическим.
<b>monitoring.applicationHealth.goodHealthThreshold</b>	0.85	Если отношение занятой памяти к общему её объему меньше чем установленное значение, состояние приложения считается хорошим.
<b>monitoring.applicationHealth.refreshPeriodInSeconds</b>	10	Как часто (в секундах) проверять состояние приложения.
<b>monitoring.applicationHealth.http.httpPort</b>	8080	Http-порт, на котором ведется статистика активных подключений.
<b>monitoring.applicationHealth.http.httpsPort</b>	8443	Https-порт, на котором ведется статистика активных подключений.
<b>monitoring.applicationHealth.http.connectionsToLogThreshold</b>	100	Число активных подключений, при превышении которого в лог будет записываться информация о размере входящей очереди сообщений.
<b>monitoring.applicationHealth.http.workerQueueSizeToLogThreshold</b>	10	Число очередей приложения, при превышении которого в лог будет записываться информация о размере входящей очереди сообщений.
<b>monitoring.configuration.getLocalHostTimeout</b>	1s	Время идентификации имени машины, на которой разворачивается «Единый клиент». При превышении в лог пишется сообщение уровня ERROR.
<b>monitoring.nodeSync.jmsLatencyThresholdMs</b>	5000	Время (в миллисекундах) задержки сообщений в очереди . При превышении в лог пишется сообщение уровня
<b>monitoring.nodeSync.loggingPeriodSec</b>	300	Как часто (в секундах) логировать размер очереди сообщений между нодами в ГР и время задержки сообщений в очереди.
<b>monitoring.applicationHealth.queueSizeAlert.alertPeriodSec</b>	86400 (24 часа)	Периодичность логирования в лог предупреждения о размере очереди
<b>monitoring.applicationHealth.queueSizeAlert.queueSizeThreshold</b>	30000	Порог отправки почтового уведомления о превышении размера очереди синхронизации нод
<b>monitoring.jms.loggingPeriodSec</b>	60	Как часто (в секундах) логировать статистику отправки событий в обратный поток.

# 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
<b>Общие параметры</b>		
<b>mail.from</b>	<b>automation@hflabs.ru</b>	Имя почтового ящика, с которого будет производиться рассылка
<b>mail.host</b>	<b>smtp.gmail.com</b>	Адрес <b>smtp</b> сервера, с которого будет производиться рассылка
<b>mail.debug</b>	<b>false</b>	Флаг, отвечающий за отправку сообщений <b>DEBUG</b> -уровня
<b>mail.user</b>	<b>automation@hflabs.ru</b>	Логин для подключения к серверу рассылки
<b>mail.password</b>	<b>***</b>	Пароль для подключения к серверу рассылки
<b>mail.port</b>	<b>-1</b>	Порт для отправки почты (если отличается от умолчательного для протокола)
<b>mail.protocol</b>	<b>smtps</b>	Протокол для отправки почты ( <b>smtp</b> / <b>smtps</b> )
<b>Диагностическая информация</b>		
<b>mail.diagnostic.enabled</b>	<b>false</b>	Флаг, отвечающий за отправку диагностики на почту
<b>mail.diagnostic.to</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") собранной диагностики
<b>mail.diagnostic.subject</b>	Единый клиент <b>{nodeld}</b> - диагностика	Формат заголовка сообщения о собранной диагностике
<b>mail.diagnostic.body</b>	Профили: <b>{profiles}</b> . Параметры: <b>{parameters}</b> .	Содержание сообщения о собранной диагностике
<b>mail.diagnostic.defaultProfiles</b>	<b>config,db.connection,logs,search,sql.counts,sql.metadata,versions</b>	Умолчательный набор профилей
<b>Автоматическая миграция приложения</b>		
<b>mail.migration.enabled</b>	<b>false</b>	Флаг, отвечающий за отправку оповещений миграции на почту
<b>mail.migration.to</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") оповещений миграции
<b>mail.migration.</b>	Единый клиент <b>{nodeld}</b>	Формат заголовка оповещения о перестроении индексов



<b>subject. rebuildIndexes</b>	обновляется на версию <b>{newVersion}</b>	
<b>mail.migration. body. rebuildIndexes</b>	Обновление с версии <b>{oldVersion}</b> на <b>{newVersion}</b> . Будут перестроены индексы ( <b>{indexes}</b> ).	Содержание оповещения о перестроении индексов
<b>mail.migration. subject. diagnostic</b>	Единый клиент <b>{nodeId}</b> обновился на версию <b>{newVersion}</b>	Формат заголовка оповещения об окончании миграции
<b>mail.migration. body.diagnostic</b>	Обновление с версии <b>{oldVersion}</b> на <b>{newVersion}</b> . Диагностика во вложении ( <b>{profiles}</b> ).	Содержание оповещения об окончании миграции

#### Автоматический мониторинг логов

<b>mail.logging. enabled</b>	<b>false</b>	Флаг, отвечающий за отправку сообщений об ошибках на почту
<b>mail.logging.to</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") сообщений об ошибках
<b>mail.logging. subject.single</b>	<b>CDI Error Occurred @ \${cdi.customer.name} ({errorName})</b>	Формат заголовка сообщения об одиночной ошибке
<b>mail.logging. subject.multiple</b>	<b>CDI Errors Occurred @ \${cdi.customer.name} during {reportDate} ({lastTime} - {reportTime})</b>	Формат заголовка агрегированного сообщения об ошибках, произошедших за определенный период
<b>mail.logging. bufferSize</b>	<b>10</b>	Количество блоков информации, предшествующих ошибке
<b>mail.logging. accumulatePeriod</b>	<b>3600</b>	Частота оповещения в секундах
<b>mail.logging. skipEventPredicate1(2-5)</b>	<b>errorClass.equals ('IncorrectVersionException')</b>	Фильтр событий ошибок к отправке по почте. Если ошибка попадает под шаблон, то оповещение о ней не будет выслано.



Изменения данных параметров вступает в силу без рестарта ЕК.

#### Оповещение о необходимости проверить связи с ЧС

<b>mail.notify. blacklist. enabled</b>	<b>false</b>	Флаг, отвечающий за отправку оповещений на почту
<b>mail.notify. blacklist. partyLinksCountThreshold</b>	<b>100</b>	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.
<b>mail.notify.blacklist.subject</b>	Найдены клиенты в черном списке @ {nodeId}	Формат заголовка сообщения о найденных в ЧС контрагентах
<b>mail.notify.blacklist.to</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") сообщений
<b>Запросы в поддержку</b>		
<b>mail.supportRequest.subject</b>	Единый клиент - запрос в поддержку	Формат заголовка обращения в поддержку
<b>mail.supportRequest.to</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") запросов в поддержку
<b>Оповещение о загрузке ЧС</b>		
<b>mail.import.checkList.stat.enabled</b>	<b>false</b>	Включение/отключение отправки писем по загрузке ЧС задачей listFilesImport
<b>mail.import.checkList.stat.addresses</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") писем по загрузке ЧС
<b>Оповещение о результатах сверки с ЧС (только для модуля Журналы ЧС)</b>		
<b>mail.checkByList.stat.enabled</b>	<b>false</b>	Включение/отключение отправки писем по результатам сверки с ЧС
<b>mail.checkByList.stat.addresses</b>	<b>log.catcher@hflabs.ru</b>	Список получателей (разделитель - ";") писем по результатам сверки с ЧС

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 **{nodeId}** - диагностика.
- В **standalone.conf.bat** задан параметр **-Dcdi.app.nodeId=MY\_NODE\_1**.
- Тема отправленного письма — «Единый клиент MY\_NODE\_1 - диагностика».

# МЕХАНИЗМ ОТПРАВКИ ПИСЕМ ОБ ОШИБКАХ НА ПОЧТУ

## *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 приходиться 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.

## *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:

<b>NotAuthorizedLocalUserAlert</b>	1
<b>IllegalArgumentException</b>	1
<b>AccessDeniedException</b>	1

## 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:

<b>NotAuthorizedLocalUserAlert</b>	1
<b>IllegalArgumentException</b>	1
<b>AccessDeniedException</b>	1

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

<b>NotAuthorizedLocalUserAlert</b>	1
<b>AccessDeniedException</b>	2

# ПАРАМЕТРЫ ПОДКЛЮЧЕНИЯ К ФАКТОРУ




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


Название параметра	Значение по умолчанию	Описание
<b>factor.blacklist.cdi.context</b>	<b>/factor-service-{customer}/services/CdiBlackListService</b>	WSDL сервиса поиска по Черным спискам
<b>factor.blacklist.cdi.timeout</b>	<b>600000</b>	Тайм-аут ожидания ответа от Фактора во время поиска по Черным спискам, в миллисекундах
<b>factor.clean.batch.max</b>	<b>-1</b>	Максимальное количество записей в одном сообщении при отправке записей на очистку в Фактор, по умолчанию количество неограничено
<b>factor.clean.context</b>	<b>/factor-service-{customer}/services/CleanService</b>	WSDL <b>CleanService</b>
<b>factor.clean.packetSize</b>	<b>200</b>	Размер пакета на стандартизацию / трансформацию
<b>factor.clean.timeout</b>	<b>1200000</b>	Тайм-аут ожидания ответа от Фактора во время стандартизации / трансформации (для каждого пакета), в миллисекундах
<b>factor.dedup.context</b>	<b>/factor-service-{customer}/services/DedupService</b>	WSDL сервиса поиска дубликатов
<b>factor.dedup.full.threadsCount</b>	<b>-1</b>	Количество потоков для выполнения полной дедубликации в Факторе (автоматически, если указан "-1"). Эти же настройки используются для оффлайн проверки по ЧС / ЕГРЮЛ.
<b>factor.dedup.full.threadsPriority</b>	<b>-1</b>	Приоритет потоков для выполнения полной дедубликации в Факторе (автоматически, если указан "-1"). Эти же настройки используются для оффлайн проверки по ЧС / ЕГРЮЛ. Все приоритеты настраиваются в диапазоне [1, 10], приоритет по умолчанию (если не указан или указан отрицательный): 5
<b>factor.dedup.incremental.queueSize</b>	<b>10000</b>	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)
<b>factor.dedup.timeout</b>	<b>600000</b>	Тайм-аут ожидания ответа от Фактора во время поиска дубликатов, в миллисекундах
<b>factor.factor.context</b>	<b>/factor-service-{customer}/services/FactorService</b>	WSDL Фактора
<b>factor.factor.timeout</b>	<b>600000</b>	Тайм-аут ожидания ответа от Фактора

<b>factor.health.checkIndex</b>	<b>true</b>	<p>Включение/отключение проверки валидации индексов и хэшей дедапа в хэлс-чеке</p> <p>✔ Изменение данного параметра применяется без рестарта модуля CDI</p>
<b>factor.health.timeoutMs</b>	<b>1000</b>	<p>Время ожидания ответа на проверку хэлс-чека Фактора</p> <p>✔ Изменение данного параметра применяется без рестарта модуля CDI</p>
<b>factor.host</b>	<b>http://host:port</b>	<p>Базовый URL Фактора в формате <b>http://host:port</b> (используется для поиска дубликатов)</p>
<b>factor.hosts</b>	<b>http://host:port</b>	<p>Перечень URL Фактора для пакетной очистки (может быть несколько), разделитель - ";".</p> <p>Если на конкретный фактор требуется отправлять данные в несколько потоков, то после его URL следует указать "[<b>количество_потоков</b>]".</p> <p>Например:</p> <div style="border: 1px dashed blue; padding: 5px; display: inline-block; margin: 10px 0;"> <b>http://host1:18080[2]; http://host2:18080</b> </div> <p>означает, что 2 потока будут посылать запросы на <b>http://host1:18080</b> и 1 на <b>http://host2:18080</b>.</p> <p>При включенном factor.ws.useLoadBalancingOnline ещё используются и для онлайн-запросов</p>
<b>factor.ws.numberOfRetries</b>	<b>1</b>	<p>Количество попыток повторного вызова Фактора при наличии сетевой ошибки</p>
<b>factor.ws.delayBetweenRetries</b>	<b>1000</b>	<p>Задержка в мс между попытками повторного вызова Фактора при наличии сетевой ошибки</p>
<b>factor.ws.chunkingMode</b>	<b>ALWAYS</b>	<p>Режим использования Chunked Encoding при обращении к серверу:</p> <ul style="list-style-type: none"> <li>• ALWAYS — всегда использовать;</li> <li>• REMOTE — использовать только для удаленных серверов;</li> <li>• NEVER — никогда не использовать.</li> </ul> <p>При возникновении ошибок "<i>Marshalling Error: Error writing request body to server</i>" рекомендуется переключить в REMOTE.</p>
<b>factor.ws.useLoadBalancingOnline</b>	<b>false</b>	<p>Включение/отключение использования нескольких Факторов из hosts для online очистки</p> <p>✔ Изменение данного параметра применяется без рестарта модуля CDI</p>


# ПАРАМЕТРЫ ПОИСКОВОГО ИНДЕКСА

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

Название параметра	Значение по умолчанию	Описание
<b>engine.rebuild.freeSpaceRatio</b>	<b>1.5</b>	Коэффициент свободного места, которое должно быть при очередном запуске задачи перестроения индекса Если значение <1, то проверка свободного места отключена
<b>engine.rebuild.pageSize</b>	<b>10000</b>	Размер страницы (количество контрагентов) при перестроении индекса
<b>engine.rebuild.poolSize</b>	<b>-1</b>	Количество потоков для перестроения индекса Если равно -1, устанавливается по формуле: $\min(\text{количество процессоров (ядер)} * 2, 8)$
<b>engine.rebuild.skipOptimize</b>	<b>false</b>	Пропускать ли оптимизацию при перестроении
<b>engine.rebuild.sortedIterator</b>	<b>true</b>	Добавлять ли сортировку в итератор по актуальным контрагентам. 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.
<b>engine.rebuild.strategy</b>	<b>BLOCKING</b>	Стратегия перестроения индекса: <ul style="list-style-type: none"><li>• <b>BLOCKING</b> — во время перестроения блокируются все модификации индекса, рядом строится новый свежий индекс, которым потом подменяем старый;</li><li>• <b>CONCURRENT</b> — состоит из двух шагов (применима только для индексов контрагентов; граф / группы дублей / конфликты перестраиваются по <b>BLOCKING</b> алгоритму):<ul style="list-style-type: none"><li>• в существующем индексе обновляются все документы, индекс оптимизируется;</li><li>• 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.</li></ul></li></ul> <p>Если у вас нет проблем с онлайн-потоками, то эффективнее использовать <b>BLOCKING</b>. 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 <b>CONCURRENT</b></p>
<b>engine.</b>	<b>1000</b>	Размер буфера коммита изменений поискового индекса на диск.

<b>search. bufferSize</b>		<p>Все операции из очереди группируются в памяти, пока не будет заполнен данный буфер.</p> <p>Буфер измеряется в контрагентах (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>
<b>engine. search. commitInterval</b>	<b>5000</b>	<p>Частота коммита изменений поискового индекса на диск.</p> <p>Все операции из очереди группируются в памяти, пока не сработает таймер коммита. Измеряется в миллисекундах. Не рекомендуется устанавливать менее 1000 мс.</p>
<b>engine. search. maxBufferedDocs</b>	<b>1000</b>	Максимальное количество документов в памяти до сброса на диск
<b>engine. search. needOptimizeWhenOpen</b>	<b>false</b>	Проводить оптимизацию индекса при старте системы?
<b>engine. search. queueSize</b>	<b>20000</b>	<p>Размер очереди операций по модификации индекса.</p> <p>Если очередь полностью заполняется, на все последующие операции модификации индекса возвращается ошибка.</p>
<b>engine. search. indexDirectoryType</b>	<b>AUTO</b>	<p>Способ чтения файлов индекса (тип директории — <b>AUTO / MMAP / SIMPLE / NIO / RAM</b>) для всех индексов, кроме групп дубликатов.</p> <p>Можно переключать с <b>AUTO</b> (который на современных серверах всегда <b>MMAP</b>) на <b>NIO</b>. Это замедлит скорость отдельно взятых процессов, но в сумме дает большую стабильность сервера и прогнозируемость.</p>
<b>engine. search. graph. indexDirectoryType</b>	<b>AUTO</b>	Способ чтения файлов индекса (тип директории — <b>AUTO / MMAP / SIMPLE / NIO / RAM</b> ) для графа связей.
<b>engine. search. groups. indexDirectoryType</b>	<b>AUTO</b>	<p>Способ чтения файлов индекса (тип директории — <b>AUTO / MMAP / SIMPLE / NIO / RAM</b>) для групп дубликатов.</p> <p>Если есть проблемы с <b>MMAP</b>, то в первую очередь стоит поменять доступ на <b>NIO</b> для групп дубликатов.</p>
<b>engine. search. groups. maxFilteredPartyCount</b>	<b>100000</b>	<p>Максимальное допустимое количество контрагентов, которое может вернуть поиск по хидам КА, инициированный фильтрацией на группах дубликатов.</p> <p>Если будет найдено больше, то ошибка: <b>Ваш запрос находит слишком много записей. Уточните условие запроса.</b></p>
<b>engine. search. supportLegacyNumericFields</b>	<b>true</b>	<p>Флаг для переключения формата полей, которые пишутся в индекс на новый Lucene 6.</p> <p>Если <b>false</b>, то все запросы начинают работать по новым полям Lucene 6, старый формат полей не пишется в индекс.</p> <p> Параметр удален в релизе 18.15.</p>
<b>engine.</b>	<b>false</b>	Флаг использования доступов через <b>NIO</b> для итераторов по всему индексу, в



<b>search. useNIOForIte rateIndex</b>		частности, при выгрузке кандидатов для поиска дубликатов.
<b>engine. validation. maxAccept ableMismatch</b>	<b>-1</b>	Максимально допустимое расхождение при валидации между индексом и БД. По умолчанию <b>-1</b> , на любое расхождение будет <b>ERROR</b> -сообщение в лог.   Параметр удален в релизе 23.14.
<b>engine. validation. maxPercent Acceptable Mismatch</b>	<b>0.01</b>	Максимально допустимое расхождение при валидации между индексом и БД. По умолчанию <b>0.01</b> , на любое расхождение будет <b>ERROR</b> -сообщение в лог.
<b>engine. search. queueWork ersCount</b>	<b>1</b>	Количество потоков, сбрасывающих информацию об изменениях контрагентов из памяти приложения в поисковый индекс.  Увеличение количества ведет к ускорению разбора очереди модификаций индекса, но генерирует повышенную нагрузку на диск.

# НЕБЛОКИРУЮЩЕЕ ПЕРЕСТРОЕНИЕ ПОИСКОВОГО ИНДЕКСА

**i engine.rebuild.strategy = CONCURRENT**

## *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.

## *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**, где PartyType имеет формат **PHYSICAL, LEGAL** и т.д.

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.**

## ***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.

## ***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**.

# ПАРАМЕТРЫ ПРОВЕРКИ В ФНС

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

Взаимодействие с ФНС происходит через коннектор (про него подробнее в [Установка и настройка fns-connector](#)).

Настройки соединения с коннектором:


Название параметра	Значение по умолчанию	Описание
<b>fns.connector.endpoint</b>	http://localhost:1111	URL коннектора.
<b>fns.connector.auth.username</b>		Логин для авторизации к коннектору.
<b>fns.connector.auth.password</b>		Пароль для авторизации к коннектору. Не шифруется.
<b>fns.connector.connection.timeoutMs</b>	60000	Время ожидания установки соединения с коннектором (в миллисекундах).
<b>fns.check.supportedDocumentTypes</b>	PASSPORT_RU: 21; OVERSEAS_PASSPORT_RU: 22; BIRTH_CERTIFICATE: 03; PASSPORT_FOREIGNER: 10; TEMPORARY_CERTIFICATE: 14; RESIDENCY: 12; PASSPORT_USSR: 01; FOREIGN_BIRTH_CERT: 23; TEMPORARY_RESIDENCE: 15; CERTIFICATE_OF_ASYLUM: 19	Список допустимых типов ДУЛ (разделитель - ";").  Указаны в порядке уменьшения приоритета.  Формат "литера ДУЛ по ЕК: код по СПДУЛ".
<b>fns.connector.batch.waitTimeOutMin</b>	5	Время ожидания результата ответа от ФНС (в минутах).
<b>fns.connector.batch.retryCount</b>	3	Число попыток отправки пачки в ФНС, если не дождалась результата.

# ПАРАМЕТРЫ СЕРВИСОВ



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

Название параметра	Значение по умолчанию	Описание
<b>services.actualityDate.truncateTime</b>	<b>true</b>	Если <b>true</b> , то для даты актуальности используется тип данных <b>DATE</b> , иначе — <b>TIMESTAMP</b> (для переключения на <b>TIMESTAMP</b> нужна перепоставка релиза).  Важно: Перед включением настройки необходимо обратиться в поддержку "Единого клиента".
<b>services.actualityDate.enableRangeCheck</b>	<b>false</b>	Если <b>true</b> , попытка обновления даты актуальности на дату из будущего упадет с ошибкой <b>Actuality date value cannot be greater then current date</b> .  ✔ Рестарт CDI не требуется
<b>services.applicationMigration.enabled</b>	<b>false</b>	Флаг включения миграции версий приложения (автозапуска задач и перестроений индексов).  На миграцию БД флаг не влияет.
<b>services.applicationMigration.rebuildInvalidIndexes</b>	<b>false</b>	Флаг автоматического перестроения невалидных индексов на миграции приложения
<b>services.cache.refreshIntervalMin</b>	<b>60</b>	Периодичность сбора кэша справочников (в минутах).
<b>services.consent.defaultValidityPeriodYears</b>	<b>-1</b>	Умолчательный срок действия согласия (для модуля ЦУС)
<b>services.dao.versioning.defaultPhase</b>	<b>—</b>	Умолчательная фаза миграции для таблиц, отсутствующих в <b>MIGRATION_PHASE</b>
<b>services.deduplication.autoMatchScope</b>	<b>98</b>	Пороговое значения коэффициента схожести для гарантированных дубликатов по умолчанию.
<b>services.deduplication.autoMatch</b>	<b>—</b>	Пороговое значения коэффициента схожести для гарантированных дубликатов для конкретных типов <b>PARTY</b> (заполняется только в случае, если значения разные, например - <b>PHYSICAL:85;LENDER:95</b> ).

<b>ScopeByPartyType</b>		
<b>services.deduplication.byPartyHidsFullScalIteratorThreshold</b>	<b>200000</b>	Максимальный размер контрагентов для итерирования фулсканом по дубликатам при инкрементальной загрузке
<b>services.deduplication.matchScopeThreshold</b>	<b>0</b>	Минимальный порог схожести дубликатов для создания дубликата (все, что ниже — уже не дубли, а домохозяйства)
<b>services.deduplication.matchScopeThresholdByPartyType</b>	<b>PHYSICAL : 50</b>	Минимальный порог схожести дубликатов для создания дубликата по типам контрагентов
<b>services.duplicateGroup.bigGroup.duplicatesCountToLoad</b>	<b>10000000</b>	Максимальное количество дубликатов, загружаемое для большой группы.
<b>services.duplicateGroup.bigGroup.partyCountThreshold</b>	<b>200</b>	Количество контрагентов, начиная с которого группа считается большой и обрабатывается особым образом.
<b>services.deduplication.maxDuplicatesForGroupsCalculation</b>	<b>200000</b>	Максимальный размер множества дублей, загружаемых в память, для построения группы дубликатов.
<b>services.lifecycleEvents.persistent</b>	<b>true</b>	Сохранять ли в таблице <b>logging</b> события старта-остановки приложения
<b>services.maintenance.nodeIds</b>	<b>null</b>	<p>Список идентификаторов нод, которые находятся в режиме тех. обслуживания и не принимают онлайн-запросы. Также для нод, находящихся в maintenance mode останавливаются входящие сообщения синхронизации.</p> <p>Указывать можно через запятую, если нужно вывести сразу несколько.</p> <p>Указывать нужно ip-адреса или идентификаторы нод, если они указаны в standalone.conf через <i>-Dcdi.app.nodeId=node1</i></p> <p> Чтобы балансировщик не отправлял запросы на ноду в режиме обслуживания, необходимо в его настройках для проверки доступности указать адрес api «Единого</p>

---

клиента» **http://<hostname>:<port>/cdi/api/manage/health**. В противном случае нода будет считаться активной.

Режимом тех. обслуживания можно управлять с помощью неавторизованных запросов по адресу **http://<hostname>:<port>/cdi/api/manage/maintenance**. **POST** запрос переключает ноду в maintenance mode, **DELETE** выводит ноду из этого режима, а **GET** показывает текущий статус режима тех. обслуживания.

✔ Рестарт CDI не требуется

---

<b>services.merge.attribute.mergePackSize</b>	<b>50</b>	Допустимое количество атрибутов, которые можно сливать "каждый с каждым" (при превышении делить на группы)
<b>services.merge.findCycles</b>	<b>true</b>	Если <b>false</b> , то поиск циклов не производится. Группа дубликатов отмечается как ошибочная при подозрении на цикл. Если <b>true</b> , то определяются атрибуты-победители без поиска циклов. Если в результате есть несовпадение ожидаемого и фактического количества атрибутов, происходит поиск циклов.
<b>services.party.includeSourceToOriginal</b>	<b>false</b>	Возвращать в ответах <b>SOAP-методов</b> для исходной карточки tag source?
<b>services.pastValues.enabled</b>	<b>false</b>	Включить сервис использования прошлых атрибутов для поиска дублей
<b>services.personalData.mode</b>	<b>DISABLED</b>	<ul style="list-style-type: none"><li>• <b>DISABLED</b> – логируем все в основной лог, ничего не скрываем</li><li>• <b>SEPARATE</b> – логируем перс.данные в отдельный лог cdi-personal-data.log, а в основном ставим ссылку, чтобы можно было найти</li><li>• <b>HIDDEN</b> – в основном логе перс.данные скрываем, и никуда их не пишем (paranoid mode)</li></ul>
<b>services.relation.enabled</b>	<b>false</b>	Включить сервис расчета графа взаимосвязей.
<b>services.relations.fillSourceIdByDefault</b>	<b>false</b>	Флаг включения расширенного оповещения JMS очереди о создании связей. Если <b>true</b> , то в сообщении о создании/удалении связи для исходных карточек помимо <b>hid</b> будет указываться <b>rawId</b> и <b>sourceSystem</b> . Для золотых карточек и вспомогательных сущностей (например, Сегментов) всегда будет указан только <b>hid</b> . Важно: Включение данной настройки может увеличить скорость выполнения запросов.
<b>services.relation.maxRelatedEssencesCount</b>	<b>100</b>	Максимальное число связанных сущностей, которое уходит в индекс
<b>services.sql.paramsLoggingEnabled</b>	<b>true</b>	Флаг записи параметров sql-запросов в cdi.log.
<b>services.sqlProfiler</b>	<b>60</b>	Пороговое значение для логирования sql-запросов. Запрос, выполняющийся дольше установленного значения, будет сохранен в логе.

---

<b>thresholdIn Seconds</b>		✓ Рестарт CDI не требуется
<b>services. state. mapdb. useMmap</b>	<b>true</b>	Использовать mmap для работы с локальной базой (файловая), в которой хранятся даты последнего обновления индекса.  Если <b>false</b> , то mmap не используется — используется обычный доступ к файлам
<b>services. suggestion s.enabled</b>	<b>false</b>	Флаг включения подсказок для адресов
<b>services. suggestion s.token</b>	<b>{api-ключ}</b>	Ключ доступа к облачным подсказкам
<b>services. suggestion s. upperCase</b>	<b>true</b>	Флаг перевода значений из подсказок в верхний регистр при подстановке
<b>services. suggestion s.url</b>	<b>https://data.ru/api/v2</b>	<b>URL</b> сервиса подсказок. При использовании <b>standalone</b> -подсказок заменить на: <b>http://{ip}:{port}/suggestions/api/4_1/rs.</b>
<b>services. transform. closeMissedAttributes</b>	<b>false</b>	Закрывать ли при обновлении атрибуты, которые отсутствуют в обновлении?
<b>services. transform. updateConflictStrategy</b>	<b>DISABLED</b>	Стратегия мониторинга конфликтующих изменений. Для подключения выставить в значение <b>SKIP_UPDATE_AND_SAVE</b> .
<b>services. transform. updateConflictStrategy ByPartyType</b>		Стратегия мониторинга конфликтующих изменений по типам контрагентов. Формат: «PARTY_TYPE: STRATEGY»
<b>services. rules. update.log</b>	<b>false</b>	Флаг включения сработавших правил обновления  ✓ Рестарт CDI не требуется
<b>services. rules. merge.log</b>	<b>false</b>	Флаг включения сработавших правил слияния  ✓ Рестарт CDI не требуется
<b>services. rules. skipLoggingForRules</b>	<b>oldWins, reqIfEqualsThenOldWins</b>	Перечень событий (правил), которые не нужно логировать  ✓ Рестарт CDI не требуется
<b>services. duplicateGroup. actualFilteringMode</b>	DAO	Режим фильтрации при отборе дубликатов: <ul style="list-style-type: none"> <li>• DAO — через БД.</li> <li>• GRAPH — через граф связей. Перестроение групп дублей работает быстрее, чем при фильтрации через БД.</li> <li>• DISABLED — фильтрация отключена. Есть вероятность отобразить неактуальный дубликат.</li> </ul>



<b>services.party.massive.sourceCountThreshold</b>	1000	Параметр, определяющий, что карточка большая
<b>services.view.maxAllowedContractFilesCount</b>	10	Количество отображаемых ссылок с каждого контракта. ✔ Ресурсы CDI не требуется
<b>services.relation.maxOriginalNodesCount</b>	2000	Максимальное количество исходных КА, связи которых отразятся в UI ✔ Ресурсы CDI не требуется
<b>services.relation.requestedCountMultiplier</b>	1.5	Коэффициент для запрашиваемого числа связей. После схлопывания оставит количество связей не менее указанного. Работает в паре с services.relation.maxOriginalNodesCount ✔ Ресурсы CDI не требуется
<b>mail.notify.sendPeriodInMinutes</b>	1	Интервал оповещения об изменениях FATCA/CRS (по умолчанию 1 минута) ✔ Ресурсы CDI не требуется
<b>services.metrics.source.quality.enabled</b>	<b>false</b>	Подключение сбора метрик качества данных ✔ Ресурсы CDI не требуется
<b>services.maintenance.nodeIds.sync</b>	<b>false</b>	Для нод, находящихся в maintenance mode останавливаются входящие сообщения синхронизации.  Чтобы включить синхронизацию нод в режиме в maintenance mode сделали отдельный режим "обслуживание с синхронизацией нод", для его включения необходимо отправить POST запрос:  <b>http://&lt;hostname&gt;:&lt;port&gt;/cdi/api/manage/sync</b> . В противном случае нода будет считаться неактивной и запросы с балансировщика не будут проходить. <b>{"enabled":"true"}</b> – все штатно работает в обычном режиме синхронизации.  <b>GET</b> запрос показывает текущий статус синхронизации ноды в maintenance mode, <b>DELETE</b> выводит ноду из этого режима. <b>⚠</b> Режим работает только в паре с <b>services.maintenance.nodeIds</b>
<b>services.kyc.physicalRiskRule</b>		Факторы, влияющие на уровень риска ФЛ.  Формат заполнения поля:  <b>&lt;Уровень риска 1&gt;:&lt;Номер правила 1&gt;, &lt;Номер правила 2&gt;...&lt;Номер правила N&gt;/&lt;Литера списка 1&gt;,&lt;Литера списка 2&gt;...&lt;Литера списка M&gt;; &lt;Уровень риска 2&gt;:&lt;Номер правила 1&gt;, &lt;Номер правила 2&gt;...&lt;Номер правила N&gt;/&lt;Литера списка 1&gt;,&lt;Литера списка 2&gt;...&lt;Литера списка M&gt;</b>

---

**Уровень риска** — литера из справочника Уровни риска. Настройки возможны только для высокого (**HIGH**) и среднего (**MEDIUM**) уровня риска. В случае отсутствия факторов, влияющих на оценку степени (уровня) риска клиента, клиенту присваивается «низкий» уровень.

**Номер правила** — идентификатор [правила проверки по черным спискам ФЛ](#), совпадение по которому должно приводить к назначению соответствующего уровня риска.

**Литера списка** — код исходной системы, от имени которой загружаются списки в CDI. При выявлении ФЛ в упомянутом списке по правилу, приведенному до "/" клиенту назначается соответствующий уровень риска. Для таких перечней как Недействительные паспорта и список стран и территорий ФАТФ должны использоваться наименования INVALID\_PASSPORT и FATF\_BLACKLIST\_ADDRESS, FATF\_BLACKLIST\_CITIZENSHIP соответственно.

Пример: **HIGH:1,2,3,4,5,11/TERRORIST,OON,MVK; MEDIUM:1,2,3,4,5,11,12,14/IPDL,RPDL,MPDL,FATF\_BLACKLIST\_ADDRESS,INVALID\_PASSPORT**

✔ Рестарт CDI не требуется

---

services.  
kyc.  
legalRiskRule

Факторы, влияющие на уровень риска ЮЛ.

Формат заполнения поля:

**<Уровень риска 1>:<Номер правила 1>, <Номер правила 2>...<Номер правила N>/<Литера списка 1>,<Литера списка 2>...<Литера списка M>; <Уровень риска 2>:<Номер правила 1>, <Номер правила 2>...<Номер правила N>/<Литера списка 1>,<Литера списка 2>...<Литера списка M>**

**Уровень риска** — литера из справочника Уровни риска. Настройки возможны только для высокого (**HIGH**) и среднего (**MEDIUM**) уровня риска. В случае отсутствия факторов, влияющих на оценку степени (уровня) риска клиента, клиенту присваивается «низкий» уровень.

**Номер правила** — идентификатор [правила проверки по черным спискам ЮЛ](#), совпадение по которому должно приводить к назначению соответствующего уровня риска.

**Литера списка** — код исходной системы, от имени которой загружаются списки в CDI. При выявлении ЮЛ в упомянутом списке по правилу, приведенному до "/" клиенту назначается соответствующий уровень риска.

Пример: **HIGH:1,2,3,4,5,11/TERRORIST,OON,MVK; MEDIUM:1,2,3,4,5,11,12,14/EU**

✔ Рестарт CDI не требуется

---

# СТРАТЕГИЯ РЕШЕНИЯ КОНФЛИКТОВ

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

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

После смены стратегии перезапустите «Единый клиент».

# ПАРАМЕТРЫ СИНХРОНИЗАЦИИ НОД



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

Параметры синхронизации нод

Название параметра	Значение
nodeSync.jms.enabled	true
nodeSync.jms.queue.name	cdi.nodeSync
nodeSync.jms.receive.connectionFactoryJndiName	nodeSyncRemoteConnectionFactory
nodeSync.jms.receive.maxInterval	300000
nodeSync.jms.receive.password	cdi
nodeSync.jms.receive.recoveryInterval	30000
nodeSync.jms.receive.username	sync
nodeSync.jms.send.connectionFactoryJndiName	jmsConnectionFactory

Сохранить

Название параметра	Значение по умолчанию	Описание
<b>nodeSync.jms.enabled</b>	<b>true</b>	Флаг включения синхронизации между нодами
<b>nodeSync.jms.queue.name</b>	<b>cdi.nodeSync</b>	Название очереди
<b>nodeSync.jms.send.connectionFactory.jndiName</b>	<b>jmsConnectionFactory</b>	Название JMS-фабрики для отправки сообщений на вторую ноду
<b>nodeSync.jms.receive.connectionFactory.jndiName</b>	<b>nodeSyncRemoteConnectionFactory</b>	Название JMS-фабрики для получения сообщений со второй ноды
<b>nodeSync.jms.receive.username</b>	<b>sync</b>	Логин авторизации для соединения с очередью второй ноды
<b>nodeSync.jms.receive.password</b>	<b>cdi</b>	Пароль авторизации для соединения с очередью второй ноды
<b>nodeSync.jms.receive.recoveryInterval</b>	<b>30000</b>	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).
<b>nodeSync.jms.receive.maxInterval</b>	<b>300000</b>	Максимальный интервал ожидания между попытками возобновить соединение (в миллисекундах).

# ПАРАМЕТРЫ ЭКСПОРТА ДАННЫХ

## Параметры экспорта данных

Название параметра	Значение
incrementalExportRelations.lastDate	<input type="text" value="10.02.2015 02:23:47"/>
<input type="button" value="Сохранить"/>	

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

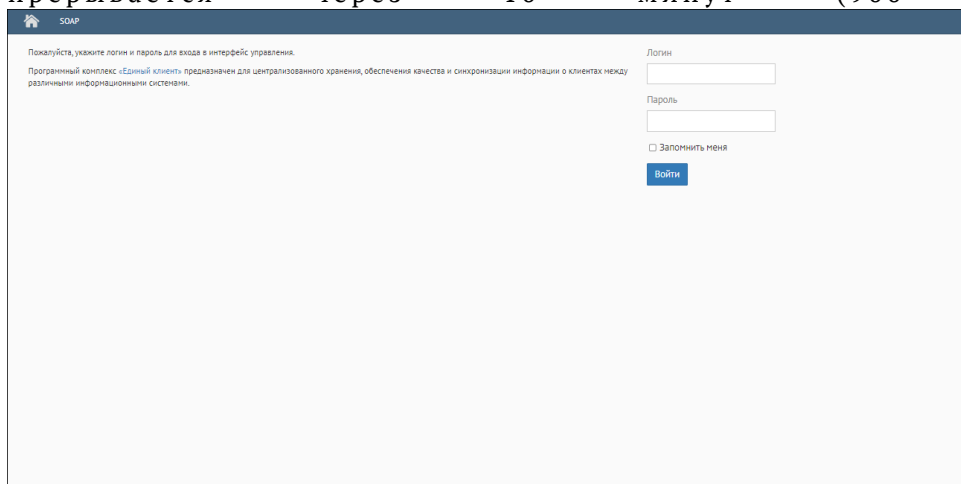
При запуске задачи *fullExportRelations* (Полный экспорт эталонных связей) параметр сбрасывается.

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

# СЕССИИ ПОЛЬЗОВАТЕЛЕЙ

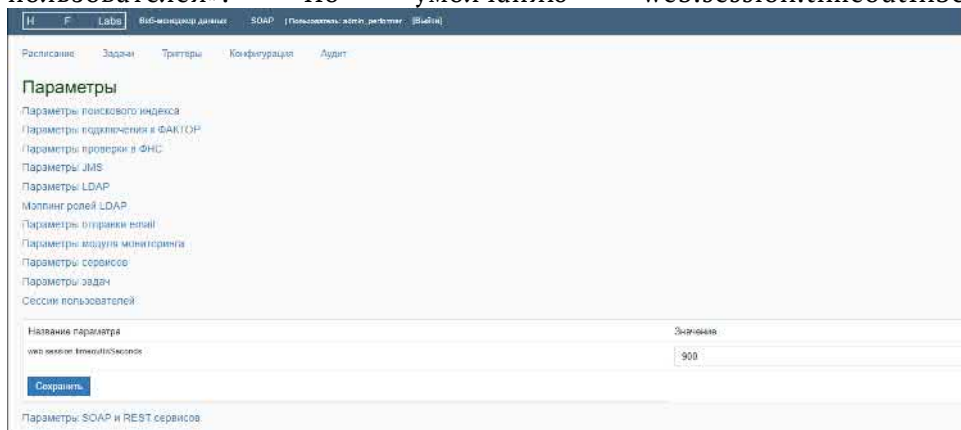
**⚠** Изменения настроек вступают в силу только после рестарта сервиса «Единого клиента»!

- Если снят чекбокс «Запомнить меня», то сессия пользователя при неактивности прерывается через 10 минут (900 сек).



The screenshot shows the SOAP login page. At the top, there is a header with a home icon and the text 'SOAP'. Below the header, there is a message: '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.' To the right of the message, there are two input fields for 'Логин' (Login) and 'Пароль' (Password). Below these fields is a checkbox labeled 'Запомнить меня' (Remember me) and a blue button labeled 'Войти' (Login).

- Время таймаута задается в параметре `web.session.timeoutInSeconds` в блоке «Сессии пользователей». По умолчанию `web.session.timeoutInSeconds=900`




The screenshot shows the SOAP configuration page. At the top, there is a header with a home icon, a user icon, and the text 'LabS'. Below the header, there is a navigation menu with items: 'Расписание', 'Задачи', 'Триггеры', 'Конфигурация', and 'Аудит'. The main content area is titled 'Параметры' (Parameters) and lists various configuration categories: 'Параметры поискового индекса', 'Параметры подключения к ФАКТОР', 'Параметры проверки в ФНС', 'Параметры JMS', 'Параметры LDAP', 'Мэппинг ролей LDAP', 'Параметры отправки email', 'Параметры модуля мониторинга', 'Параметры сорасов', 'Параметры задач', and 'Сессии пользователей'. The 'Сессии пользователей' section is expanded, showing a table with columns 'Название параметра' (Parameter name) and 'Значение' (Value). The parameter 'web.session.timeoutInSeconds' is listed with a value of '900'. Below the table is a blue button labeled 'Сохранить' (Save).

**⚠** Новое значение параметра применяется только после рестарта «Единого клиента»

- Если проставлен чекбокс «Запомнить меня», то это дает право работать без прерывания сессии пользователя на месяц без перелогина

# 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).

- Добавление в справочник новой записи
- Модификация существующей в справочнике записи
- Представление с содержимым справочников
- Список справочников, которые можно пополнять и редактировать
- Примеры использования процедур (Oracle)
- Примеры использования процедур (MariaDB)

## Добавление в справочник новой записи

Для добавления новых записей в справочники предназначена процедура **create\_reference\_record**.

Параметр	Обязательность	Значение по умолчанию	Описание
p_ref_type	Да		Тип справочника
p_code	Да		Литера справочного значения
p_ordinal	Да		Числовой код справочного значения
p_label	Да		Краткое описание справочного значения
p_description	Нет	null	Развернутое описание справочного значения
p_party_type	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).
p_author	Нет	HFLabs-CDI	Автор последнего изменения

## 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
p_ref_type	Да		Тип справочника
p_code	Да		Литера справочного значения
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

## Представление с содержимым справочников

Представление **reference\_view** содержит всю информацию, существующую в справочниках. Модель данных аналогична таблице [REFERENCE](#).

## Список справочников, которые можно пополнять и редактировать

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

## 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;
```

## 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);
```

# ЖУРНАЛИРОВАНИЕ И АУДИТ

- Журналы «Единого клиента»
- Журналы Фактора
- Уровни протоколирования
- Аудит

# ЖУРНАЛЫ «ЕДИНОГО КЛИЕНТА»

Журналы расположены на сервере приложений в каталоге **JBOSS\_CDI\_HOME/standalone/****log**.

Все логи кроме console.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. Не содержит стек-трейсов.	всегда при старте cdi
cdi-full.log	Основной журнал работы Единого клиента. Содержит стек-трейсы. Доступен по адресу <b>СЕРВЕР_ЭК:ПОРТ_ЭК/cdi/api/manage/logfile</b>	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-jms-event-trace.log	Журнал трассировки JMS-сообщений (входящие и исходящие). Подключается по запросу на короткое время.	по умолчанию отключен
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-minor-summary.log	Журнал количества 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 — <b>webservice.logging.filters</b>	всегда при старте cdi
cdi-lifecycle.log	Обновления версий сервиса и события старта/стопа приложения. Лог не пересоздается.	При первом старте cdi

<code>cdi-personal-data.log</code>	Журнал-дополнение к основному логу, в котором сохраняются персональные данные с целью его деперсонализации. Не отправляется в рамках задачи сбора диагностики.	всегда при старте cdi
<code>cdi-task.log</code>	Журнал, в который копируется вся информация о работе задач из <code>cdi.log</code>	всегда при старте cdi
<code>cdi-distributed-stats.log</code>	Журнал метрик размера очереди и задержки сообщений в очереди между нодами в горячем резерве.	всегда при старте cdi
<code>cdi-stats-avg.log</code>	Журнал производительности	всегда при старте cdi
<code>cdi-distributed-state.log</code>	Журнал информационных сообщений, относящихся к индексам. Сообщения вынесены из основного лога.	всегда при старте cdi
<code>cdi-ui-longops.log</code>	Журнал трассировки долгих запросов веб-интерфейса. Настройка критичного времени ответа — <code>webservice.logging.longops.ui.criticalTimeThresholdInMillis</code>	всегда при старте cdi
<code>cdi-ui-stats.log</code>	Журнал статистики запросов веб-интерфейса	всегда при старте cdi
<code>cdi-stats-datasource.log</code>	Журнал доступности базы данных	всегда при старте cdi
<code>cdi-stats-system.log</code>	Общая информация о системе. При запуске cdi создаётся также журнал <code>cdi-stats-system-&lt;PID&gt;.log</code> , где PID является системным идентификатором процесса cdi.	всегда при старте cdi
<code>cdi-stats-cputop.log</code>	Журнал загрузки центрального процессора.	всегда при старте cdi
<code>cdi-migration.log</code>	Журнал статистики миграции БД.	всегда при старте cdi
<code>cdi-stats-jms.log</code>	Журнал статистики ОП	всегда при старте cdi если есть ОП

# ЖУРНАЛ (ЛОГ) ДОСТУПНОСТИ БАЗЫ ДАННЫХ

## cdi-stats-datasource.log

В лог пишется время ответа базы данных.

Большие значения времени ping означают проблемы с доступностью базы данных.

При недоступности бд или таймауте в пять секунд значение пинга будет равно -1.

### Формат

```
{datetime} {ping, ms}
```

### Описание колонок

Название	Значение	Описание
datetime	datetime	Дата и время логирования
ping	ms	Время ответа базы данных

### Пример

```
datetime          ping, ms
2020-03-12 '21:45' 122
2020-03-12 '21:50' 45
```

### Настройки

По умолчанию логируется каждые 10 секунд.

# ЖУРНАЛ (ЛОГ) ПРОИЗВОДИТЕЛЬНОСТИ



Данный лог доступен в версии «Единого клиента» 20.9 и выше.

## cdi-stats-avg.log

Статистика как общей, так и создаваемой «Единым клиентом» текущей нагрузки на CPU, память, диск и сеть для первичной диагностики проблем, связанных с производительностью:

- нагрузка на процессор;
- занятость памяти;
- нагрузка на диск;
- нагрузка на сеть.

## Описание параметров

название параметра	единица измерения	описание	возможные проблемы/рекомендации
Дата и время в формате: уууу- ММ-dd 'НН:мм:ss'			
cpu_proc	от 0 до 1, где 1 — это 100%	нагрузка на cpu от ЕК	
cpu_user	от 0 до 1, где 1 — это 100%	Нагрузка на cpu от всех приложений, работающих на сервере	Если на сервере стоит только ЕК и показатель > 1%, проблемы с нагрузкой от других приложений, подключать админов со стороны заказчика.  Если стоят ЕК+Фактор или Фактор+Подсказки, то анализировать ситуацию по всем логам.
cpu_sys	от 0 до 1, где 1 — это 100%	Нагрузка на cpu на обслуживание переключений между задачами	
cpu_io	от 0 до 1 где 1 — это 100%	Нагрузка на cpu на обслуживание ввода-вывода	
cpu_irq	от 0 до 1, где 1 — это 100%	Нагрузка на cpu на прерывания	
cpu_steal	от 0 до 1, где 1 — это 100%	Нагрузка на cpu на обслуживание виртуализации (только для виртуальных машин linux)	Проблемы с виртуализацией, если в динамике показатель больше 0, подключать админов со стороны заказчика.
gc_pause	миллисекунды	Максимальные паузы GarbageCollector	Не справляется GC, если он растет в динамике или показатель больше 1000

files	число дескрипторов	Количество открытых файловых дескрипторов Фактором	
threads	число потоков	Количество активных потоков, которые использует ЕК	
jvm_eden	ГБ	Динамическая память, которая используется «Единым клиентом»	Проблемы с утилизацией памяти, если в динамике на графиках не получается "ровная пила".
jvm_old	ГБ	Память, занятая под долгосрочные объекты (справочники, фильтры, классы и т.д.)	
mem_rss	ГБ	Кол-во памяти, выделенное приложению, которое реально находится в оперативной памяти.	
mem_vss	ГБ	Кол-во используемой виртуальной памяти.  Используется для люсиновских индексов, хранения хешей (зависит от реализации).	
mem_swap	ГБ	Сколько памяти находится в swap-разделе	
disk_q	число	Длина очереди запросов к диску ( I/O's в ходе выполнения).	
disk_bi	килобайты	Количество килобайт, прочитанных с дисков.	
disk_bo	килобайты		
disk_root	сколько занято места/сколько свободно (в ГБ)	Показывает значение для переменной CDI_ROOT  По умолчанию путь:	
disk_tmp	сколько занято места/сколько свободно (в ГБ)	Показывает значение для переменной CDI_TMP_PATH  По умолчанию путь:	
net_rb	байты	Получено байт за промежуток логирования	
net_re	число ошибок	Количество ошибок при получении данных через сетевой интерфейс	Если ошибок > 0:  проблемы с сетью, подключать админов со стороны заказчика.
net_sb	байты	Отправлено байт за промежуток логирования.	
net_se	число ошибок	Количество ошибок при отправке данных через сетевой интерфейс.	Если ошибок > 0:  проблемы с сетью, подключать админов со стороны заказчика.

## Как настроить

По умолчанию статистика собирается каждые 10 секунд.

### Настройка в панели администратора


`monitoring.applicationHealth.refreshPeriodInSeconds`

## Пример журнала

	<b>cpu_proc</b>	<b>cpu_user</b>	<b>cpu_sys</b>	<b>cpu_io</b>	<b>cpu_irq</b>		
<b>cpu_steal</b>	<b>gc_pause</b>	<b>files</b>	<b>threads</b>	<b>jvm_eden</b>	<b>jvm_old</b>		
<b>mem_rss</b>	<b>mem_vss</b>	<b>mem_swap</b>	<b>disk_q</b>	<b>disk_bi</b>	<b>disk_bo</b>	<b>disk_root</b>	
<b>disk_tmp</b>	<b>net_rb</b>	<b>net_re</b>	<b>net_sb</b>	<b>net_se</b>			
2020-06-03 '21:09:02'	0	0.1	0	0	0	0	0
0	4871	880	7.5	7.38	36.2	331.34	0
0	2027	1849	6253.91/7096.55	441.94/468.36	1396169	0	0
1141017	0						
2020-06-03 '21:09:12'	0	0.09	0	0	0	0	0
0	4829	888	7.86	7.46	36.2	331.34	0
1	0	174	6253.51/7096.55	441.94/468.36	1457102	0	0
1325446	0						



# ЖУРНАЛ (ЛОГ) СЕТЕВЫХ СОЕДИНЕНИЙ

 Данный лог доступен в версии «Единого клиента» 21.12 и выше.

## cdi-stats-netstat.log

Вывод в лог системной утилиты `netstat -an` раз в N секунд.

Собирается информация о соединениях системы по протоколам UDP и TCP, использующих через сокет номер порта, на котором запущен «Единый клиент».

Для диагностики проблем, связанных с сетью:

- поиск сетевых проблем;
- определение производительности сети.

## Описание параметров

значения состояния сокета	единица измерения	описание
		Дата и время в формате: <code>yyyy-MM-dd 'HH:mm:ss'</code>
<code>listen</code>	количество сокетов	Ожидает входящих соединений.
<code>syn_sent</code>	количество сокетов	Активно пытается установить соединение.
<code>syn_received</code>	количество сокетов	Идет начальная синхронизация соединения.
<code>established</code>	количество сокетов	Соединение установлено.
<code>fin_wait_1</code>	количество сокетов	Сокет закрыт; отключение соединения.
<code>fin_wait_2</code>	количество сокетов	Сокет закрыт; ожидание отключения удаленной стороны.
<code>close_wait</code>	количество сокетов	Удаленная сторона отключилась; ожидание закрытия сокета.
<code>closing</code>	количество сокетов	Сокет закрыт, затем удаленная сторона отключилась; ожидание подтверждения.
<code>last_ack</code>	количество сокетов	Удаленная сторона отключилась, затем сокет закрыт; ожидание подтверждения.
<code>time_wait</code>	количество сокетов	Сокет закрыт, но ожидает пакеты, ещё находящиеся в сети для

	сокетов	обработки.
closed	количество сокетов	Закрыт. Сокет не используется.

## Как настроить

По умолчанию статистика собирается каждые 10 секунд.

Порт для мониторинга 8080:

<b>Настройка в панели администратора</b>
<b>monitoring.netstat.port=8080</b>

## Пример журнала

	listen	syn_sent	syn_received	established	fin_wait_1	fin_wait_2
close_wait	closing	last_ack	time_wait	closed		
2021-04-01 '00:07:						
00'	1	0	0	0	0	0
	4	0				
2021-04-01 '00:07:						
10'	1	0	0	2	0	0
	4	0				

# ЖУРНАЛ (ЛОГ) СИНХРОНИЗАЦИИ НОД

## cdi-distributed-stats.log

В лог пишутся метрики работы очередей синхронизации нод.

Большие значения времени задержки `max_latency` говорят о значительной рассинхронизации состояний нод.

Большие значения размера очереди `max_queue_size` говорят о накопленном объеме не примененных изменений.

### Формат

```
{datetime} {max_queue_size} {max_latency, ms}
```

### Описание колонок

Название	Значение	Описание
<code>datetime</code>	<code>datetime</code>	Дата и время логирования
<code>max_queue_size</code>	шт	Максимальный размер очереди за период
<code>max_latency</code>	ms	Максимальное время задержки применения сообщения за период (разница между временем применения и временем создания сообщения)

### Пример

```
datetime          max_queue_size  max_latency, ms
2020-03-12 '21:45'    34              122
2020-03-12 '21:50'    2               45
```

### Настройки

По умолчанию логируется каждые 5 минут.

Частоту логирования можно изменить в параметре **`monitoring.nodeSync.loggingPeriodSec`**

Отсечка на логирование аларма в `cdi.log` **`monitoring.nodeSync.jmsLatencyThresholdMs`** — при ее превышении аларм пишется в общий `cdi.log`

# ЖУРНАЛ АУТЕНТИФИКАЦИЙ

 cdi-security.log

## 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}**
4. Попытка войти в систему под пользователем, имеющим конфликтные роли:

**{Timestamp} {Имя потока} ERROR UsernamePasswordAuthenticationFilter - An internal error occurred while trying to authenticate the user.  
ru.hflabs.cdi.exception.security.ConflictingRolesAuthenticationException: Found conflicting roles [{список ролей}] for user '{пользователь, с которого производилась попытка логина}'**

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
2017-08-24 10:35:17,641 [default task-9] ERROR UsernamePasswordAuthenticationFilter - An internal error occurred while trying to authenticate the user. ru.hflabs.cdi.exception.security.ConflictingRolesAuthenticationException: Found conflicting roles [ROLE_OPERATOR, ROLE_PERFORMER] for user 'superuser'
```

## 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.

# ЖУРНАЛ ДОСТУПА К ПЕРСОНАЛЬНЫМ ДАННЫМ

 cdi-data-access.log

 По умолчанию лог отключен. Для подключения обратитесь в поддержку

## 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}

## 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]
```

# ЖУРНАЛ ЗАГРУЗКИ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

 cdi-stats-cputop.log

## Описание журнала

Логирование CPU метрик. Этот журнал позволяет узнать загруженность системы и посмотреть соседние процессы в системе. Логирование общей информации о системе осуществляется [журналом общей информации о системе](#).

## Описание параметров

Частота и другие параметры сбора метрик и их логирования настраивается группой параметров модуля мониторинга `monitoring.log.cpu.*`.

Параметр	Описание
<code>monitoring.log.cpu.step</code>	Определяет частоту вывода метрик CPU. Измеряется в интервалах в 6 секунд – по умолчанию имеет значение 10, то есть вывод метрик происходит каждые 60 секунд.
<code>monitoring.log.cpu.preserve</code>	Количество сохраняемых метрик.
<code>monitoring.log.cpu.print</code>	Максимальное количество строк для печати метрик CPU.
<code>monitoring.log.cpu.avg.depthm</code>	Количество записей, по которым будет считаться скользящее среднее значение для метрик CPU.
<code>monitoring.log.cpu.avg.threshold</code>	Пороговое значение нагрузки ЦПУ для метрик CPU, %. Значения до этого порога игнорируются.
<code>monitoring.log.cpu.avg.multiplier</code>	Количество раз, на которое новое значение должно превысить среднее (MA), чтобы триггер сработал для метрик CPU.
<code>monitoring.log.cpu.periodical.period</code>	Частота срабатывания триггера по логированию метрик CPU.

# ЖУРНАЛ ИСКЛЮЧЕННЫХ ОШИБОК

 cdi-minor-errors.log

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**.

С релиза 20.11 «Единый клиент» научился фильтровать в лог cdi-minor-errors предупреждения – сообщения с типом WARN.

## *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 логе — **cdi.log**.



# ЖУРНАЛ КОЛИЧЕСТВА ОШИБОК, ИСКЛЮЧЕННЫХ ИЗ ОСНОВНОГО ЛОГА

 cdi-minor-summary.log

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.

Они логируются подробно в **cdi-minor-errors.log**. Правила для переноса ошибок в данный лог достаточно общие и по ним могут быть отловлены в дальнейшем не только ошибки конкретной интеграции, но и новые.

Для мониторинга количества ошибок сделан лог **cdi-minor-summary.log**.

## Особенности

Лог создается по результату работы задачи **diagnosticLogs** и не хранится на диске вместе с остальными логами.

Форматирование и информация об ошибке задается по маске:

**CODE (номер задачи в JIRA, под который зарегистрирована проблема заказчика) - количество ошибок такого типа**

Пример:

**UNKNOWN - 1\***  
**OPEN-1018 - 1**  
**OPEN-797 - 4**

\* UNKNOWN можно встретить в логе, если поменяется формат **cdi-minor-errors.log** и мы перестанем понимать код задач.

# ЖУРНАЛ ОБНОВЛЕНИЙ ВЕРСИЙ СЕРВЕРА

 cdi-lifecycle.log

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 автоматической миграции. 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 логе:

```
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

При старте и остановке сервиса помимо прочего указывается уникальный идентификатор процесса:

```
2022-08-26 12:02:05,799 INFO start - CDI application [Demo 22.13-SNAPSHOT 2022-08-26 08:
56:44(e29bf396, core 628578d3)] [127.0.0.1] started in 43 s with PID 432531
2022-08-26 12:04:24,473 INFO stop - CDI application [Demo 22.13-SNAPSHOT 2022-08-26 08:
56:44(e29bf396, core 628578d3)] [127.0.0.1] stopped with PID 432531
2022-08-26 12:06:04,620 INFO start - CDI application [Demo 22.13-SNAPSHOT 2022-08-26 08:
56:44(e29bf396, core 628578d3)] [127.0.0.1] started in 43 s with PID 433235
```

# ЖУРНАЛ С ПЕРСОНАЛЬНЫМИ ДАННЫМИ

 cdi-personal-data.log

Этот лог по умолчанию отключён, его работа регулируется параметром `services.personalData.mode`.

В общий лог (`cdi.log`) пишется, например

```
Error saving party with data #777
```

а в этот журнал пишутся персональные данные:

```
2018-07-11 13:01:32,181 INFO PersonalDataService - Personal data #777:  
FSSPSearchHelper.PhysicalData(requestId=15, fullNameRawSource=null, surname=Евгенов,  
name=Павел, patronymic=Григорьевич, birthdate=Thu Oct 08 00:00:00 GMT+03:00 1987,  
birthYear=null, inn=null, documentType=PASSPORT_RU, documentSeries=4509,  
documentNumber=531732)
```

# ЛОГИ ЗАДАЧИ ТРАНСФОРМАЦИИ

- Логирование результатов
- Задача дошла до конца без ошибок
- Задача дошла до конца, но были ошибки
- Задача остановилась из-за ошибок
- Подсчет ошибок

## 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 разливание)
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**

### **Задача дошла до конца без ошибок**

- АРМ Администратора, статус - *FINISHED*
- АРМ Администратора, комментарий - *OK*
- 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**

### **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**

### **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
```

## *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

# ЛОГ ТРИГТЕРА

- Все задачи прошли успешно
- Задача упала, триггер остановился
- Задача упала, триггер продолжил работу

## 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.
```

## 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.
```

## 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.
```

# ОБЩАЯ ИНФОРМАЦИЯ О СИСТЕМЕ

 cdi-stats-system.log  
cdi-stats-system-{PID}.log

Логирование общей информации о системе. Логирование метрик CPU осуществляется [журналом загрузки центрального процессора](#).

Название раздела	Описание
<b>Configured system properties</b>	Перечень переменных, с которыми запущена JVM.
<b>Configured GC options</b>	Настройки сборщика мусора (garbage collector) JVM.
<b>Configured JVM modules</b>	Информация для разработчиков об открытых модулях JDK.
<b>Effective context properties</b>	Информация об эффективных свойствах контекста.
<b>Operating system</b>	Информация об операционной системе: <ul style="list-style-type: none"><li>• <b>name</b> — название ОС,</li><li>• <b>uptime</b> — время работы ОС с момента последнего запуска.</li></ul>
<b>Computer</b>	Информация о компьютере, причем <code>manufacturer</code> — производитель компьютера. Если значение <code>VMware</code> , <code>Xen</code> , то это виртуальная машина.
<b>Processor</b>	Информация о процессоре: <ul style="list-style-type: none"><li>• <b>name</b> — название модели процессора,</li><li>• <b>identifier</b> — идентификатор процессора,</li><li>• <b>S/N</b> — серийный номер процессора,</li><li>• <b>count.package</b> — количество кристаллов процессора,</li><li>• <b>count.processor</b> — физическое количество процессоров,</li><li>• <b>count.logical</b> — логическое количество процессоров.</li></ul>
<b>Limits</b>	<a href="#">Параметры ОС</a> . Отображаются только для Linux.
<b>OS memory</b>	Информация о количестве и состоянии оперативной памяти компьютера: <ul style="list-style-type: none"><li>• <b>ram.total</b> — общее количество оперативной памяти на сервере,</li><li>• <b>ram.free</b> — свободное количество оперативной памяти,</li><li>• <b>swap.total</b> — общее количество swap-памяти на сервере,</li><li>• <b>swap.used</b> — используемая swap-память.</li></ul>
<b>JVM heap memory</b>	Настройки heap памяти JVM:



- 
- **min** — минимальное количество оперативной памяти, которое выделено «Единому клиенту»,
  - **max** — максимальное количество оперативной памяти, которое выделено «Единому клиенту».

---

**JVM non heap memory**

Настройки stack памяти JVM:

- **min** — объем виртуальной памяти, к которому может получить доступ процесс,
- **max** — объем памяти процесса в оперативной памяти в момент старта приложения.

---

**Disks**

Информация о дисках. В Докере информация о дисках не предоставляется.

Для диска определяется:

- модель
- серийный номер
- размер диска
- разделы для диска:
  - тип файловой системы
  - размер
  - опции монтирования

---

**Bind points**

Информация о точках монтирования с типом *bind* из */etc/fstab*. Отображаются только для Linux.

---

**Network info**

Описание сетевых интерфейсов сервера:

- **ipv4** — ip адрес сервера.
  - **mac** — аппаратный адрес сетевой карты.
  - **mtu** — максимальный размер пакет или фрейма в байтах, который может быть отправлен одновременно без фрагментации.
  - **speed** — фактическая скорость сетевой карты на момент старта приложения.
-

# СТАТИСТИКА SOAP-ЗАПРОСОВ

## Единый клиент (cdi-soap-stats.log)

В журнале содержится агрегированная информация о вызовах SOAP-методов.

В релизе 19.15 добавлены дополнительная информация об IP, авторе запроса и системе и агрегирование запросов по ним.

Для каждого наименования метода добавлена общая строка, содержащая **<ALL>** вместо **{ip}**, **{login}**, **{system}**, в которой объединена информация в рамках одного метода.

Статистика по запросам в рамках периода отделена пустой строкой.

### Формат:

```
{datetime} {endpoint} {method} {ip} {login} {system} call {call_qty} fail {fail_qty} avg {avg_qty} p90 {p90_qty} p95 {p95_qty} p99 {p99_qty} max {max_qty} min {min_qty} qps {qps_qty}
```

### Описание колонок:

Колонка	Значение	Описание
1	<b>date</b>	Дата и время сбора статистики с точностью до минут. Время отображается в одинарных кавычках.
2	<b>endpoint</b>	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]).
3	<b>method</b>	Название метода.
4	<b>ip</b>	IP-адрес, с которого пришел запрос.
5	<b>login</b>	Учетная запись, от которой пришел запрос.
6	<b>system</b>	Наименование исходной системы.
8	<b>call_qty</b>	Число вызовов.
10	<b>fail_qty</b>	Число вызовов, завершившихся с ошибкой.
12	<b>avg_qty</b>	Среднее время обработки успешных вызовов, в мс.
14	<b>p90_qty</b>	Время обработки, в которое укладывается 90% успешных вызовов, в мс.
16	<b>p95_qty</b>	Время обработки, в которое укладывается 95% успешных вызовов, в мс.

	<b>qty</b>	
18	<b>p99_ qty</b>	Время обработки, в которое укладывается 99% успешных вызовов, в мс.
20	<b>max_ qty</b>	Максимальное время обработки успешных вызовов, в мс.
22	<b>min_ qty</b>	Минимальное время обработки успешных вызовов, в мс.
24	<b>qps_ qty</b>	Количество вызовов в секунду.

Колонки 8, 10, 12, 14, 16, 18, 20, 22, 24 содержат фиксированный текст.

Примечание: Если запись логов происходит не по расписанию, а при закрытии приложения, тогда вместо 24 и 25 колонок будет строка типа: (forced close at 5-03-2018 11:26:52).

### Пример:

```

2019-08-15 '13:45' soapOverJms_CFT_80_89 save
192.168.0.1 author1 AL call 1 fail 0
p90 3050 p95 3050 p99 3050 avg 3050 max 3050
min 3050 qps 0.00
2019-08-15 '13:45' soapOverJms_CFT_80_89 save
<ALL> <ALL> <ALL> call 1 fail 0
p90 3050 p95 3050 p99 3050 avg 3050 max 3050
min 3050 qps 0.00
2019-08-15 '13:50' soap_PartyWS save
192.168.0.1 author1 BT call 1 fail
0 p90 78 p95 78 p99 78 avg 78
max 78 min 78 qps 0,02
2019-08-15 '13:50' soap_PartyWS save
<ALL> <ALL> <ALL> call 1 fail 0
p90 78 p95 78 p99 78 avg 78 max
78 min 78 qps 0,02
2019-08-15 '13:50'
2019-08-15 '13:51' soap_PartyWS save
192.168.0.1 author1 AL call 1 fail
0 p90 78 p95 78 p99 78 avg 78
max 78 min 78 qps 0,02
2019-08-15 '13:51' soap_PartyWS save
192.168.0.1 author1 BT call 1 fail
0 p90 62 p95 62 p99 62 avg 62
max 62 min 62 qps 0,02
2019-08-15 '13:51' soap_PartyWS save
192.168.0.1 author1 GM call 1 fail
0 p90 62 p95 62 p99 62 avg 62
max 62 min 62 qps 0,02
2019-08-15 '13:51' soap_PartyWS save
<ALL> <ALL> <ALL> call 3 fail 0
p90 78 p95 78 p99 78 avg 67 max
78 min 62 qps 0,05
2019-08-15 '13:51' soap_PartyWS saveAndMerge
192.168.0.1 author1 BT call 1 fail
0 p90 1404 p95 1404 p99 1404 avg 1404
max 1404 min 1404 qps 0,02
2019-08-15 '13:51' soap_PartyWS saveAndMerge
192.168.0.1 author1 GM call 1 fail

```

0	p90	3807	p95	3807	p99	3807	avg	3807
max	3807	min	3807	qps	0,02			
2019-08-15 '13:51'		soap_PartyWS			saveAndMerge			
<ALL>	<ALL>	<ALL>	<ALL>	call	2	fail	0	
p90	3807	p95	3807	p99	3807	avg	2606	max
3807	min	1404	qps	0,03				
2019-08-15 '13:51'		soap_PartyWS			save			
2019-08-15 '14:53'		soap_PartyWS			save			
192.168.0.1	author1			BT	call	1	fail	
0	p90	172	p95	172	p99	172	avg	172
max	172	min	172	(force closed at 15.08.2019 14:53:17)				
2019-08-15 '14:53'		soap_PartyWS			save			
<ALL>	<ALL>	<ALL>	<ALL>	call	1	fail	0	
p90	172	p95	172	p99	172	avg	172	max
172	min	172	(force closed at 15.08.2019 14:53:17)					

### Настройки:

По умолчанию собирается раз в 1800 секунд (30 минут).

Для изменения частоты записи требуется изменить параметр `webservice.stats.periodInSeconds` (Подробнее про параметр тут: [Параметры SOAP и REST сервисов](#)) в *APM Администратора — Конфигурация — Параметры SOAP и REST сервисов*.

# СТАТИСТИКА ЗАПРОСОВ ВЕБ-ИНТЕРФЕЙСА

## cdi-ui-stats.log

В журнале содержится агрегированная информация о вызовах веб-интерфейса.

Статистика по запросам в рамках периода отделена пустой строкой.

### Формат:

```
{datetime} {controller} {method} {protocol} call {call_qty} fail {fail_qty} cancel {cancel_qty} timeout {timeout_qty} p90 {p90_qty} p95 {p95_qty} p99 {p99_qty} avg {avg_qty} max {max_qty} qps {qps_qty}
```

### Описание колонок:

Колонка	Значение	Описание
1	<b>date</b>	Дата и время сбора статистики с точностью до минут. Время отображается в одинарных кавычках.
2	<b>controller</b>	Название название контроллера. Необходимо для обеспечения уникальности записей, т.к. могут быть одинаковые методы для разных контроллеров.
3	<b>method</b>	Название метода.
4	<b>protocol</b>	Протокол (для ui всегда rest)
6	<b>call_qty</b>	Число вызовов.
8	<b>fail_qty</b>	Число вызовов, завершившихся с ошибкой.
10	<b>cancel_qty</b>	Число отменённых вызовов.
12	<b>timeout_qty</b>	Число вызовов, завершившихся с ошибкой по таймауту.
14	<b>p90_qty</b>	Время обработки, в которое укладывается 90% успешных вызовов, в мс.
16	<b>p95_qty</b>	Время обработки, в которое укладывается 95% успешных вызовов, в мс.
18	<b>p99_qty</b>	Время обработки, в которое укладывается 99% успешных вызовов, в мс.
20	<b>avg_qty</b>	Среднее время обработки успешных вызовов, в мс.
22	<b>max_qty</b>	Максимальное время обработки успешных вызовов, в мс.
24	<b>qps_qty</b>	Количество вызовов в секунду.

Колонки 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 содержат фиксированный текст.



# СТАТИСТИКА ОБРАТНОГО ПОТОКА

## cdi-stats-jms.log

В журнале содержится агрегированная информация об отправке событий в обратный поток

### Формат:

```
{datetime} {pending} {errors} {sent} {max_sent_time}
```

### Описание колонок:

Колонка	Значение	Описание
1	<b>date</b>	Дата и время сбора статистики с точностью до минут. Время отображается в одинарных кавычках.
2	<b>pending</b>	Количество событий в статусе pending.
3	<b>errors</b>	Количество событий, отправка которых завершилась с ошибкой.
4	<b>sent</b>	Количество отправленных событий в секунду.
6	<b>max_sent_time</b>	Максимальное время успешной отправки событий вызовов, в мс.

### Пример:

```
                pending_restEventSender.sblife
2023-09-10 '18:00:00'    0
2023-09-10 '18:05:00'    0
2023-09-10 '18:10:00'    5
2023-09-10 '18:15:00'    5
                pending_restEventSender.sblife    sent_restEventSender.sblife
max_sent_time_restEventSender.sblife,ms    errors_restEventSender.sblife
2023-09-10 '18:20:00'    4                0.02
127                3
```

### Настройки:

По умолчанию собирается раз в минуту.

Для изменения частоты записи требуется изменить параметр **monitoring.jms.loggingPeriodSec** в АРМ Администратора — Конфигурация — Параметры модуля мониторинга

# ТРАССИРОВКА JMS-СООБЩЕНИЙ

Временно включить журналирование JMS-сообщений (ЕК версии 19.13 и выше):

1. Найти файл logback.xml текущей сборки ЕК. Местоположение:

- **JBOSS\_CDI\_HOME\standalone\tmp\vfs\temp\temp\*\*\* (с наибольшей датой актуальности)\content-\*\*\*\WEB-INF\classes\logback.xml**

2. Найти настройку и изменить **level** с **OFF** на **INFO** / **DEBUG**:

- в режиме **INFO** в лог **cdi-jms-event-trace.log** начнут писаться все входящие и уходящие JMS-сообщения
- в режиме **DEBUG** дополнительно еще в лог будут выводиться стек-трейс ошибок

```
<!-- Switch to INFO to enable JMS_EVENTS trace (DEBUG will enable error stacktraces) -->  
<logger name="ru.hflabs.cdi.jms.EVENT_TRACE" level="OFF" additivity="false">  
  <appender-ref ref="JMS_EVENT_TRACE_FILE" />  
</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.



# ТРАССИРОВКА SOAP-ЗАПРОСОВ

Временно включить журналирование SOAP-запросов:

1. Найти файл logback.xml текущей сборки ЕК. Местоположение:

**WILDFLY\_CDI\_HOME\standalone\tmp\vfs\tmp\temp\*\*\*(с наибольшей датой актуальности)\content-\*\*\*\WEB-INF\classes\logback.xml**

2. Найти настройку: **LoggingFeature\_TRACE**:


```
<!-- Trace SOAP and REST messages [OFF/INFO] -->
<logger name="org.apache.cxf.ext.logging.LoggingFeature_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.ext.logging.LoggingFeature_TRACE" level="INFO" additivity="false">
  <appender-ref ref="SOAP_TRACE" />
</logger>
```

Перезапуск JBoss / WildFly не требуется, журналирование будет автоматически включено в течение нескольких секунд. Логи будут записываться в файл **cdi-soap-trace.log**. Чтобы отключить журналирование, достаточно заменить в указанной настройке уровень протоколирования INFO на OFF, либо перезапустить JBoss/WildFly.

Включать журналирование на длительное время не рекомендуется, т. к. это снижает быстродействие системы и существенно увеличивает объем журналов.

 Включение трассировки временная опция, журналирование будет автоматически выключено при перезапуске сервиса ЕК.

# ТРАССИРОВКА ПРОИЗВОЛЬНОГО ПАКЕТА (КАК ВКЛЮЧИТЬ TRACE В ЛОГАХ)

Доступно с релиза 21.7.

Результат команд применяется сразу, рестарт ЕК не требуется. После рестарта ЕК установленные настройки сбрасываются к умолчательным.

Для установки уровня логирования необходимо выполнить команду из консоли:

```
curl 'http://localhost:8080/cdi/api/manage/loggers/{package}' -i -X POST \  
-H 'Content-Type: application/json' \  
-d '{"configuredLevel": "{level}"}'  
  
# Например, установить уровень DEBUG для пакета org.springframework.security можно так  
curl 'http://localhost:8080/cdi/api/manage/loggers/org.springframework.security' -i -X POST \  
-H 'Content-Type: application/json' \  
-d '{"configuredLevel": "DEBUG"}'
```

Для просмотра текущего уровня необходимо выполнить команду из консоли:

```
curl 'http://localhost:8080/cdi/api/manage/loggers/{package}' -i -X GET
```

Для сброса уровня логирования к умолчательному значению:

```
curl 'http://localhost:8080/cdi/api/manage/loggers/{package}' -i -X POST \  
-H 'Content-Type: application/json' \  
-d '{}'
```

# ЖУРНАЛЫ ФАКТОРА

Журналы расположены на сервере приложений в каталоге **JBOSS\_FACTOR\_HOME** **/standalone/log**.

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

## ЖУРНАЛЫ КОНТЕЙНЕРА ПРИЛОЖЕНИЙ

- `console.log` — Консольный вывод джавы. Создаётся заново при рестарте приложения.
- `server.log` — Системные сообщения и сообщения от сервера приложений, в т.ч. сообщения об ошибках при загрузке контейнера.
- `gc.log` — Временные метки запуска сборщика мусора и размер кучи (heap), время потраченное на сборку мусора.
- `access.log` — Статистика всех запросов, приходящих в WildFly. Есть информация о вызывающей стороне.

## ЖУРНАЛЫ «ФАКТОРА»

### Общие

- `factor.log` — Общая информация: о старте приложения, о количестве запущенных экземпляров внутри подсети, о статусе загрузки справочников, о технических ошибках при вызове интерфейсов.
- `factor-lifecycle.log` — Содержит версию «Фактора», время старта и остановки.

### Журналы мониторинга и диагностики «Фактора»

- `factor-stats-system.log` и `factor-stats-system-{PID}.log` — Информация о лицензии, технические сведения о сервере (ОС, CPU, память, диски, сетевые интерфейсы) и переменных запуска JVM.
- `factor-stats-avg.log` — Статистика как общей, так и создаваемой «Фактором» текущей нагрузки на CPU, память, диск и сет.
- `factor-stats-task.log` — Статистика запуска и выполнения оффлайн задач.
- `factor-stats-api.log` — Статистика вызовов «Фактора» через онлайн-API.
- `factor-trace-clean.log` — Журнал временного логирования данных запроса и ответа сервиса онлайн-очистки данных (SOAP, REST). Подключается временно для диагностики.
- `factor-stats-netstat.log` — Вывод в лог системной утилиты `netstat -an https://ru.wikipedia.org/wiki/Netstat` раз в N секунд. Собирается информация о соединениях системы по протоколам UDP и TCP, использующих через сокет номер порта, на котором запущен «Фактор».
- `factor-stats-cputop.log` — Вывод процессов, нагружающих CPU на хост-машине.
- `factor-stats-dedup.log`

### Журналы поиска по чёрным спискам (ЧС)

- `factor-blacklist-dictionary.log` — Содержит информацию об этапах загрузки чёрных списков.
- `factor-blacklist-offline.log` — Содержит информацию об этапах и статусе полного поиска по ЧС.
- `factor-blacklist-online.log` — Содержит запросы и ответы сервиса онлайн поиска по ЧС.

Журнал	Описание	Условие создания
<code>console.log</code>	Консольный вывод джавы. Перезатируется при рестарте приложения.	всегда при старте jboss

factor-lifecycle.log	Обновления версий сервиса и события старта/стопа приложения. Лог не пересоздается.	при первом старте factor
server.log	Системные сообщения и сообщения от сервера JBOSS.	всегда при старте jboss
factor.log	Основной журнал работы Фактора.	всегда при старте jboss
factor-blacklist-online.log	Журнал онлайн поиска по черным спискам.	если подключен модуль проверки по стоп-листам
factor-blacklist-offline.log	Журнал полного поиска по черным спискам.	если подключен модуль проверки по стоп-листам
factor-blacklist-dictionary.log	Журнал загрузки данных по черным спискам.	если подключен модуль проверки по стоп-листам
factor-blacklist.log	Журнал поиска по черным спискам. (упразднен с релиза 20.12+)	если подключен модуль проверки по стоп-листам
factor-stats.log	Журнал статистики вызовов SOAP-методов.	всегда при старте jboss
factor-dedup-offline-<deduprule>.log	Журнал полного поиска дубликатов для конкретного дедап процессора, напр.: <ul style="list-style-type: none"> <li>factor-dedup-offline-physical.log</li> <li>factor-dedup-offline-egr_check.log</li> </ul>	всегда при старте jboss
factor-dedup-online-<deduprule>.log	Журнал онлайн поиска дубликатов для конкретного дедап процессора, напр.: <ul style="list-style-type: none"> <li>factor-dedup-online-physical.log</li> <li>factor-dedup-online-egr_check.log</li> </ul>	всегда при старте jboss
factor-dedup.log	Общий журнал поиска дубликатов	при вызове неизвестного Фактору дедап процессора
factor-longops.log	Журнал трассировки SOAP-запросов, выполняющихся дольше максимального порога.	всегда при старте jboss
cdi-recode.log	Журнал перекодировок. Логирует значения, которых нет в классификаторах.	если подключены перекодирующие фильтры

```

27.02.2019/18:06:09 127.0.0.1#anonymous
(MethodStatisticsCollector#dictionary) [INFO]
(NotFoundStatisticsCollector) Recoding fails:
dictionary='legalEntity.txt' value='SIBL - AGAT' count=1
27.02.2019/18:06:09 127.0.0.1#anonymous
(MethodStatisticsCollector#dictionary) [INFO]
(NotFoundStatisticsCollector) Recoding fails:

```

---

**dictionary='legalEntity.txt' value='SIBL - BRILLIANT' count=8**  
**27.02.2019/18:06:09 127.0.0.1#anonymous**  
**(MethodStatisticsCollector#dictionary) [INFO]**  
**(NotFoundStatisticsCollector) Recoding fails:**  
**dictionary='legalEntity.txt' value='SIBL - IZUMRUD' count=1**

---

# 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.

**TRACE** — наиболее детальная информация о ходе процесса обработки данных, которая включает в себя все описанные выше уровни и добавляет еще больше детализации, например, показывает правило, которое победило при выборе реквизита золотой карточки.

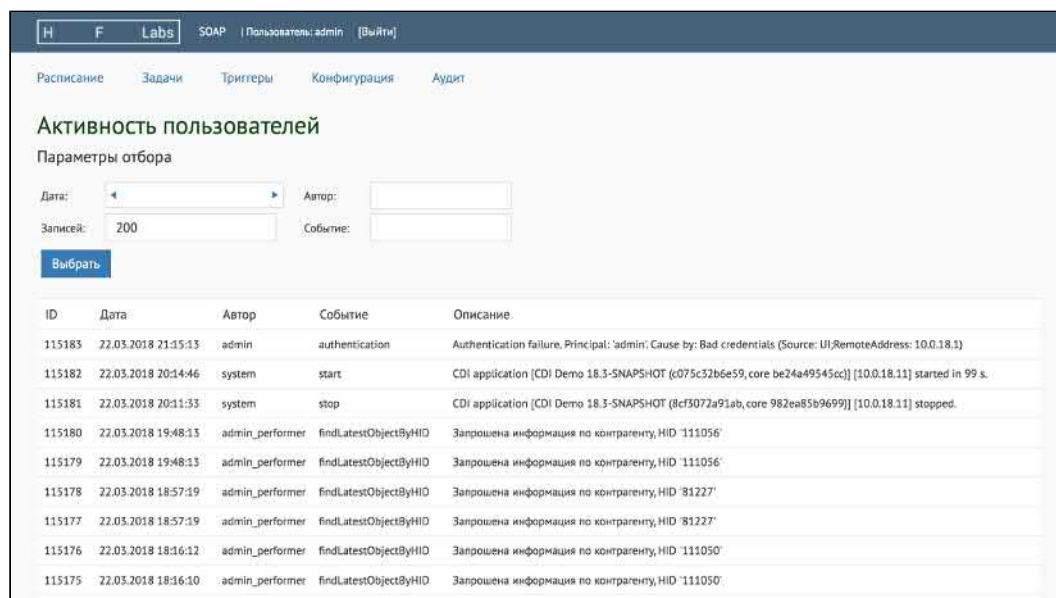
Для повседневного использования в штатном режиме рекомендуется уровень протоколирования WARN или INFO.

Для отладки и отслеживания причин возникновения проблем — уровень DEBUG.



# АУДИТ

Экран аудита предназначен для мониторинга действий, производимых пользователями в системе Единый Клиент.



The screenshot shows a web interface for monitoring user activity. At the top, there are navigation tabs: 'Расписание', 'Задачи', 'Триггеры', 'Конфигурация', and 'Аудит'. The main heading is 'Активность пользователей'. Below it, there are filter parameters: 'Дата:' with a date range selector, 'Автор:' with a text input, 'Записей:' with a numeric input set to '200', and 'Событие:' with a dropdown menu. A 'Выбрать' button is located below the filters. The main part of the interface is a table with the following columns: ID, Дата, Автор, Событие, and Описание. The table contains several rows of log entries, including authentication failures, application starts/stops, and database queries.

ID	Дата	Автор	Событие	Описание
115183	22.03.2018 21:35:13	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 20:11:53	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.

## 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.

## Типы событий

В системе определены следующие типы событий:

Код события	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 атрибута контрагента	Удаление прошлого значения атрибута '{атрибут}', ID значения '{идентификатор_атрибута}'
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}'
getOriginalWithWinnersFlags	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, НІДы контрагентов '{идентификатор_контрагента}'
performDuplicate	Работа с парами дубликатов	Подтверждение/отклонение/откладывание пары дубликатов, НІД '{идентификатор_контрагента}'-{идентификатор_контрагента}'
performDuplicates	Работа с парами транзитивных дубликатов	Подтверждение/отклонение/откладывание пары дубликатов, НІД '{идентификатор_контрагента}'-{идентификатор_контрагента}', в случае присутствия транзитивного дубля
performUnmerge	Разъединение контрагентов	Разъединение контрагентов, объединенный НІД '{идентификатор_контрагента}', отделяемый НІД '{идентификатор_контрагента}'
getDiff	Сравнение контрагентов из групп дубликатов	Запрошена информация по контрагенту, НІД '{идентификатор_контрагента}', '{идентификатор_контрагента}'
performRecover	Восстановление контрагента на дату	Восстановление контрагента, НІД '{идентификатор_контрагента}' на дату '{дата}'
performRecoverRequire	Восстановление реквизитов контрагента на дату	Восстановление реквизитов контрагента, НІД '{идентификатор_контрагента}' на дату '{дата}'
performRecoverAttribute	Восстановление атрибута контрагента на дату	Восстановление атрибута '{атрибут}', НІД '{идентификатор_контрагента}' на дату '{дата}'
propertiesChange	Изменение свойства конфигурации	Добавлено/изменено/удалено свойство '{свойство}'
search	Поиск контрагента	Поиск контрагента ['{запрос}'] - Всего найдено: '{количество}' КА. Размер страницы данных: '{количество}' <b>Если найдено более 3-х карточек</b>  Поиск контрагента ['{запрос}'] - Результат: ['{идентификатор(ы)_контрагента}'] <b>Если найдено 1-3 карточки</b>
search_scroll	Поиск контрагента (при прокрутке результатов выдачи)	Скролл поиска (Оффсет: '{смещение}') ['{запрос}'] - Всего найдено: '{количество}' КА. Размер страницы данных: '{количество}'

# УСТАНОВКА ОБНОВЛЕНИЙ

**i** Данная инструкция содержит стандартный порядок действий, выполняемый при обновлении Единого Клиента.

При необходимости выполнения дополнительных действий предоставляется отдельная подробная инструкция по обновлению.

## 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 **JBOS S AS (JBOS Application Server)**, каждый в своём инстансе.

Серверы приложений **JBOS** 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.

## ОБНОВЛЕНИЕ МОДУЛЕЙ

**i** С релиза 20.16 изменился формат имени war файла, были добавлены дата и время сборки файла: **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**

### Обновление модуля «Единый клиент»

1. Сделать резервные копии предыдущей сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.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. Например, выполнив команду: **md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**
3. Остановить сервис «Единого клиента» (**HFLABS CDI**)
  - a. linux: **service cdi stop**
  - b. win: **net stop 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**).
  - a. linux: **service cdi start**
  - b. win: **net start cdi**
7. Проверить, что в АРМ Администратора отображается корректная версия продукта (номер версии указан в начале инструкции).

### Обновление модуля «Фактор»

1. Сделать резервную копию предыдущей сборки «Фактора» **factor-{CUSTOMER}-{version}.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. Например, выполнив команду: **md5sum -c factor-{CUSTOMER}-{version}.war.md5**
3. Остановить сервис **HFLABS FACTOR**
  - a. linux: **service factor stop**
  - b. win: **net stop 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**
  - a. linux: **service factor start**
  - b. win: **net start factor**
7. Проверить, что в АРМ Администратора отображается корректная версия Фактора (номер версии указан в начале инструкции).

## 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
```



# 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.

# ОБНОВЛЕНИЕ МОДУЛЯ «ЕДИНЫЙ КЛИЕНТ»

**i** С релиза 20.16 изменился формат имени war файла, были добавлены дата и время сборки файла: **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**

## Обновление модуля «Единый клиент»

1. Сделать резервные копии предыдущей сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.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. Например, выполнив команду: **md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**
3. Остановить сервис «Единого клиента» (**HFLABS CDI**)
  - a. linux: **service cdi stop**
  - b. win: **net stop 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**).
  - a. linux: **service cdi start**
  - b. win: **net start cdi**
7. Проверить, что в АРМ Администратора отображается корректная версия продукта (номер версии указан в начале инструкции).

# ОБНОВЛЕНИЕ МОДУЛЯ «ФАКТОР»

## Обновление модуля «Фактор»

1. Сделать резервную копию предыдущей сборки «Фактора» **factor-{CUSTOMER}-{version}.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. Например, выполнив команду: **md5sum -c factor-{CUSTOMER}-{version}.war.md5**
3. Остановить сервис **HFLABS FACTOR**
  - a. linux: **service factor stop**
  - b. win: **net stop 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**
  - a. linux: **service factor start**
  - b. win: **net start factor**
7. Проверить, что в АРМ Администратора отображается корректная версия Фактора (номер версии указан в начале инструкции).

# АВТОМАТИЧЕСКАЯ МИГРАЦИЯ НА НОВУЮ ВЕРСИЮ

При переходе на новую версию ЕК автоматически обновляет БД, поисковые индексы и собирает диагностику:

- 1. При старте ЕК понимает, что нужно мигрировать на новую версию*
- 2. Обновляется БД*

ЕК изменяет схему данных, создает новые и удаляет устаревшие объекты.

- 3. Перестраиваются поисковые индексы*

ЕК определяет, какие индексы нужно перестроить.

Запускает перестроение индексов. Список индексов отправляет администратору по почте:

<b>Тема: Единый клиент обновляется на версию 3.22</b>
<b>Обновление с версии 3.21 на 3.22.</b>
<b>Будут перестроены индексы (GRAPH, PHYSICAL, MANUAL_DUPLICATE_GROUP).</b>

- 4. Собирается диагностика*

ЕК собирает стандартную диагностику (профили **mail.diagnostic.defaultProfiles**).

Отправляет диагностику администратору по почте:

<b>Тема: Единый клиент обновился на версию 3.22</b>
<b>Обновление с версии 3.21 на 3.22. Диагностика во вложении (config,db.connection,logs,search,sql.counts,sql.metadata,versions).</b>

# АВТОМАТИЧЕСКАЯ МИГРАЦИЯ НА НОВУЮ ВЕРСИЮ (ТЕХПОДДЕРЖКА)

- Включить автоматическую миграцию
- Перестраивать невалидные индексы при миграции
- Задать индексы для автоматической миграции
  - Где настраивается
  - Как формируется список индексов для перестроения
- Настроить оповещение по почте
- Логирование

## Включить автоматическую миграцию

В параметрах сервисов установить **services.applicationMigration.enabled = true**.

## 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](#) **service s.applicationMigration.rebuildInvalidIndexes = true**. Тогда после запуска новой версии ЕК придется ждать, пока ЕК проверит все индексы. Это может занять больше часа.

## Задать индексы для автоматической миграции

### 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>
```

## 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 включим флаг **services.applicationMigration.enabled = true**, список пополняется невалидными индексами.

## Настроить оповещение по почте

Задать параметры отправки email, блок *Автоматическая миграция приложения*.

## 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.



# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ С JBOSS 4 НА JBOSS 7 ДЛЯ АДМИНИСТРАТОРОВ

- [Инструкция для Linux](#)
- [Инструкция для Windows](#)
- [Настройка datasource](#)



# 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**.

## 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**

## 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
```

## 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\_CD\_I\_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
```

## 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/
```

## 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>
[...]
```

## Настройка datasource для заказчиков, использующих зашифрованный пароль к БД

Для шифрования 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>
```

# ИНСТРУКЦИЯ ДЛЯ WINDOWS

⚠ Под пользователем **HFL\_USER** подразумевается пользователь, от имени которого запускались службы CDI и ФАКТОР на JBoss 4

## 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**

## 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.

## Установка JBOSS

Распакуйте архивы с JBoss в директорию, в которой они располагались ранее.

## 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
```

## 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"}
```

## 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
```

## 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}"
```

## 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>
[...]
```

## Настройка datasource для заказчиков, использующих зашифрованный пароль к БД

Для шифрования 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>
```

# НАСТРОЙКА DATASOURCE

## 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>  
[...]
```

## Настройка datasource для заказчиков, использующих зашифрованный пароль к БД

Для шифрования 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>
```





# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ С JBOSS 7 НА WILDFLY 8 ДЛЯ АДМИНИСТРАТОРОВ

- [Инструкция для Linux \(переход на WildFly 8\)](#)
- [Инструкция для Windows \(переход на WildFly 8\)](#)

# ИНСТРУКЦИЯ ДЛЯ LINUX (ПЕРЕХОД НА WILDFLY 8)

- Установочный пакет
- Создание резервной копии предыдущей инсталляции JBoss
- Установка WildFly
- Настройка сервисов
- Настройка параметров доступа к БД системы
- Настройка параметров запуска JBoss
- Настройка горячего резерва (при необходимости)
- Удаление установленного JBoss

⚠ 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**.

## 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».

## 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
```

## 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
```

## 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/
```

## 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](#).

## 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»:
  - a. Удалить параметр **-ea**.
  - b. 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.

## 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>
```

## 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
```

# ИНСТРУКЦИЯ ДЛЯ WINDOWS (ПЕРЕХОД НА WILDFLY 8)

- Установочный пакет
- Создание резервной копии предыдущей инсталляции JBoss
- Удаление старых системных служб запуска JBoss
- Установка WildFly
- Создание системных служб запуска WildFly
- Настройка параметров доступа к БД системы
- Настройка параметров запуска WildFly
- Удаление установленного JBoss

## *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».

## *Создание резервной копии предыдущей инсталляции JBoss*

1. Остановить службы **CDI** и **Фактор**.
2. Заархивировать предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

## *Удаление старых системных служб запуска JBoss*

```
sc delete cdi
sc delete factor
```

## *Установка WildFly*

Распаковать архивы с WildFly в директорию, где ранее лежал JBoss.

## *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.

## ***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](#).

## ***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.

## ***Удаление установленного JBoss***

Удалить директории, в которых располагались предыдущие инсталляции JBoss для «Единого клиента» и «Фактора».

# ИНСТРУКЦИЯ ПО ПЕРЕНОСУ ДАМПА БД

- Экспорт
- Импорт
- Создание oracle-директории

## Экспорт

1. Найти подходящую oracle-директорию или создать новую
2. Запустить команду для экспорта:

```
<ORACLE_HOME>/bin/expdp <USERNAME>/<PASSWORD>  
FLASHBACK_TIME=systemtimestamp 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> – , Oracle ,
- FLASHBACK\_TIME=systemtimestamp – , ( , )
- <USERNAME>/<PASSWORD> – ( , CDI\_TEST) .  
sysDBA
- <ORACLE\_DIRECTORY> – oracle- 1.

После успешного экспорта в директории появится файл дампа схемы БД и лог.

## 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> - , Oracle .
- <USERNAME>/<PASSWORD> - , , ,  
, - .
- <ORACLE\_DIRECTORY> - oracle-, 2.
- <OLD\_SCHEMA> - , ( )
- <NEW\_SCHEMA> - ,

## 6. После успешного окончания импорта.

- а. Выполнить скрипт `<CDI_mercurial_root>\cdi-services\src\main\schema\schema_oracle_tuning.sql`, предварительно заменив в нем `"${jdbc.username}"` на имя пользователя БД.
- б. Собрать статистику по схеме в целом
- в. 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).

# 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:

- а. 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> - , , "EXPORT\_DIR"
- <path\_to\_directory> - ,  
2a, , "C:/tmp/export"
- <username> - ( , CDI\_TEST)

 Если дамп с прода поднимали 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

# АВТОМАТИЧЕСКОЕ СОЗДАНИЕ СХЕМЫ CDI

## *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 **SYSD BA**, а также connection-string БД.

Windows:

```
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:

```
standalone.conf

# Init Database
JAVA_OPTS="$JAVA_OPTS -Djdbc.dbo.username=dbaAdminName -Djdbc.dbo.
password=dbaAdminPass -Djdbc.url=jdbc:oracle:thin:@server:port:sid"
```

## *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](#))

## *Пересоздание существующей схемы*

Запустить WildFly с параметром **-Djdbc.recreate**.

Windows:

```
standalone.bat -Djdbc.recreate
```

Linux:

```
sh standalone.sh -Djdbc.recreate
```

# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ С WILDFLY 8 НА WILDFLY 10 ДЛЯ АДМИНИСТРАТОРОВ

- [Инструкция для Linux \(переход на WildFly 10\)](#)
- [Инструкция для Windows \(переход на WildFly 10\)](#)

# ИНСТРУКЦИЯ ДЛЯ LINUX (ПЕРЕХОД НА WILDFLY 10)

- Установочный пакет
- Создание резервной копии предыдущей инсталляции JBoss
- Установка WildFly
- Настройка параметров доступа к БД системы
- Настройка параметров запуска JBoss
- Опциональные настройки
  - Горячий резерв
  - Безопасное соединение (SSL)
  - Доступ к внешней БД для Фактора
- Удаление предыдущей инсталляции JBoss

! 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**.

## 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».

## 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
```

## 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 {} \;
```

## 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.

## 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"
```

## Опциональные настройки

### 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 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 ЕК}" port="8088"/>
</outbound-socket-binding>
```

### 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>
```



```
</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.

### *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
```

### *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
```

# ИНСТРУКЦИЯ ДЛЯ WINDOWS (ПЕРЕХОД НА WILDFLY 10)

- Установочный пакет
- Создание резервной копии предыдущей инсталляции WildFly
- Удаление старых системных служб запуска WildFly
- Установка WildFly
- Создание системных служб запуска WildFly
- Настройка параметров доступа к БД системы
- Настройка параметров запуска WildFly
- Настройка горячего резерва (при необходимости)
- Удаление предыдущей инсталляции WildFly

## *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».

## *Создание резервной копии предыдущей инсталляции WildFly*

1. Остановить службы **CDI** и **Фактор**.
2. Заархивировать предыдущие инсталляции WildFly для «Единого клиента» и «Фактора».

## *Удаление старых системных служб запуска WildFly*

```
sc delete cdi  
sc delete factor
```

## *Установка WildFly*

Распаковать архивы с WildFly в директорию, где ранее лежал WildFly 8.

## 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.

## 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.

## 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` и з 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"
```

## 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 ЕК}" port="8088"/>  
</outbound-socket-binding>
```

## *Удаление предыдущей инсталляции WildFly*

Удалить директории, в которых располагались предыдущие инсталляции WildFly для «Единого клиента» и «Фактора».

# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ НА OPENJDK 11 И WILDFLY 16 ДЛЯ АДМИНИСТРАТОРОВ

**Начиная с версии 19.16, «Единый клиент» может быть запущен только на openJDK11 и WildFly16.**

С 16 апреля 2019 года Oracle изменил лицензионную политику, начиная с этой даты использование сборок Java SE, а значит, и получение патчей с обновлениями становится платным. В связи с этим HFLabs переходит в своих продуктах на использование OpenJDK.

**При переходе на версию «Единого клиента» 19.16 и выше необходимо дополнительно установить OpenJDK, обновить версию Java и Wildfly на всех серверах приложений по инструкции:**

- [Инструкция по обновлению для Windows;](#)
- [Инструкция по обновлению для Linux.](#)

# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ НА OPENJDK 11 И WILDFLY 16 (LINUX)

- Шаг 1. Проверить актуальность ОС  
Проверка актуальности версии
- Шаг 2. Установить OpenJDK 11  
Установка OpenJDK 11  
Через репозиторий  
Сборка AdoptOpenJDK  
  
Обновление значения JAVA\_HOME
- Шаг 3. Обновить Wildfly  
Переход на WildFly 16 для Linux  
Установочный пакет  
Установка WildFly  
Настройка параметров доступа к БД системы  
Настройка параметров запуска JBoss  
Для «Единого клиента»  
Для «Фактора»  
  
Перенос хешей дедапа и индексов «Фактора»  
Дополнительные настройки (при необходимости)  
Изменение смещения портов «Единого клиента» и «Фактора»  
Горячий резерв  
Безопасное соединение (SSL)  
Доступ к внешней БД для «Фактора»  
  
Переключение на новую версию WildFly  
Установка «Единого клиента» и «Фактора» и запуск служб
- Шаг 4. Перейти к инструкции по обновлению «Единого клиента» на новую версию

## Шаг 1. Проверить актуальность ОС

Linux развивается быстрее любой другой операционной системы. Регулярно появляются новые версии, реализующие новые функции и содержащие исправление различных ошибок и доработки по оптимизации выполнения процессов. При этом старые версии снимаются с поддержки и, как следствие, не выпускаются новые версии даже в случае обнаружения каких-либо ошибок.

Поэтому важно поддерживать сервера в актуальном состоянии и периодически обновлять операционную систему до актуальной версии.

На текущий момент актуальными и рекомендованными для использования на продуктовых серверах являются CentOS 7-й версии и выше, Red Hat 7 и выше.

Так как в рамках текущего обновления осуществляется переход на OpenJDK, обновление Java и WildFly, рекомендуем также сразу же выполнить работы по обновлению операционной системы до актуальной версии.

## Проверка актуальности версии

Проверить текущую установленную версию Linux можно одной из команд:

```
cat /etc/*-release
```

```
cat /proc/version
```

Если версия ОС соответствует рекомендованной — CentOS 7 и выше / Red Hat 7 и выше, — тогда ваши сервера находятся в актуальном состоянии и можно переходить к шагу 2.

Если установлена более старая версия ОС, тогда рекомендуем обновить ОС перед выполнением дальнейших шагов.

**Важно!** Не всегда обновить ОС можно без форматирования дисков и установки всего с нуля. Если обновление выполняется с форматированием дисков необходимо перед обновлением ОС сделать следующее:

- сделать бэкап директорий **JBOSS\_CDI\_HOME** и **JBOSS\_FACTOR\_HOME**, это позволит сохранить настройки ПО и сохранить справочники и индексы «Фактора»;
- по возможности сделать бэкап поискового индекса, находящийся в папке **CDI\_SEARCH\_DIR**, этот шаг необходим, чтобы не пришлось тратить время на перестроение индекса «Единого клиента» после запуска обновленной системы.


После обновления ОС выполнить установку, следуя [руководству по развертыванию](#), но не запускать систему.

Далее:

- Скопировать папку **standalone/data** из бэкапа «Фактора» в папку standalone новой инсталляции.
- Скопировать папку с поисковым индексом из бэкапа в папку **CDI\_SEARCH\_DIR**.
- Перенести необходимые настройки из бэкапа старой версии WildFly, следуя пункту «[Настройка параметров запуска JBoss](#)» инструкции по обновлению WildFly.
- При необходимости выполнить дополнительные настройки следовать пункту «[Дополнительные настройки](#)».
- По возможности, скопировать логи из папок **standalone/log** бэкапа в папки **standalone/log** новой инсталляции; это облегчит жизнь администраторам и поддержке.
- Произвести запуск системы и проверить, что она работает.

Если нет возможности выполнить обновление ОС, пожалуйста, напишите нам в поддержку.

## Шаг 2. Установить OpenJDK 11

 Поскольку Oracle начал брать деньги за использование своих сборок JDK, было принято решение перейти на open-source сборку openJDK.

При обновлении обязательно проверьте, что устанавливается именно openJDK!

### Установка OpenJDK 11

Существует несколько вариантов получения и установки пакета:

- Основной вариант установки: через репозиторий операционной системы (необходим доступ к репозиториям ОС);
- Альтернативный: сборка AdoptOpenJDK (если нет возможности использовать репозиторий для установки).

#### Через репозиторий

 Данный способ является приоритетным. Если нет возможности использовать репозитории ОС, см. следующий пункт «Используя сборку AdoptOpenJDK».

Установка

```
yum update
yum install java-11-openjdk-devel
```

После этого необходимо переключить default-версию java, выполнив команды

```
alternatives --config java
alternatives --config javac
```

В каждом случае нужно выбрать **java-11-openjdk**.

После этого перейти к пункту «Обновление значения JAVA\_HOME».

#### Сборка AdoptOpenJDK

Проверить версию [последней сборки](#) на сайте разработчика.

Создать папку для сборки, выкачать ее и указать расположение новой Java (при наличии более новой версии заменить ссылку [https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/OpenJDK11U-jdk\\_x64\\_linux\\_hotspot\\_11.0.4\\_11.tar.gz](https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz) на ссылку на новую версию):



```

mkdir /usr/java/
cd /usr/java/
wget https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11
/OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
tar zxvf OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
rm OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
alternatives --install /usr/bin/java java /usr/java/jdk-11.0.4+11/bin/java 2
alternatives --install /usr/bin/jar jar /usr/java/jdk-11.0.4+11/bin/jar 2
alternatives --install /usr/bin/javac javac /usr/java/jdk-11.0.4+11/bin/javac 2
alternatives --set java /usr/java/jdk-11.0.4+11/bin/java
alternatives --set jar /usr/java/jdk-11.0.4+11/bin/jar
alternatives --set javac /usr/java/jdk-11.0.4+11/bin/javac

```

## Обновление значения JAVA\_HOME

Чтобы установить новые значения для переменной JAVA\_HOME, необходимо:

1. Проверить, прописано ли значение JAVA\_HOME для пользователей «ЕК» и «Фактора» в файлах **/home/<user>/.bash\_profile** и **/home/<user>/.profile**, где **<user>** – пользователи «ЕК» и «Фактора» (обычно это **jboss**, **cdi**, **factor**, etc).  
Узнать используемых пользователей можно, например, выполнив команду **ls -l** с указанием пути к папке с инсталляцией WildFly. Например, для инсталляции cdi при условии, что WildFly установлен в папке **/home/cdi/jboss**:

команда	
<b>ls -l /home/cdi/</b>	
результат выполнения	
<b>drwxr-xr-x 9 cdi cdi</b>	<b>4096 авг 14 14:00 jboss</b>
....	

В третьем столбце указан пользователь-владелец каталога, в четвертом – его группа. Если путь указан, но он ведет к старой версии Java (например **/usr/lib/jvm/java-8-oracle**), тогда изменить значение на актуальное (**/usr/java/jdk-11.0.4+11**).

2. Проверить, прописано ли значение JAVA\_HOME в **/etc/cdi/wildfly.conf** и **/etc/factor/wildfly.conf**.
3. Если прописано старое значение, то изменить значение на актуальное. Если значение не задано, можно оставить файл без изменений.
4. Проверить, что возвращаются правильные версии java и javac и что они совпадают:

```

javac -version
java -version

```

## Шаг 3. Обновить Wildfly

## Переход на WildFly 16 для Linux

❗ В инструкции предполагается, что «Единый клиент» запускается от имени пользователя **cdi**, а «Фактор» от имени пользователя **factor**.

### Установочный пакет

- **wildfly-16.0.0.Final-8080.zip** — для «Единого клиента».
- **wildfly-16.0.0.Final-8080-HotReserve.zip** — для «Единого клиента» с горячим резервированием.
- **wildfly-16.0.0.Final-18080.zip** — для «Фактора».

Данные пакеты можно получить из указанных ресурсов:

- скачать из нашего [общего хранилища](#)
- взять архивы в составе поставки.

ℹ Инструкция предполагает в качестве предыдущего места установки каталоги **/opt/cdi** для «Единого клиента» и **/opt/factor** для «Фактора».

Если для установки использовались другие каталоги, например **/home/cdi**, **/home/factor** или **/data/wildfly**, то необходимо заменить предложенные в инструкции пути на актуальные для вашей системы.

### Установка WildFly

1. Распаковать архивы с WildFly в каталог **/opt**:

```
unzip wildfly-16.0.0.Final-8080*.zip -d /opt/cdi
unzip wildfly-16.0.0.Final-18080.zip -d /opt/factor

mv /opt/cdi/wildfly* /opt/cdi/jboss_new
mv /opt/factor/wildfly* /opt/factor/jboss_new
```

2. Скопировать файлы **launch.sh** из старых инсталляций JBoss «Единого клиента» и JBoss «Фактора» для обеспечения запуска приложений:

```
cp /opt/cdi/jboss/bin/launch.sh /opt/cdi/jboss_new/bin/
cp /opt/factor/jboss/bin/launch.sh /opt/factor/jboss_new/bin/
```

## Настройка параметров доступа к БД системы

Скопировать файл **cdi-oracle-ds.xml**, **cdi-postgresql-ds.xml** или **cdi-mariadb-ds.xml** из старой инсталляции JBoss для «Единого клиента» в директорию **/opt/cdi/jboss\_new/standalone/deployments/**

### Oracle

```
cp /opt/cdi/jboss/standalone/deployments/cdi-oracle-ds.xml /opt/cdi/jboss_new/standalone/deployments
```

### Postgresql

```
cp /opt/cdi/jboss/standalone/deployments/cdi-postgresql-ds.xml /opt/cdi/jboss_new/standalone/deployments
```

### Mariadb

```
cp /opt/cdi/jboss/standalone/deployments/cdi-mariadb-ds.xml /opt/cdi/jboss_new/standalone/deployments
```

**При необходимости настроить шифрование пароля к БД.**

Определить, что система использует шифрование пароля к БД можно по содержанию файлов с параметрами доступа к БД, указанным выше. Если в файле логин и пароль указаны в явном виде

```
<user-name>username</user-name>
<password>password</password>
```

то настройка шифрования не требуется, если указано использование зашифрованного пароля

```
<security-domain>EncryptedPassword</security-domain>
```

то потребуется настройка согласно инструкции, ссылка на которую дана выше.

## Настройка параметров запуска JBoss

### Для «Единого клиента»

1. Скопировать файл **standalone.conf** из предыдущей инсталляции JBoss для «Единого клиента».

```
cp /opt/cdi/jboss/bin/standalone.conf /opt/cdi/jboss_new/bin/
```

2. Заменить в нем старые параметры Garbage collection:

```
# Garbage collection tuning
JAVA_OPTS="$JAVA_OPTS -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode -
XX:+CMSIncrementalPacing -XX:+CMSClassUnloadingEnabled -XX:
CMSIncrementalDutyCycleMin=0 -XX:CMSIncrementalDutyCycle=10
```

на новые:

```
# Garbage collection tuning
JAVA_OPTS="$JAVA_OPTS -XX:+UseG1GC -XX:G1HeapRegionSize=16m"
```

3. Если это не прод (на продажах дебаг выключен), обновить строку:

```
### Debugging options
JAVA_OPTS="$JAVA_OPTS -Xnoagent -Djava.compiler=NONE -Xdebug -Xrunjdwp:
transport=dt_socket,server=y,suspend=n,address=5005"
```

на

```
### Debugging options
JAVA_OPTS="$JAVA_OPTS -Xnoagent -Djava.compiler=NONE -Xdebug -Xrunjdwp:
transport=dt_socket,server=y,suspend=n,address=*:5005"
```

4. Проверить, что выставлены правильные настройки локализации, в файле должны присутствовать следующие настройки:

```
# Locale settings
JAVA_OPTS="$JAVA_OPTS -Duser.country=ru -Duser.language=ru -Duser.
timezone=GMT+3"
```

Если не задать эти настройки, то система будет использовать английскую локализацию.

Вместо часового пояса GMT+3 можно указать другой, если необходимо.

5. Проверить, что задан **CDI\_ROOT\_DIR** (обычно, это /home/cdi, но возможно использование иных каталогов):

```
# CDI root folder
JAVA_OPTS="$JAVA_OPTS -Dcdi.root.folder=/home/cdi"
```

Отдельно может быть задан каталог для поисковых индексов **CDI\_SEARCH\_DIR** — параметр **-Dcdi.search.folder**, и общий каталог для дедупликации — параметр **-Dcdi.dedup.folder**.

6. Найти настройки использования памяти, которые выглядят следующим образом:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms6g -Xmx12g -XX:PermSize=128m -XX:
MaxPermSize=256m -XX:-UseCodeCacheFlushing -XX:ReservedCodeCacheSize=256m"
```

и заменить их на новые настройки, сохранив старые значения **-Xms** и **-Xmx**, которые отвечают за пределы используемой памяти:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms6g -Xmx12g -XX:-UseCodeCacheFlushing -XX:
```

```
ReservedCodeCacheSize=256m -XX:MaxMetaspaceSize=512m -XX:-OmitStackTraceInFastThrow -XX:+UseCompressedOops -XX:+ExitOnOutOfMemoryError"
```

## Для «Фактора»

1. Отредактировать файл **standalone.conf**, перенеся в него недостающие настройки из предыдущей инсталляции JBoss для «Фактора» (**/opt/factor/jboss/bin/standalone.conf**).
2. Если раньше в опциях Java был прописан флаг **-ea**, проверить, что в новом файле **standalone.conf** он не используется. Использование этого флага в новой версии WildFly приведет к невозможности запустить «Фактор».
3. Перенести настройки используемой памяти из старого Wildfly с «Фактором».

Параметр `-Xms{x}g` отвечает за минимальное количество используемой памяти, параметр `-Xmx{X}g` — за максимальное.

Необходимо найти строки, указанные ниже, и заменить значения для `-Xms` `-Xmx` на те, которые были указаны в настройках старого Wildfly:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms8g -Xmx8g"
```

4. Указать пути к справочникам и индексам, если они были заданы в предыдущей инсталляции. Пример ( необходимо использовать пути, указанные в старом файле настроек):

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_INDEXES_PATH=/data/factor/indexes"
JAVA_OPTS="$JAVA_OPTS -DFACTOR_DICTIONARIES_PATH=/data/factor/dictionaries"
JAVA_OPTS="$JAVA_OPTS -DFACTOR_DEDUP_PATH=/data/factor/dedup"
```

## Перенос хешей дедупа и индексов «Фактора»

Скопировать папку **/data** из папки старой инсталляции **/opt/factor/jboss/standalone/** в папку **/opt/cdi/jboss\_new/standalone/**. Это поможет избежать полного перестроения хешей и индексов и лишних операций.

```
cp -a /opt/factor/jboss/standalone/data/ /opt/factor/jboss_new/standalone/
```

## Дополнительные настройки (при необходимости)

### Изменение смещения портов «Единого клиента» и «Фактора»

Если требуется изменить смещение портов «Фактора» относительно «Единого клиента» заданное в сборках WildFly (по умолчанию выставлено смещение 10000), необходимо отредактировать соответствующую настройку в **standalone.conf** для «Фактора»:

```
### Port offset
JAVA_OPTS="$JAVA_OPTS -Djboss.socket.binding.port-offset={port offset}"
```

Текущее используемое смещение указано в старом файле настроек.

Если смещение портов для «Фактора» не требуется (не указано в настройках предыдущей инсталляции), удалить ключ **-Djboss.socket.binding.port-offset={port offset}** из настроек или закомментировать его.

Отредактировать настройки отладки, указав в них адрес со смещением.

Например, адрес=5005, смещение=8, адрес+смещение=5013:

```
### Debugging options
#JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,
address=*:5013"
```

Необходимо, чтобы смещение всех портов на всех адресах было одинаковым.

Если необходимо задать смещение портов для «Единого клиента», то необходимо добавить ключ **-Djboss.socket.binding.port-offset={port offset}** в настройки **standalone.conf** для «Единого клиента» аналогично примеру, описанному выше для «Фактора», и задать остальные смещения портов по аналогии с примерами выше.

## Горячий резерв

В новый конфигурационный файл **/opt/cdi/jboss\_new/standalone/configuration/standalone.xml** перенести доменное имя второго экземпляра «ЕК» и его http-порт из блока **outbound-socket-binding name="node-sync-binding"** старого файла **/opt/cdi/jboss/standalone/configuration/standalone.xml**:

```
<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>
```

## Безопасное соединение (SSL)

Перенести настройки доступа к хранилищу ключей из **/opt/cdi/jboss/standalone/configuration/standalone.xml** в файл **/opt/cdi/jboss\_new/standalone/configuration/standalone.xml**.

В новый конфигурационный файл **/opt/cdi/jboss\_new/standalone/configuration/standalone.xml** скопировать блок **security-realm name="HttpsSecuredRealm"** из старого файла **/opt/cdi/jboss/standalone/configuration/standalone.xml**, удостовериться, что для коннектора **https-listener** блока **<server name="default-server">** указан

**параметр security-realm** — `HttpsSecuredRealm` (в данном примере указан общий параметр `<ssl>` блока `<server-identities>`, актуальные настройки находятся в конфигурационном файле старой инсталляции):

```
<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>
```

Блок `<security-realm name="HttpsSecuredRealm">` описывает настройки шифрования, блок коннектора `https-listener` описывает, когда его применять.

### Описание параметров `HttpsSecuredRealm`

- **keystore.path** — путь к хранилищу ключей, корневая директория определяется параметром **relative.to**;
- **alias** — алиас, под которым ключи доступны в хранилище;
- **keystore-password** — пароль к хранилищу;
- **key-password** — пароль к ключам (если не указан, используется **keystore-password**).

Скопировать в директорию `/opt/cdi/jboss_new/standalone/configuration/` файлы `*.jks`, на которые ссылается конфигурационный файл.

### *Доступ к внешней БД для «Фактора»*

Если для каких-либо справочников «Фактора» используется база данных, то доступ к ней настраивается с помощью файла `{customer}-ds.xml`, который находится в каталоге `standalone/deployments` инсталляции JBoss.

Если подобный **datasource**-файл использовался, то его необходимо скопировать из предыдущей инсталляции:

```
cp /opt/factor/jboss/standalone/deployments/factor-ds.xml /opt/factor/jboss_new/standalone
/deployments
```

---

## Переключение на новую версию WildFly

1. Остановить службы cdi и factor:

```
service cdi stop
service factor stop
```

2. Переместить логи из старой инсталляции в новую

```
mv /opt/cdi/jboss/standalone/log /opt/cdi/jboss_new/standalone/
mv /opt/factor/jboss/standalone/log /opt/factor/jboss_new/standalone/
```

Перенос логов позволит продолжить использование существующих логов, сохранит неразрывность информации и облегчит жизнь администраторам и поддержке. Логи могут занимать довольно много места и их лучше именно переносить, а не копировать.

3. Переименовать папки и заархивировать старые версии:

```
mv /opt/cdi/jboss /opt/cdi/jboss_10
mv /opt/factor/jboss /opt/factor/jboss_10

mv /opt/cdi/jboss_new /opt/cdi/jboss
mv /opt/factor/jboss_new /opt/factor/jboss

zip -rT /opt/cdi/old-jboss-cdi /opt/cdi/jboss_10
zip -rT /opt/factor/old-jboss-factor /opt/factor/jboss_10
```

4. Выдать соответствующим пользователям права доступа к каталогам jboss «Единого клиента» и jboss «Фактора», в том числе на редактирование:

```
chown -R cdi:cdi /opt/cdi/jboss
chown -R factor:factor /opt/factor/jboss
```

5. Назначить права на запуск исполняемых файлов:

```
find /opt/{factor,cdi}/jboss/ -type d -exec chmod 755 {} \;
find /opt/{factor,cdi}/jboss/ -type f -exec chmod 644 {} \;
find /opt/{factor,cdi}/jboss/ -type f -name "*.sh" -exec chmod 755 {} \;
```

6. Если раньше не осуществлялась настройка расширенного логирования работ системных служб, то необходимо в файлах **/etc/systemd/system/factor.service** и **/etc/systemd/system/cdi.service** заменить строку:

```
StandardOutput=null
```

на строки

```
StandardOutput=syslog
StandardError=syslog
```

и выполнить команду

---



```
systemctl daemon-reload
```

В противном случае дополнительных манипуляций со службами производить не нужно.

## **Установка «Единого клиента» и «Фактора» и запуск служб**

1. Скопировать файлы `cdi-web-<имя заказчика>-<номер версии>-SNAPSHOT.war` и `factor-<имя заказчика>-<номер версии>.war`, переданные в поставке с данной инструкцией, в папки **`/opt/cdi/jboss/standalone/deployments`** и **`/opt/factor/jboss/standalone/deployments`** соответственно.
2. Запустить службы `cdi` и `factor`:

```
service cdi start  
service factor start
```

3. Удостовериться, что комплекс запустился и работает нормально.

## **Шаг 4. Перейти к инструкции по обновлению «Единого клиента» на новую версию**

# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ НА OPENJDK 11 (LINUX)

⚠ Поскольку Oracle начал брать деньги за использование своих сборок JDK, было принято решение перейти на open-source сборку openJDK.

При обновлении обязательно проверьте, что устанавливается именно openJDK!

## Установка OpenJDK 11

Существует несколько вариантов получения и установки пакета:

- Основной вариант установки: через репозиторий операционной системы (необходим доступ к репозиториям ОС);
- Альтернативный: сборка AdoptOpenJDK (если нет возможности использовать репозиторий для установки).

### Через репозиторий

⚠ Данный способ является приоритетным. Если нет возможности использовать репозитории ОС, см. следующий пункт «Используя сборку AdoptOpenJDK».

Установка

```
yum update
yum install java-11-openjdk-devel
```

После этого необходимо переключить default-версию java, выполнив команды

```
alternatives --config java
alternatives --config javac
```

В каждом случае нужно выбрать **java-11-openjdk**.

После этого перейти к пункту «Обновление значения JAVA\_HOME».

### Сборка AdoptOpenJDK

Проверить версию [последней сборки](#) на сайте разработчика.

Создать папку для сборки, выкачать ее и указать расположение новой Java (при наличии более новой версии заменить ссылку [https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/OpenJDK11U-jdk\\_x64\\_linux\\_hotspot\\_11.0.4\\_11.tar.gz](https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz) на ссылку на новую версию):

```
mkdir /usr/java/
cd /usr/java/
```

```
wget https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11
/OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
tar zxvf OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
rm OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
alternatives --install /usr/bin/java java /usr/java/jdk-11.0.4+11/bin/java 2
alternatives --install /usr/bin/jar jar /usr/java/jdk-11.0.4+11/bin/jar 2
alternatives --install /usr/bin/javac javac /usr/java/jdk-11.0.4+11/bin/javac 2
alternatives --set java /usr/java/jdk-11.0.4+11/bin/java
alternatives --set jar /usr/java/jdk-11.0.4+11/bin/jar
alternatives --set javac /usr/java/jdk-11.0.4+11/bin/javac
```

## Обновление значения `JAVA_HOME`

Чтобы установить новые значения для переменной `JAVA_HOME`, необходимо:

1. Проверить, прописано ли значение `JAVA_HOME` для пользователей «ЕК» и «Фактора» в файлах `/home/<user>/.bash_profile` и `/home/<user>/.profile`, где `<user>` – пользователи «ЕК» и «Фактора» (обычно это **jboss**, **cdi**, **factor**, etc).  
Узнать используемых пользователей можно, например, выполнив команду `ls -l` с указанием пути к папке с инсталляцией WildFly. Например, для инсталляции cdi при условии, что WildFly установлен в папке `/home/cdi/jboss`:

команда	
<code>ls -l /home/cdi/</code>	

результат выполнения	
<code>drwxr-xr-x 9 cdi cdi</code>	<code>4096 авг 14 14:00 jboss</code>
<code>....</code>	

В третьем столбце указан пользователь-владелец каталога, в четвертом – его группа.  
Если путь указан, но он ведет к старой версии Java (например `/usr/lib/jvm/java-8-oracle`), тогда изменить значение на актуальное (`/usr/java/jdk-11.0.4+11`).

2. Проверить, прописано ли значение `JAVA_HOME` в `/etc/cdi/wildfly.conf` и `/etc/factor/wildfly.conf`.
3. Если прописано старое значение, то изменить значение на актуальное. Если значение не задано, можно оставить файл без изменений.
4. Проверить, что возвращаются правильные версии `java` и `javac` и что они совпадают:

```
javac -version
java -version
```

# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ НА WILDFLY 16 ДЛЯ LINUX

## Переход на WildFly 16 для Linux

❗ В инструкции предполагается, что «Единый клиент» запускается от имени пользователя **cdi**, а «Фактор» от имени пользователя **factor**.

### Установочный пакет

- **wildfly-16.0.0.Final-8080.zip** — для «Единого клиента».
- **wildfly-16.0.0.Final-8080-HotReserve.zip** — для «Единого клиента» с горячим резервированием.
- **wildfly-16.0.0.Final-18080.zip** — для «Фактора».

Данные пакеты можно получить из указанных ресурсов:

- скачать из нашего [общего хранилища](#)
- взять архивы в составе поставки.

ℹ Инструкция предполагает в качестве предыдущего места установки каталоги **/opt/cdi** для «Единого клиента» и **/opt/factor** для «Фактора».

Если для установки использовались другие каталоги, например **/home/cdi**, **/home/factor** или **/data/wildfly**, то необходимо заменить предложенные в инструкции пути на актуальные для вашей системы.

### Установка WildFly

1. Распаковать архивы с WildFly в каталог **/opt**:

```
unzip wildfly-16.0.0.Final-8080*.zip -d /opt/cdi
unzip wildfly-16.0.0.Final-18080.zip -d /opt/factor
```

```
mv /opt/cdi/wildfly* /opt/cdi/jboss_new
mv /opt/factor/wildfly* /opt/factor/jboss_new
```

2. Скопировать файлы **launch.sh** из старых инсталляций JBoss «Единого клиента» и JBoss «Фактора» для обеспечения запуска приложений:

```
cp /opt/cdi/jboss/bin/launch.sh /opt/cdi/jboss_new/bin/
cp /opt/factor/jboss/bin/launch.sh /opt/factor/jboss_new/bin/
```

## Настройка параметров доступа к БД системы

Скопировать файл **cdi-oracle-ds.xml**, **cdi-postgresql-ds.xml** или **cdi-mariadb-ds.xml** из старой инсталляции JBoss для «Единого клиента» в директорию **/opt/cdi/jboss\_new/standalone/deployments/**

### Oracle

```
cp /opt/cdi/jboss/standalone/deployments/cdi-oracle-ds.xml /opt/cdi/jboss_new/standalone/deployments
```

### Postgresql

```
cp /opt/cdi/jboss/standalone/deployments/cdi-postgresql-ds.xml /opt/cdi/jboss_new/standalone/deployments
```

### Mariadb

```
cp /opt/cdi/jboss/standalone/deployments/cdi-mariadb-ds.xml /opt/cdi/jboss_new/standalone/deployments
```

**При необходимости настроить шифрование пароля к БД.**

Определить, что система использует шифрование пароля к БД можно по содержанию файлов с параметрами доступа к БД, указанным выше. Если в файле логин и пароль указаны в явном виде

```
<user-name>username</user-name>
<password>password</password>
```

то настройка шифрования не требуется, если указано использование зашифрованного пароля

```
<security-domain>EncryptedPassword</security-domain>
```

то потребуется настройка согласно инструкции, ссылка на которую дана выше.

## Настройка параметров запуска JBoss

### Для «Единого клиента»

1. Скопировать файл **standalone.conf** из предыдущей инсталляции JBoss для «Единого клиента».

```
cp /opt/cdi/jboss/bin/standalone.conf /opt/cdi/jboss_new/bin/
```

2. Заменить в нем старые параметры Garbage collection:

```
# Garbage collection tuning
JAVA_OPTS="$JAVA_OPTS -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode -
XX:+CMSIncrementalPacing -XX:+CMSClassUnloadingEnabled -XX:
CMSIncrementalDutyCycleMin=0 -XX:CMSIncrementalDutyCycle=10
```

на новые:

```
# Garbage collection tuning
JAVA_OPTS="$JAVA_OPTS -XX:+UseG1GC -XX:G1HeapRegionSize=16m"
```

3. Если это не прод (на продажах дебаг выключен), обновить строку:

```
### Debugging options
JAVA_OPTS="$JAVA_OPTS -Xnoagent -Djava.compiler=NONE -Xdebug -Xrunjdwp:
transport=dt_socket,server=y,suspend=n,address=5005"
```

на

```
### Debugging options
JAVA_OPTS="$JAVA_OPTS -Xnoagent -Djava.compiler=NONE -Xdebug -Xrunjdwp:
transport=dt_socket,server=y,suspend=n,address=*:5005"
```

4. Проверить, что выставлены правильные настройки локализации, в файле должны присутствовать следующие настройки:

```
# Locale settings
JAVA_OPTS="$JAVA_OPTS -Duser.country=ru -Duser.language=ru -Duser.
timezone=GMT+3"
```

Если не задать эти настройки, то система будет использовать английскую локализацию.

Вместо часового пояса GMT+3 можно указать другой, если необходимо.

5. Проверить, что задан **CDI\_ROOT\_DIR** (обычно, это /home/cdi, но возможно использование иных каталогов):

```
# CDI root folder
JAVA_OPTS="$JAVA_OPTS -Dcdi.root.folder=/home/cdi"
```

Отдельно может быть задан каталог для поисковых индексов **CDI\_SEARCH\_DIR** — параметр **-Dcdi.search.folder**, и общий каталог для дедупликации — параметр **-Dcdi.dedup.folder**.

6. Найти настройки использования памяти, которые выглядят следующим образом:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms6g -Xmx12g -XX:PermSize=128m -XX:
MaxPermSize=256m -XX:-UseCodeCacheFlushing -XX:ReservedCodeCacheSize=256m"
```

и заменить их на новые настройки, сохранив старые значения **-Xms** и **-Xmx**, которые отвечают за пределы используемой памяти:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms6g -Xmx12g -XX:-UseCodeCacheFlushing -XX:
```

```
ReservedCodeCacheSize=256m -XX:MaxMetaspaceSize=512m -XX:-OmitStackTraceInFastThrow -XX:+UseCompressedOops -XX:+ExitOnOutOfMemoryError"
```

## Для «Фактора»

1. Отредактировать файл **standalone.conf**, перенеся в него недостающие настройки из предыдущей инсталляции JBoss для «Фактора» (**/opt/factor/jboss/bin/standalone.conf**).
2. Если раньше в опциях Java был прописан флаг **-ea**, проверить, что в новом файле **standalone.conf** он не используется. Использование этого флага в новой версии WildFly приведет к невозможности запустить «Фактор».
3. Перенести настройки используемой памяти из старого Wildfly с «Фактором».

Параметр `-Xms{x}g` отвечает за минимальное количество используемой памяти, параметр `-Xmx{X}g` — за максимальное.

Необходимо найти строки, указанные ниже, и заменить значения для `-Xms` `-Xmx` на те, которые были указаны в настройках старого Wildfly:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms8g -Xmx8g"
```

4. Указать пути к справочникам и индексам, если они были заданы в предыдущей инсталляции. Пример ( необходимо использовать пути, указанные в старом файле настроек):

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_INDEXES_PATH=/data/factor/indexes"
JAVA_OPTS="$JAVA_OPTS -DFACTOR_DICTIONARIES_PATH=/data/factor/dictionaries"
JAVA_OPTS="$JAVA_OPTS -DFACTOR_DEDUP_PATH=/data/factor/dedup"
```

## Перенос хешей дедупа и индексов «Фактора»

Скопировать папку **/data** из папки старой инсталляции **/opt/factor/jboss/standalone/** в папку **/opt/cdi/jboss\_new/standalone/**. Это поможет избежать полного перестроения хешей и индексов и лишних операций.

```
cp -a /opt/factor/jboss/standalone/data/ /opt/factor/jboss_new/standalone/
```

## Дополнительные настройки (при необходимости)

### Изменение смещения портов «Единого клиента» и «Фактора»

Если требуется изменить смещение портов «Фактора» относительно «Единого клиента» заданное в сборках WildFly (по умолчанию выставлено смещение 10000), необходимо отредактировать соответствующую настройку в **standalone.conf** для «Фактора»:

```
### Port offset
JAVA_OPTS="$JAVA_OPTS -Djboss.socket.binding.port-offset={port offset}"
```

Текущее используемое смещение указано в старом файле настроек.

Если смещение портов для «Фактора» не требуется (не указано в настройках предыдущей инсталляции), удалить ключ **-Djboss.socket.binding.port-offset={port offset}** из настроек или закомментировать его.

Отредактировать настройки отладки, указав в них адрес со смещением.

Например, адрес=5005, смещение=8, адрес+смещение=5013:

```
### Debugging options
#JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,
address=*:5013"
```

Необходимо, чтобы смещение всех портов на всех адресах было одинаковым.

Если необходимо задать смещение портов для «Единого клиента», то необходимо добавить ключ **-Djboss.socket.binding.port-offset={port offset}** в настройки **standalone.conf** для «Единого клиента» аналогично примеру, описанному выше для «Фактора», и задать остальные смещения портов по аналогии с примерами выше.

## Горячий резерв

В новый конфигурационный файл **/opt/cdi/jboss\_new/standalone/configuration/standalone.xml** перенести доменное имя второго экземпляра «ЕК» и его http-порт из блока **outbound-socket-binding name="node-sync-binding"** старого файла **/opt/cdi/jboss/standalone/configuration/standalone.xml**:

```
<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>
```

## Безопасное соединение (SSL)

Перенести настройки доступа к хранилищу ключей из **/opt/cdi/jboss/standalone/configuration/standalone.xml** в файл **/opt/cdi/jboss\_new/standalone/configuration/standalone.xml**.

В новый конфигурационный файл **/opt/cdi/jboss\_new/standalone/configuration/standalone.xml** скопировать блок **security-realm name="HttpsSecuredRealm"** из старого файла **/opt/cdi/jboss/standalone/configuration/standalone.xml**, удостовериться, что для коннектора **https-listener** блока **<server name="default-server">** указан



**параметр security-realm** — `HttpsSecuredRealm` (в данном примере указан общий параметр `<ssl>` блока `<server-identities>`, актуальные настройки находятся в конфигурационном файле старой инсталляции):

```
<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>
```

Блок `<security-realm name="HttpsSecuredRealm">` описывает настройки шифрования, блок коннектора `https-listener` описывает, когда его применять.

### Описание параметров `HttpsSecuredRealm`

- **keystore.path** — путь к хранилищу ключей, корневая директория определяется параметром **relative.to**;
- **alias** — алиас, под которым ключи доступны в хранилище;
- **keystore-password** — пароль к хранилищу;
- **key-password** — пароль к ключам (если не указан, используется **keystore-password**).

Скопировать в директорию `/opt/cdi/jboss_new/standalone/configuration/` файлы `*.jks`, на которые ссылается конфигурационный файл.

### *Доступ к внешней БД для «Фактора»*

Если для каких-либо справочников «Фактора» используется база данных, то доступ к ней настраивается с помощью файла `{customer}-ds.xml`, который находится в каталоге `standalone/deployments` инсталляции JBoss.

Если подобный **datasource**-файл использовался, то его необходимо скопировать из предыдущей инсталляции:

```
cp /opt/factor/jboss/standalone/deployments/factor-ds.xml /opt/factor/jboss_new/standalone
/deployments
```

---

## Переключение на новую версию WildFly

1. Остановить службы cdi и factor:

```
service cdi stop
service factor stop
```

2. Переместить логи из старой инсталляции в новую

```
mv /opt/cdi/jboss/standalone/log /opt/cdi/jboss_new/standalone/
mv /opt/factor/jboss/standalone/log /opt/factor/jboss_new/standalone/
```

Перенос логов позволит продолжить использование существующих логов, сохранит неразрывность информации и облегчит жизнь администраторам и поддержке. Логи могут занимать довольно много места и их лучше именно переносить, а не копировать.

3. Переименовать папки и заархивировать старые версии:

```
mv /opt/cdi/jboss /opt/cdi/jboss_10
mv /opt/factor/jboss /opt/factor/jboss_10

mv /opt/cdi/jboss_new /opt/cdi/jboss
mv /opt/factor/jboss_new /opt/factor/jboss

zip -rT /opt/cdi/old-jboss-cdi /opt/cdi/jboss_10
zip -rT /opt/factor/old-jboss-factor /opt/factor/jboss_10
```

4. Выдать соответствующим пользователям права доступа к каталогам jboss «Единого клиента» и jboss «Фактора», в том числе на редактирование:

```
chown -R cdi:cdi /opt/cdi/jboss
chown -R factor:factor /opt/factor/jboss
```

5. Назначить права на запуск исполняемых файлов:

```
find /opt/{factor,cdi}/jboss/ -type d -exec chmod 755 {} \;
find /opt/{factor,cdi}/jboss/ -type f -exec chmod 644 {} \;
find /opt/{factor,cdi}/jboss/ -type f -name "*.sh" -exec chmod 755 {} \;
```

6. Если раньше не осуществлялась настройка расширенного логирования работ системных служб, то необходимо в файлах **/etc/systemd/system/factor.service** и **/etc/systemd/system/cdi.service** заменить строку:

```
StandardOutput=null
```

на строки

```
StandardOutput=syslog
StandardError=syslog
```

и выполнить команду

---

```
systemctl daemon-reload
```

В противном случае дополнительных манипуляций со службами производить не нужно.

## ***Установка «Единого клиента» и «Фактора» и запуск служб***

1. Скопировать файлы `cdi-web-<имя заказчика>-<номер версии>-SNAPSHOT.war` и `factor-<имя заказчика>-<номер версии>.war`, переданные в поставке с данной инструкцией, в папки **`/opt/cdi/jboss/standalone/deployments`** и **`/opt/factor/jboss/standalone/deployments`** соответственно.
2. Запустить службы `cdi` и `factor`:

```
service cdi start  
service factor start
```

3. Удостовериться, что комплекс запустился и работает нормально.

# РАЗВЕРТЫВАНИЕ ПОСЛЕ ПОЛНОГО ОБНОВЛЕНИЯ ОС LINUX ДО ВЕРСИИ RED HAT 7/CENTOS7

## 1. Требования к системе

### ОС и программное обеспечение

- alma linux 8+ или Red Hat Enterprise Linux 8+, x64.
- Java SE Development Kit (OpenJDK) 11, с установленными актуальными обновлениями.

### Установка и настройка

1. Активирована служба ssh.

### Доступы и права

1. Открыт доступ к серверу СУБД по порту, на котором слушает Oracle (1521) или PostgreSQL (5432).
2. Открыт доступ к серверам приложений для очистки данных по порту 8080.
3. Открыт доступ к серверу Active Directory по порту 3269.
4. Открыт доступ к SMTP-серверу по порту 25.
5. Открыты порты:
  - 22 (ssh);
  - 8080 (HTTP-порт «Единого клиента»);
  - 18080 (HTTP-порт «Фактора»);
  - 9990 (порт для мониторинга «Единого клиента»);
  - 19990 (порт для мониторинга «Фактора»);
  - 9001 (порт для синхронизации нод при наличии горячего резерва).
6. Установка антивируса запрещена.

## 2. Настройка системы

### *Создание пользователей для «Единого клиента» и «Фактора»*

Создайте пользователей, под которыми будут работать сервера приложений «Единого клиента» и «Фактора» — **cdi** и **factor**:

```
useradd cdi
useradd factor
passwd -l cdi
passwd -l factor
```

Пользователей нужно объединить в одну группу:

```
groupadd hfl_cdi
usermod -a -G hfl_cdi cdi
usermod -a -G hfl_cdi factor
```

## **Запрет на выделение памяти сверх того, что есть и отключение SWAP**

1. Откройте файл **/etc/sysctl.conf** и добавьте в него строки:

```
/etc/sysctl.conf
#Do not overcommit memory
vm.overcommit_memory=2
vm.overcommit_ratio=100

#Max map count
vm.max_map_count = 16777216

# Low swap level
vm.swappiness = 10
```

2. Использование SWAP сильно тормозит работу приложений, поэтому его использование лучше честно отключить. Открывайте файл **/etc/fstab** и закомментируйте в нем монтирования раздела swar вида. На DEV-стендах из соображения экономии ресурсов swar лучше не отключать.

```
/etc/fstab
%SOME_TEXT% swap          swap defaults 0 0
```

3. Перезагрузите операционную систему.
4. Проверьте параметры с помощью команды:

```
sysctl -p
либо
systemctl -p
```

Расшифровка параметров:

**vm.swappiness** ? при каком значении нужно пытаться перекладывать куски памяти в swar (в процентах). По умолчанию это 60 (то есть если осталось свободно 60 процентов оперативки, то начать пытаться перекладывать давно неиспользуемые куски в swar). Установка этого параметра в 10 позволит максимально использовать оперативку, без задействования swar-a (0

ставить нельзя, так как хоть по документации это и отключает swar вообще, но многие ОС это игнорируют)

## Увеличение предела открытых дескрипторов файлов

1. Откройте файл **/etc/security/limits.conf** и добавьте в него строки:

```

/etc/security/limits.conf
cdi      hard  nofile 65535
cdi      soft  nofile 65535
cdi      hard  nproc  8192
cdi      soft  nproc  4096
cdi      hard  as     unlimited
cdi      soft  as     unlimited
cdi      hard  rss   unlimited
cdi      soft  rss   unlimited
factor   hard  nofile 65535
factor   soft  nofile 65535
factor   hard  nproc  8192
factor   soft  nproc  4096
factor   hard  as     unlimited
factor   soft  as     unlimited
factor   hard  rss   unlimited
factor   soft  rss   unlimited
```

где **factor** и **cdi** — имена пользователей, под которыми работают «Фактор» и «Единый клиент».

2. Перезагрузите ОС, чтобы настройки вступили в силу
3. Залогиньтесь под пользователем «Фактора» и убедитесь, что настройки применены:

```

su factor -s /bin/sh

sysctl vm.max_map_count
ulimit -n
ulimit -u
ulimit -v
ulimit -m
```

Должно получиться следующее:

```

$ sysctl vm.max_map_count
vm.max_map_count = 16777216

$ ulimit -n
65535

$ ulimit -u
4096

$ ulimit -v
unlimited

$ ulimit -m
unlimited
```

4. Залогиньтесь под пользователем «Единого клиента» и убедитесь, что настройки применены:

```
su cdi -s /bin/sh
```

```
$ ulimit -n  
65535
```

```
$ ulimit -u  
4096
```

```
$ ulimit -v  
unlimited
```

```
$ ulimit -m  
unlimited
```

## Настройка для работы с SSD-дисками

### Отключение времени модификации файлов

Если приложение часто и многократно пишет и читает файлы (именно так делают «ЕК», «Фактор», «Подсказки») то на файловых системах **ext3** и **ext4** нужно отключить дополнительные функции работы метаданными файлов.

Для этого нужно изменить параметры монтирования диска, добавив следующие опции:

1. **noatime** — полностью отключает запись времени доступа к файлу. Большинство программ не используют это поле.
2. **data=ordered** — журналирует только изменения метаданных, но обновления данных сбрасываются на жесткий диск до совершения транзакции. Данные записываются не атомарно, но этот режим гарантирует, что после падения файлы не будут содержать блоки данных из устаревших файлов.

В итоге строка в `/etc/fstab` должна выглядеть примерно следующим образом (**sdX** — устройство SSD)

```
                                /etc/fstab  
# <fs> <mountpoint> <type> <opts> <dump/pass>  
/dev/sdX /opt ext4 defaults,noatime,data=ordered,errors=remount-ro 0 2
```


### Выключите IO Scheduler для SSD

Выполните команду и добавьте ее в скрипт автозапуска

```
echo noop > /sys/block/sdX/queue/scheduler
```

для каждого устройства (заменяя **sdX** на нужное имя)

### 3. Установка Java

 Для работы системы должен использоваться **openJDK 11** версии не ниже **11.0.4**

#### Установочный пакет

В случае выбора операционной системы Linux приоритетным вариантом установки является установка из репозитория ОС. Альтернативно возможно использование архива AdoptOpenJDK.

Установочный пакет можно скачать с сайта проекта AdoptOpenJDK по ссылке <https://adoptopenjdk.net/?variant=openjdk11&jvmVariant=hotspot>:

- удостовериться, что выбрана версия **OpenJDK 11 (LTS)** и **JVM Hotspot**;
- Нажать кнопку Latest release;
- Выбрать вид тип установочного пакета, подходящего для ОС сервера.
- Скачать установочный пакет.

#### Установка JDK

Приоритетным является вариант установки через репозиторий ОС.

Пример (CentOS 7 и Red Hat 7):

```
sudo yum install java-11-openjdk-devel
```

После этого можно перейти к проверке правильности установки JDK.

Если доступа к репозиториям нет, то возможно использовать альтернативный вариант установки: установка вручную из архива AdoptOpenJDK (при необходимости, заменить ссылку [https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/OpenJDK11U-jdk\\_x64\\_linux\\_hotspot\\_11.0.4\\_11.tar.gz](https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz) на ссылку на более новую версию установочного пакета):

```
mkdir /usr/java/  
cd /usr/java/  
wget https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.4+11/  
/OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz  
tar zxvf OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz  
rm OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz  
alternatives --install /usr/bin/java java /usr/java/jdk-11.0.4+11/bin/java 2  
alternatives --install /usr/bin/jar jar /usr/java/jdk-11.0.4+11/bin/jar 2  
alternatives --install /usr/bin/javac javac /usr/java/jdk-11.0.4+11/bin/javac 2  
alternatives --set java /usr/java/jdk-11.0.4+11/bin/java  
alternatives --set jar /usr/java/jdk-11.0.4+11/bin/jar  
alternatives --set javac /usr/java/jdk-11.0.4+11/bin/javac
```



## Проверка правильности установки JDK

Выполнить в командной строке команду

```
javac -version
```

Должно появиться сообщение вида

```
javac 11.0.4
```

Версия **JDK** должна соответствовать версии установочного пакета.

Также нужно проверить версию самой java-машины:

```
java -version
```

Она должна быть идентична версии javac.

## Установка переменных окружения

Для пользователей **cdi**, **factor** в файлах **/home/cdi/.bash\_profile**, **/home/cdi/.bash\_profile** (также в **/etc/profile** или **/etc/skel/profile**) добавьте строки, предварительно заменив **/usr/java/jdk-11.0.4+11/** на полный путь к каталогу, в который установлен JDK:

```
export JAVA_HOME=/usr/java/jdk-11.0.4+11/  
export PATH=$JAVA_HOME/bin:$PATH
```

После добавления строк выполнить одну из команд, приведенных ниже:

```
source etc/profile  
source /etc/skel/profile
```

Проверить, что путь был добавлен, можно выполнив команду:

```
echo $PATH
```

В ответе должен быть замечен путь к каталогу с Java.

После этого нужно повторно проверить версии java и javac.

## 4. Установка WildFly

### Установочный пакет

Установочные пакеты Jboss поставляются совместно с системой в архивах:

- **wildfly-16.0.0.Final-8080.zip** — для «Единого клиента».
- **wildfly-16.0.0.Final-8080-HotReserve.zip** — для «Единого клиента» с горячим резервированием.
- **wildfly-16.0.0.Final-18080.zip** — для «Фактора».

## Установка JBOSS

1. Распаковать архив с WildFly в каталог **JBOSS\_HOME**. Здесь и далее используется каталог **/opt** как рекомендуемый. Может быть заменен на /home или /data, если основная память примонтирована сюда.

```
unzip wildfly-16.0.0.Final-8080.zip -d /opt/cdi
unzip wildfly-16.0.0.Final-18080.zip -d /opt/factor
```

```
mv /opt/cdi/wildfly* /opt/cdi/appserver
mv /opt/factor/wildfly* /opt/factor/appserver
```

2. Назначить созданным директориям соответствующих владельцев – **cdi** для **/opt/cdi/** и **factor** для **/opt/factor/**

```
chown -R cdi:cdi /opt/cdi/
chown -R factor:factor /opt/factor/
```

3. Назначить права на запуск исполняемых файлов:

```
find /opt/{factor,cdi}/appserver/ -type d -exec chmod 755 {} \;
find /opt/{factor,cdi}/appserver/ -type f -exec chmod 644 {} \;
find /opt/{factor,cdi}/appserver/ -type f -name "*.sh" -exec chmod 755 {} \;
```

## Настройка сервисов

1. Создать директорию в **/etc** с названием будущей службы (**factor, cdi**), скопировать файлы

```
mkdir /etc/cdi
mkdir /etc/factor

cp /opt/cdi/appserver/docs/contrib/scripts/systemd/wildfly.conf /etc/cdi/
cp /opt/cdi/appserver/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/cdi.service
cp /opt/cdi/appserver/docs/contrib/scripts/systemd/launch.sh /opt/cdi/appserver/bin/

cp /opt/factor/appserver/docs/contrib/scripts/systemd/wildfly.conf /etc/factor/
cp /opt/factor/appserver/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/systemfactor.service
cp /opt/factor/appserver/docs/contrib/scripts/systemd/launch.sh /opt/factor/appserver/bin/
```

2. В `/etc/systemd/system/factor.service` и `/etc/systemd/system/cdi.service` переменные заданы по умолчанию. При необходимости заменить параметры **Limit\*** и путь к **launch.sh** (параметр **ExecStart**). При необходимости заменить пути на используемые при установке

```
cdi.service

[Unit]
Description=CDI WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=-/etc/cdi/wildfly.conf
User=cdi
OOMScoreAdjust=-1000
PIDFile=/var/run/cdi/wildfly.pid
ExecStart=/opt/cdi/appserver/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG
$WILDFLY_BIND
StandardOutput=syslog
StandardError=syslog
LimitNOFILE=65535
LimitNPROC=8192
LimitAS=infinity
LimitRSS=infinity

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

```
factor.service

[Unit]
Description=Factor WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=-/etc/factor/wildfly.conf
User=factor
OOMScoreAdjust=-1000
PIDFile=/var/run/factor/wildfly.pid
ExecStart=/opt/factor/appserver/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG
$WILDFLY_BIND
StandardOutput=syslog
StandardError=syslog
LimitNOFILE=65535
LimitNPROC=8192
LimitAS=infinity
LimitRSS=infinity

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

3. Указать путь к домашней директории Jboss в `/opt/factor/appserver/bin/launch.sh`

```
WILDFLY_HOME="/opt/factor/appserver"
```

и в **/opt/cdi/appserver/bin/launch.sh**

```
WILDFLY_HOME="/opt/cdi/appserver"
```

4. На файлы **launch.sh** и **standalone.sh** выдать права на запуск:

```
chmod +x /opt/cdi/appserver/bin/launch.sh
chmod +x /opt/cdi/appserver/bin/standalone.sh

chmod +x /opt/factor/appserver/bin/launch.sh
chmod +x /opt/factor/appserver/bin/standalone.sh
```

5. Перезагрузить список доступных сервисов, чтобы systemd мог управлять новым сервисом:

```
systemctl daemon-reload
```

6. Добавить службы в автозапуск:

```
systemctl enable cdi.service
systemctl enable factor.service
```

## 5. Создание рабочих каталогов приложения

1. Создайте рабочий каталог CDI и дайте на него права пользователю **cdi**:

```
mkdir -p /opt/cdi
chown -R cdi:cdi /opt/cdi
```

2. Создайте каталог, используемый при поиске дубликатов:

```
mkdir /opt/cdi/dedup
```

3. Если вы используете отдельных пользователей для служб CDI и FACTOR, выдайте права на каталог для дубликатов обоим пользователям.

Пример, когда пользователи входят в одну группу с именем "hfl\_cdi":

```
chgrp hfl_cdi /opt/cdi/dedup
```

# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ НА OPENJDK 11 И WILDFLY 16 (WINDOWS)

- Шаг 1. Установить OpenJDK 11
  - Установка openJDK
  - Обновление значения JAVA\_HOME
- Шаг 2. Обновить Wildfly
  - Установочный пакет
  - Установка WildFly
  - Настройка параметров доступа к БД системы
  - Настройка параметров запуска WildFly
    - Для «Единого клиента»
    - Для «Фактора»
  - Перенос хешей дедапа и индексов «Фактора»
  - Дополнительные настройки (при необходимости)
    - Изменение смещения портов «Единого клиента» и «Фактора»
    - Горячий резерв
    - Безопасное соединение (SSL)
    - Доступ к внешней БД для «Фактора»
  - Переключение Wildfly на новую версию
- Шаг 3. Перейти к инструкции по обновлению «Единого клиента» на новую версию

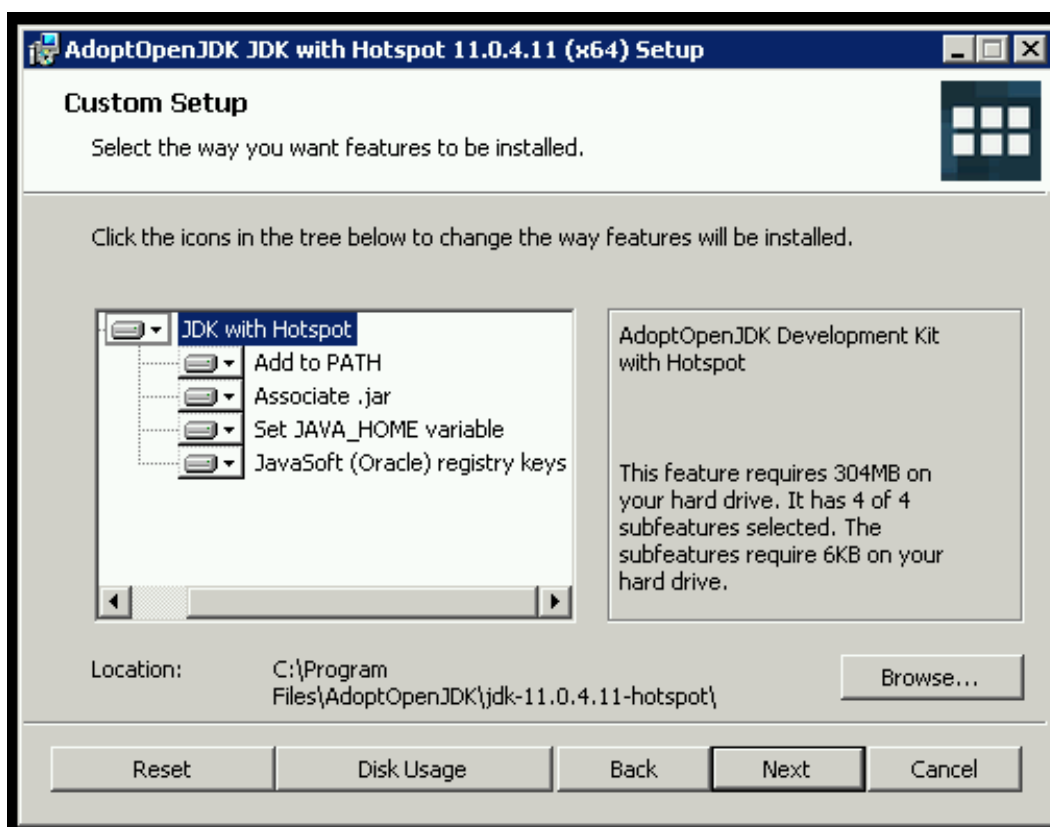
## Шаг 1. Установить OpenJDK 11

### *Установка openJDK*

**При обновлении обязательно проверьте, что устанавливается именно openJDK**

Загрузить установочный файл.

Запустить установочный файл, принять лицензионное соглашение, указать путь и устанавливаемые компоненты как на скриншоте:

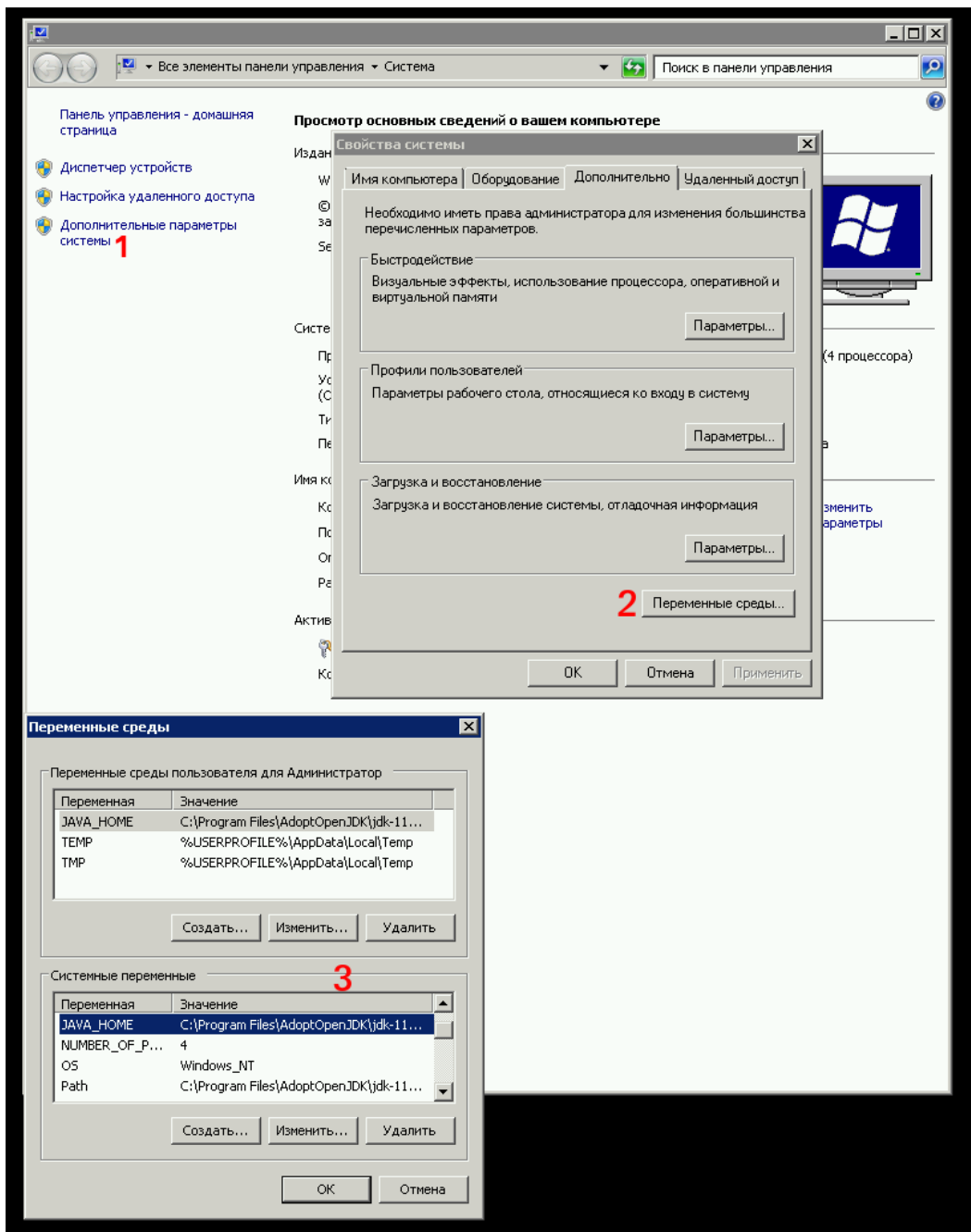


## Обновление значения *JAVA\_HOME*

Выполнить следующие команды в командной строке:

```
setx JAVA_HOME "C:\Program Files\AdoptOpenJDK\jdk-11.0.4.11-hotspot"  
setx PATH "%PATH%;%JAVA_HOME%\bin"
```

Проверить, что переменные среды обновились можно выполнив в командной строке команду `set` или через графический интерфейс (панель управления ? система ? Дополнительные параметры системы ? вкладка "Дополнительно" ? переменные среды):



При необходимости, отредактировать переменные через графический интерфейс.

Проверить, что возвращаются правильные версии java и javac (11.0.4) и что они совпадают:

```
javac -version  
java -version
```

## Шаг 2. Обновить Wildfly

## Установочный пакет

- **wildfly-16.0.0.Final-8080.zip** — для «Единого клиента».
- **wildfly-16.0.0.Final-8080-HotReserve.zip** — для «Единого клиента» с горячим резервированием.
- **wildfly-16.0.0.Final-18080.zip** — для «Фактора».

Данные пакеты можно получить из указанных ресурсов:

- скачать из нашего [общего хранилища](#);
- взять архивы в составе поставки.

## Установка WildFly

Распаковать архивы с WildFly 16 в директорию рядом с инсталляциями WildFly 10.

## Настройка параметров доступа к БД системы

Скопировать файл **cdi-oracle-ds.xml**, **cdi-postgresql-ds.xml** или **cdi-mariadb-ds.xml** из инсталляции WildFly для «Единого клиента» в директорию **wildfly-16.0.0.Final-cdi\standalone\deployments**

**При необходимости настроить шифрование пароля к БД.**

Определить, что система использует шифрование пароля к БД можно по содержанию файлов с параметрами доступа к БД, указанным выше. Если в файле логин и пароль указаны в явном виде

```
<user-name>username</user-name>  
<password>password</password>
```

то настройка шифрования не требуется, если указано использование зашифрованного пароля

```
<security-domain>EncryptedPassword</security-domain>
```

то потребуется настройка согласно инструкции, ссылка на которую дана выше.

## Настройка параметров запуска WildFly

### Для «Единого клиента»

1. В директорию **wildfly-16.0.0.Final-cdi\bin** скопировать файл **standalone.conf.bat** из предыдущей инсталляции WildFly для «Единого клиента».
2. В скопированном файле необходимо заменить старые настройки GC:



```
: Garbage collection tuning
set "JAVA_OPTS=%JAVA_OPTS% -XX:+UseConcMarkSweepGC -XX:
+CMSIncrementalMode -XX:+CMSIncrementalPacing -XX:+CMSClassUnloadingEnabled -
XX:CMSIncrementalDutyCycleMin=0 -XX:CMSIncrementalDutyCycle=10"
```

на новые

```
: Garbage collection tuning
set "JAVA_OPTS=%JAVA_OPTS% -XX:+UseG1GC -XX:G1HeapRegionSize=16m"
```

3. Проверить, что выставлены правильные настройки локализации, в файле должны присутствовать следующие настройки:

```
: Locale settings
set "JAVA_OPTS=%JAVA_OPTS% -Duser.country=ru -Duser.language=ru -Duser.
timezone=GMT+3"
```

Если не задать эти настройки, то система будет использовать английскую локализацию.

Вместо часового пояса GMT+3 можно указать другой, если необходимо.

4. Если отдельно задавался параметр **CDI\_ROOT\_DIR** (обычно, это C:\cdi, но возможно использование иных каталогов):

```
set "JAVA_OPTS=%JAVA_OPTS% -Dcdi.root.folder=C:\cdi"
```

Отдельно может быть задана папка для поисковых индексов **CDI\_SEARCH\_DIR** — параметр **-Dcdi.search.folder**, и общая папка для дедупликации — параметр **-Dcdi.dedup.folder**.

5. Найти настройки использования памяти, которые выглядят следующим образом:

```
: Basic settings, memory usage
set "JAVA_OPTS=%JAVA_OPTS% -server -Xms6g -Xmx12g -XX:-UseCodeCacheFlushing -
XX:ReservedCodeCacheSize=256m"
```

и заменить их на новые настройки, сохранив старые значения **-Xms** и **-Xmx**, которые отвечают за пределы используемой памяти:

```
: Basic settings, memory usage
set "JAVA_OPTS=%JAVA_OPTS% -server -Xms6g -Xmx12g -XX:-UseCodeCacheFlushing -
XX:ReservedCodeCacheSize=256m -XX:MaxMetaspaceSize=512m -XX:-
OmitStackTraceInFastThrow -XX:+UseCompressedOops -XX:+ExitOnOutOfMemoryError"
```

## Для «Фактора»

В директории **wildfly-16.0.0.Final-factor\bin** отредактировать файл **standalone.conf.bat** и перенести в него недостающие настройки из предыдущей инсталляции WildFly для «Фактора»:

1. Если раньше в опциях Java был прописан флаг **-ea**, проверить, что в новом файле **standalone.conf.bat** он не используется. Использование этого флага в новой версии WildFly приведет к невозможности запустить «Фактор».

2. Перенести настройки используемой памяти из старого Wildfly с «Фактором».

Параметр `-Xms{x}g` отвечает за минимальное количество используемой памяти, параметр `-Xmx{X}g` — за максимальное.

Необходимо найти строки, указанные ниже, и заменить значения для `-Xms` `-Xmx` на те, которые были указаны в настройках старого Wildfly:

```
rem # Basic settings, memory usage
set "JAVA_OPTS_MEMORY=-server -Xms8g -Xmx8g"
```

3. Указать пути к справочникам и индексам, если они были заданы в предыдущей инсталляции. Пример (необходимо использовать пути, указанные в старом файле настроек):

```
rem # Source of blacklists
set "JAVA_OPTS=%JAVA_OPTS% -DFACTOR_PASSPORT_INVALID_PATH=C:
\cdi\invalid_doc -Dfactor.blacklistPath=C:\cdi\blacklist"
```

## *Перенос хешей дедана и индексов «Фактора»*

Скопировать папку **data** из папки старой инсталляции **wildfly-10.1.0.Final-18080\standalone** в папку **wildfly-16.0.0.Final-factor\standalone**. Это поможет избежать полного перестроения хешей и индексов и лишних операций.

## *Дополнительные настройки (при необходимости)*

### *Изменение смещения портов «Единого клиента» и «Фактора»*

Если требуется изменить смещение портов «Фактора» относительно «Единого клиента», заданное в сборках WildFly (по умолчанию выставлено смещение 10000), необходимо отредактировать соответствующую настройку в **standalone.conf.bat** для «Фактора»:

```
rem # Port offset
set "JAVA_OPTS=%JAVA_OPTS% -Djboss.socket.binding.port-offset={port offset}"
```

Текущее используемое смещение указано в старом файле настроек.

Если смещение портов для «Фактора» не требуется (не указано в настройках предыдущей инсталляции), удалить ключ **-Djboss.socket.binding.port-offset={port offset}** из настроек или закомментировать его.

Отредактировать настройки отладки, указав в них адрес со смещением.

Например, адрес=5005, смещение=8, адрес+смещение=5013:

```
rem # Debugging options
rem # set "JAVA_OPTS_JPDA=-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,
address=*:5013"
```

В папке **wildfly-16.0.0.Final-factor\bin\service** отредактировать файл **service.bat**.

Найти строку

```
set CONTROLLER=localhost:19990
```

и заменить 19990 на 9990+смещение.

Например, если смещение=8,  $9990+8=9998$ :

```
set CONTROLLER=localhost:9998
```

Необходимо, чтобы смещение всех портов на всех адресах было одинаковым.

Если необходимо задать смещение портов для «**Единого клиента**», то необходимо **добавить ключ `-Djboss.socket.binding.port-offset={port offset}`** в настройки **standalone.conf.bat** для «Единого клиента» аналогично примеру, описанному выше для «Фактора», и задать остальные смещения портов по аналогии с примерами выше, включая смещение в **wildfly-16.0.0.Final-cdi\bin\service\service.bat**.

### Горячий резерв

В новый конфигурационный файл **wildfly-16.0.0.Final-cdi\standalone\configuration\standalone.xml** скопировать блок **outbound-socket-binding name="node-sync-binding"** из старого файла **wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml**, удостовериться, что доменное имя и порт заданы правильно, смещение порта (если используется) учтено:

```
<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>
```

### Безопасное соединение (SSL)

Перенести **настройки доступа к хранилищу ключей** из **wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml** в файл **wildfly-16.0.0.Final-cdi\standalone\configuration\standalone.xml**.

В новый конфигурационный файл **wildfly-16.0.0.Final-cdi\standalone\configuration\standalone.xml** скопировать блок **security-realm name="HttpsSecuredRealm"** из старого файла **wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml**, удостовериться, что для **коннектора https-listener** блока **<server name="default-server">** **указан параметр security-realm="HttpsSecuredRealm"** (в данном примере указан общий параметр **<ssl>** блока **<server-identities>**, актуальные настройки находятся в конфигурационном файле старой инсталляции):

```

<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>

```

Блок **<security-realm name="HttpsSecuredRealm">** описывает настройки шифрования, блок коннектора **https-listener** описывает, когда его применять.

### Описание параметров HttpsSecuredRealm

- **keystore.path** — путь к хранилищу ключей, корневая директория определяется параметром **relative.to**;
- **alias** — алиас, под которым ключи доступны в хранилище;
- **keystore-password** — пароль к хранилищу;
- **key-password** — пароль к ключам (если не указан, используется **keystore-password**).

Скопировать в директорию **wildfly-16.0.0.Final-cdi\standalone\configuration\** файлы **\*.jks**, на которые ссылается конфигурационный файл.

### *Доступ к внешней БД для «Фактора»*

Если для каких-либо справочников «Фактора» используется база данных, то доступ к ней настраивается с помощью файла **{customer}-ds.xml**, который находится в каталоге **standalone\deployments** инсталляции JBoss.

Если подобный **datasource**-файл использовался, то его необходимо скопировать из предыдущей инсталляции в папку **wildfly-16.0.0.Final-factor\standalone\deployments**

## *Переключение Wildfly на новую версию*

1. Остановить службы CDI и FACTOR.

2. Перенести логи из старых инсталляций «Единого клиента» и «Фактора» (находятся в папке **standalone\log**) в новую инсталляцию( **wildfly-16.0.0.Final-cdi\standalone\log** для «Единого клиента» и **wildfly-16.0.0.Final-factor\standalone\log** для «Фактора»). Это позволит продолжить использовать существующие логи, обеспечит неразрывность данных и облегчит жизнь администраторам и поддержке. Логи могут занимать много места, так что лучше их именно перемещать, а не копировать.
3. Заархивировать предыдущие инсталляции WildFly для «Единого клиента» и «Фактора».
4. Удалить старые системные службы запуска WildFly:

```
sc delete cdi
sc delete factor
```

5. Установить новые системные службы (команду следует выполнять из директории **\bin\service** WildFly 16 «Единого клиента»):

```
cd wildfly-16.0.0.Final-cdi\bin\service
service.bat install
cd ..\..\..\wildfly-16.0.0.Final-factor\bin\service
service.bat install
```

*Примечание. Если возникает ошибка " **ERROR: The service already exists** ", то перед установкой системных служб перезагрузить ОС.*

6. Скопировать файлы **cdi-web-*<имя заказчика>-*<номер версии>*-SNAPSHOT.war*** и **factor-*<имя заказчика>-*<номер версии>*.war*** из поставки в папки **wildfly-16.0.0.Final-cdi\standalone\deployments** и **wildfly-16.0.0.Final-factor\standalone\deployments** соответственно.
7. Запустить службы CDI и FACTOR.
8. Проверить, что программы запущены и работают.

### Шаг 3. Перейти к инструкции по обновлению «Единого клиента» на новую версию

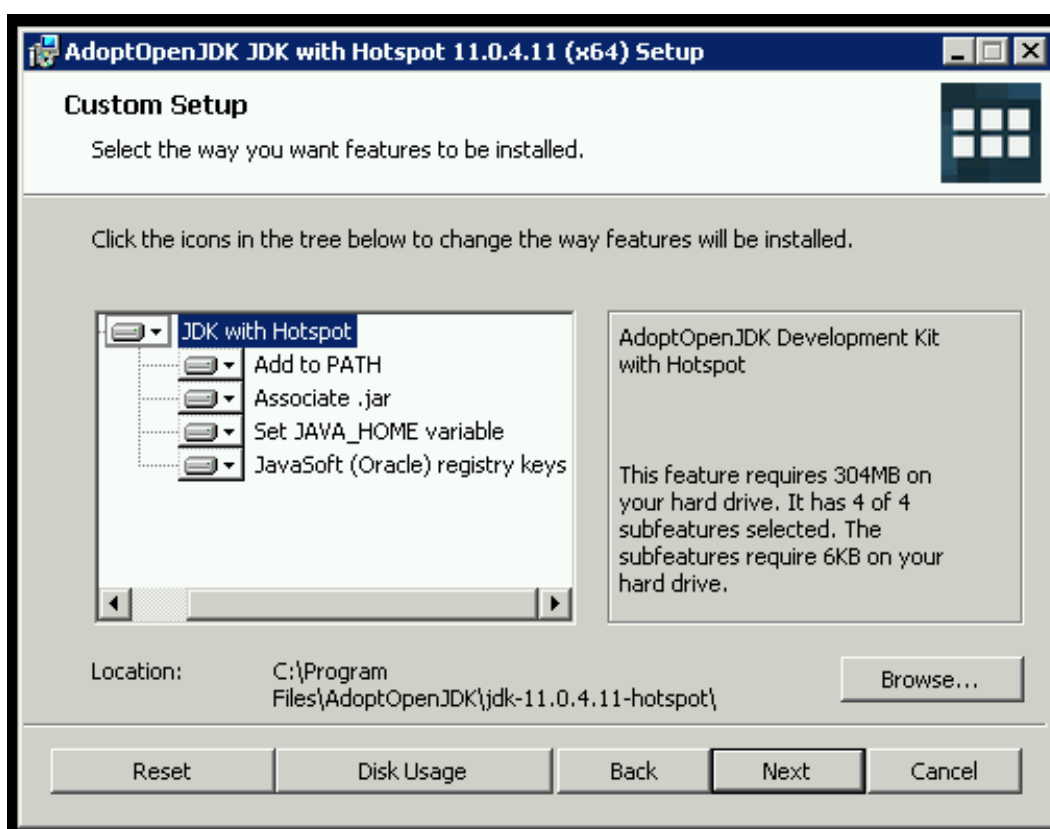
# ИНСТРУКЦИЯ ПО ПЕРЕХОДУ НА OPENJDK 11 (WINDOWS)

## Установка openJDK

При обновлении обязательно проверьте, что устанавливается именно openJDK

Загрузить установочный файл.

Запустить установочный файл, принять лицензионное соглашение, указать путь и устанавливаемые компоненты как на скриншоте:

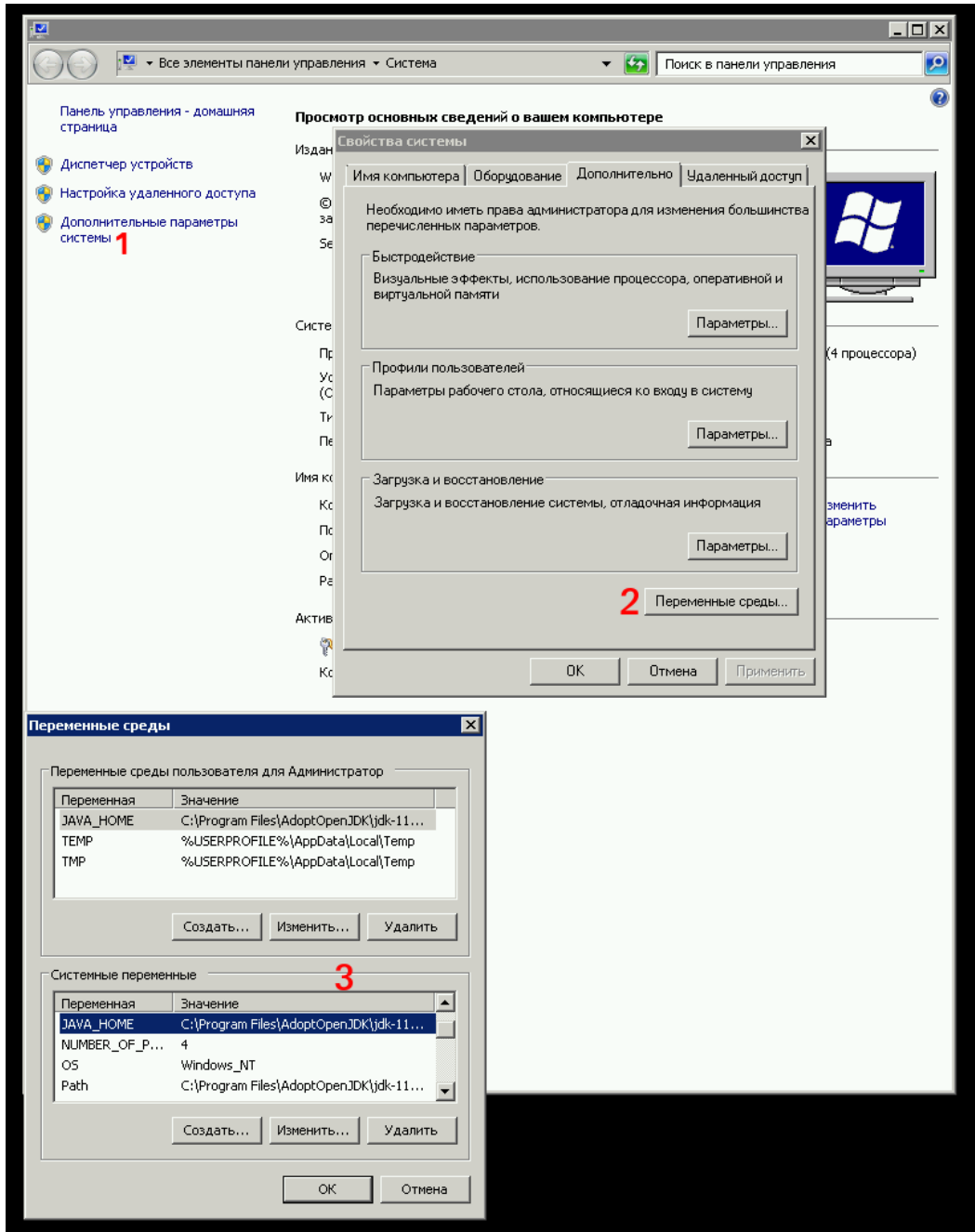


## Обновление значения JAVA\_HOME

Выполнить следующие команды в командной строке:

```
setx JAVA_HOME "C:\Program Files\AdoptOpenJDK\jdk-11.0.4.11-hotspot"  
setx PATH "%PATH%;%JAVA_HOME%\bin"
```

Проверить, что переменные среды обновились можно выполнив в командной строке команду set или через графический интерфейс (панель управления ? система ? Дополнительные параметры системы ? вкладка "Дополнительно" ? переменные среды):



При необходимости, отредактировать переменные через графический интерфейс.

Проверить, что возвращаются правильные версии java и javac (11.0.4) и что они совпадают:

```
javac -version  
java -version
```





# ИНСТРУКЦИЯ ДЛЯ WINDOWS (ПЕРЕХОД НА WILDFLY 16)

## Установочный пакет

- **wildfly-16.0.0.Final-8080.zip** — для «Единого клиента».
- **wildfly-16.0.0.Final-8080-HotReserve.zip** — для «Единого клиента» с горячим резервированием.
- **wildfly-16.0.0.Final-18080.zip** — для «Фактора».

Данные пакеты можно получить из указанных ресурсов:

- скачать из нашего [общего хранилища](#);
- взять архивы в составе поставки.

## Установка WildFly

Распаковать архивы с WildFly 16 в директорию рядом с инсталляциями WildFly 10.

## Настройка параметров доступа к БД системы

Скопировать файл **cdi-oracle-ds.xml**, **cdi-postgresql-ds.xml** или **cdi-mariadb-ds.xml** из инсталляции WildFly для «Единого клиента» в директорию **wildfly-16.0.0.Final-cdi\standalone\deployments**

**При необходимости настроить шифрование пароля к БД.**

Определить, что система использует шифрование пароля к БД можно по содержанию файлов с параметрами доступа к БД, указанным выше. Если в файле логин и пароль указаны в явном виде

```
<user-name>username</user-name>
<password>password</password>
```

то настройка шифрования не требуется, если указано использование зашифрованного пароля

```
<security-domain>EncryptedPassword</security-domain>
```

то потребуется настройка согласно инструкции, ссылка на которую дана выше.

## Настройка параметров запуска WildFly

### Для «Единого клиента»

1. В директорию **wildfly-16.0.0.Final-cdi\bin** скопировать файл **standalone.conf.bat** из предыдущей инсталляции WildFly для «Единого клиента».
2. В скопированном файле необходимо заменить старые настройки GC:

```
: Garbage collection tuning
set "JAVA_OPTS=%JAVA_OPTS% -XX:+UseConcMarkSweepGC -XX:
+CMSIncrementalMode -XX:+CMSIncrementalPacing -XX:+CMSClassUnloadingEnabled -
XX:CMSIncrementalDutyCycleMin=0 -XX:CMSIncrementalDutyCycle=10"
```

на новые

```
: Garbage collection tuning
set "JAVA_OPTS=%JAVA_OPTS% -XX:+UseG1GC -XX:G1HeapRegionSize=16m"
```

3. Проверить, что выставлены правильные настройки локализации, в файле должны присутствовать следующие настройки:

```
: Locale settings
set "JAVA_OPTS=%JAVA_OPTS% -Duser.country=ru -Duser.language=ru -Duser.
timezone=GMT+3"
```

Если не задать эти настройки, то система будет использовать английскую локализацию.

Вместо часового пояса GMT+3 можно указать другой, если необходимо.

4. Если отдельно задавался параметр **CDI\_ROOT\_DIR** (обычно, это C:\cdi, но возможно использование иных каталогов):

```
set "JAVA_OPTS=%JAVA_OPTS% -Dcdi.root.folder=C:\cdi"
```

Отдельно может быть задана папка для поисковых индексов **CDI\_SEARCH\_DIR** — параметр **-Dcdi.search.folder**, и общая папка для дедупликации — параметр **-Dcdi.dedup.folder**.

5. Найти настройки использования памяти, которые выглядят следующим образом:

```
: Basic settings, memory usage
set "JAVA_OPTS=%JAVA_OPTS% -server -Xms6g -Xmx12g -XX:-UseCodeCacheFlushing -
XX:ReservedCodeCacheSize=256m"
```

и заменить их на новые настройки, сохранив старые значения **-Xms** и **-Xmx**, которые отвечают за пределы используемой памяти:

```
: Basic settings, memory usage
set "JAVA_OPTS=%JAVA_OPTS% -server -Xms6g -Xmx12g -XX:-UseCodeCacheFlushing -
XX:ReservedCodeCacheSize=256m -XX:MaxMetaspaceSize=512m -XX:-
OmitStackTraceInFastThrow -XX:+UseCompressedOops -XX:+ExitOnOutOfMemoryError"
```

## Для «Фактора»

В директории **wildfly-16.0.0.Final-factor\bin** отредактировать файл **standalone.conf.bat** и перенести в него недостающие настройки из предыдущей инсталляции WildFly для «Фактора»:

1. Если раньше в опциях Java был прописан флаг **-ea**, проверить, что в новом файле **standalone.conf.bat** он не используется. Использование этого флага в новой версии WildFly приведет к невозможности запустить «Фактор».
2. Перенести настройки используемой памяти из старого Wildfly с «Фактором».

Параметр `-Xms{x}g` отвечает за минимальное количество используемой памяти, параметр `-Xmx{X}g` — за максимальное.

Необходимо найти строки, указанные ниже, и заменить значения для `-Xms` `-Xmx` на те, которые были указаны в настройках старого Wildfly:

```
rem # Basic settings, memory usage
set "JAVA_OPTS_MEMORY=-server -Xms8g -Xmx8g"
```

3. Указать пути к справочникам и индексам, если они были заданы в предыдущей инсталляции. Пример (необходимо использовать пути, указанные в старом файле настроек):

```
rem # Source of blacklists
set "JAVA_OPTS=%JAVA_OPTS% -DFACTOR_PASSPORT_INVALID_PATH=C:\
cdi\invalid_doc -Dfactor.blacklistPath=C:\cdi\blacklist"
```

## Перенос хешей дедана и индексов «Фактора»

Скопировать папку **data** из папки старой инсталляции **wildfly-10.1.0.Final-18080\standalone** в папку **wildfly-16.0.0.Final-factor\standalone**. Это поможет избежать полного перестроения хешей и индексов и лишних операций.

## Дополнительные настройки (при необходимости)

### Изменение смещения портов «Единого клиента» и «Фактора»

Если требуется изменить смещение портов «Фактора» относительно «Единого клиента», заданное в сборках WildFly (по умолчанию выставлено смещение 10000), необходимо отредактировать соответствующую настройку в **standalone.conf.bat** для «Фактора»:

```
rem # Port offset
set "JAVA_OPTS=%JAVA_OPTS% -Djboss.socket.binding.port-offset={port offset}"
```

Текущее используемое смещение указано в старом файле настроек.

Если смещение портов для «Фактора» не требуется (не указано в настройках предыдущей инсталляции), удалить ключ **-Djboss.socket.binding.port-offset={port offset}** из настроек или закомментировать его.

Отредактировать настройки отладки, указав в них адрес со смещением.

Например, адрес=5005, смещение=8, адрес+смещение=5013:

```
rem # Debugging options
rem # set "JAVA_OPTS_JPDA=-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,
address=*:5013"
```

В папке **wildfly-16.0.0.Final-factor\bin\service** отредактировать файл **service.bat**.

Найти строку

```
set CONTROLLER=localhost:19990
```

и заменить 19990 на 9990+смещение.

Например, если смещение=8, 9990+8=9998:

```
set CONTROLLER=localhost:9998
```

Необходимо, чтобы смещение всех портов на всех адресах было одинаковым.

Если необходимо задать смещение портов для «Единого клиента», то необходимо добавить ключ **-Djboss.socket.binding.port-offset={port offset}** в настройки **standalone.conf.bat** для «Единого клиента» аналогично примеру, описанному выше для «Фактора», и задать остальные смещения портов по аналогии с примерами выше, включая смещение в **wildfly-16.0.0.Final-cdi\bin\service\service.bat**.

## Горячий резерв

В новый конфигурационный файл **wildfly-16.0.0.Final-cdi\standalone\configuration\standalone.xml** скопировать блок **outbound-socket-binding name="node-sync-binding"** из старого файла **wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml**, удостовериться, что доменное имя и порт заданы правильно, смещение порта (если используется) учтено:

```
<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>
```

## Безопасное соединение (SSL)

Перенести настройки доступа к хранилищу ключей из **wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml** в файл **wildfly-16.0.0.Final-cdi\standalone\configuration\standalone.xml**.

В новый конфигурационный файл **wildfly-16.0.0.Final-cdi\standalone\configuration\standalone.xml** скопировать блок **security-realm name="HttpsSecuredRealm"** из старого файла **wildfly-10.1.0.Final-8080\standalone\configuration\standalone.xml**, удостовериться, что для коннектора **https-listener** блока **<server name="default-server">** указан параметр **security-realm="HttpsSecuredRealm"** (в данном примере указан общий параметр **<ssl>** блока **<server-identities>**, актуальные настройки находятся в конфигурационном файле старой инсталляции):

```
<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>
```

Блок **<security-realm name="HttpsSecuredRealm">** описывает настройки шифрования, блок коннектора **https-listener** описывает, когда его применять.

### Описание параметров HttpsSecuredRealm

- **keystore.path** — путь к хранилищу ключей, корневая директория определяется параметром **relative.to**;
- **alias** — алиас, под которым ключи доступны в хранилище;
- **keystore-password** — пароль к хранилищу;
- **key-password** — пароль к ключам (если не указан, используется **keystore-password**).

Скопировать в директорию **wildfly-16.0.0.Final-cdi\standalone\configuration\** файлы **\*.jks**, на которые ссылается конфигурационный файл.

## Доступ к внешней БД для «Фактора»

Если для каких-либо справочников «Фактора» используется база данных, то доступ к ней настраивается с помощью файла **{customer}-ds.xml**, который находится в каталоге **standalone\deployments** инсталляции JBoss.

Если подобный **datasource**-файл использовался, то его необходимо скопировать из предыдущей инсталляции в папку **wildfly-16.0.0.Final-factor\standalone\deployments**

## Переключение Wildfly на новую версию

1. Остановить службы CDI и FACTOR.
2. Перенести логи из старых инсталляций «Единого клиента» и «Фактора» (находятся в папке **standalone\log**) в новую инсталляцию( **wildfly-16.0.0.Final-cdi\standalone\log** для «Единого клиента» и **wildfly-16.0.0.Final-factor\standalone\log** для «Фактора»). Это позволит продолжить использовать существующие логи, обеспечит неразрывность данных и облегчит жизнь администраторам и поддержке. Логи могут занимать много места, так что лучше их именно перемещать, а не копировать.
3. Заархивировать предыдущие инсталляции WildFly для «Единого клиента» и «Фактора».
4. Удалить старые системные службы запуска WildFly:

```
sc delete cdi
sc delete factor
```

5. Установить новые системные службы (команду следует выполнять из директории **bin\service** WildFly 16 «Единого клиента»):

```
cd wildfly-16.0.0.Final-cdi\bin\service
service.bat install
cd ..\..\wildfly-16.0.0.Final-factor\bin\service
service.bat install
```

*Примечание. Если возникает ошибка " **ERROR: The service already exists** ", то перед установкой системных служб перезагрузить ОС.*

6. Скопировать файлы **cdi-web-<имя заказчика>-<номер версии>-SNAPSHOT.war** и **factor-<имя заказчика>-<номер версии>.war** из поставки в папки **wildfly-16.0.0.Final-cdi\standalone\deployments** и **wildfly-16.0.0.Final-factor\standalone\deployments** соответственно.
7. Запустить службы CDI и FACTOR.
8. Проверить, что программы запущены и работают.

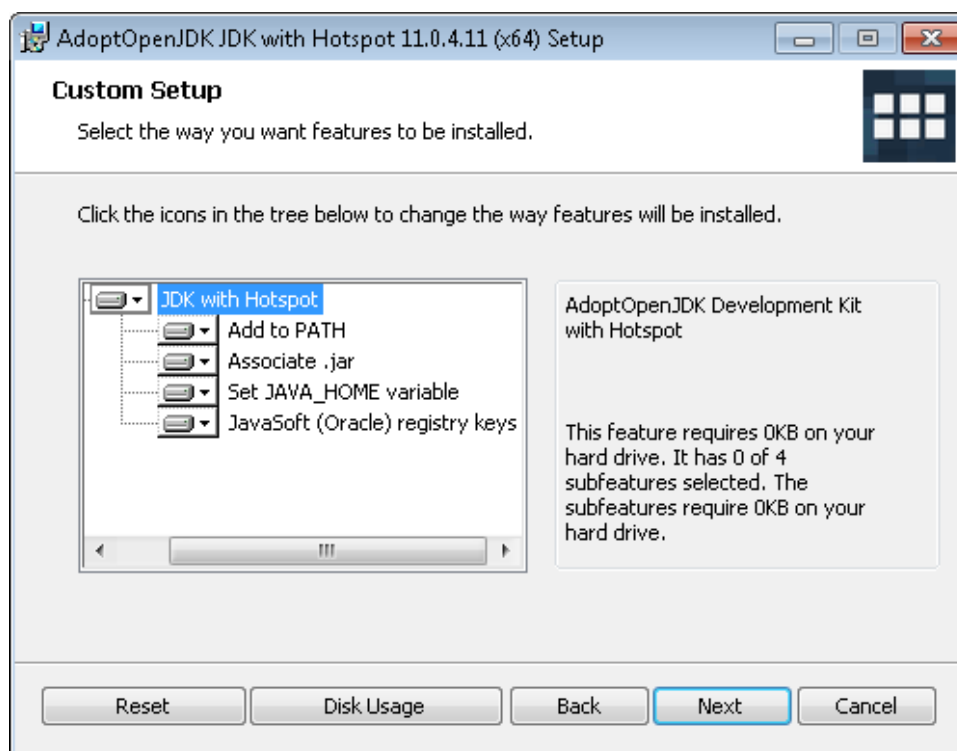
# ИНСТРУКЦИЯ ДЛЯ ПОДСКАЗОК НА WINDOWS (ПЕРЕХОД НА WILDFLY 16)

## Обновление «Подсказок»

### 1. Скачайте и распакуйте OpenJDK 11:

Загрузите [установочный файл](#) или используйте установочный файл из состава поставки.

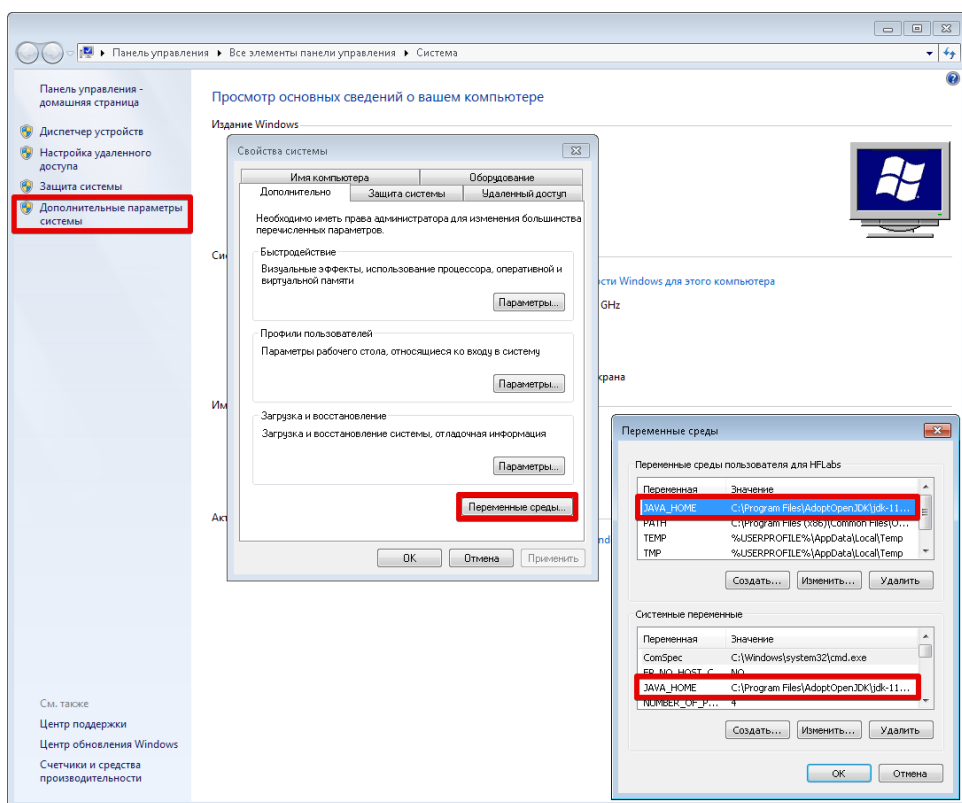
Запустите установочный файл, примите лицензионное соглашение, укажите путь и устанавливаемые компоненты как на скриншоте:



Выполните следующие команды в командной строке:

```
setx JAVA_HOME "C:\Program Files\AdoptOpenJDK\jdk-11.0.4.11-hotspot"  
setx PATH "%PATH%;%JAVA_HOME%\bin"
```

Проверьте, что переменные среды обновились, выполнив в командной строке команду **set** или через графический интерфейс (Панель управления ? Система ? Дополнительные параметры системы ? вкладка "Дополнительно" ? Переменные среды):



При необходимости отредактируйте переменные через графический интерфейс.

Проверьте, что возвращаются правильные версии **java** и **javac** и что они совпадают:

```
javac -version
java -version
```

2. Остановите службу **HFLABS SUGGESTIONS**.
3. Переименуйте директорию используемой версии WildFly **c:\suggestions** в **c:\suggestions\_wildfly10**.
4. Распакуйте WildFly 16 из архива **suggestions.zip** в **c:\suggestions**.
5. Скопируйте в директорию **c:\suggestions\bin** файл **standalone.conf.bat** из **c:\suggestions\_wildfly10\bin**.
6. В **новый** конфигурационный файл **c:\suggestions\standalone\configuration\standalone.xml** перенесите значения портов и **port-offset** из блока **socket-binding-group** старого файла **c:\suggestions\_wildfly10\standalone\configuration\standalone.xml**:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-
offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-http" interface="management" port="${jboss.
management.http.port:9990}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:9993}"/>
  <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
  <socket-binding name="http" port="${jboss.http.port:8080}"/>
  <socket-binding name="https" port="${jboss.https.port:8443}"/>
```



```
<socket-binding name="txn-recovery-environment" port="4712"/>
<socket-binding name="txn-status-manager" port="4713"/>
<outbound-socket-binding name="mail-smtp">
  <remote-destination host="localhost" port="25"/>
</outbound-socket-binding>
</socket-binding-group>
```

7. Распакуйте архив **service.zip** в **c:\suggestions\bin**. В указанном каталоге должна появиться директория **service**.
8. Удалите старую системную службу запуска WildFly:

```
sc delete sgt
```

9. Установите новую системную службу (команду следует выполнять из директории **c:\suggestions\bin\service**):

```
cd c:\suggestions\bin\service
service.bat install
```

*Примечание. Если возникает ошибка " **ERROR: The service already exists** ", то перед установкой системной службы следует перезагрузить ОС.*

10. Скопируйте файл **suggestions-web-19.6-SNAPSHOT.war** из поставки в директорию **c:\suggestions\standalone\deployments**.
11. Запустите службу **HFLABS SUGGESTIONS**.

# ИНСТРУКЦИИ ПО ВОЗВРАТУ НА JAVA 8 И ВЕРСИЮ «ЕДИНОГО КЛИЕНТА» НИЖЕ 19.16

Если что-то пошло не так и возникла необходимость вернуться к более старой версии «Единого клиента», необходимо будет переключить используемую Java на 8 версию и использовать WildFly 10.

При наличии бэкапов «Единого клиента» и «Фактора» восстановление должно пройти быстро и без особых проблем.

Ниже представлены инструкции, описывающие процесс перехода.

- Откат к openJDK 8, WildFly 10.1.0 и «Единому клиенту» 19.15 после переустановки системы
- Откат на WildFly 10 и «Единый клиент» версии 19.15 или ниже (Для Linux)
- Откат на WildFly 10 и «Единый клиент» версии 19.15 или ниже (Для Windows)

# ОТКАТ К OPENJDK 8, WILDFLY 10.1.0 И «ЕДИНОМУ КЛИЕНТУ» 19.15 ПОСЛЕ ПЕРЕУСТАНОВКИ СИСТЕМЫ

- - 1. Требования к системе
  - 2. Настройка системы
    - Создание пользователей для «Единого клиента» и «Фактора»
    - Запрет на выделение памяти сверх того, что есть и отключение SWAP
    - Увеличение предела открытых дескрипторов файлов
    - Настройка для работы с SSD-дисками
    - Отключение времени модификации файлов
    - Выключите IO Scheduler для SSD
  - 3. Установка Java
    - Установочный пакет
    - Установка JDK
    - Проверка правильности установки JDK
    - Установка переменных окружения
  - 4. Установка WildFly
    - Установочный пакет
    - Установка JBOSS
    - Настройка сервисов
  - 5. Создание рабочих каталогов приложения
  - 6. Настройка параметров запуска JBoss
    - Базовые настройки
    - Дополнительные настройки (при необходимости)
    - Изменение смещения портов «Единого клиента» и «Фактора»
    - Горячий резерв
    - Безопасное соединение (SSL)
    - Доступ к внешней БД для «Фактора»
  - 7. Установка «Единого клиента» и «Фактора» и запуск служб

## 1. Требования к системе

Для установки «Единого клиента» и «Фактора» до версий 19.15 и 19.7 включительно необходимо, чтобы система удовлетворяла следующим требованиям:

- CentOS 7+ или Red Hat Enterprise Linux 7+, x64.
- Java SE Development Kit (OpenJDK) 8, с установленными актуальными обновлениями.

- Wildfly 10.1.0

## 2. Настройка системы

### Создание пользователей для «Единого клиента» и «Фактора»

Создайте пользователей, под которыми будут работать сервера приложений «Единого клиента» и «Фактора» — **cdi** и **factor**:

```
useradd cdi
useradd factor
passwd -l cdi
passwd -l factor
```

Пользователей нужно объединить в одну группу:

```
groupadd hfl_cdi
usermod -a -G hfl_cdi cdi
usermod -a -G hfl_cdi factor
```

### Запрет на выделение памяти сверх того, что есть и отключение SWAP

1. Откройте файл **/etc/sysctl.conf** и добавьте в него строки:

```
/etc/sysctl.conf

#Do not overcommit memory
vm.overcommit_memory=2
vm.overcommit_ratio=100

#Max map count
vm.max_map_count = 16777216

# Low swap level
vm.swappiness = 10
```

2. Использование SWAP сильно тормозит работу приложений, поэтому его использование лучше честно отключить. Открывайте файл **/etc/fstab** и прокомментируйте в нем монтирования раздела swap вида. На DEV-стендах из соображения экономии ресурсов swap лучше не отключать.

```
/etc/fstab

%SOME_TEXT% swap          swap defaults 0 0
```

3. Перезагрузите операционную систему.
4. Проверьте параметры с помощью команды:

```
sysctl -p  
либо  
systemctl -p
```

Расшифровка параметров:

**vm.swappiness** ? при каком значении нужно пытаться перекладывать куски памяти в swap (в процентах). По умолчанию это 60 (то есть если осталось свободно 60 процентов оперативки, то начать пытаться перекладывать давно неиспользуемые куски в swap). Установка этого параметра в 10 позволит максимально использовать оперативку, без задействования swap-а (0 ставить нельзя, так как хоть по документации это и отключает swap вообще, но многие ОС это игнорируют)

## Увеличение предела открытых дескрипторов файлов

1. Откройте файл `/etc/security/limits.conf` и добавьте в него строки:

```
                                /etc/security/limits.conf  
cdi      hard  nofile 65535  
cdi      soft  nofile 65535  
cdi      hard  nproc  8192  
cdi      soft  nproc  4096  
cdi      hard  as     unlimited  
cdi      soft  as     unlimited  
cdi      hard  rss   unlimited  
cdi      soft  rss   unlimited  
factor   hard  nofile 65535  
factor   soft  nofile 65535  
factor   hard  nproc  8192  
factor   soft  nproc  4096  
factor   hard  as     unlimited  
factor   soft  as     unlimited  
factor   hard  rss   unlimited  
factor   soft  rss   unlimited
```

где **factor** и **cdi** — имена пользователей, под которыми работают «Фактор» и «Единый клиент».

2. Перезагрузите ОС, чтобы настройки вступили в силу
3. Залогиньтесь под пользователем «Фактора» и убедитесь, что настройки применены:

```
su factor -s /bin/sh  
  
sysctl vm.max_map_count  
ulimit -n  
ulimit -u  
ulimit -v  
ulimit -m
```

Должно получиться следующее:

```
$ sysctl vm.max_map_count  
vm.max_map_count = 16777216
```

```
$ ulimit -n
65535

$ ulimit -u
4096

$ ulimit -v
unlimited

$ ulimit -m
unlimited
```

4. Залогиньтесь под пользователем «Единого клиента» и убедитесь, что настройки применены:

```
su cdi -s /bin/sh

$ ulimit -n
65535

$ ulimit -u
4096

$ ulimit -v
unlimited

$ ulimit -m
unlimited
```

## Настройка для работы с SSD-дисками

### Отключение времени модификации файлов

Если приложение часто и многократно пишет и читает файлы (именно так делают «ЕК», «Фактор», «Подсказки») то на файловых системах **ext3** и **ext4** нужно отключить дополнительные функции работы метаданными файлов.

Для этого нужно изменить параметры монтирования диска, добавив следующие опции:

1. **noatime** — полностью отключает запись времени доступа к файлу. Большинство программ не используют это поле.
2. **data=ordered** — журналирует только изменения метаданных, но обновления данных сбрасываются на жесткий диск до совершения транзакции. Данные записываются не атомарно, но этот режим гарантирует, что после падения файлы не будут содержать блоки данных из устаревших файлов.

В итоге строка в */etc/fstab* должна выглядеть примерно следующим образом (**sdX** — устройство SSD)

```
/etc/fstab
```

```
# <fs> <mountpoint> <type> <opts> <dump/pass>
/dev/sdX /opt ext4 defaults,noatime,data=ordered,errors=remount-ro 0 2
```

## Выключите IO Scheduler для SSD

Выполните команду и добавьте ее в скрипт автозапуска

```
echo noop > /sys/block/sdX/queue/scheduler
```

для каждого устройства (заменяя **sdX** на нужное имя)

## 3. Установка Java

 Для работы системы должен использоваться **openJDK 8 последней версии**

### Установочный пакет

Приоритетным вариантом установки является установка из репозитория ОС. Альтернативно возможно использование архива AdoptOpenJDK.

Установочный пакет можно скачать с сайта проекта AdoptOpenJDK по ссылке <https://adoptopenjdk.net/?variant=openjdk8&jvmVariant=hotspot>:

- удостовериться, что выбрана версия **OpenJDK 8 (LTS)** и **JVM Hotspot**;
- Нажать кнопку Latest release;
- Выбрать вид тип установочного пакета, подходящего для ОС сервера.
- Скачать установочный пакет.

### Установка JDK

Приоритетным является вариант установки через репозиторий ОС.

Пример (CentOS 7 и Red Hat 7):

```
sudo yum install java-1.8.0-openjdk-devel
```

После этого можно перейти к проверке правильности установки JDK.

Если доступа к репозиториям нет, то возможно использовать альтернативный вариант установки: установка вручную из архива AdoptOpenJDK (при необходимости, заменить ссылку [https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10/OpenJDK8U-jdk\\_x64\\_linux\\_hotspot\\_8u222b10.tar.gz](https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10/OpenJDK8U-jdk_x64_linux_hotspot_8u222b10.tar.gz) на ссылку на более новую версию установочного пакета):

```
mkdir /usr/java/
cd /usr/java/
```



```
wget https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10
/OpenJDK8U-jdk_x64_linux_hotspot_8u222b10.tar.gz
tar zxvf OpenJDK8U-jdk_x64_linux_hotspot_8u222b10.tar.gz
rm OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz
alternatives --install /usr/bin/java java /usr/java/jdk8u222-b10/bin/java 2
alternatives --install /usr/bin/jar jar /usr/java/jdk8u222-b10/bin/jar 2
alternatives --install /usr/bin/javac javac /usr/java/jdk8u222-b10/bin/javac 2
alternatives --set java /usr/java/jdk8u222-b10/bin/java
alternatives --set jar /usr/java/jdk8u222-b10/bin/jar
alternatives --set javac /usr/java/jdk8u222-b10/bin/javac
```

## Проверка правильности установки JDK

Выполнить в командной строке команду

```
javac -version
```

Должно появиться сообщение вида:

```
javac 1.8.0_222
```

Версия **JDK** должна соответствовать версии установочного пакета.

Также нужно проверить версию самой java-машины:

```
java -version
```

Она должна быть идентична версии javac.

## Установка переменных окружения

Для пользователей **cdi, factor** в файлах **/home/cdi/.bash\_profile, /home/cdi/.bash\_profile** (также в **/etc/profile** или **/etc/skel/profile**) добавьте строки, предварительно заменив **/usr/java/jdk8u222-b10/** на полный путь к каталогу, в который установлен JDK:

```
export JAVA_HOME=/usr/java/jdk8u222-b10/
export PATH=$JAVA_HOME/bin:$PATH
```

После добавления строк выполнить одну из команд, приведенных ниже:

```
source etc/profile
source /etc/skel/profile
```

Проверить, что путь был добавлен, можно выполнив команду:

```
echo $PATH
```

В ответе должен быть заметен путь к каталогу с Java.

После этого нужно повторно проверить версии java и javac.

## 4. Установка WildFly

### Установочный пакет

Установочные пакеты Jboss 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».

### Установка JBOSS



Если у вас имеются архивы со старыми версиями, то в следующих шагах следует использовать их вместо чистых установочных пакетов. В предложенном ниже скрипте нужно заменить имена архивов на те, которые использовались у вас.

```
unzip old-jboss-cdi -d /opt/cdi
unzip old-jboss-factor -d /opt/factor
```

Может возникнуть необходимость переименовать разархивированные папки, чтобы они назывались просто **jboss**.

После этого следует проверить настройки и обновить значения путей к **служебным папкам** в файлах **standalone.conf** (находятся в папках **jboss/bin/**) для «Единого клиента» и «Фактора».

После выполнения разархивированной версии надо проверить, что файлы **.war** «Единого клиента» и «Фактора» находятся в папках **standalone/deployments** инсталляций.

Затем перейти к пункту настройки сервисов и проследовать по инструкции, не используя пункты настройки параметров запуска JBoss и установки приложений.

Запустить сервисы:

```
service cdi start
service factor start
```

Если архивы были утрачены, то необходимо произвести установку чистых пакетов по инструкции.

1. Распаковать архив с WildFly в каталог **JBOSS\_HOME**. Здесь и далее используется каталог **/opt** в качестве примера. Вы можете использовать любую другую, например **/home** или **/data**.

```
unzip wildfly-10.1.0.Final-8080.zip -d /opt/cdi
unzip wildfly-10.1.0.Final-18080.zip -d /opt/factor
```

```
mv /opt/cdi/wildfly* /opt/cdi/jboss
mv /opt/factor/wildfly* /opt/factor/jboss
```

2. Назначить созданным директориям соответствующих владельцев – **cdi** для **/opt/cdi/** и **factor** для **/opt/factor/**

```
chown -R cdi:cdi /opt/cdi/
chown -R factor:factor /opt/factor/
```

3. Назначить права на запуск исполняемых файлов:

```
find /opt/{factor,cdi}/jboss/ -type d -exec chmod 755 {} \;
find /opt/{factor,cdi}/jboss/ -type f -exec chmod 644 {} \;
find /opt/{factor,cdi}/jboss/ -type f -name "*.sh" -exec chmod 755 {} \;
```

## Настройка сервисов

1. Создать директорию в **/etc** с названием будущей службы (**factor, cdi**), скопировать файлы

```
mkdir /etc/cdi
mkdir /etc/factor

cp /opt/cdi/jboss/docs/contrib/scripts/systemd/wildfly.conf /etc/cdi/
cp /opt/cdi/jboss/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/cdi.service
cp /opt/cdi/jboss/docs/contrib/scripts/systemd/launch.sh /opt/cdi/jboss/bin/

cp /opt/factor/jboss/docs/contrib/scripts/systemd/wildfly.conf /etc/factor/
cp /opt/factor/jboss/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/factor.service
cp /opt/factor/jboss/docs/contrib/scripts/systemd/launch.sh /opt/factor/jboss/bin/
```

2. В **/etc/systemd/system/factor.service** и **/etc/systemd/system/cdi.service** переменные заданы по умолчанию. При необходимости заменить параметры **Limit\*** и путь к **launch.sh** (параметр **ExecStart**). При необходимости заменить пути на используемые при установке

### cdi.service

```
[Unit]
Description=CDI WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=-/etc/cdi/wildfly.conf
User=cdi
OOMScoreAdjust=-1000
```

```
PIDFile=/var/run/cdi/wildfly.pid
ExecStart=/opt/cdi/jboss/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG
$WILDFLY_BIND
StandardOutput=syslog
StandardError=syslog
LimitNOFILE=65535
LimitNPROC=8192
LimitAS=infinity
LimitRSS=infinity

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

#### factor.service

```
[Unit]
Description=Factor WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=-/etc/factor/wildfly.conf
User=factor
OOMScoreAdjust=-1000
PIDFile=/var/run/factor/wildfly.pid
ExecStart=/opt/factor/jboss/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG
$WILDFLY_BIND
StandardOutput=syslog
StandardError=syslog
LimitNOFILE=65535
LimitNPROC=8192
LimitAS=infinity
LimitRSS=infinity

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

3. Указать путь к домашней директории Jboss в **/opt/factor/jboss/bin/launch.sh**

```
WILDFLY_HOME="/opt/factor/jboss"
```

- и в **/opt/cdi/jboss/bin/launch.sh**

```
WILDFLY_HOME="/opt/cdi/jboss"
```

4. На файлы **launch.sh** и **standalone.sh** выдать права на запуск:

```
chmod +x /opt/cdi/jboss/bin/launch.sh
chmod +x /opt/cdi/jboss/bin/standalone.sh

chmod +x /opt/factor/jboss/bin/launch.sh
chmod +x /opt/factor/jboss/bin/standalone.sh
```

5. Перезагрузить список доступных сервисов, чтобы systemd мог управлять новым сервисом:

```
systemctl daemon-reload
```

6. Добавить службы в автозапуск:

```
systemctl enable cdi.service
systemctl enable factor.service
```

## 5. Создание рабочих каталогов приложения

1. Создайте рабочий каталог CDI и дайте на него права пользователю **cdi**:

```
mkdir -p /home/cdi
chown -R cdi:cdi /home/cdi
```

2. Создайте каталог, используемый при поиске дубликатов:

```
mkdir /home/cdi/dedup
```

3. Если вы используете отдельных пользователей для служб CDI и FACTOR, выдайте права на каталог для дубликатов обоим пользователям.

Пример, когда пользователи входят в одну группу с именем "hfl\_cdi":

```
chgrp hfl_cdi /home/cdi/dedup
```

## 6. Настройка параметров запуска JBoss

### Базовые настройки

Как для «Единого клиента», так и для «Фактора» необходимо настроить параметры использования памяти:

```
# Basic settings, memory usage
JAVA_OPTS="$JAVA_OPTS -server -Xms8g -Xmx8g"
```

Xms указывает минимальное количество используемой памяти, Xmx указывает максимальное.

Для единого клиента нужно указать путь к **CDI\_ROOT\_DIR**:

```
# CDI root folder
JAVA_OPTS="$JAVA_OPTS -Dcdi.root.folder=/home/cdi"
```

Отдельно может быть задан каталог для поисковых индексов **CDI\_SEARCH\_DIR** — параметр **-Dcdi.search.folder**, и общий каталог для дедупликации — параметр **-Dcdi.dedup.folder**.

Для «Фактора» можно задать пути к справочникам:

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_INDEXES_PATH=/data/factor/indexes"  
JAVA_OPTS="$JAVA_OPTS -DFACTOR_DICTIONARIES_PATH=/data/factor/dictionaries"  
JAVA_OPTS="$JAVA_OPTS -DFACTOR_DEDUP_PATH=/data/factor/dedup"
```

## Дополнительные настройки (при необходимости)

### Изменение смещения портов «Единого клиента» и «Фактора»

Если требуется изменить смещение портов «Фактора» относительно «Единого клиента» заданное в сборках WildFly (по умолчанию выставлено смещение 10000), необходимо отредактировать соответствующую настройку в **standalone.conf** для «Фактора»:

```
### Port offset  
JAVA_OPTS="$JAVA_OPTS -Djboss.socket.binding.port-offset={port offset}"
```

Если смещение портов для «Фактора» не требуется (не указано в настройках предыдущей инсталляции), удалить ключ **-Djboss.socket.binding.port-offset={port offset}** из настроек или закомментировать его.

Отредактировать настройки отладки, указав в них адрес со смещением.

Например, адрес=5005, смещение=8, адрес+смещение=5013:

```
### Debugging options  
#JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,  
address=5013"
```

Необходимо, чтобы смещение всех портов на всех адресах было одинаковым.

Если необходимо задать смещение портов для «Единого клиента», то необходимо добавить ключ **-Djboss.socket.binding.port-offset={port offset}** в настройки **standalone.conf** для «Единого клиента» аналогично примеру, описанному выше для «Фактора», и задать остальные смещения портов по аналогии с примерами выше.

### Горячий резерв

Вконфигурационный файл **/opt/cdi/jboss/standalone/configuration/standalone.xml** перенести доменное имя второго экземпляра «ЕК» и его http-порт:

```
<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>
```

## Безопасное соединение (SSL)

Указать настройки доступа к хранилищу ключей в `/opt/cdi/jboss/standalone/configuration/standalone.xml`.

В конфигурационный файл `/opt/cdi/jboss_new/standalone/configuration/standalone.xml` добавить блок `security-realm name="HttpsSecuredRealm"`, удостовериться, что для коннектора `https-listener` блока `<server name="default-server">` указан параметр `security-realm` — `HttpsSecuredRealm` (в данном примере указан общий параметр `<ssl>` блока `<server-identities>`):

```
<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>
```

Блок `<security-realm name="HttpsSecuredRealm">` описывает настройки шифрования, блок коннектора `https-listener` описывает, когда его применять.

### Описание параметров `HttpsSecuredRealm`

- **keystore.path** — путь к хранилищу ключей, корневая директория определяется параметром **relative.to**;
- **alias** — алиас, под которым ключи доступны в хранилище;
- **keystore-password** — пароль к хранилищу;
- **key-password** — пароль к ключам (если не указан, используется **keystore-password**).

Скопировать в директорию `/opt/cdi/jboss_new/standalone/configuration/` файлы `*.jks`, на которые ссылается конфигурационный файл.

## Доступ к внешней БД для «Фактора»

Если для каких-либо справочников «Фактора» используется база данных, то доступ к ней настраивается с помощью файла **{customer}-ds.xml**, который необходимо поместить в папку `/opt/factor/jboss/standalone/deployments/`.

## 7. Установка «Единого клиента» и «Фактора» и запуск служб

1. Скопировать файлы `cdi-web-<имя заказчика>-<номер версии>-SNAPSHOT.war` и `factor-<имя заказчика>-<номер версии>.war`, переданные в поставке с данной инструкцией, в папки **`/opt/cdi/jboss/standalone/deployments`** и **`/opt/factor/jboss/standalone/deployments`** соответственно.
2. Запустить службы `cdi` и `factor`:

```
service cdi start
service factor start
```

3. Удостовериться, что комплекс запустился и работает нормально.



# ОТКАТ НА WILDFLY 10 И «ЕДИНЫЙ КЛИЕНТ» ВЕРСИИ 19.15 ИЛИ НИЖЕ (ДЛЯ LINUX)

- Подготовка
- Замена WildFly
- Запуск служб

## Подготовка

Подготовьте заархивированные при переходе на новую версию старые инсталляции JBoss.

Для работы WildFly 10.1.0 необходима JDK версии 8. Рекомендуется использовать openJDK.

Необходимый пакет можно установить из репозитория ОС:

```
yum install java-1.8.0-openjdk-devel
```

Если нет доступа к репозиториям ОС, можно либо использовать имеющуюся установку java 8 (ВНИМАНИЕ! Это чревато возможными претензиями со стороны Oracle, если не использовалась openJDK), либо использовать сборку AdoptOpenJDK.

(при необходимости, заменить ссылку [https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10/OpenJDK8U-jdk\\_x64\\_linux\\_hotspot\\_8u222b10.tar.gz](https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10/OpenJDK8U-jdk_x64_linux_hotspot_8u222b10.tar.gz) на ссылку на более новую версию установочного пакета):

```
mkdir /usr/java/  
cd /usr/java/  
wget https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u222-b10/  
OpenJDK8U-jdk_x64_linux_hotspot_8u222b10.tar.gz  
tar zxvf OpenJDK8U-jdk_x64_linux_hotspot_8u222b10.tar.gz  
rm OpenJDK11U-jdk_x64_linux_hotspot_11.0.4_11.tar.gz  
alternatives --install /usr/bin/java java /usr/java/jdk8u222-b10/bin/java 2  
alternatives --install /usr/bin/jar jar /usr/java/jdk8u222-b10/bin/jar 2  
alternatives --install /usr/bin/javac javac /usr/java/jdk8u222-b10/bin/javac 2  
alternatives --set java /usr/java/jdk8u222-b10/bin/java  
alternatives --set jar /usr/java/jdk8u222-b10/bin/jar  
alternatives --set javac /usr/java/jdk8u222-b10/bin/javac
```

Проверить, и при необходимости выставить используемую Java вручную (необходимо выбрать java-1.8.0):

```
alternatives --config java
```

Проверить правильность изменения версии.

Выполнить в командной строке команду

```
javac -version
```

Должно появиться сообщение вида:

```
javac 1.8.0_222
```

Версия **JDK** должна соответствовать версии установочного пакета.

Также нужно проверить версию самой java-машины:

```
java -version
```

Она должна быть идентична версии javac.

После нужно обновить значение JAVA\_HOME для пользователей «Единого Клиента» и «Фактора» в файлах **/home/<user>/.bash\_profile** и **/home/<user>/.profile**, где **<user>** – пользователи «Единого Клиента» и «Фактора» (обычно это **jboss**, **cdi**, **factor**, etc), указав там путь для используемой версии Java (например /usr/java/jdk8u222-b10).

Проверить, прописано ли значение JAVA\_HOME в **/etc/cdi/wildfly.conf** и **/etc/factor/wildfly.conf** и изменить его, если необходимо

## Замена WildFly

Разархивировать старые инсталляции. Инструкция подразумевает, что перед архивированием папки были переименованы в jboss\_10 (при необходимости, заменить названия файлов и пути), а установка производится в директорию /opt (необходимо изменить пути в командах на те, которые используются у вас):

```
unzip old-jboss-cdi -d /opt/cdi
unzip old-jboss-factor -d /opt/factor
```

Остановить службы cdi и factor:

```
service cdi stop
service factor stop
```

Заменить wildfly:

```
mv /opt/cdi/jboss /opt/cdi/jboss_16
mv /opt/factor/jboss /opt/factor/jboss_16

mv /opt/cdi/jboss_10 /opt/cdi/jboss
mv /opt/factor/jboss_10 /opt/factor/jboss
```

Переместить накопившиеся логи из новой инсталляции в старую

```
mv /opt/cdi/jboss_16/standalone/log/* /opt/cdi/jboss/standalone/log
mv /opt/factor/jboss_16/standalone/log/* /opt/factor/jboss/standalone/log
```

Выдать все необходимые права:

```
chown -R cdi:cdi /opt/cdi/jboss
chown -R factor:factor /opt/factor/jboss

find /opt/{factor,cdi}/jboss/ -type d -exec chmod 755 {} \;
find /opt/{factor,cdi}/jboss/ -type f -exec chmod 644 {} \;
find /opt/{factor,cdi}/jboss/ -type f -name "*.sh" -exec chmod 755 {} \;
```

## Запуск служб

Если необходимо установить «Единый клиент» и «Фактор», скопировать файлы cdi-web-<имя заказчика>-<номер версии>-SNAPSHOT.war и factor-<имя заказчика>-<номер версии>.war, переданные в поставке, в папки **/opt/cdi/jboss/standalone/deployments** и **/opt/factor/jboss/standalone/deployments** соответственно.

1. Запустить службы cdi и factor:

```
service cdi start
service factor start
```

2. Удостовериться, что комплекс запустился и работает нормально.

# ОТКАТ НА WILDFLY 10 И «ЕДИНЫЙ КЛИЕНТ» ВЕРСИИ 19.15 ИЛИ НИЖЕ (ДЛЯ WINDOWS)

- Подготовка.
- Переключение WildFly и запуск служб

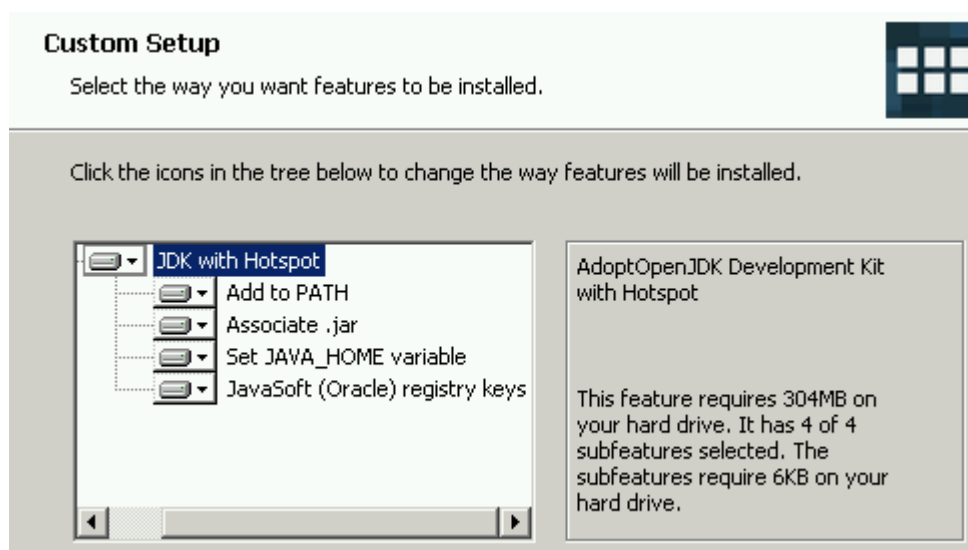
## Подготовка.

Удалить openJDK 11 средствами системы.

Для даунгрейда лучше так же использовать сборку AdoptOpenJDK, но версии 8, чтобы избежать проблем с лицензией Oracle.

Необходимо загрузить [установочный файл](#).

Запустить установочный файл, принять лицензионное соглашение, указать путь и устанавливаемые компоненты как на скриншоте:



Выполнить следующие команды в командной строке(заменяв, если надо, путь к папке установки Java):

```
setx JAVA_HOME "C:\Program Files\AdoptOpenJDK\jdk8u222-b10"  
setx PATH "%PATH%;%JAVA_HOME%\bin"
```

Проверить, что переменные среды обновились можно выполнив в командной строке команду set или через графический интерфейс (панель управления ? система ? Дополнительные параметры системы ? вкладка "Дополнительно" ? переменные среды).

Если папки со старыми инсталляциями были заархивированы, разархивировать их рядом с папками новой инсталляции.

## Переключение WildFly и запуск служб

1. Остановить службы CDI и FACTOR.
2. Удалить старые системные службы запуска WildFly:

```
sc delete cdi  
sc delete factor
```

3. Установить новые системные службы (команду следует выполнять из директории **bin\service** WildFly 10 «Единого клиента», при необходимости следует заменить названия папок и пути):

```
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 перезагрузить ОС.*

4. Если необходимо, скопировать файлы cdi-web-*<имя заказчика>*-*<номер версии>*-SNAPSHOT.war и factor-*<имя заказчика>*-*<номер версии>*.war из поставки в папки **wildfly-10.1.0.Final-8080\standalone\deployments** и **wildfly-10.1.0.Final-18080\standalone\deployments** соответственно
5. Добавить последние логи новой инсталляции к логам старой инсталляции.
6. Запустить службы CDI и FACTOR.
7. Проверить, что программы запущены и работают.

# ИНСТРУКЦИЯ ПО ОБНОВЛЕНИЮ МОДУЛЕЙ «ЕК» И «ФАКТОР» ДЛЯ LINUX

## *Инструкция по обновлению модулей CDI и Factor для Linux*

1-8. Выполните стандартное обновление версий «Единого клиента» и «Фактора».

### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**md5sum -c factor-{CUSTOMER}-{version}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» и БД «Единого клиента». Резервная копия должна располагаться **вне** директории **/opt/cdi** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Единого клиент»: **systemctl stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **/opt/factor** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **systemctl stop factor**

### 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **/opt/cdi/appserver**.
2. Скопировать новую сборку «Единого клиента» в директорию **/opt/cdi/appserver**.

### 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **/opt/factor/appserver**.
2. Скопировать новую сборку «Фактора» в директорию **/opt/factor/appserver**.

## 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **systemctl start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

## 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **systemctl start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

## 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{cdi\_hostname}:{cdi\_port}/cdi/api/manage/health**
2. В разделе "**factorIndex**" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

# ИНСТРУКЦИЯ ПО ОБНОВЛЕНИЮ МОДУЛЕЙ «ЕК» И «ФАКТОР» ДЛЯ WINDOWS

## *Инструкция по обновлению модулей CDI и Factor для Windows*

Обновление модулей «Единый клиент» и «Фактор» выполняется в следующем порядке: стоп ЕК ? стоп Фактор ? обновление ЕК и Фактора ? старт Фактор ? ждем, когда Фактор «поднимется» ? старт ЕК.

1-8. Выполните стандартное обновление версий «Единого клиента» и «Фактора».

### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**certutil -hashfile cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war MD5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**certutil -hashfile factor-{CUSTOMER}-{version}.war MD5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» 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. Остановить сервис «Единого клиент»: **net stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **{JBOSS\_FACTOR\_HOME}\standalone\deployments** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **net stop factor**

### 4. Замена war-файла CDI



1. Удалить старый war-файл «Единого клиента» из директории **{JBOSS\_CDI\_HOME}\standalone\deployments**.
2. Скопировать новую сборку «Единого клиента» в директорию **deployments**.

#### 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **{JBOSS\_FACTOR\_HOME}\standalone\deployments**.
2. Скопировать новую сборку «Фактора» в директорию **deployments**.

#### 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **net start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

#### 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **net start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

#### 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{hostname}:{port}/cdi/api/manage/health**
2. В разделе "factorIndex" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

# ИНСТРУКЦИЯ ПО ОБНОВЛЕНИЮ МОДУЛЕЙ «ЕК» И «ФАКТОР» ДЛЯ WINDOWS НА ДВУХ НОДАХ

## *Инструкция по обновлению модулей «CDI» и «Factor» для Windows на двух нодах*

Выполните стандартное обновление «Единого клиента» и «Фактора» поочередно на каждой из нод.

1. Убедитесь, что все задачи завершены на обеих нодах.
2. Выведите первую ноду из балансировки.
3. Выполните стандартное обновление версий «Единого клиента» и «Фактора» на первой ноде.

### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**certutil -hashfile cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war MD5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**certutil -hashfile factor-{CUSTOMER}-{version}.war MD5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» 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. Остановить сервис «Единого клиент»: **net stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **{JBOSS\_FACTOR\_HOME}\standalone\deployments** и её поддиректорий, иначе нормальная работа сервера не гарантируется.

2. Остановить сервис «Фактор»: **net stop factor**

4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **{JBOSS\_CDI\_HOME}\standalone\deployments**.
2. Скопировать новую сборку «Единого клиента» в директорию **deployments**.

5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **{JBOSS\_FACTOR\_HOME}\standalone\deployments**.
2. Скопировать новую сборку «Фактора» в директорию **deployments**.

6. Старт модуля Factor

1. Запустить сервис «Фактор»: **net start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **net start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{hostname}:{port}/cdi/api/manage/health**
2. В разделе "factorIndex" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

4. Первую ноду введите в балансировку. Выведите вторую ноду из балансировки.

5. Выполните стандартное обновление версий «Единого клиента» и «Фактора» на второй ноде.

1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**certutil -hashfile cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war MD5**

полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:

**certutil -hashfile factor-{CUSTOMER}-{version}.war MD5**

полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

## 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» 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. Остановить сервис «Единого клиент»: **net stop cdi**

## 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **{JBOSS\_FACTOR\_HOME}\standalone\deployments** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **net stop factor**

## 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **{JBOSS\_CDI\_HOME}\standalone\deployments**.
2. Скопировать новую сборку «Единого клиента» в директорию **deployments**.

## 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **{JBOSS\_FACTOR\_HOME}\standalone\deployments**.
2. Скопировать новую сборку «Фактора» в директорию **deployments**.

## 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **net start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

## 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **net start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

## 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{hostname}:{port}/cdi/api/manage/health**
2. В разделе "factorIndex" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

## 6. Введите вторую ноду в балансировку.

# ИНСТРУКЦИЯ ПО ОБНОВЛЕНИЮ МОДУЛЕЙ «ЕК» И «ФАКТОР» ДЛЯ LINUX НА ДВУХ НОДАХ

## Инструкция по обновлению модулей «CDI» и «Factor» для Linux на двух нодах

Выполните стандартное обновление «Единого клиента» и «Фактора» поочередно на каждой из нод.

На балансировщике одновременно не должен находиться старый и новый UI.

1. Убедитесь, что все задачи завершены на обеих нодах.
2. Выведите первую ноду из балансировки.
3. Выполните стандартное обновление версий «Единого клиента» и «Фактора» на первой ноде.

### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**md5sum -c factor-{CUSTOMER}-{version}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» и БД «Единого клиента». Резервная копия должна располагаться **вне** директории **/opt/cdi** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Единого клиент»: **systemctl stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **/opt/factor** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **systemctl stop factor**

### 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **/opt/cdi/appserver**.
2. Скопировать новую сборку «Единого клиента» в директорию **/opt/cdi/appserver**.

#### 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **/opt/factor/appserver**.
2. Скопировать новую сборку «Фактора» в директорию **/opt/factor/appserver**.

#### 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **systemctl start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

#### 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **systemctl start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

#### 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{cdi\_hostname}:{cdi\_port}/cdi/api/manage/health**
2. В разделе "**factorIndex**" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

4. Первую ноду введите в балансировку. Выведите вторую ноду из балансировки.

5. Выполните стандартное обновление версий «Единого клиента» и «Фактора» на второй ноде.

#### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**md5sum -c factor-{CUSTOMER}-{version}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

#### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» и БД «Единого клиента». Резервная копия должна располагаться **вне** директории **/opt/cdi** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Единого клиент»: **systemctl stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **/opt/factor** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **systemctl stop factor**

### 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **/opt/cdi/appserver**.
2. Скопировать новую сборку «Единого клиента» в директорию **/opt/cdi/appserver**.

### 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **/opt/factor/appserver**.
2. Скопировать новую сборку «Фактора» в директорию **/opt/factor/appserver**.

### 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **systemctl start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

### 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **systemctl start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

### 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{cdi\_hostname}:{cdi\_port}/cdi/api/manage/health**
2. В разделе "**factorIndex**" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

### 6. Введите вторую ноду в балансировку.



# ИНСТРУКЦИЯ ПО ОБНОВЛЕНИЮ МОДУЛЕЙ «ЕК» И «ФАКТОР» ДЛЯ LINUX (23.19+)

## *Инструкция по обновлению модулей CDI и Factor для Linux*

1-8. Выполните стандартное обновление версий «Единого клиента» и «Фактора».

### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**md5sum -c factor-{CUSTOMER}-{version}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» и БД «Единого клиента». Резервная копия должна располагаться вне директории **/opt/cdi** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Единого клиент»: **systemctl stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **/opt/factor** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **systemctl stop factor**

### 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **/opt/cdi/appserver**.
2. Скопировать новую сборку «Единого клиента» в директорию **/opt/cdi/appserver**.

### 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **/opt/factor/appserver**.
2. Скопировать новую сборку «Фактора» в директорию **/opt/factor/appserver**.

## 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **systemctl start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

## 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **systemctl start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

## 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{cdi\_hostname}:{cdi\_port}/cdi/api/manage/health**
2. В разделе "**factorIndex**" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

# ИНСТРУКЦИЯ ПО ОБНОВЛЕНИЮ МОДУЛЕЙ «ЕК» И «ФАКТОР» ДЛЯ LINUX НА ДВУХ НОДАХ (23.19+)

## *Инструкция по обновлению модулей «CDI» и «Factor» для Linux на двух нодах*

Выполните стандартное обновление «Единого клиента» и «Фактора» поочередно на каждой из нод.

На балансировщике одновременно не должен находиться старый и новый UI.

1. Убедитесь, что все задачи завершены на обеих нодах.
2. Выведите первую ноду из балансировки.
3. Выполните стандартное обновление версий «Единого клиента» и «Фактора» на первой ноду.

### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**md5sum -c factor-{CUSTOMER}-{version}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

### 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» и БД «Единого клиента». Резервная копия должна располагаться вне директории **/opt/cdi** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Единого клиент»: **systemctl stop cdi**

### 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **/opt/factor** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **systemctl stop factor**

#### 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **/opt/cdi/appserver**.
2. Скопировать новую сборку «Единого клиента» в директорию **/opt/cdi/appserver**.

#### 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **/opt/factor/appserver**.
2. Скопировать новую сборку «Фактора» в директорию **/opt/factor/appserver**.

#### 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **systemctl start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

#### 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **systemctl start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

#### 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{cdi\_hostname}:{cdi\_port}/cdi/api/manage/health**
2. В разделе "**factorIndex**" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

#### 4. Первую ноду введите в балансировку. Выведите вторую ноду из балансировки.

#### 5. Выполните стандартное обновление версий «Единого клиента» и «Фактора» на второй ноде.

##### 1. Скачивание war-файлов

1. Скачать с FS war-файл новой сборки «Единого клиента» **cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war**.
2. Скачать с FS war-файл новой сборки «Фактора» **factor-{CUSTOMER}-{version}.war**.
3. Проверить MD-5 сумму war-файла новой сборки «Единого клиента». Например, в командной строке выполнить команду:  
**md5sum -c cdi-web-{customer}-{version}-SNAPSHOT-{data}\_{time}-{revision}-{environment}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).
4. Проверить MD-5 сумму war-файла новой сборки «Фактора». Например, в командной строке выполнить команду:  
**md5sum -c factor-{CUSTOMER}-{version}.war.md5**  
полученный результат сравнить с контрольной суммой в файле **.md5** (открыть любым текстовым редактором).

## 2. Остановка модуля CDI

1. Сделать резервные копии предыдущей сборки «Единого клиента» и БД «Единого клиента». Резервная копия должна располагаться вне директории **/opt/cdi** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Единого клиент»: **systemctl stop cdi**

## 3. Остановка модуля Factor

1. Сделать резервную копию предыдущей сборки «Фактора». Резервная копия должна располагаться **вне** директории **/opt/factor** и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Остановить сервис «Фактор»: **systemctl stop factor**

## 4. Замена war-файла CDI

1. Удалить старый war-файл «Единого клиента» из директории **/opt/cdi/appserver**.
2. Скопировать новую сборку «Единого клиента» в директорию **/opt/cdi/appserver**.

## 5. Замена war-файла Factor

1. Удалить старый war-файл «Фактора» из директории **/opt/factor/appserver**.
2. Скопировать новую сборку «Фактора» в директорию **/opt/factor/appserver**.

## 6. Старт модуля Factor

1. Запустить сервис «Фактор»: **systemctl start factor**
2. Только после того, как «Фактор» «поднялся», переходим к следующему пункту.

## 7. Старт модуля CDI

1. Запустить сервис «Единого клиента»: **systemctl start cdi**
2. Проверить, что в АРМ Администратора отображается корректная версия ЕК и Фактора (номер версии указан в шапке инструкции).

## 8. (с релиза 23.16+) Проверка состояния индексов и хэшей дедапа

1. Вызвать эндпоинт **http://{cdi\_hostname}:{cdi\_port}/cdi/api/manage/health**
2. В разделе "**factorIndex**" проверить состояние хэшей:
  - a. status: "UP" – ошибок нет, можно переходить к следующему пункту инструкции;
  - b. status: "DOWN" – необходимо приостановить дальнейшее обновление и обратиться в поддержку.

## 6. Введите вторую ноду в балансировку.

# FAQ

- Основное (FAQ)
  - CPU под 100% на задаче
  - Индекс неактуальный (Index may be corrupted)
  - Как переместить прод с одной машины на другую
  - Как поднять бекап базы с PROD на TEST стенде?
  - Почему лучше делать разных пользователей для ЕК и Фактора
  - Что такое "схлопывание" атрибутов?
- Менеджер Данных (FAQ)
  - Не могу зайти в АРМ Менеджера данных
  - Отслеживание логинов в CDM
  - Ошибка «Разрешите противоречия между подтвержденными и отклоненными дубликатами, прежде чем выполнять слияние»
- АРМ Администратора (FAQ)
  - В IE\* нельзя создать триггер - не удастся выбрать задачи из списка
  - Изменение параметра oldMaxId
- Дополнительно (FAQ)
  - Запросы в БД, соответствующие фильтрам в CDM
  - Как пополнить справочник имен / фамилий?
- Инструменты для проверки (FAQ)
  - Как снять Thread Dump
  - Как снять Thread Dump через командную строку в Unix
  - Как можно вызвать метод SOAP-интерфейса?
    - Создание проекта в SoapUI
    - Авторизация запроса в SoapUI
  - Как можно вызвать REST-метод вручную
  - Как проверить учетную запись LDAP?
  - Как посмотреть JMS, уходящие в RabbitMQ
  - Как посмотреть JMS, уходящие в IBM MQ
    - Запуск и мониторинг в MQ
  - Как посмотреть JMS, уходящие в Tibco
  - Как послать в Tibco входящее JMS-сообщение
  - Как настроить приветствие на Linux
  - Как по ID в исходной системе понять с кем слит контрагент
- Запуск задач / триггеров
  - Фазы миграции на \_HIST таблицы
  - Что делать, если инкремент упал с ошибкой

- FAQ для сопровождаемых
- Фактор (FAQ)
  - Есть ли doBatchClean для телефонов?
- Интеграция (faq)
  - Как передать двух клиентов с одной связью
- Обновление справочников (FAQ)
  - Я обновил перечень ПНП на Linux, но он не подхватывается

## ОСНОВНОЕ (FAQ)

- CPU под 100% на задаче
- Индекс неактуальный (Index may be corrupted)
- Как переместить прод с одной машины на другую
- Как поднять бекап базы с PROD на TEST стенде?
- Почему лучше делать разных пользователей для ЕК и Фактора
- Что такое "схлопывание" атрибутов?



## 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

# 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*.

# КАК ПЕРЕМЕСТИТЬ ПРОД С ОДНОЙ МАШИНЫ НА ДРУГОЮ

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-{customer}-{version}.war** со старой машины (действующий прод) из папки **{JBOSS\_CDI\_HOME}/standalone/deployments** на новую в соответствующую папку.
6. Скопировать файл **factor-service-{customer}.war** со старой машины из папки **{JBOSS\_FACTOR\_HOME}/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 **CDI\_SEARCH\_DIR** и **{JBOSS\_FACTOR\_HOME}/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 кэши фактора на новую машину:
  - Скопировать папку **{JBOSS\_FACTOR\_HOME}**, задать владельцем пользователя `cdi_user` и раздать на нее права `RWX`.
  - Скопировать папку **{JBOSS\_FACTOR\_HOME}/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.

# 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. Сделать бэкап таблиц с настройками для TEST, чтобы не настраивать повторно после накатывания дампа (см. примечание).
3. Развернуть бекап .
4. Выполнить очистку забэкапленных таблицы из п. 2 (см. примечание)
5. Удалить старый поисковый индекс.
6. Запустить сервис «Единого клиента».
7. В APM Администратора перейти за закладку *Триггеры* и выполнить триггер `afterDatabaseBackupTrigger`.

В случае горячего резерва операции необходимо выполнить на каждой ноде по отдельности. На тесте возможно запускать одновременно на двух нодах, но нужно иметь ввиду повышенную нагрузку на базу.



Список таблиц, которые **нельзя** восстанавливать из бэкапа на TEST (в них содержатся PROD настройки).

Перед накатываем дампа ознакомьтесь с описанием этих таблиц и примечаниями. Возможно вам потребуется сделать бэкап этих таблиц с тестового контура и после накатывания дампа развернуть данные из бэкапа.

**Обязательно** очищаем/исключаем из дампа:

- **CONFIG\_PROPERTY** - тут содержатся настройки **PROD**. Чтобы случайно не пустить нагрузку на Фактор **PROD**, не начать вычитывать **JMS** очередь или отправлять в **JMS** очередь сообщения и прочее данную таблицу **НИКОГДА** нельзя переносить с **PROD** и накатывать на **TEST**.  
**Рекомендация:** чтобы не прописывать все настройки с нуля, рекомендуем сделать бэкап данной таблицы с теста до накатывания дампа **PROD**. После накатывания дампа, очистить эту таблицу и заполнить данными из сделанного бэкапа с **TEST** до дампа прода.
- **TASK\_LOCK** - здесь хранятся запущенные на окружении таблицы, триггеры, для предотвращения повторного запуска этих же задач, а также запуска конфликтных задач.  
**Дамп может быть снят во время выполнения каких-либо задач. В итоге записи могут быть перенесены на тестовый контур вместе с**

дампом и блокировать запуск задач/триггеров, хотя ничего не запущено. Смело чистим эту табличку после того как накатали дампа.

Данные таблички очищаем опционально (можно написать нам в поддержку для уточнения), но **лучше тоже почистить:**

- **TASK\_PARAMETER** - параметры, которые выставлены в задачах. Критично для тех заказчиков, у кого есть задачи которые читают /изменяют данные вне окружения. Например, задача импортирует данные с внешнего sftp - сервера. Если не очистить таблицу, то параметры в задаче останутся как на PROD и случайно можем прогрузить файлы на тест вместо прод. Если в задачах уже настроены какие-либо параметры для работы именно с тестовыми источниками, то стоит сделать бэкап таблицы и накатить его поверх дампа, чтобы не настраивать вручную.
- **JMS\_EVENT** - события, отправляемые в обратные поток. Актуально только для заказчиков с обратным потоком. Не критично если таблица будет на тесте, но обычно в ней очень много данных, которые занимают большой объем физической памяти. Лучше совсем исключать данную таблицу из дампа, чтобы не переносить лишнее. Если ее накатывать дампа вместе с ней может не хватить места под БД тестового контура, так как там обычно меньше ресурсов.
- **USER - JDBC** учетные записи. В целом не критично, если переедут на тест. Но если на тестовом контуре были созданы какие-либо УЗ специально созданные для этого контура, они могут потеряться. Тут тоже лучше сделать бэкап таблички до работ и накатить его поверх дампа обратно.
- **AUTHORITIES** - Роли пользователей. Идет как дополнение к **USERS**, поэтому если для **USERS** решили делать бэкап, то и для этой таблички делаем тоже.

## ПОЧЕМУ ЛУЧШЕ ДЕЛАТЬ РАЗНЫХ ПОЛЬЗОВАТЕЛЕЙ ДЛЯ ЕК И ФАКТОРА

**Q:** Хотим поднимать ЕК и Фактор под одним пользователем, зачем делать разных?

**A:** Единая группа сильно упрощает администрирование прав на директории и т.д. А разные пользователи позволяют диагностировать непредвиденные случаи. Именно с этим пользователем в системных журналах будут сообщения.

Если служба безопасности всячески ограждается, устанавливая на сервера антивирусы и урезая права, от несанкционированного доступа, то они первые, кто должен быть заинтересован в разных пользователях. У нас поднимаются два (а подсказками ещё и три) разных приложения. При 0-day уязвимости и плодению эксплоита на сервере невозможно будет понять, через какое именно приложение он попал в систему, если будет один пользователь.

## ЧТО ТАКОЕ "СХЛОПЫВАНИЕ" АТТРИБУТОВ?



**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 прислать 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**).

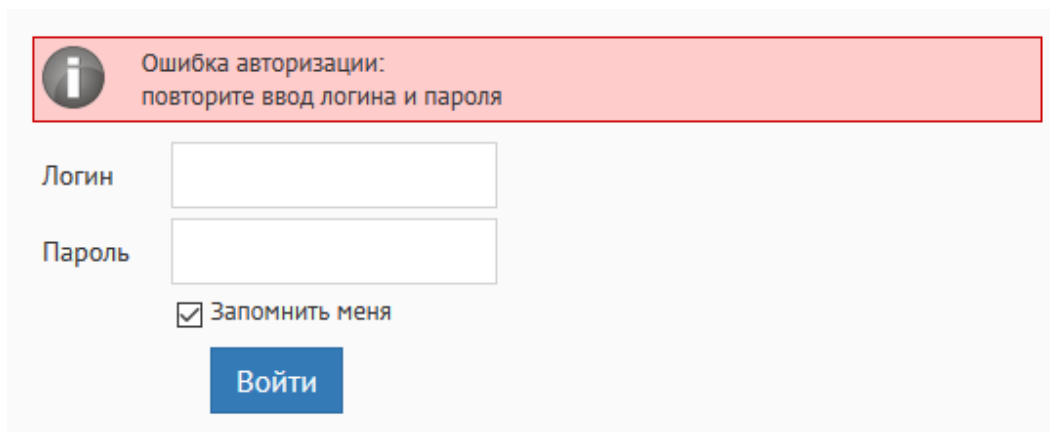
## МЕНЕДЖЕР ДАННЫХ (FAQ)

- Не могу зайти в АРМ Менеджера данных
- Отслеживание логинов в CDM
- Ошибка «Разрешите противоречия между подтвержденными и отклоненными дубликатами, прежде чем выполнять слияние»



# НЕ МОГУ ЗАЙТИ В АРМ МЕНЕДЖЕРА ДАННЫХ

## Ошибка



The screenshot shows a login form with a red error banner at the top. The banner contains an information icon and the text: "Ошибка авторизации: повторите ввод логина и пароля". Below the banner are two input fields labeled "Логин" and "Пароль". There is a checkbox labeled "Запомнить меня" which is checked. At the bottom of the form is a blue button labeled "Войти".

## Возможные причины

- Неверный пароль.
- Недостаточно прав.
- Пользователь заблокирован.
- Истек срок действия пароля и т.д.

## Что делать

Прочитать текст сообщения об ошибке.

# 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.

# ОШИБКА «РАЗРЕШИТЕ ПРОТИВОРЕЧИЯ МЕЖДУ ПОДТВЕРЖДЕННЫМИ И ОТКЛОНЕННЫМИ ДУБЛИКАТАМИ, ПРЕЖДЕ ЧЕМ ВЫПОЛНЯТЬ СЛИЯНИЕ»

## Ошибка



## Возможные причины

- Группы дубликатов формируются из пачек по 2 млн (задача `duplicateGroupActualize`), 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."
- На новом механизме перестроения групп такое возможно, если отклоненный дубль не успел попасть в группу дублей.

## Что делать

- Лечится перестроение групп дубликатов, задача `duplicateGroupActualize`.

## АРМ АДМИНИСТРАТОРА (FAQ)

- В IE\* нельзя создать триггер - не удается выбрать задачи из списка
- Изменение параметра oldMaxId

# В IE\* НЕЛЬЗЯ СОЗДАТЬ ТРИГГЕР - НЕ УДАЕТСЯ ВЫБРАТЬ ЗАДАЧИ ИЗ СПИСКА

## Ошибка



## Возможные причины

- Не включен **JavaScript**.

## 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 интрасети**" (**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.

## 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.

## ДОПОЛНИТЕЛЬНО (FAQ)

- Запросы в БД, соответствующие фильтрам в CDM
- Как пополнить справочник имен / фамилий?

# 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):

## 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 );
```

## 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 );
```

## 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 );
```

## 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 );
```

#### **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 );
```

## КАК ПОПОЛНИТЬ СПРАВОЧНИК ИМЕН / ФАМИЛИЙ?

В сложных ФИО возможен неправильный разбор Фактором, так как в наших справочниках может не быть одного из компонентов.

Для исправления ситуации можно:

- скорректировать ФИО клиента в веб-менеджере данных СДІ и подтвердить введенное значение.
- прислать значения нам в Поддержку, если считаете ФИО корректными и часто встречающимися, добавим разбор в следующих релизах.

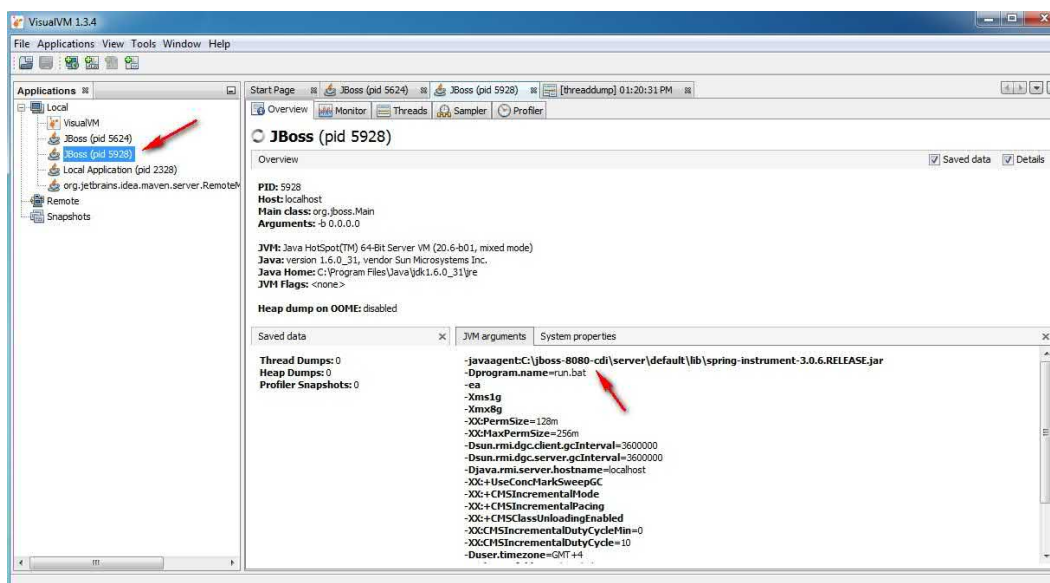
## ИНСТРУМЕНТЫ ДЛЯ ПРОВЕРКИ (FAQ)

- Как снять Thread Dump
- Как снять Thread Dump через командную строку в Unix
- Как можно вызвать метод SOAP-интерфейса?
- Как можно вызвать REST-метод вручную
- Как проверить учетную запись LDAP?
- Как посмотреть JMS, уходящие в RabbitMQ
- Как посмотреть JMS, уходящие в IBM MQ
- Как посмотреть JMS, уходящие в Tibco
- Как послать в Tibco входящее JMS-сообщение
- Как настроить приветствие на Linux
- Как по ID в исходной системе понять с кем слит контрагент

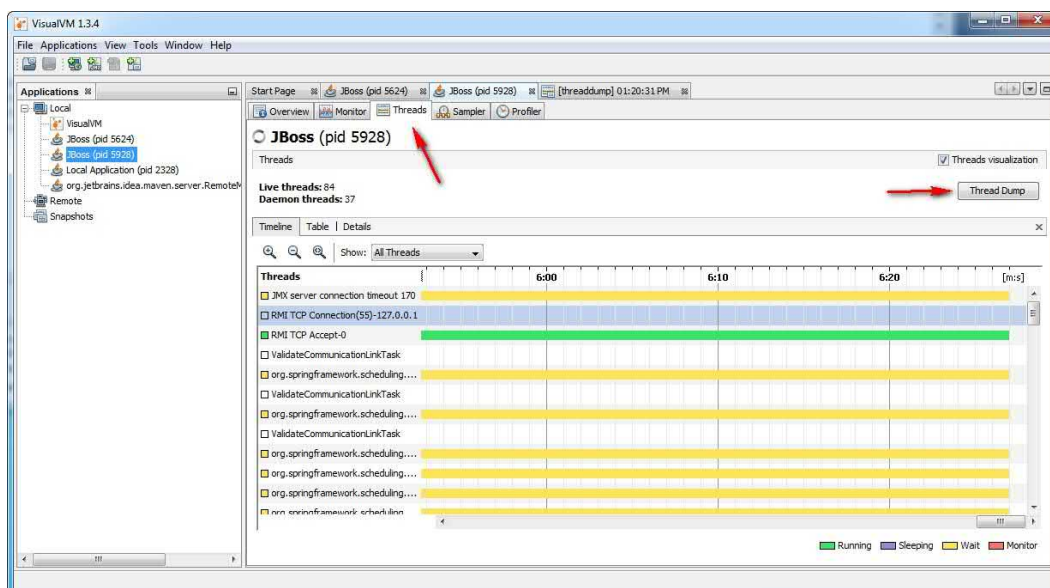
# КАК СНЯТЬ THREAD DUMP

Чтобы снять thread dump, необходимо:

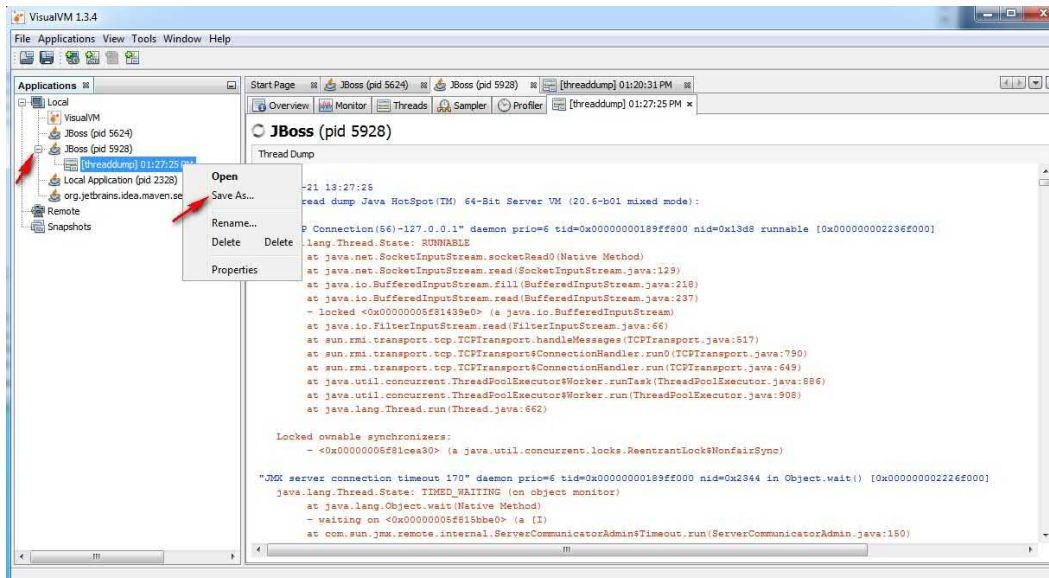
- Запустить Java VisualVM (находится в папке bin)
- Распаковать архив и запустить приложение
- В левой области под названием “Applications” выбрать jboss. Их там 2 – cdi и factor. Понять, где какой, можно в описании приложения



- 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».



# 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 -Xrunjdwp: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
```

## Первый вариант

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.

## Второй вариант

Командой вида

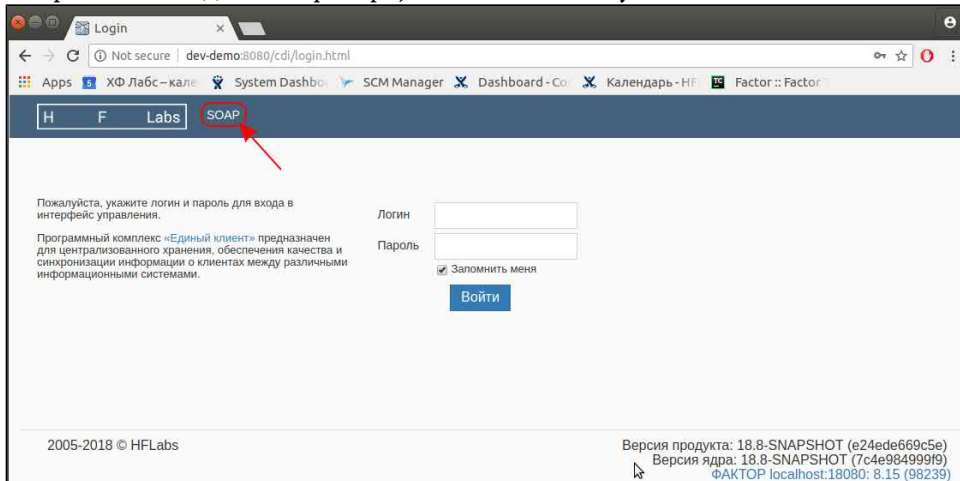
```
jstack -F 29271 > /opt/cdi/jstack.log
```

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.

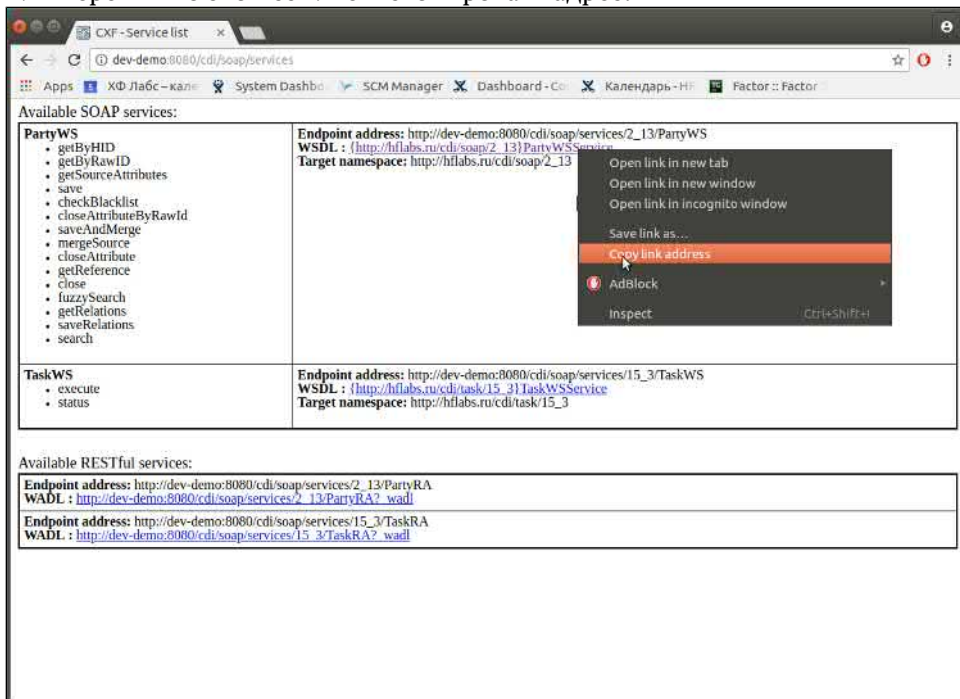
# КАК МОЖНО ВЫЗВАТЬ МЕТОД SOAP-ИНТЕРФЕЙСА?

Чтобы проверить работу онлайн-интерфейсов вручную, можно использовать инструмент SoapUI (можно скачать тут или с GitHub):

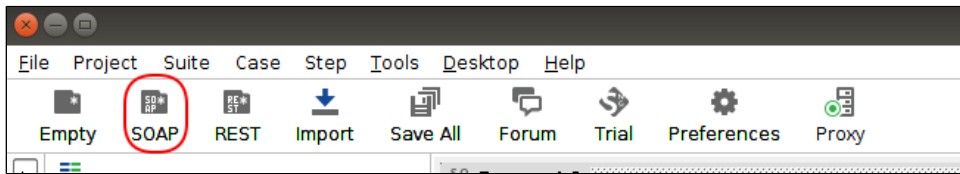
1. Открыть APM Администратора, нажать на ссылку SOAP:



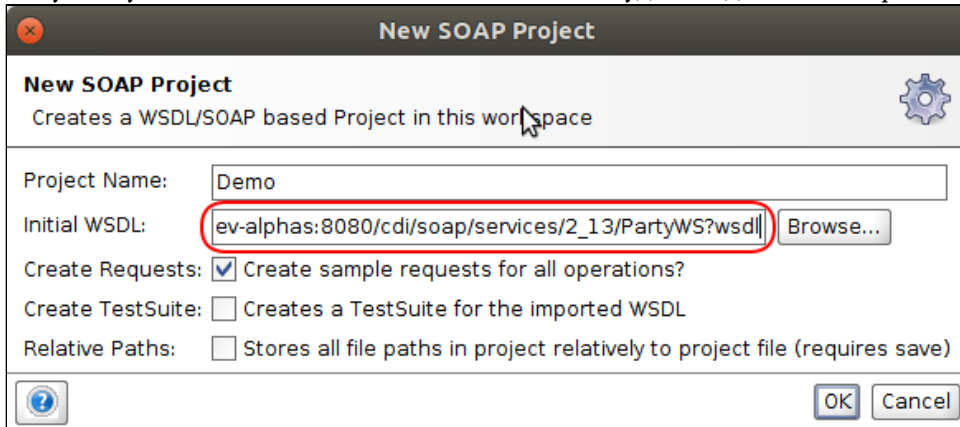
2. Система отображает все доступные пользователю сервисы (PartyWS, TaskWS и т. д.). Нажать на ссылку WSDL у необходимого сервиса правой кнопкой мыши и выбрать copy link address или перейти по этой ссылке и скопировать адрес:



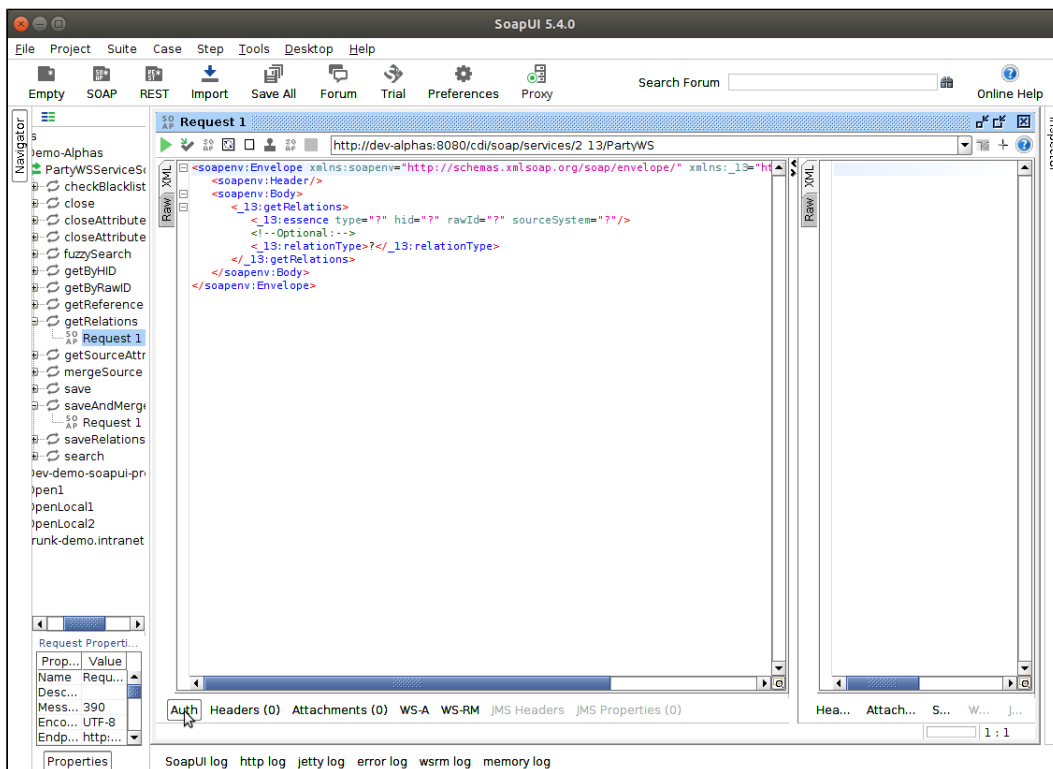
3. В SoapUI нажать на кнопку SOAP (Create a new SOAP Project):



4. В открывшемся окне необходимо задать имя новому проекту и ввести ссылку на WSDL, полученную на шаге 2. После чего нажать "OK". Будет создан новый проект.



После выбора запроса (save, close и т. д.) нажмите на кнопку Auth (Authorization) слева внизу и авторизуйтесь:

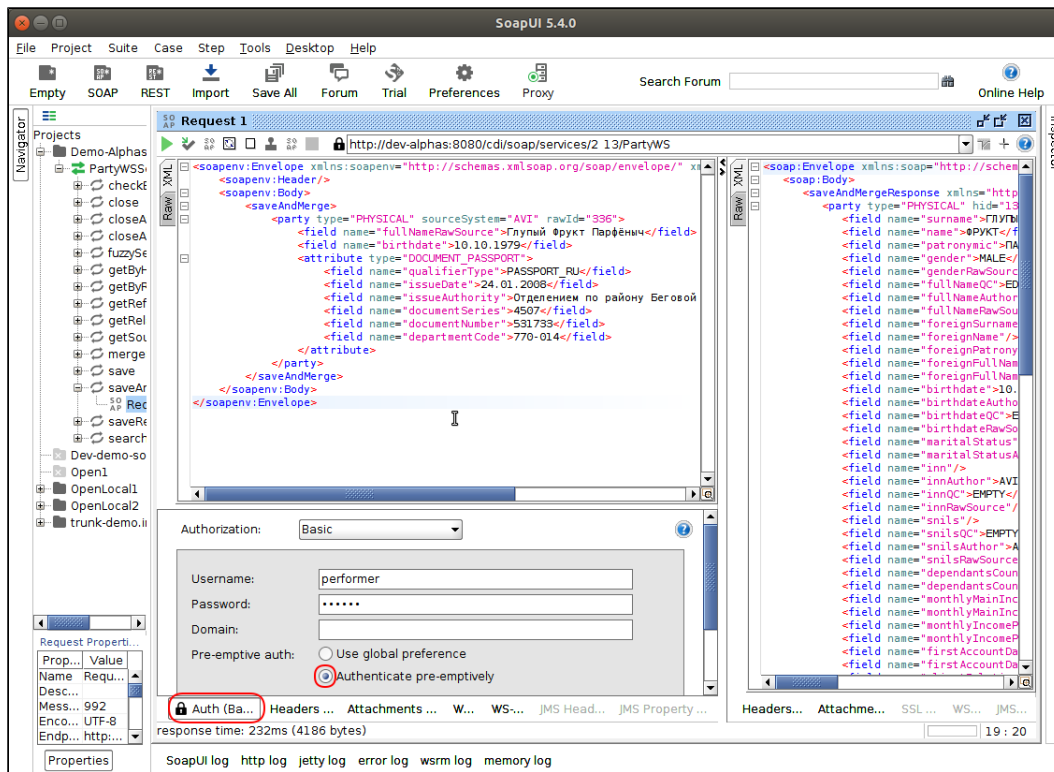




Как правило у учётной записи менеджера данных есть необходимые права (ROLE\_EXTERNAL\_SYSTEM), однако в отдельных случаях потребуется специальная учётная запись. Подробнее смотрите в спэйсе заказчика.

Чаще всего хватит логина под admin\_performer.

Необходимо также выбрать предварительную авторизацию (Authenticate pre-emptively). Если авторизация не выбрана, в доступе будет отказано.

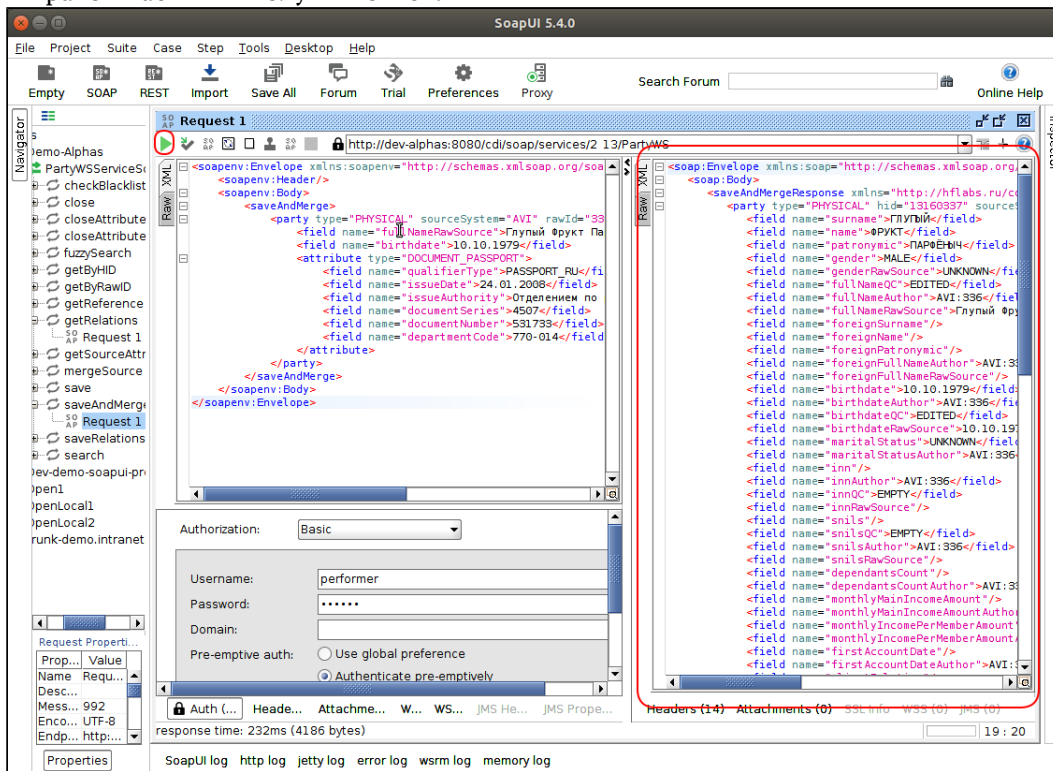


Теперь необходимо отредактировать шаблон запроса (в данном случае search):



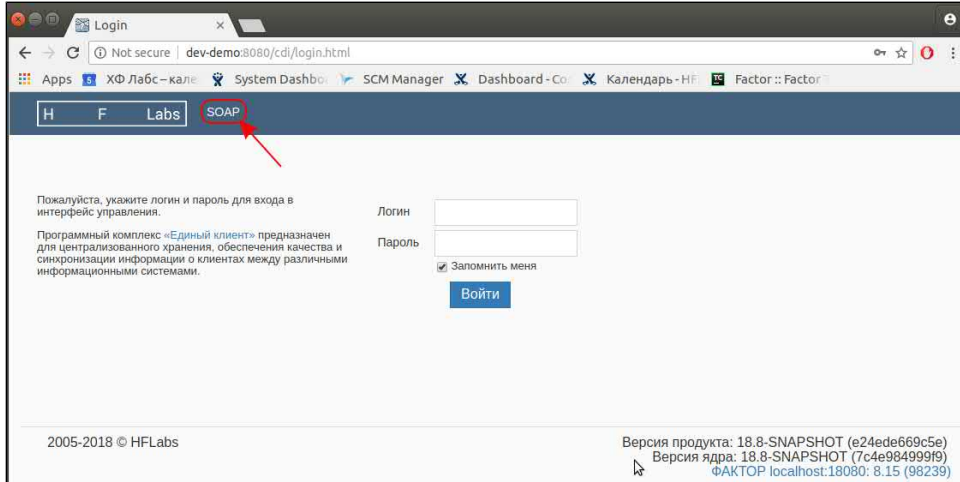
Его необходимо отредактировать согласно документации (выше приведена ссылка на пример запроса search) и нажать зелёную стрелку (кнопка submit request).

В правой части мы получим ответ:

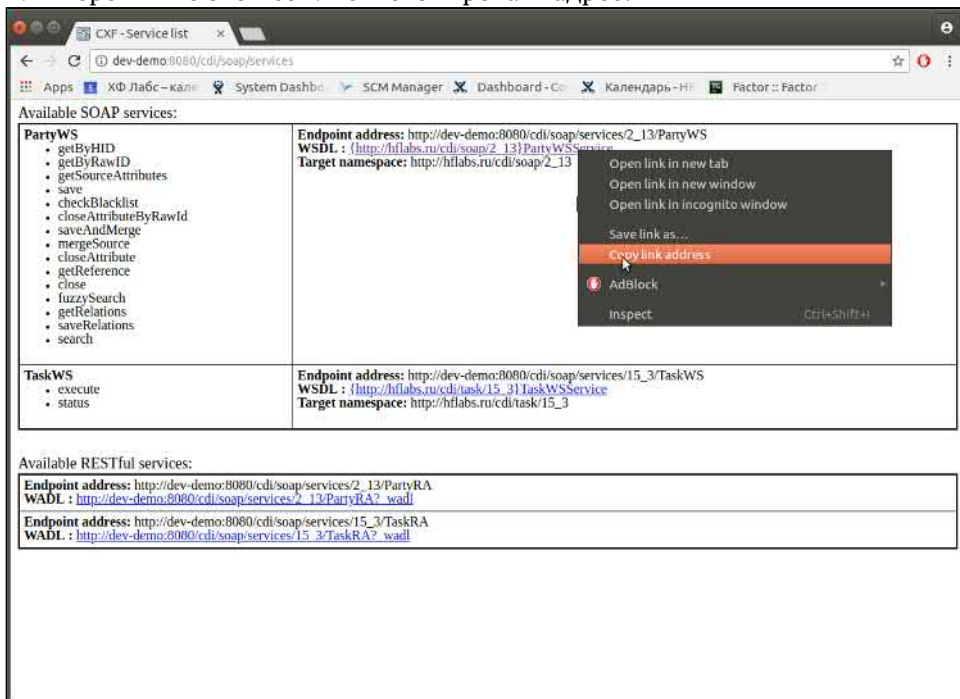


# СОЗДАНИЕ ПРОЕКТА В SOAPUI

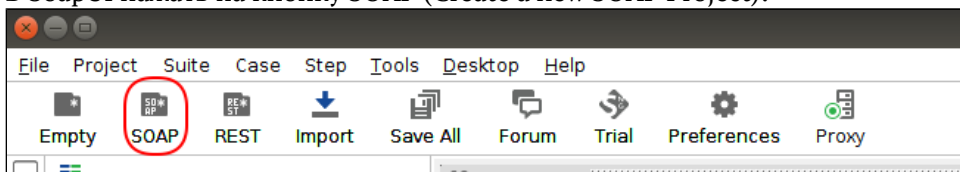
1. Открыть АРМ Администратора, нажать на ссылку SOAP:



2. Система отображает все доступные пользователю сервисы (PartyWS, TaskWS и т. д.). 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 SoapUI нажать на кнопку SOAP (Create a new SOAP Project):



4. В открывшемся окне необходимо задать имя новому проекту и ввести ссылку на WSDL, полученную на шаге 2. После чего нажать "OK". Будет создан новый проект.

**New SOAP Project**  
Creates a WSDL/SOAP based Project in this workspace

Project Name:

Initial WSDL:

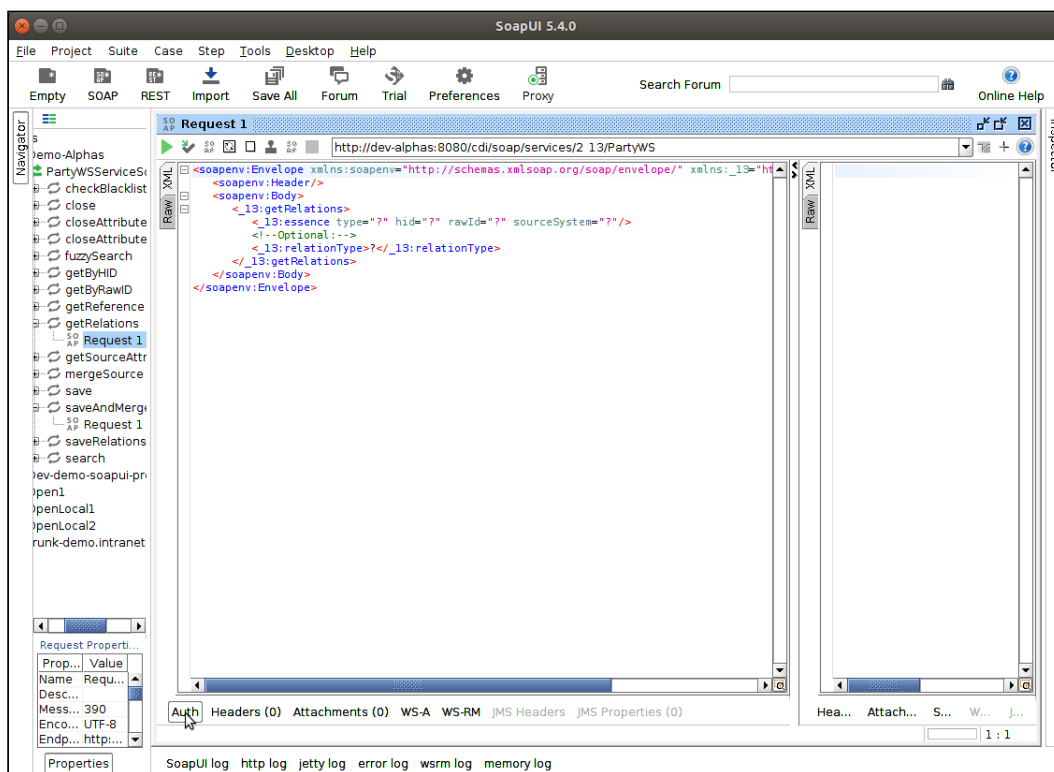
Create Requests:  Create sample requests for all operations?

Create TestSuite:  Creates a TestSuite for the imported WSDL

Relative Paths:  Stores all file paths in project relatively to project file (requires save)

# АВТОРИЗАЦИЯ ЗАПРОСА В SOAPUI

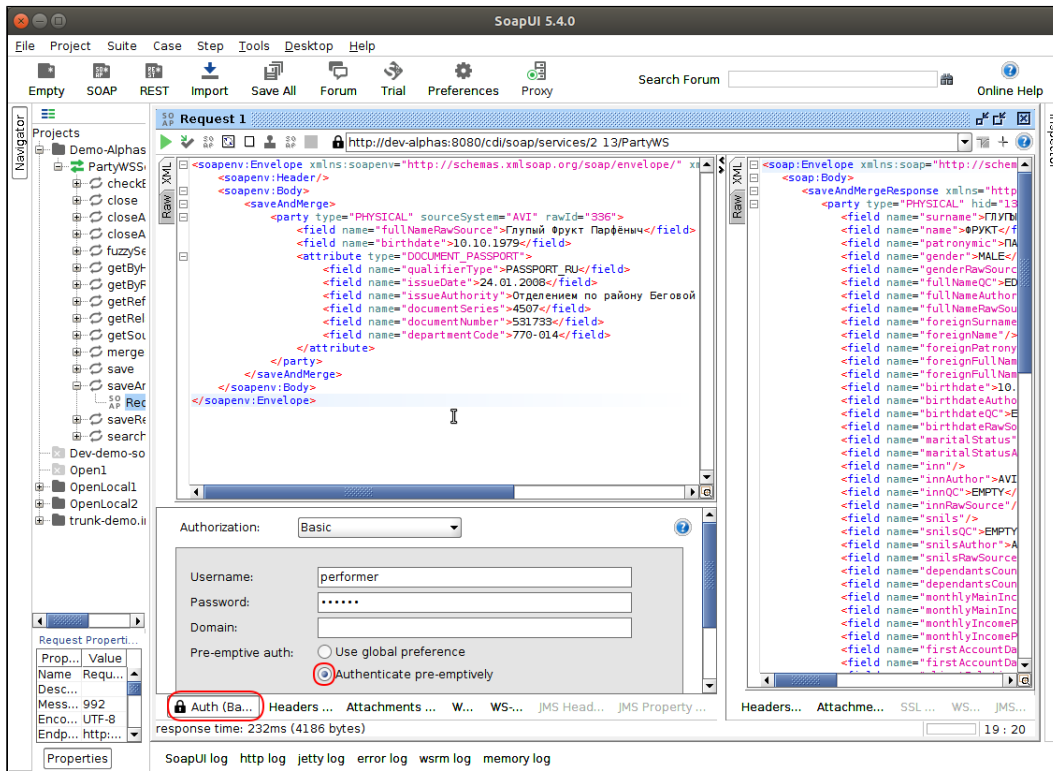
После выбора запроса (save, close и т. д.) нажмите на кнопку Auth (Authorization) слева внизу и авторизуйтесь:



Как правило у учётной записи **менеджера данных** есть необходимые права (ROLE\_EXTERNAL\_SYSTEM), однако в отдельных случаях потребуются специальная учётная запись. Подробнее смотрите в спэйсе заказчика.

Чаще всего хватит логина под `admin_performer`.

Необходимо также выбрать предварительную авторизацию (Authenticate pre-emptively). Если авторизация не выбрана, в доступе будет отказано.

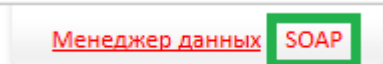


# КАК МОЖНО ВЫЗВАТЬ REST-МЕТОД ВРУЧНУЮ

Для ручного вызова REST-методов различных интерфейсов ЕК и Фактора можно использовать SoapUI. Описание на примере стенда dev-demo.

## Методы ЕК

1. Открыть АРМ Администратора, нажать на ссылку SOAP.



2. Скопировать ссылку на WADL нужного REST-интерфейс.

Available RESTful services:

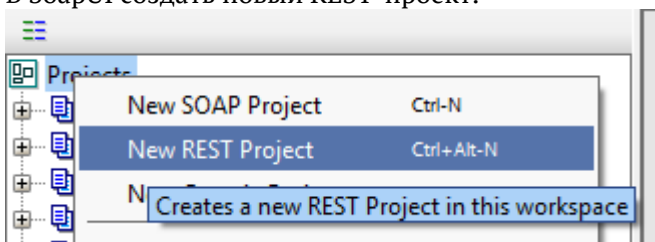
Endpoint address: [http://dev-demo:8080/cdi/soap/services/15\\_3/TaskRA](http://dev-demo:8080/cdi/soap/services/15_3/TaskRA)

WADL : [http://dev-demo:8080/cdi/soap/services/15\\_3/TaskRA? wadl](http://dev-demo:8080/cdi/soap/services/15_3/TaskRA? wadl)

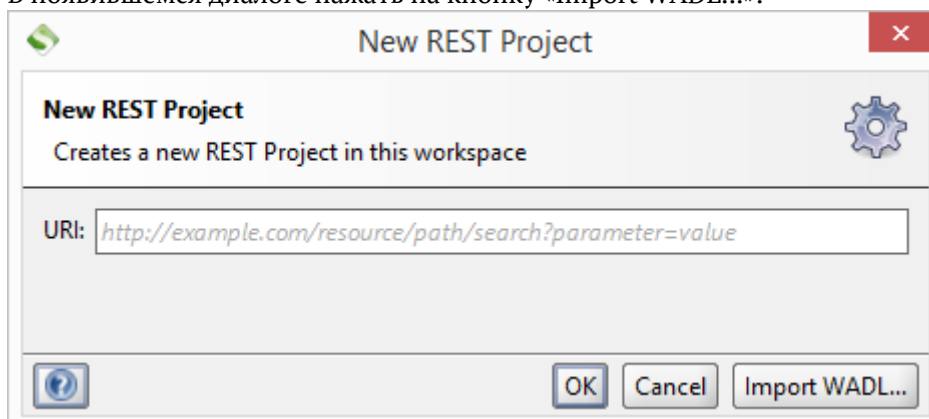
Endpoint address: [http://dev-demo:8080/cdi/soap/services/2\\_13/PartyRA](http://dev-demo:8080/cdi/soap/services/2_13/PartyRA)

WADL : [http://dev-demo:8080/cdi/soap/services/2\\_13/PartyRA? wadl](http://dev-demo:8080/cdi/soap/services/2_13/PartyRA? wadl)

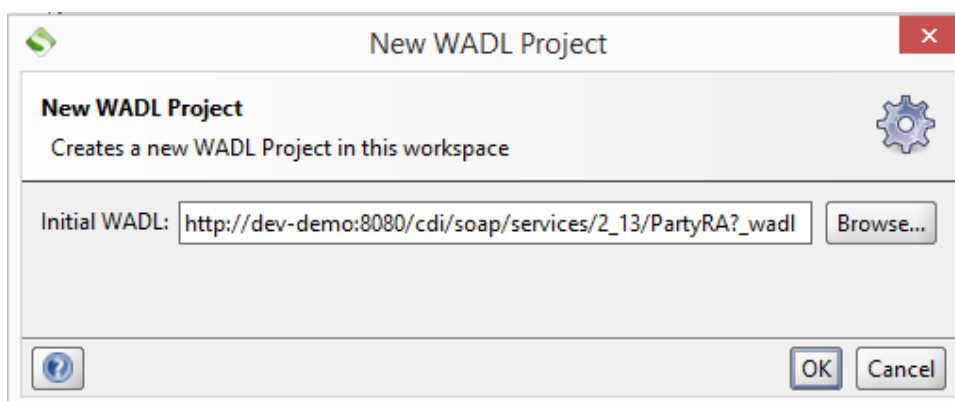
3. В SoapUI создать новый REST-проект.



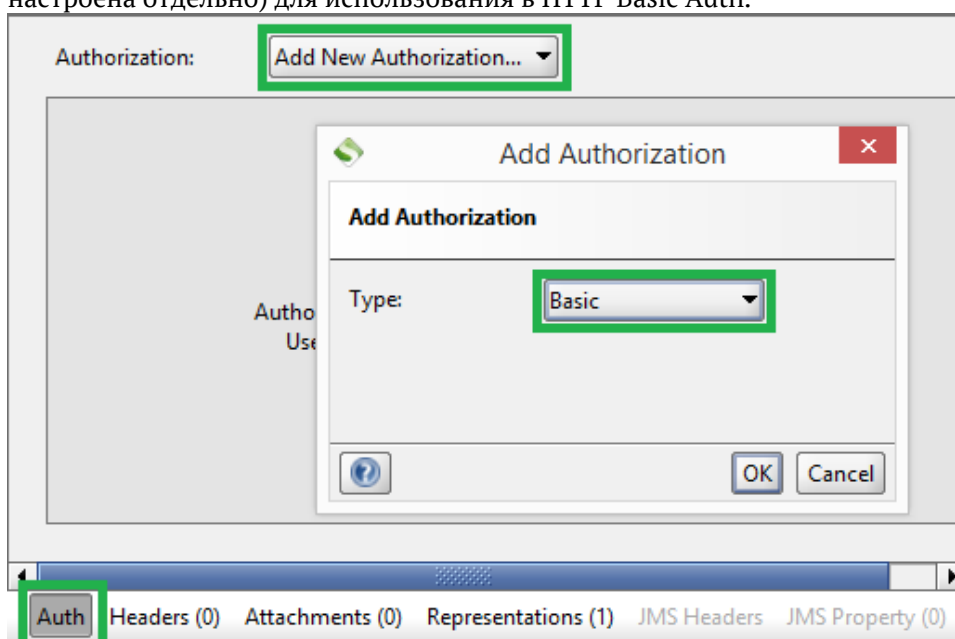
4. В появившемся диалоге нажать на кнопку «Import WADL...».



5. Вставить скопированную ранее ссылку и нажать «OK».

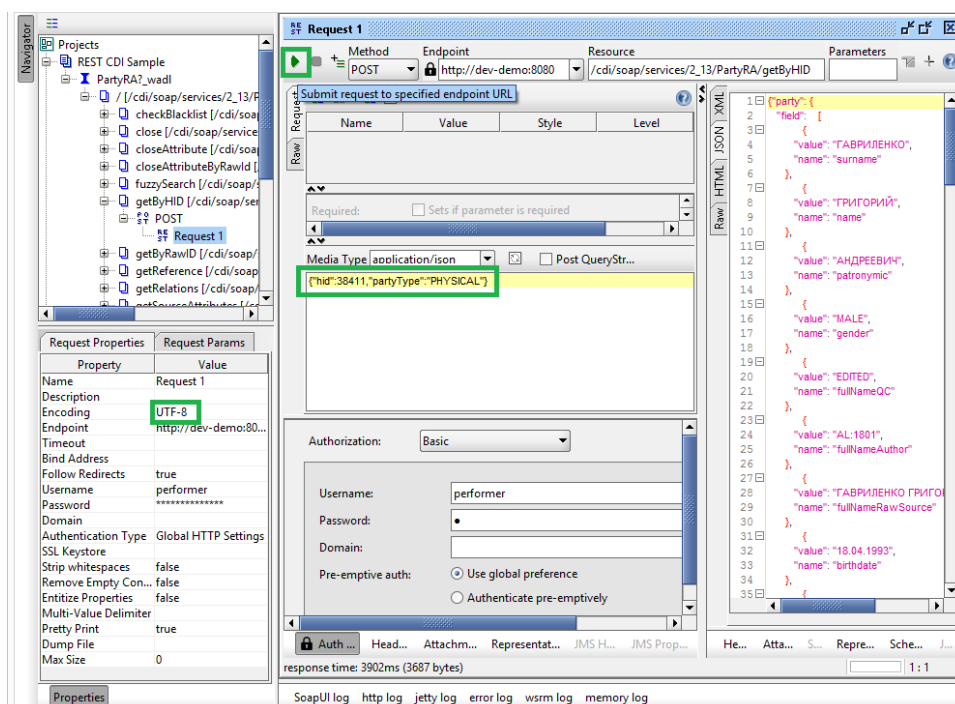


6. Указать данные пользователя с правами менеджера данных (или вебсервиса, если эта роль настроена отдельно) для использования в HTTP Basic Auth.



7. Выбрать нужную кодировку, заполнить данные запроса, нажать Submit.





## Методы Фактор

У Фактора нет WADL и авторизации, поэтому чуть по-другому. Зато можно отправлять как POST-, так и GET-запросы.

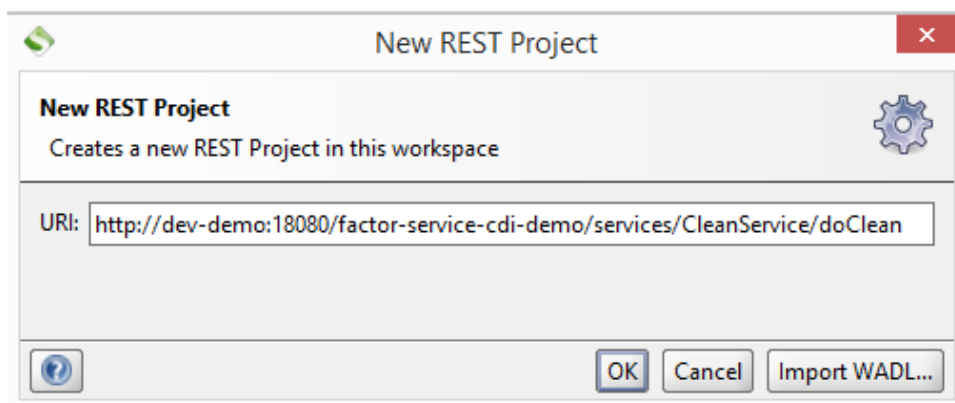
1. В SoapUI создать новый REST-проект.
2. В появившемся окне указать ссылку REST-сервиса Фактора.

-- Шаблон

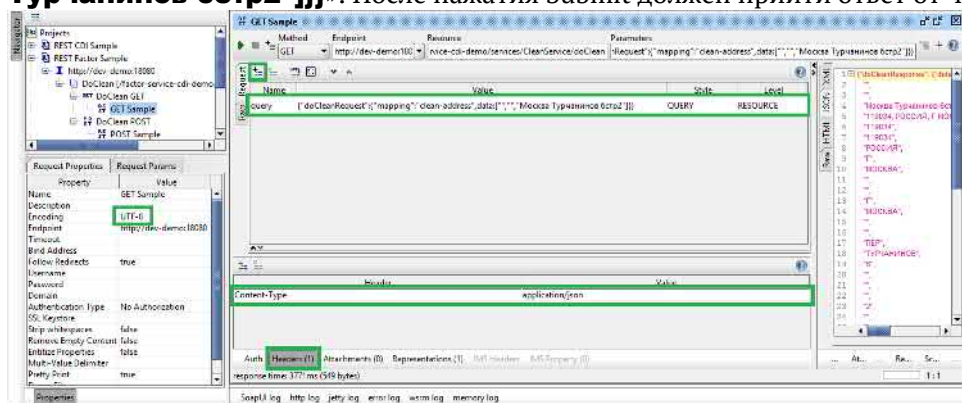
**http://{ip}:{port}/factor-service-{customer}/services/CleanService/{metod}**

-- Например, для заказчика cdi-demo и метода doClean

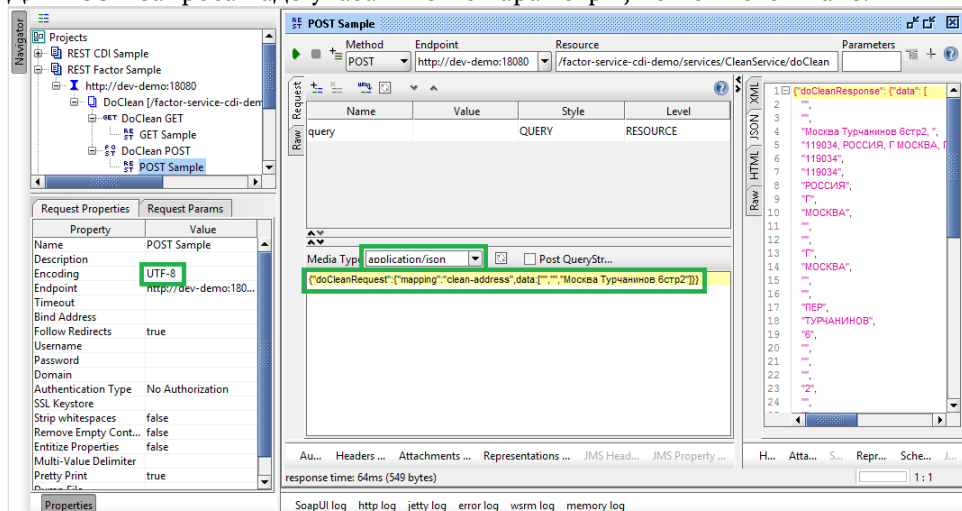
**http://dev-demo:18080/factor-service-cdi-demo/services/CleanService/doClean**



3. Для GET-запроса надо указать кодировку, добавить заголовок **Content-Type=application/json**, указать параметр **query** (можно делать через табличку 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.



# КАК ПРОВЕРИТЬ УЧЕТНУЮ ЗАПИСЬ LDAP?

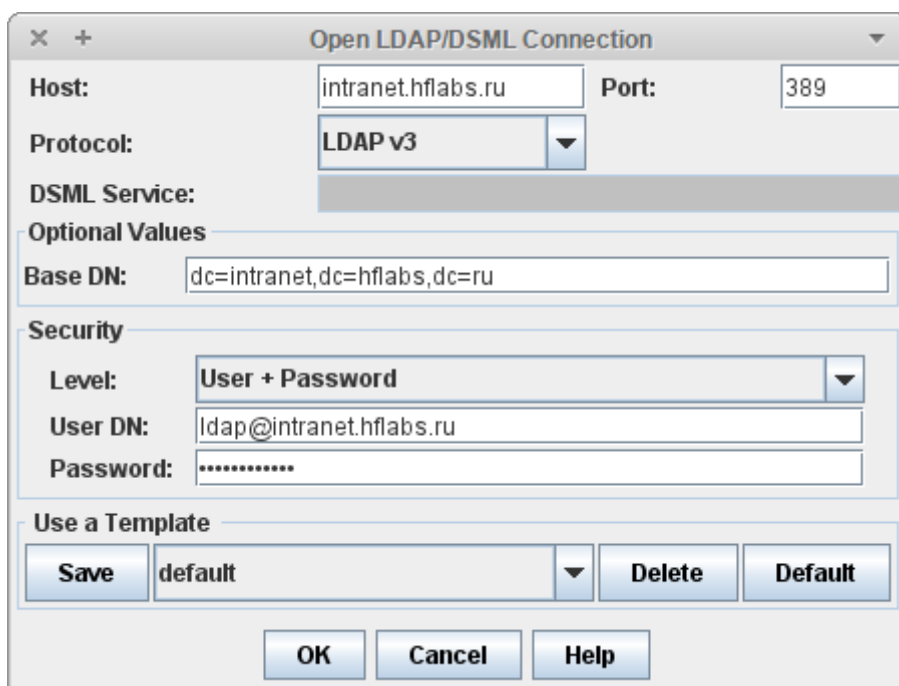
1. Открыть раздел **Параметры LDAP** на вкладке **Конфигурация** или получить данные из БД:

```
select * from config_property where source = 'ldap.properties';
```

2. Скачать и запустить программу **JXplorer**. В качестве альтернативы можно использовать **AD Explorer**.
3. Нажать на кнопку **Connect to DSA**



4. Заполнить форму (см. **Пример заполнения формы**).



5. Нажать OK
6. Если запустилось такое окно, значит, учетная запись работает:

? Неизвестное вложение

## 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**



# КАК ПОСМОТРЕТЬ JMS, УХОДЯЩИЕ В RABBITMQ

⚠ Под рестриками, чтобы ссылки давать

## Подготовка

Установить RabbitMq. У нас он развернут тут — <http://dev-yota.intranet.hflabs.ru:15672> (guest/guest)

Для настройки использовать параметры:

```
jms.events.rabbitmq.enabled = true  
jms.events.rabbitmq.destination.name = cdi.events  
jms.events.rabbitmq.destination.isTopic = false  
jms.events.rabbitmq.host = dev-yota.intranet.hflabs.ru  
jms.events.rabbitmq.username = guest  
jms.events.rabbitmq.password = guest
```

Заметьте, что в настройках порт 5672, а ссылка в админку — 15672. Это нормально, это административный порт

## Проверка сообщений

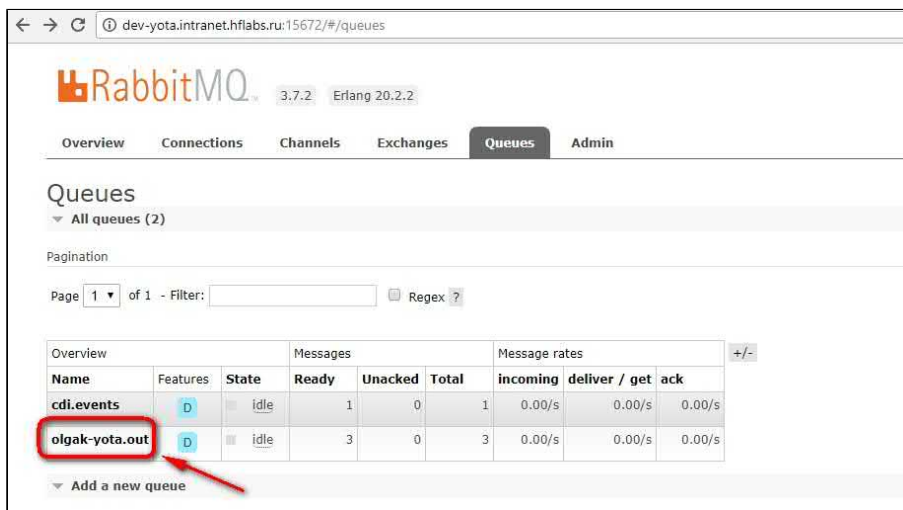
1. Открыть очереди — <http://dev-yota.intranet.hflabs.ru:15672/#/queues>
2. Создать очередь с тем же названием, что указали в параметре `jms.events.rabbitmq.destination.name`.



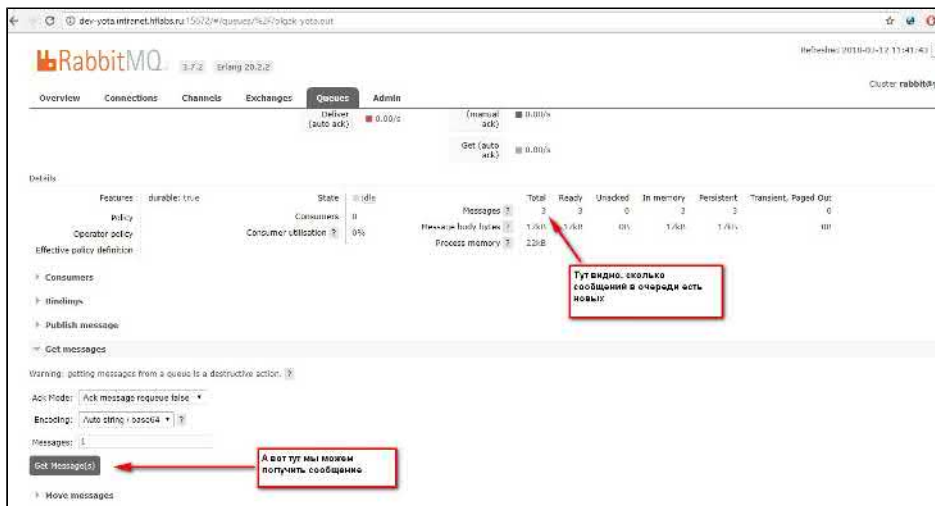
The screenshot shows the RabbitMQ Admin interface. At the top, there's a navigation bar with tabs for Overview, Connections, Channels, Exchanges, Queues, and Admin. The 'Queues' tab is active, displaying a table of existing queues. Below the table is a form to 'Add a new queue' with fields for Name, Durability (set to 'Durable'), Auto delete (set to 'No'), and Arguments. The 'Add queue' button is highlighted with a red box.

Queue Name	State	Messages	Bytes	Rate
dednik-yota.out	idle	0	0	0
kate_otp.events	idle	1,833	0	1,833 0.00/s
local-yota.out	idle	0	0	0
nikitao_otp.events	idle	0	0	0
olgak-yota.out	idle	0	0	0
tatianaqas_otp	idle	0	0	0.00/s 0.00/s 0.00/s
vadimz_sberuk.events	idle	230	0	230 0.00/s

3. Создать событие.
4. Выбрать созданную в п.2 очередь.



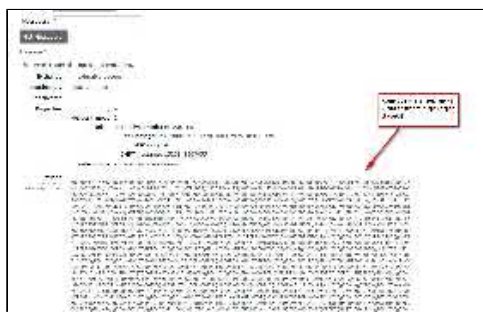
3. Нажать «Get Message» — получите сообщения или ошибку «очередь пуста». Количество сообщений видно сверху.



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/>





# КАК ПОСМОТРЕТЬ JMS, УХОДЯЩИЕ В IBM MQ

## Подготовка

Установить одну из утилит:

- **IBM MQ Explorer** (можно скачать с **overmind\soft\!\_ibm\MQ\MQ\_EXPLORER**)
- **JMSToolBox** (имхо, эта чуть удобнее)

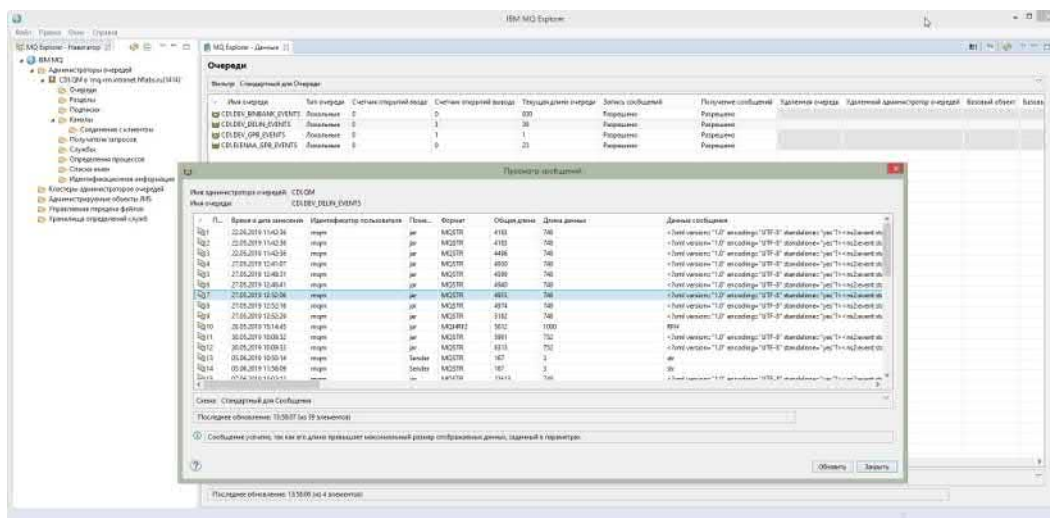
Для настройки использовать параметры:

```
jms.ibmmq.url = mq.vm.intranet.hflabs.ru
jms.ibmmq.port = 1414
jms.ibmmq.user = mqm
jms.ibmmq.password =
jms.ibmmq.sslCipherSuite =
jms.ibmmq.queueManager = CDI.QM
jms.ibmmq.channel = SYSTEM.ADMIN.SVRCONN
jms.events.destination.name = CDI.DEV_CUSTOMER_EVENTS
jms.events.enabled = true
```

## Просмотр через MQ Explorer

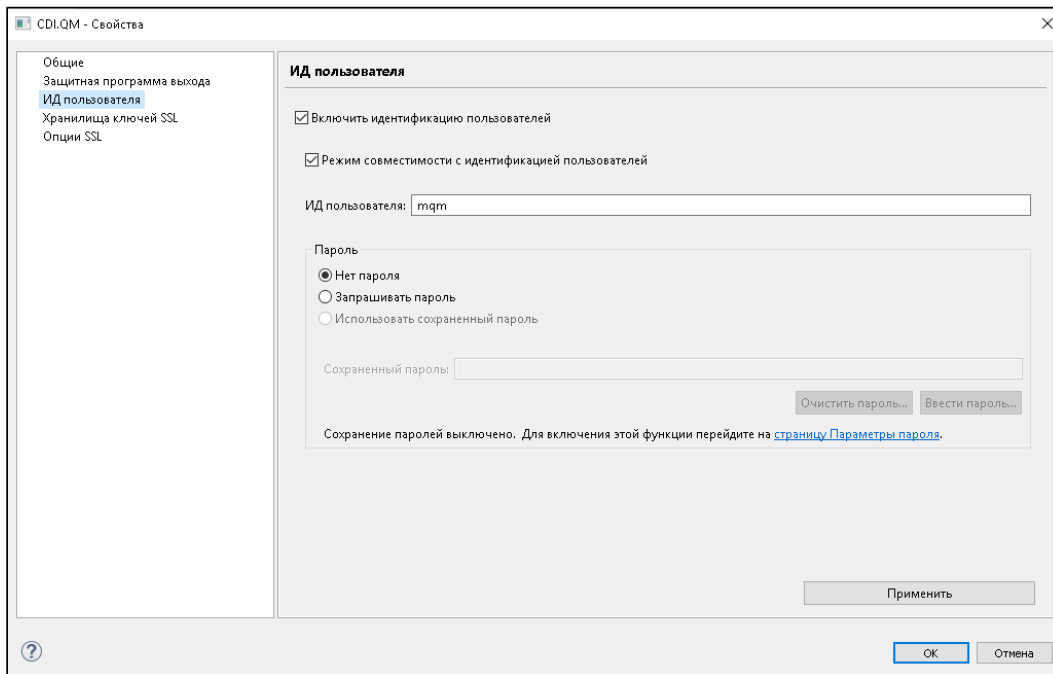
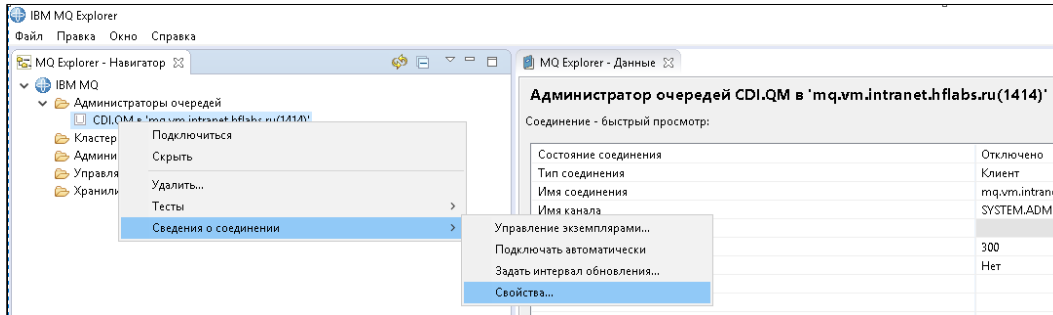
Запуск и мониторинг в MQ, следовать инструкции, но использовать ЕК параметры:

- Имя администратора очередей — **CDI.QM**
- Имя хоста — **mq.vm.intranet.hflabs.ru**
- Номер порта — **1414**
- Канал подключения к серверу — **SYSTEM.ADMIN.SVRCONN**



При подключении к администратору очереди возможна ошибка: **Доступ не разрешен. У вас нет прав доступа для выполнения этой операции. (AMQ4036).**

Закройте окно с ошибкой и задайте ИД пользователя:



Сохраните изменения и повторите попытку подключения.

## Просмотр через JMSToolBox

Скачать **jar**-ник — [com.ibm.mq.allclient.jar](#)

Скачать и запустить JMSToolBox.

В JMSToolBox перейти на вкладку Q Managers там выбрать IBM MQ, и добавить скаченный файл `com.ibm.mq.allclient.jar`

Создать новую сессию:

Update Session

Connection **Properties**

**Definition**

Session Name

Queue Manager

Folder

Session Type

**Servers**

Host / Port

Host / Port (2)

Host / Port (3)

**Security**

User id

Password

Prompt for userid/password

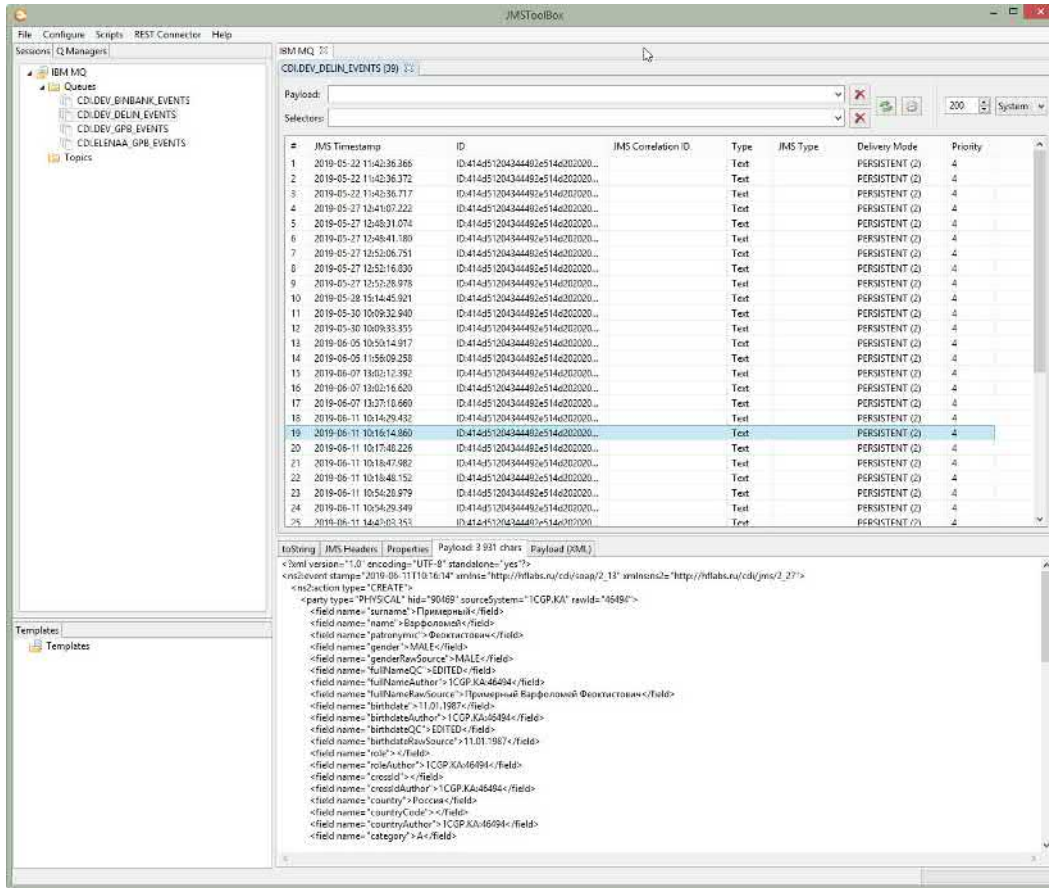
If the userid/password are NOT saved in the session, the 'REST' and 'scripts' features of JMSToolBox will not work

Update Session

Connection **Properties**

Name	Kind	Value
* channel	STRING	SYSTEM.ADMIN.SVRCONN
queueManager	STRING	CDI.QM
channelReceiveExit	STRING	
channelReceiveExitUserData	STRING	
channelSecurityExit	STRING	
channelSecurityExitUserData	STRING	
channelSendExit	STRING	
channelSendExitUserData	STRING	
com.ibm.mq.cfg.useIBMCipherMappings	BOOLEAN	
javax.net.ssl.keyStore	STRING	
javax.net.ssl.keyStorePassword	STRING	
javax.net.ssl.keyStoreType	STRING	
javax.net.ssl.trustStore	STRING	
javax.net.ssl.trustStorePassword	STRING	
javax.net.ssl.trustStoreType	STRING	
sslCipherSuite	STRING	
sslFipsRequired	BOOLEAN	

Можно пользоваться:



# ЗАПУСК И МОНИТОРИНГ В MQ

## УСТАНОВКА IBM WEBSHERE MQ EXPLORER

Это утилита от IBM с возможностью отправки/просмотра сообщений в очередь(-и) MQ.

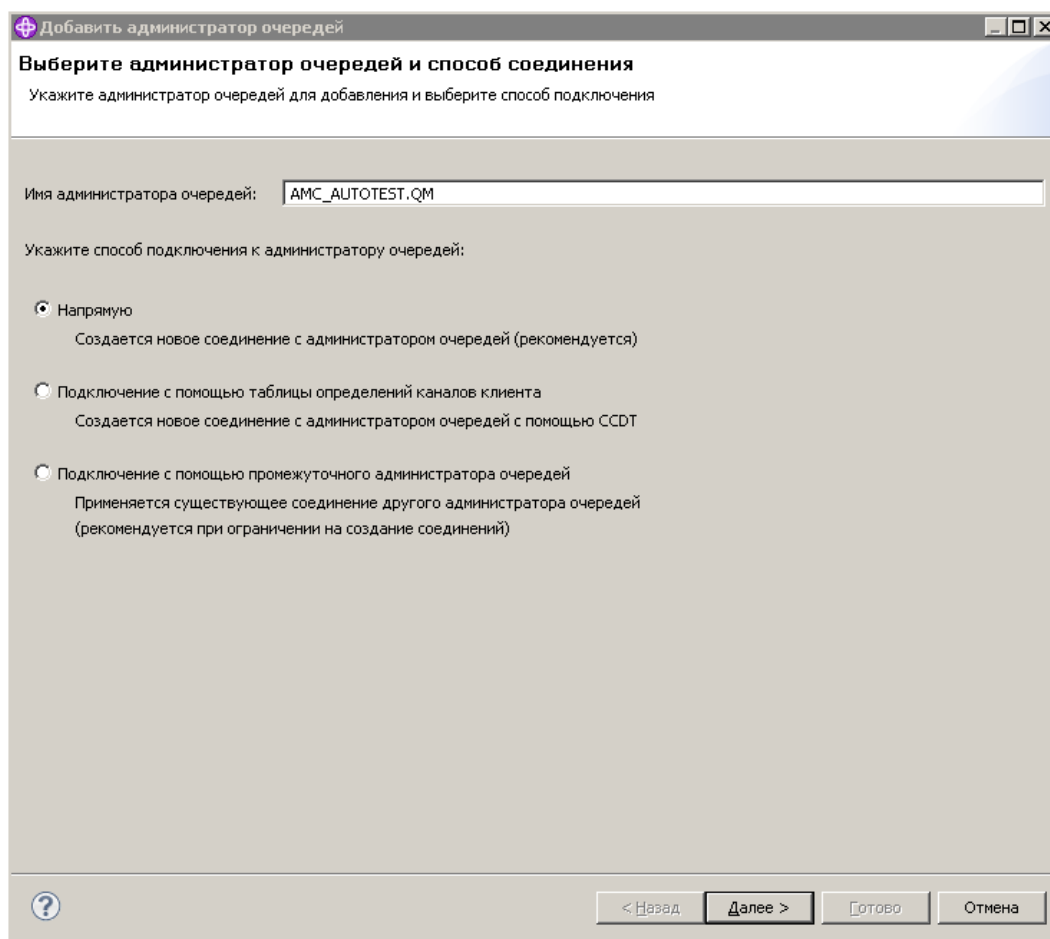
Преимущество перед HermesJMS и Apache Jmeter в том, что легко устанавливается и настраивается и позволяет сразу приступить к работе.

Дистрибутив под Windows здесь:

`\\10.0.61.20\share\distribs\ms0t_mqexplorer_75_windows.zip`  
или  
`http://cloud.hflabs.ru/index.php/apps/files/?dir=/others/mq`

Установка происходит по принципу "Далее" – "Готово" (единственное – из пути установки лучше убрать пробелы).

Затем в разделе "Администраторы очередей" – "Добавить удаленный администратор очередей" прописываем имя администратора очередей:



и параметры подключения к нему (хост, порт, канал):

**Добавить администратора очередей**

**Укажите сведения о новом соединении**  
Укажите сведения о соединении, которое необходимо настроить

Имя администратора очередей:

Сведения о соединении

Имя хоста или IP-адрес:

Номер порта:

Канал подключения к серверу:

Существует ли несколько экземпляров администратора очередей?

Сведения о соединении со вторым экземпляром

Имя хоста или IP-адрес:

Номер порта:

Канал подключения к серверу:

Подключать автоматически

Автоматически обновлять информацию, показанную для этого администратора очередей

Интервал обновления (сек):

Если для подключения к каналу не используется SSL (а на стендах в сети ХФЛ это практически всегда так), то ждем "Готово".

Открывается окошко с очередями на этом менеджере:





## МОНИТОРИНГ

Прокинуть тестовое сообщение Фактору (с корневым элементом QualityRq, примеры см. в тестах PprbTestRunner):

Queue Managers -> {Manager} -> Queues -> UCP.FACTOR.NORM.IN -> ...Put test message

Обработка прошла корректно	В очереди UCP.FACTOR.NORM.OUT увеличился счетчик Current queue depth.  ..BrowseMessages - появилось сообщение с корневым элементом QualityRs
Обработка прошла с ошибками (например, валидации)	В очереди UCP.FACTOR.NORM.OUT увеличился счетчик Current queue depth.  ..BrowseMessages - появилось сообщение = входящему (с корневым элементом QualityRq)  Причину ошибки смотрим в factor.log (чаще всего ошибки валидации)
Фактор недоступен	Содержимое UCP.FACTOR.NORM.OUT не изменилось

Проверить подключение Фактора/АМ **после обновления и рестарта сервера:**

Queue Managers -> {Manager} -> Channels -> {Channel} -> ...Status -> ...Channel Status

В списке должны присутствовать коннекты с соответствующего IP сервера.

Без рестарта сервера проверка не работает, т.к. остаются старые, уже нерабочие, соединения.

# КАК ПОСМОТРЕТЬ JMS, УХОДЯЩИЕ В TIBCO

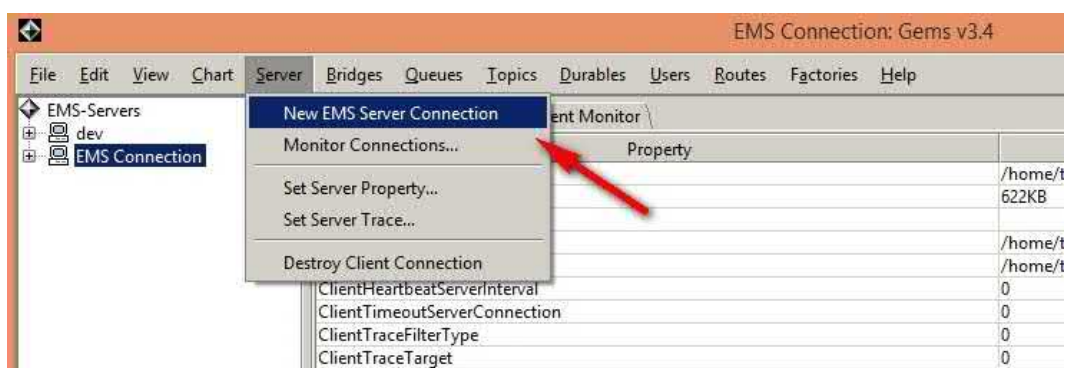
## Подготовка

Скачать утилиту **Gems**: распаковать из [архива](#). Утилита в архиве уже преднастроена, содержит библиотеки нужной версии под нашу внутреннюю tibco, а также уже содержит параметры подключения до тибко на dev-open.

## Проверка сообщений

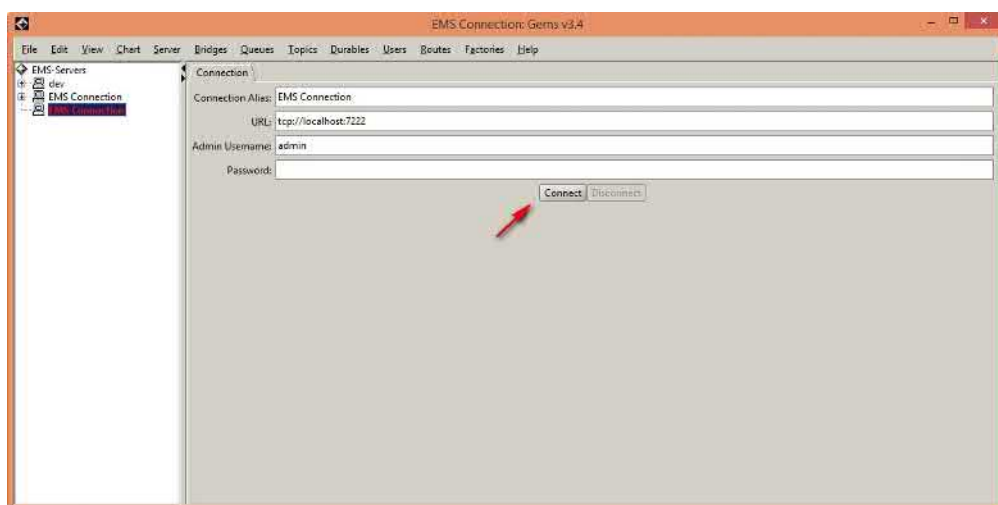
1. Запустить **Gems** — `rungems.cmd`

2. **Server** — **New EMS Server Connection**

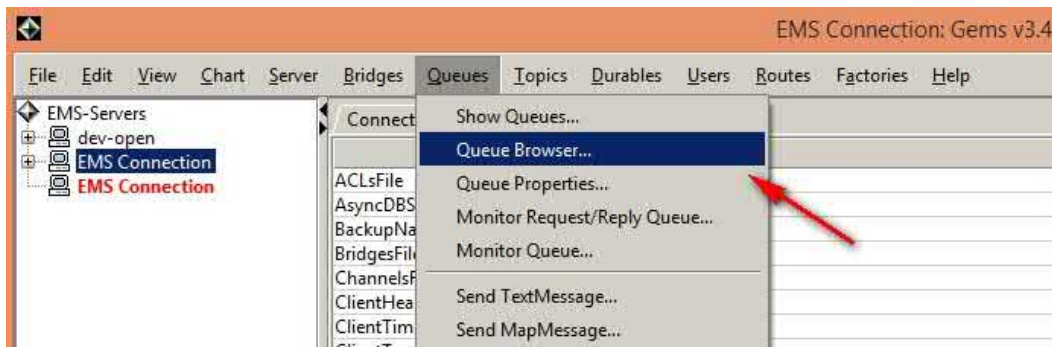


3. Указать настройки подключения, нажать **Connect**:

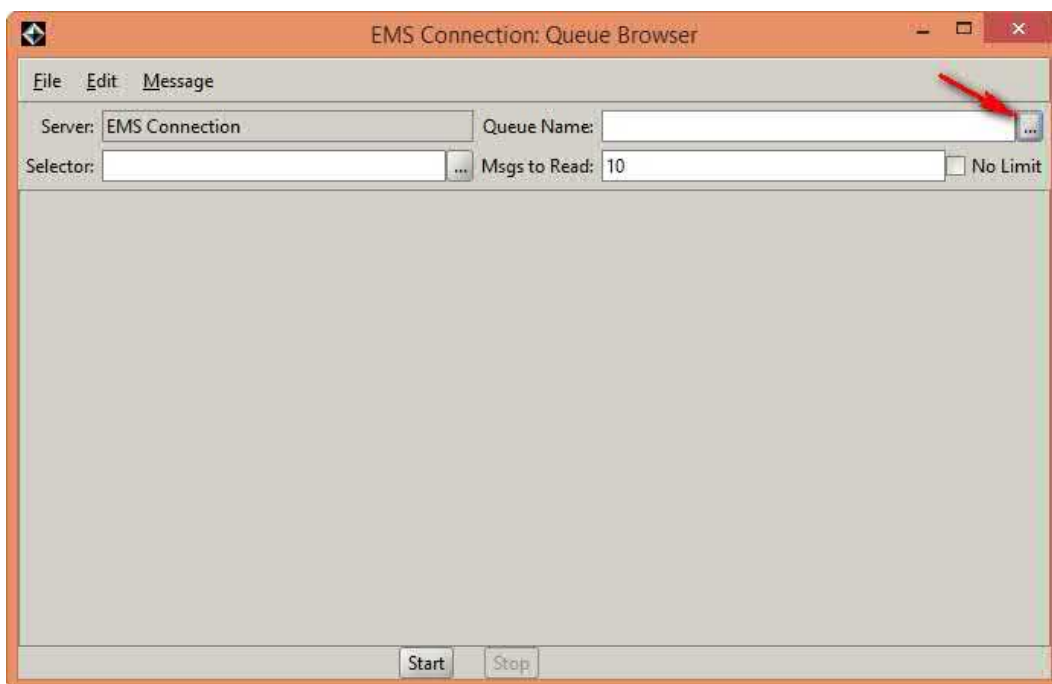
- **Alias** — имя, которое будет отображаться слева в меню
- **URL** — URL подключения к Tibco
- **Username\password** — данные для подключения



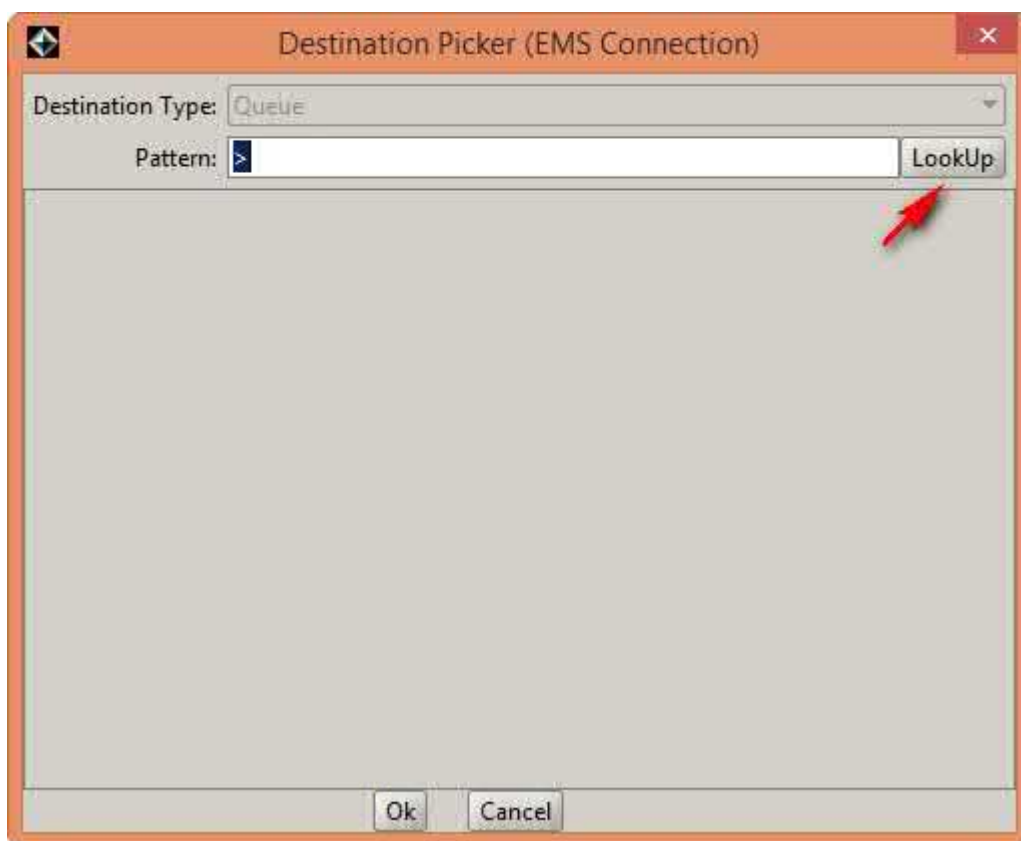
4. Выделить соединение — **Queues** — **Queue Browser**



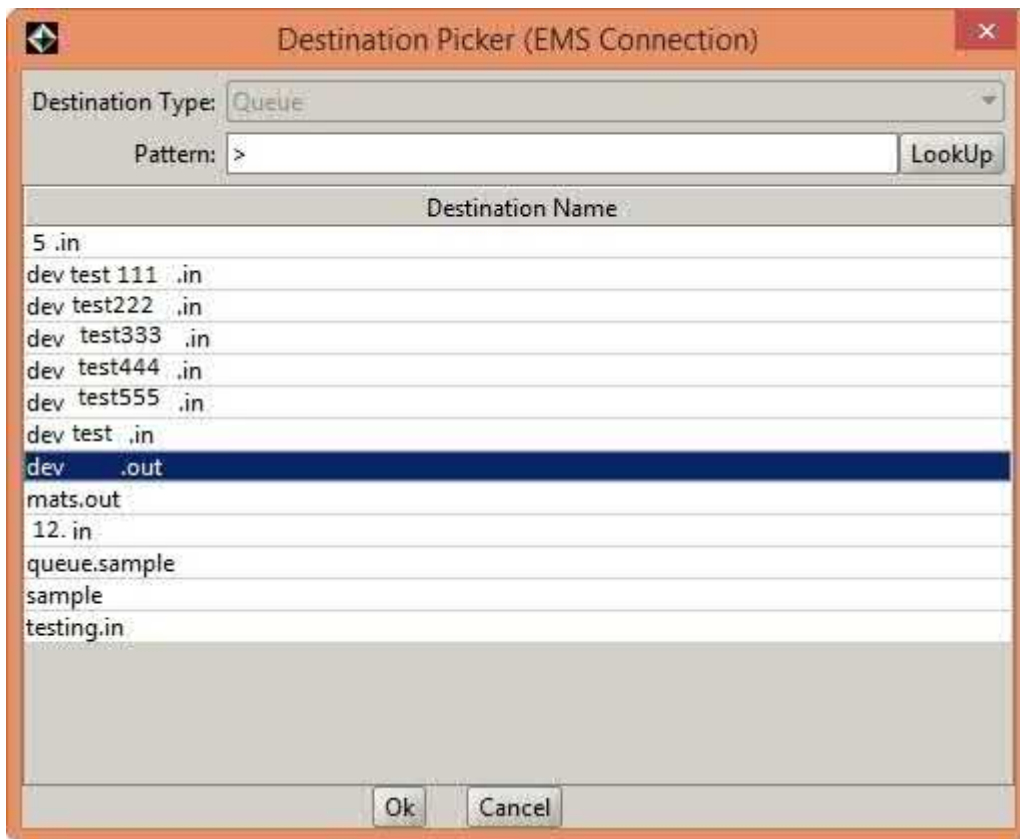
5. Нажать «...» около **Queue Name**



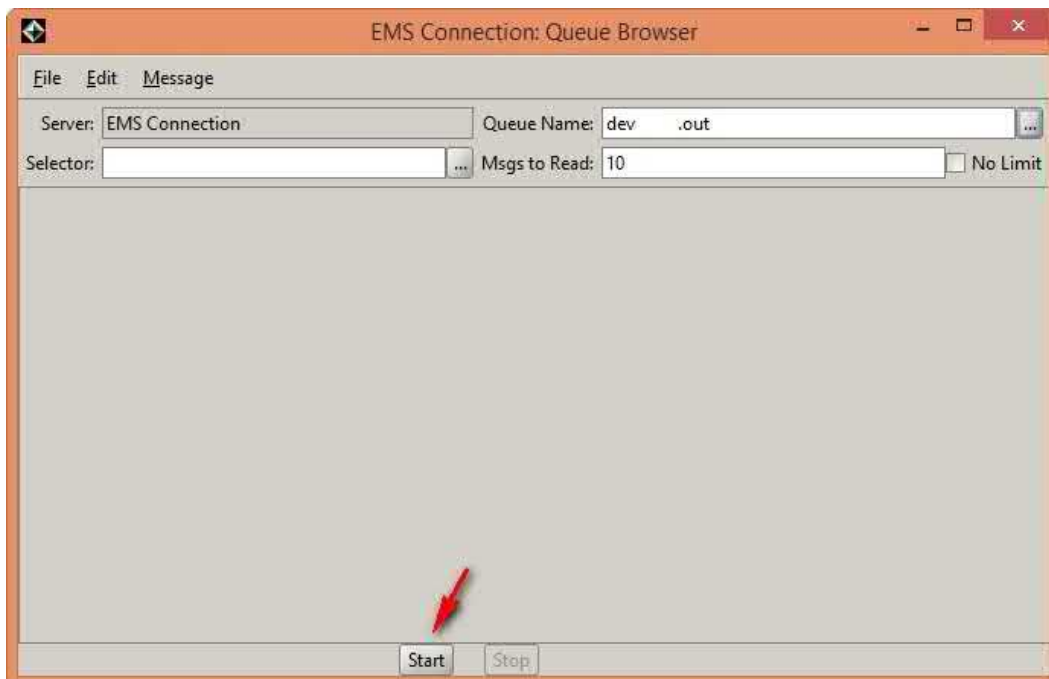
6. **LookUp** — система покажет все доступные очереди



7. Выбрать очередь, щелкнуть дважды. Выбираем исходящую очередь:

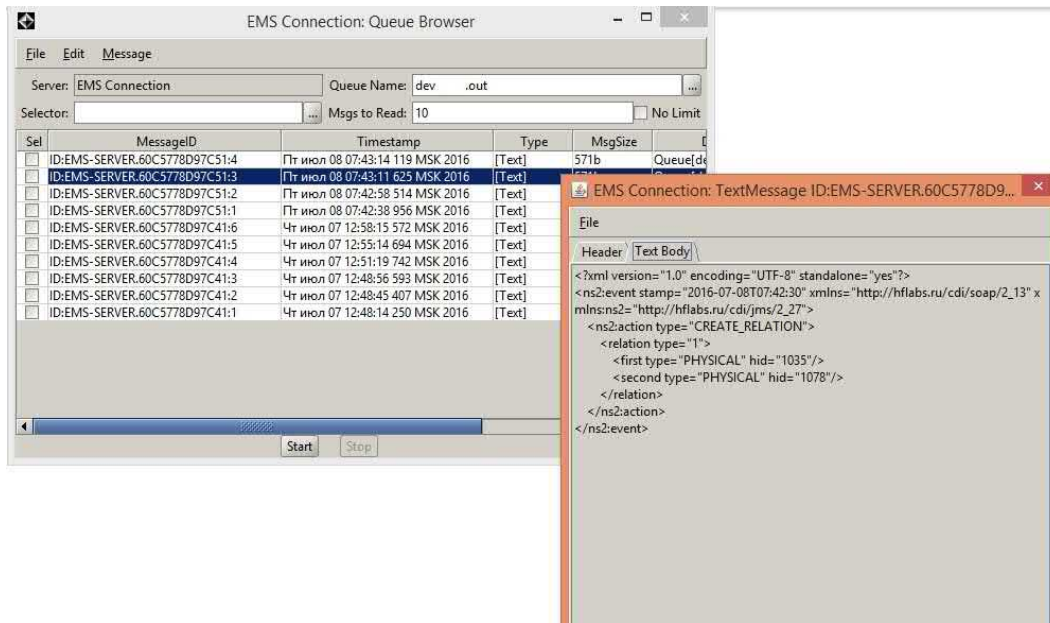


8. Нажать **Start**



9. Выбрать любое сообщение из списка. Если список пуст, сгенерить сообщение вручную, например, отправив запрос save

На вкладке **Text Body** будет текст JMS-события



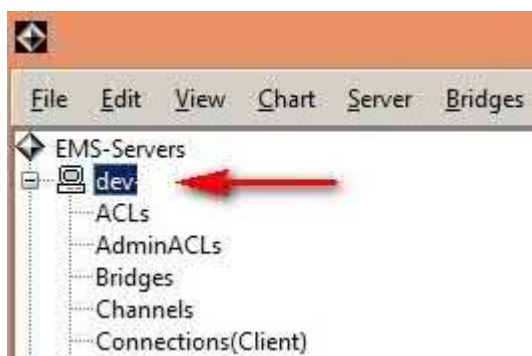
# КАК ПОСЛАТЬ В TIBCO ВХОДЯЩЕЕ JMS-СООБЩЕНИЕ

## Подготовка

Скачать утилиту **Gems**: распаковать из [архива](#). Утилита в архиве уже преднастроена, содержит библиотеки нужной версии под нашу внутреннюю tibco, а также уже содержит параметры подключения до тибко на dev-open.

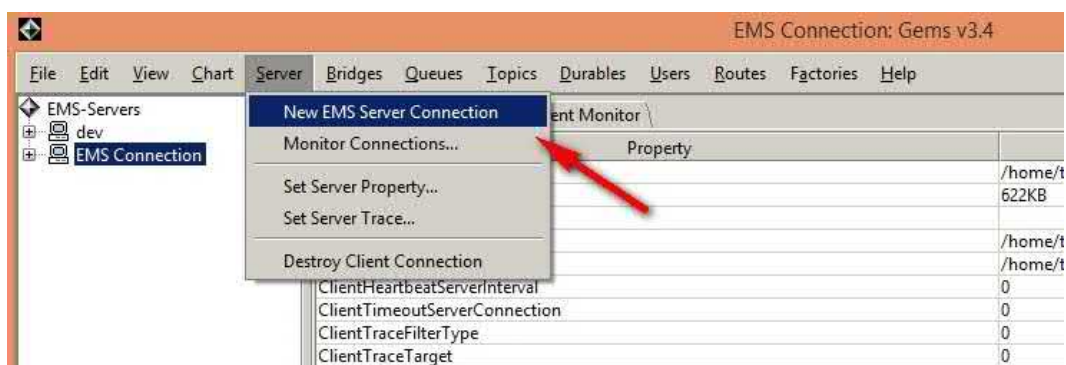
## Отправка входящего JMS-сообщения со стандартными заголовками

1. Запустить **Gems** — `rungems.cmd`
2. Если уже есть **EMS Server Connection** — выделить его



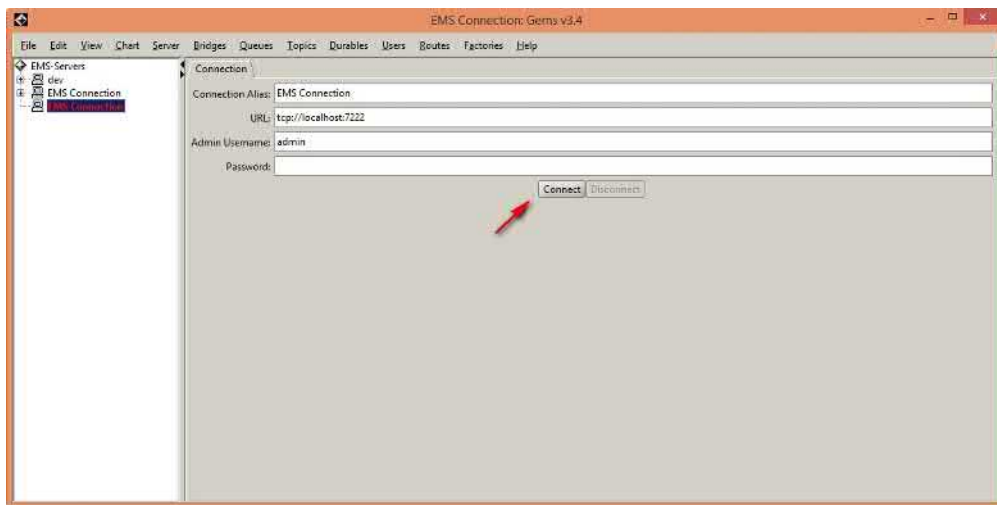
Если соединения еще нет, создаем:

### Server — New EMS Server Connection

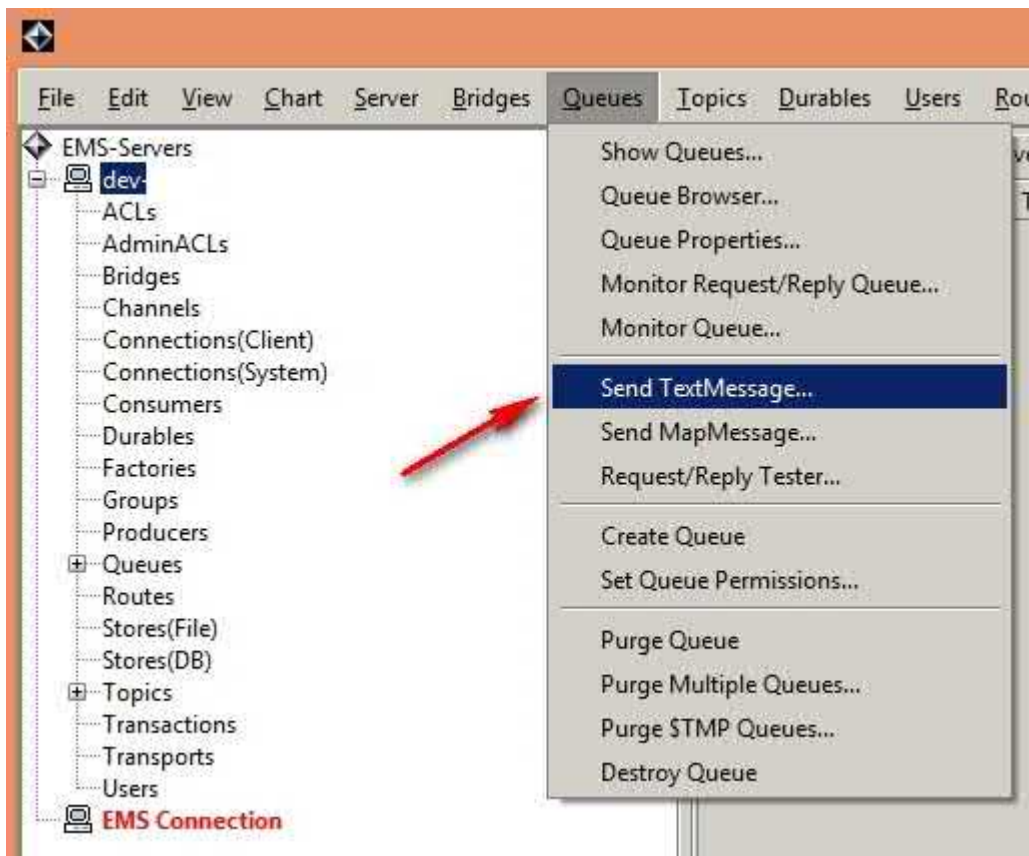


Указать настройки подключения, нажать **Connect**:

- **Alias** — имя, которое будет отображаться слева в меню
- **URL** — URL подключения к Tibco
- **Username\password** — данные для подключения



### 3. Queues — Send TextMessage



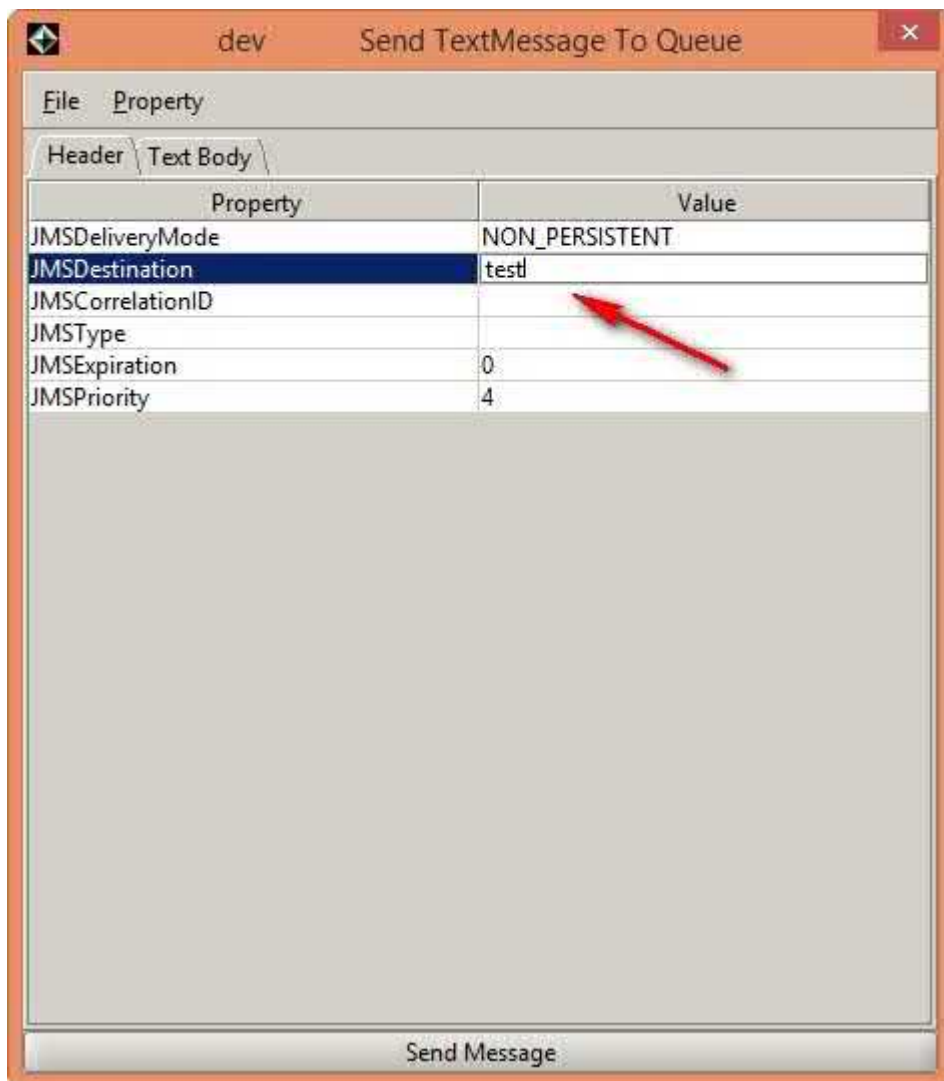
### 4. Указать параметр **JMSDestination** — место назначения.

Для ЕК в админке настроить то же место назначения:

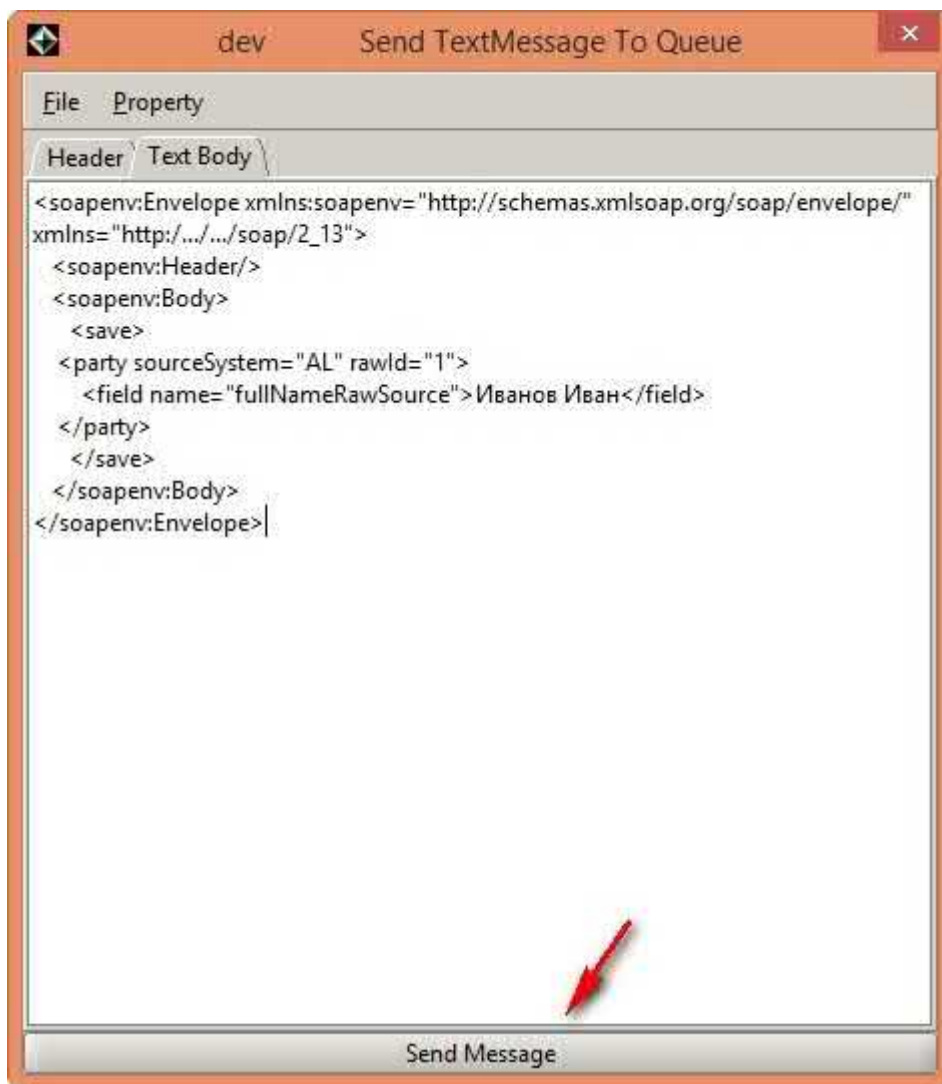
- **Конфигурация — Параметры JMS** — `.jms.soap.CFT.destination.name` (это для системы CFT, для других аналогично).



- Пример названия — **dev-open.CFT.in** (занимать уже существующее нельзя, а то разные машины будут воевать за эту очередь)

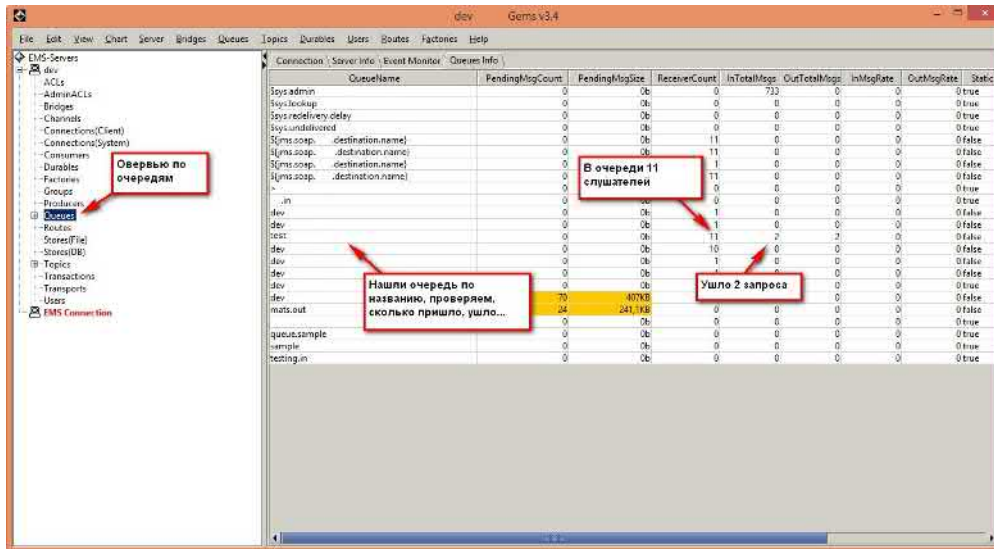


5. Все, стандартные заголовки есть, переключаемся на вкладку **Text Body** и отправляем запрос. Запрос обычный, без особых тегов, можно полностью скопировать из **Soap Ui**.



Потом проверяем внутри приложения — пришел запрос? Изменились данные?

В **Gem** можем тоже проверить — выделяем в левом меню **Queues** и видим **overview** по всем очередям

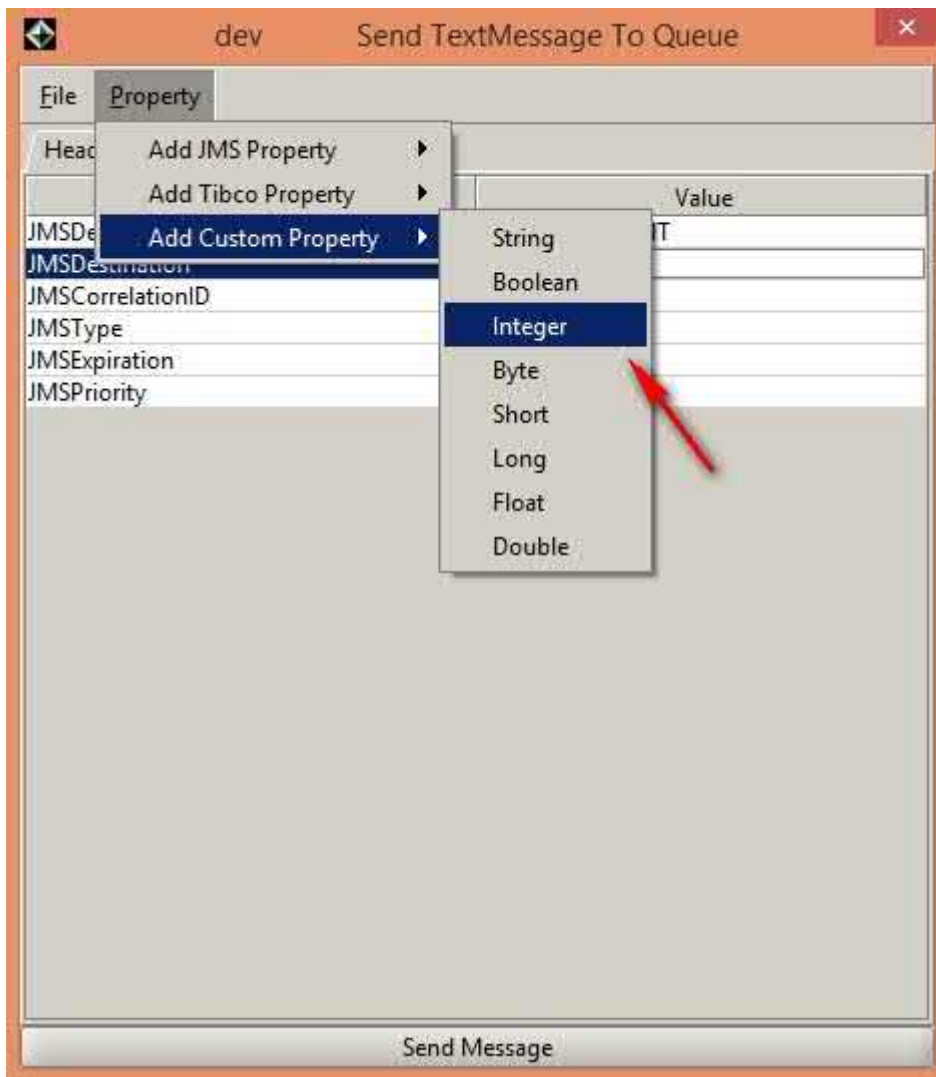


## Нестандартные заголовки

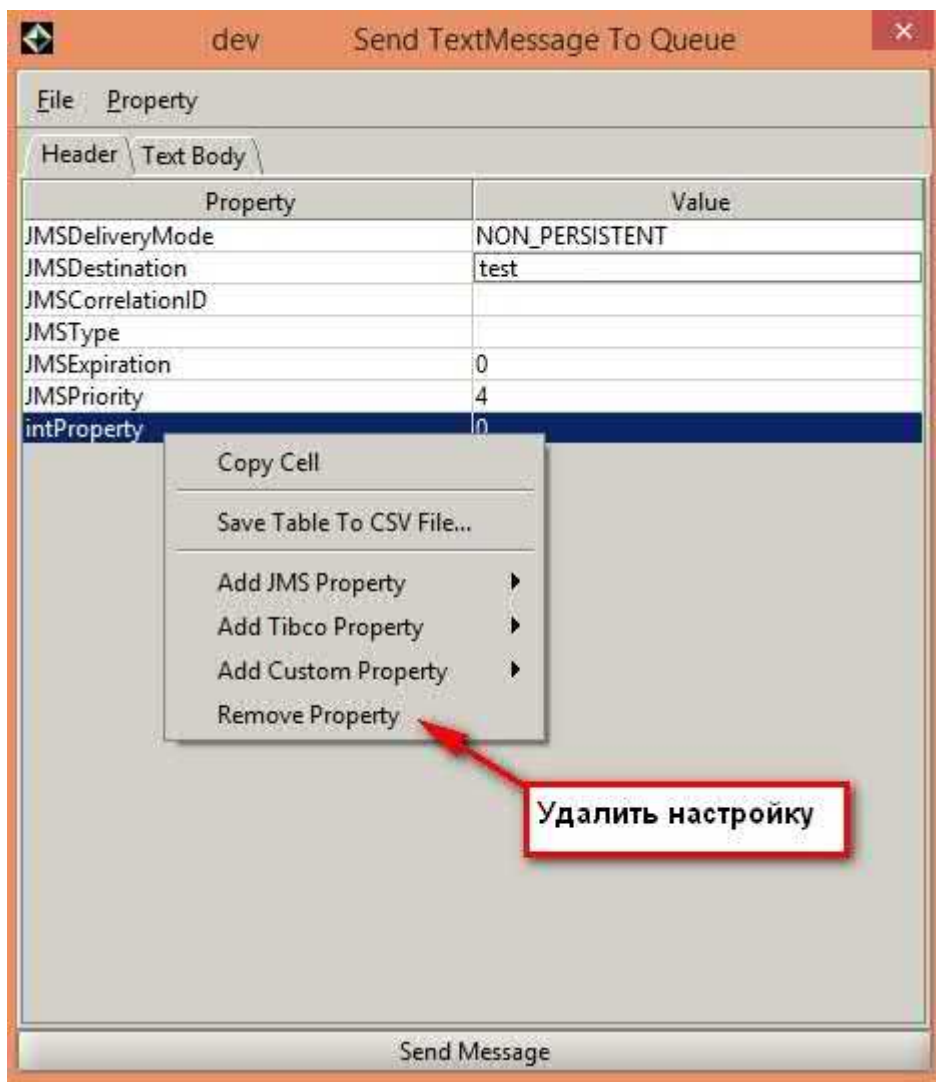
Если у нас не один читатель очереди, а 10, их надо как-то разделять. Для этого во входящее сообщение добавляется новый параметр в заголовке. Допустим, это будет ID с типом Integer.

Как проверить, что можно послать в приложение сообщение с ID в заголовке?

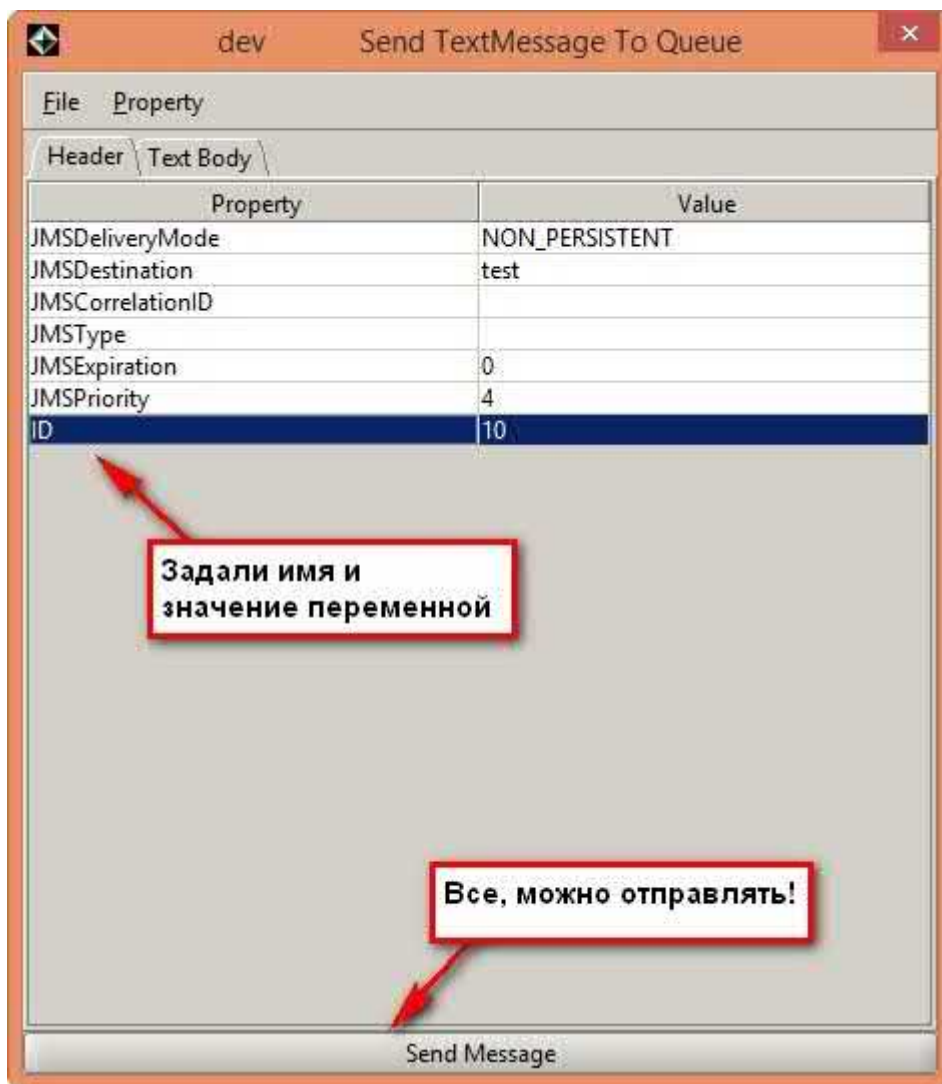
1. **Queues — Send TextMessage**
2. **Property — Add Custom Property — <выбираем тип>**



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!

**⚠ Обратите внимание:**

В заголовке важно соблюдать не только нейминг свойства, но и его тип.

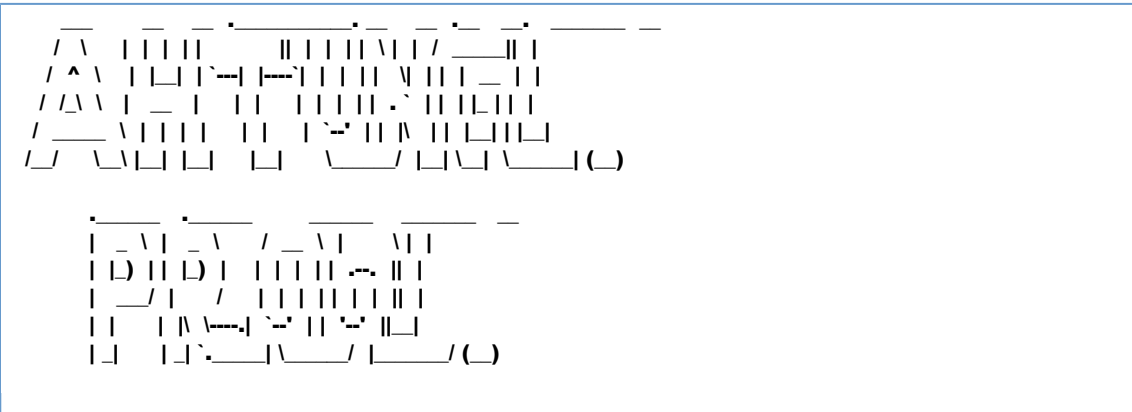
Например, в Открытии распределение по слушателям осуществляется в зависимости от значения поля clientID в заголовке. Для корректной вычитки сообщений из очереди и распределения запросов между потоками нужно указывать тип именно Integer.

# 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.

## Настройка

Указать приветствие в файле `/etc/motd`



Пример файла — `motd`

# КАК ПО ID В ИСХОДНОЙ СИСТЕМЕ ПОНЯТЬ С КЕМ СЛИТ КОНТРАГЕНТ

Если заказчику интересно межсистемное дублирование контрагентов, то можно отдать ему результат такого запроса:

```
select s.author, s.id_record, pp.merge_author  
-- в staging проиндексированы поля с идентификаторами  
from staging s  
-- из staging к ФЛ  
join physical_party pps on s.hid_staging = pps.hid_staging and s.version = 0 and pps.version  
= 0 and pps.enddate is null  
-- золотой хид  
left join merged mr on pps.hid_party = mr.hid_party and mr.canceled is null and mr.  
transitional = 0  
-- хиды исходных карточек в том же золотом хиде  
left join merged ms on mr.hid_final_scion = ms.hid_final_scion and ms.canceled is null and  
ms.transitional = 0  
-- от хида к идентификатору  
left join physical_party pp on ms.hid_party = pp.hid_party and pp.version = 0 and pp.enddate  
is null  
-- фильтрация по интересующим исходным идентификаторам  
where (s.author,s.id_record) in (select 'AL','1' from dual);
```



## 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

# ФАЗЫ МИГРАЦИИ НА \_HIST ТАБЛИЦЫ

*В: Как проходит миграция? Зачем она нужна? Что означают все эти фазы?*

- Зачем нужна миграция
- Как проходит миграция
  - Фаза 1. Создание таблиц \_ACT
  - Фаза 2. Заполнение таблиц \_ACT
  - Фаза 3. Переезд на новые таблицы
  - Фаза 4. Удаление дублированных данных

## 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:

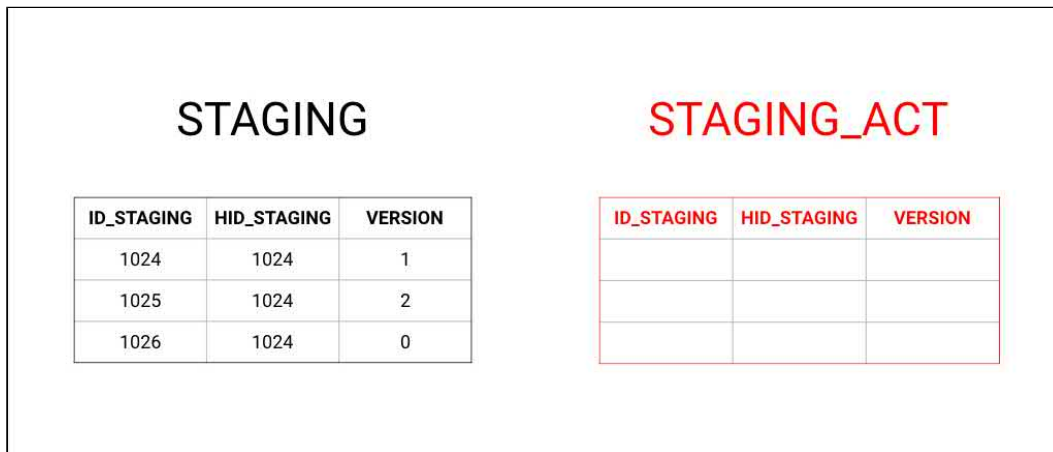
1. Ускорить задачи:
  - a. Перестроение индекса — не нужно будет делать отсев прошлых версий в селекте
  - b. Перестроение мат. view — чем меньше данных в таблице, тем быстрее работает select
2. Купить под актуальные данные в БД SSD-диск, а старые хранить на обычном. SSD стоит дорого, покупать его под большую базу невыгодно.

Ускорение будет заметно только на больших базах, где уже накопилась история: записей с **version = 0** сильно больше, чем актуальных (**version = 0**).

## Как проходит миграция

Миграция проходит в 4 фазы, чтобы избежать длительного даунтайма сервиса.

## Фаза 1. Создание таблиц **\_ACT**



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:

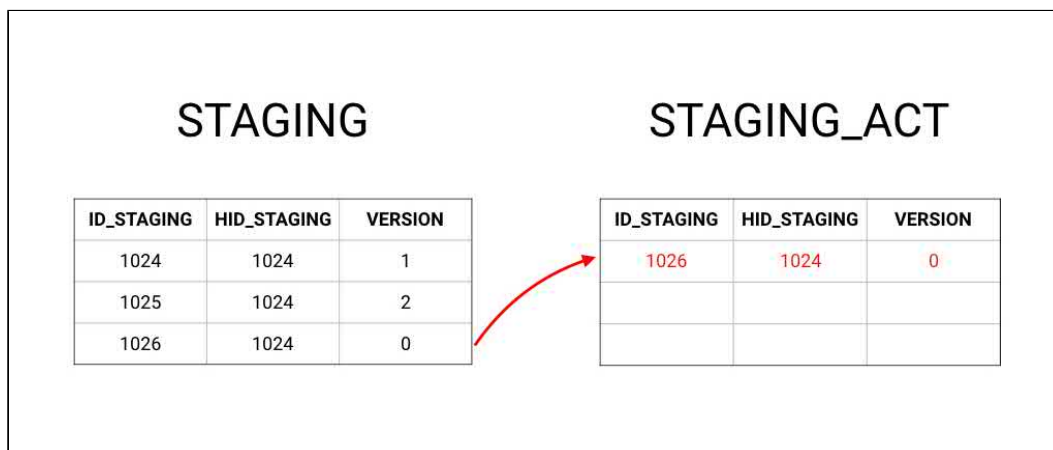
### STAGING\_ACT

ID_STAGING	HID_STAGING	VERSION
------------	-------------	---------

Операция повторяется для каждой версионной таблицы.

Фаза проходит быстро и автоматически. Сотрудники ХФЛабс добавляют скрипты автоматизации в поставку, вручную ничего делать не нужно.

## Фаза 2. Заполнение таблиц **\_ACT**



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

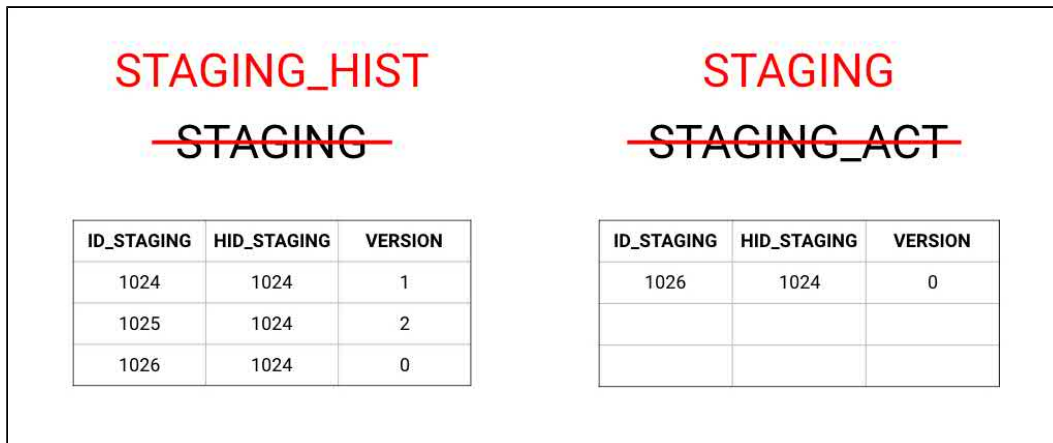
Пока это подготовительный этап, система все еще живет на старых таблицах.

Наполнение **\_ACT** 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.

### Фаза 3. Переезд на новые таблицы




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.

### 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 бэкап с базы

STAGING_HIST			STAGING		
ID_STAGING	HID_STAGING	VERSION	ID_STAGING	HID_STAGING	VERSION
1024	1024	1	1026	1024	0
1025	1024	2			
<del>1026</del>	<del>1024</del>	<del>0</del>			

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.

# ЧТО ДЕЛАТЬ, ЕСЛИ ИНКРЕМЕНТ УПАЛ С ОШИБКОЙ

## ***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.

## ***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

# FAQ ДЛЯ СОПРОВОЖДЕНЦЕВ

1. Нужно добавить задачу / триггер прямо в war-файл хотфиксом, как?

WAR-ник ? WEB-INF ? classes



## ФАКТОР (FAQ)

- Есть ли doBatchClean для телефонов?

# 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/>
        <dataFields/>
        <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>
    </ns2:doBatchCleanResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</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 "МераФон"</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>
```

## ИНТЕГРАЦИЯ (FAQ)

- Как передать двух клиентов с одной связью

# 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>
```

## ОБНОВЛЕНИЕ СПРАВОЧНИКОВ (FAQ)

- Я обновил перечень ПНП на Linux, но он не подхватывается

# 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:

```
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
```

# КАК ДОБАВИТЬ ИЛИ ИЗМЕНИТЬ НОВЫЙ ТИП АТТРИБУТА

- Задать новый тип согласия через базу данных
- Создать новый тип атрибута



# ЗАДАТЬ НОВЫЙ ТИП СОГЛАСИЯ ЧЕРЕЗ БАЗУ ДАННЫХ

В БД CDI в таблице **REFERENCE** перечислены все типы согласий, которые доступны.

Для добавления нового типа согласия необходимо выполнить следующие шаги:

1. Добавить новый тип согласия в таблицу **REFERENCE**.

a. Посмотреть последний номер ординала:

```
select max(ordinal) from reference where ref_type='CONSENT_TYPE';
```

b. Добавить следующий с номером (N+1)

**Пример запроса для добавления согласия на контакты по Push-уведомлениям:**

```
INSERT INTO reference (created, author, ref_type, ordinal, code, short_label, label)  
VALUES (SYSDATE, 'HFLabs-CDI', 'CONSENT_TYPE', 6, 'PUSH_CONTACT', 'Контакт Push',  
'Контакт по Push');
```

```
commit;
```

2. Если для Согласия необходимо установить умолчательное значение отличное от **UNKNO WN**, необходимо дополнительно указать это значение в таблице **CONSENT\_DEFAULT\_VALUE**.

**Пример запроса для добавления умолчательного значения**

**COMMON\_REQUIRED\_TO\_OBTAIN** («Требуется получить») для типа согласия «Контакт по Push»:

```
INSERT INTO CONSENT_DEFAULT_VALUE (CONSENT_TYPE, DEFAULT_VALUE)  
VALUES ('PUSH_CONTACT', 'COMMON_REQUIRED_TO_OBTAIN');
```

```
commit;
```

3. Запустить задачу **clearCache** в панели администратора.

4. Написать в поддержку HFLabs для поддержки нового Согласия в коде и приложить скрипты, которые использовали.

# СОЗДАТЬ НОВЫЙ ТИП АТТРИБУТА

Для создания нового типа необходимо добавить новый тип в ЕК и Фактор.

## 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.

При добавлении налету на стенд заказчика для значения ordinal рекомендуется использовать коды 500+.

Проверить, какие коды есть для телефонов можно:

```
select * from reference_view
where ref_type = 'PHONE_TYPE';
-- интересуется колонка ordinal
-- значения: от 0 до 7
```

## 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.

## 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

# ОБНОВЛЕНИЕ СПРАВОЧНИКОВ

# ЗАГРУЗКА И ОБНОВЛЕНИЕ ЕГРЮЛ

# ПЕРВИЧНАЯ ЗАГРУЗКА СПРАВОЧНИКОВ

## Первичная загрузка справочников ЕГРЮЛ и ЕГРИП в Единый Клиент

! Для хранения справочника ЕГРЮЛ и проверки базы клиентов по нему требуется дополнительно **100Gb** свободного места на диске, где располагаются директории CDI\_ROOT и CDI\_DEDUP и **50Gb** на диске с БД

1. В корневой директории файл-сервера находится файл, переадресовывающий на хранилище справочников для ЕГР или хранилище справочников для нормализованного ЕГР. В хранилище выбрать самую свежую сборку и загрузить из неё файл с расширением **.dic**. (без **-inc**) и чек-сумму (\***.sha**).

2. В интерфейсе администратора CDI отредактировать параметры запуска задачи **EgrDictionaryImport**:

- указать путь к директории со справочником ЕГР в параметре **sourceFileName**.
- установить **massiveChanging = true**
- Убрать параметр **essenceType**, если он присутствует.

Создать/Редактировать задачу

Группа import  
Название EgrDictionaryImport

Параметры задачи

Название параметра	Значение
massiveChanging	true
sourceFileName	/opt/cdi2/dicts
sourceDataReadBlockSize	1000
sourceDataEncoding	UTF-8
poolSize	2

Сохранить Отмена

3. Запустить задачу **EgrDictionaryImport**.

4. Отредактировать задачу, установить параметр **massiveChanging = false**

*Параметр используется только для первичной загрузки справочника. Если не вернуть ему значение false, обновления справочника не попадут в Фактор. Поиск дубликатов между ЮЛ и ЕГРЮЛ будет работать по первичной выгрузке*

# ОБНОВЛЕНИЕ СПРАВОЧНИКОВ ЕГРЮЛ

 При загрузке инкрементального ЕГРЮЛ необходимо ежедневное обновление справочника

## Загрузка справочников ЕГРЮЛ и ЕГРИП в «Единый клиент»

1. В корневой директории файл-сервера находится файл, переадресовывающий на [хранилище справочников для ЕГР](#) или [хранилище справочников для нормализованного и инкрементального ЕГР](#). В хранилище выбрать самую свежую сборку и загрузить из неё файл с расширением **.dic**.
2. В интерфейсе администратора CDI отредактировать параметры запуска задачи **EgrDictionaryImport** — указать путь к директории со справочником ЕГР в параметре **sourceFileName**.

### Создать/Редактировать задачу

Группа import  
Название EgrDictionaryImport

Параметры задачи

Название параметра	Значение
<input type="text" value="massiveChanging"/>	<input type="text" value="true"/>
<input type="text" value="sourceFileName"/>	<input type="text" value="/opt/cdi2/dicts"/>
<input type="text" value="sourceDataReadBlockSize"/>	<input type="text" value="1000"/>
<input type="text" value="sourceDataEncoding"/>	<input type="text" value="UTF-8"/>
<input type="text" value="poolSize"/>	<input type="text" value="2"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

3. Запустить задачу **EgrDictionaryImport**.

# ОБНОВЛЕНИЕ СПРАВОЧНИКОВ ПОДСКАЗОК



# СПРАВОЧНИК EMAIL

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **email-ггггммдд.zip**. Поместите скачанный файл на сервере в каталог

```
/opt/suggestions/dictionaries/email
```

2. [для версии 22.7 или раньше] Распакуйте архив.
3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{ "action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["email"] }'
```

4. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process email from '/opt/suggestions/dictionaries/email' (06.11.2018 00:00:00)
```

5. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing email completed at 10 s  
[INFO] (IndexManager) New email index (disk directory mode) from '/opt/suggestions/indexes/index_email_****' (06.11.2018 00:00:00) loaded
```

Время обновления справочника — меньше минуты.

# СПРАВОЧНИК IP-АДРЕСОВ

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **ip-ггггммдд.zip**. Поместите скачанный файл на сервере в каталог

```
/opt/suggestions/dictionaries/ip
```

2. [для версии 22.7 или раньше] Распакуйте архив.
3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{ "action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["ip"] }'
```

4. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process ip_blocks from '/opt/suggestions/dictionaries/ip' (05.09.2018 00:00:00)
```

5. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing ip_blocks completed at 1 s  
[INFO] (IndexManager) New ip_blocks index (disk directory mode) from '/opt/suggestions/indexes/index_ip_****' (05.09.2018 00:00:00) loaded
```

Время обновления справочника — меньше минуты.

## СПРАВОЧНИК ФИАС (20.2–20.8)

⚠ Используйте справочник ФИАС только в том случае, если *твердо уверены*, что вам нужен именно он. Для большинства задач намного лучше подходят [подсказки по адресам](#) — они более полные, логичные и удобные. Подсказки по ФИАС нужны, только если вам по какой-то причине важно точное, побуквенное соответствие справочнику адресов налоговой службы.

Если слова «ФИАС» и «планировочная структура» вам ничего не говорят — пожалуйста, не используйте эти подсказки. Используйте [подсказки по адресам](#).

Сервис ФИАС требует специальную лицензию, по умолчанию она выключена. Чтобы включить, напишите в техподдержку.

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **fias-ггггммдд.zip**
2. Распакуйте архив на сервере в каталог:

```
/opt/suggestions/dictionaries/address
```

3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["fias"]}'
```

4. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process fias from '/opt/suggestions/dictionaries/fias' (23.12.2018 00:00:00)
```

5. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing fias completed at 2.058 h  
[INFO] (IndexManager) New fias index (disk directory mode) from '/opt/suggestions /indexes/index_fias_***' (23.12.2018 00:00:00) loaded
```

Время обновления справочника — до 3 часов.

# СПРАВОЧНИК ФИАС (20.9+)

⚠ Используйте справочник ФИАС только в том случае, если *твердо уверены*, что вам нужен именно он. Для большинства задач намного лучше подходят [подсказки по адресам](#) — они более полные, логичные и удобные. Подсказки по ФИАС нужны, только если вам по какой-то причине важно точное, побуквенное соответствие справочнику адресов налоговой службы.

Если слова «ФИАС» и «планировочная структура» вам ничего не говорят — пожалуйста, не используйте эти подсказки. Используйте [подсказки по адресам](#).

Сервис ФИАС требует специальную лицензию, по умолчанию она выключена. Чтобы включить, напишите в техподдержку.

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **fias-ггггммдд.zip**
2. Поместите архив на сервере в каталог:

```
/opt/suggestions/dictionaries/fias
```

3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["fias"]}'
```

4. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process fias from '/opt/suggestions/dictionaries/fias' (23.12.2018 00:00:00)
```

5. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing fias completed at 2.058 h  
[INFO] (IndexManager) New fias index (disk directory mode) from '/opt/suggestions /indexes/index_fias_***' (23.12.2018 00:00:00) loaded
```

Время обновления справочника — до 3 часов.

## СПРАВОЧНИК АДРЕСОВ (20.2–20.8)

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **fias-ггггммдд.zip**
2. Распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address
```

Лучше всего распаковать не весь архив, а только те DBF-файлы, которые требуются «Подсказкам». Пример команды **unzip**:

```
unzip -o fias-20200525.zip ADDRDB*.DBF HOUSE*.DBF SOCRBASE.DBF -d /opt/suggestions/dictionaries/address
```

3. Если нужны иностранные адреса: загрузите справочник **geonames-ггггммдд.zip** из [репозитория HFLabs](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address/geonames
```

4. Если нужны ISO-коды (в подсказках версии 19.7 и выше) для стран и регионов: загрузите справочник **iso3166-ггггммдд.zip** из [репозитория HFLabs](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address/iso3166
```

5. Если нужны дополнительные дома, которых нет в ФИАС: загрузите справочник **house-ггггммдд.zip** из [репозитория HFLabs](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address/house
```

6. (в подсказках версии 20.4 и выше) Если используете поиск по географическим координатам: загрузите справочник **geo-ггггммдд.zip** из [репозитория HFLabs](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address/geo
```

7. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{ "action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["address"] }'
```

8. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process address from '/opt/suggestions/dictionaries/address' (23.12.2018 00:00:00)
```


9. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing address completed at 2.058 h  
[INFO] (IndexManager) New address index (disk directory mode) from '/opt/suggestions  
/indexes/index_address_***' (23.12.2018 00:00:00) loaded
```

Время обновления справочника — до 3 часов.

# СПРАВОЧНИК АДРЕСОВ (20.9+)

- Основной справочник адресов (Россия)
- Справочник квартир (flat)
- Справочник земельных участков (stead)
- Справочник GeoNames (иностранные адреса кроме Беларуси, Узбекистана и Казахстана)
- Справочник OSM (Беларусь, Узбекистан и Казахстан)

 Этот справочник поддерживает только административное деление.

## Основной справочник адресов (Россия)

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **fias-ггггммдд.zip**
2. Поместите архив на сервере в каталог

```
/opt/suggestions/dictionaries/address
```

3. Если нужны ISO-коды для стран и регионов: загрузите справочник **iso3166-ггггммдд.zip** из [репозитория HFLabs](#) на сервере в каталог

```
/opt/suggestions/dictionaries/address
```

распакуйте архив в каталог:

```
/opt/suggestions/dictionaries/address/iso3166
```

4. Если нужны дополнительные дома, которых нет в ФИАС: загрузите справочник **house-ггггммдд.zip** из [репозитория HFLabs](#) на сервере в каталог

```
/opt/suggestions/dictionaries/address
```

распакуйте архив в каталог:

```
/opt/suggestions/dictionaries/address/house
```

5. Если нужны географические координаты: загрузите справочник **geo-ггггммдд.zip** из [репозитория HFLabs](#) на сервере в каталог

```
/opt/suggestions/dictionaries/address
```

распакуйте архив в каталог:

```
/opt/suggestions/dictionaries/address/geo
```

6. Запустите обновление справочника адресов:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["address"]}'
```

7. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process address from '/opt/suggestions/dictionaries/address' (23.12.2018 00:00:00)
```

8. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing address completed at 2.058 h  
[INFO] (IndexManager) New address index (disk directory mode) from '/opt/suggestions/indexes/index_address_***' (23.12.2018 00:00:00) loaded
```

Время обновления справочника — до 3 часов.

## Справочник квартир (flat)

Чтобы обновить квартиры, скачайте основной справочник адресов (как описано выше), после чего выполните команду из консоли:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["flat"]}'
```

В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process flat from '/data/dictionaries/address/fias-20211210.zip' (10.12.2021 00:00:00)
```

Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing flat completed at 7.455 min  
[INFO] (IndexManager) New flat index (disk directory mode) from '/data/indexes/index_flat_***' (10.12.2021 00:00:00) loaded
```

Время обновления справочника — около 10 минут.

## Справочник земельных участков (stead)

Чтобы обновить земельные участки, скачайте основной справочник адресов (как описано выше), после чего выполните команду из консоли:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["stead"]}'
```

В журнале suggestions.log появится запись вида:



```
[INFO] (IndexManager) Process stead from '/data/dictionaries/address/fias-20211210.zip'
(10.12.2021 00:00:00)
```

Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing stead completed at 3.997 min
[INFO] (IndexManager) New stead index (disk directory mode) from '/data/indexes
/index_stead_**** (10.12.2021 00:00:00) loaded
```

Время обновления справочника — около 5 минут.

## ***Справочник GeoNames (иностранные адреса кроме Беларуси, Узбекистана и Казахстана)***

1. Загрузите справочник **geonames-ггггммдд.zip** из репозитория [HFLabs](#) на сервере в каталог

```
/opt/suggestions/dictionaries/geonames
```

Если используете подсказки версии 22.7 или раньше, распакуйте архив.

2. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-
Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names":
["geonames"]}'
```

3. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process address from '/opt/suggestions/dictionaries/geonames'
(23.12.2018 00:00:00)
```

4. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing geonames completed at 1.390 min
[INFO] (IndexManager) New geonames index (disk directory mode) from '/opt
/suggestions/indexes/index_geonames_**** (23.12.2018 00:00:00) loaded
```

Время обновления справочника — около 1 минуты.

## ***Справочник OSM (Беларусь, Узбекистан и Казахстан)***

1. Загрузите справочник **osm-ггггммдд.zip** из репозитория [HFLabs](#) на сервере в каталог

```
/opt/suggestions/dictionaries/osm
```

Если используете подсказки версии 22.7 или раньше, распакуйте архив.

2. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{ "action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["osm"] }'
```

3. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process address from '/opt/suggestions/dictionaries/osm' (23.12.2018 00:00:00)
```

4. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing osm completed at 1.390 min  
[INFO] (IndexManager) New osm index (disk directory mode) from '/opt/suggestions/indexes/index_osm_***' (23.10.2020 12:08:10) loaded
```

Время обновления справочника — около 5 минут.

# СПРАВОЧНИК БАНКОВ

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **ed807-ггггммдд.zip**. Поместите скачанный файл на сервере в каталог

```
/opt/suggestions/dictionaries/bank
```

2. [для версии 22.7 или раньше] Распакуйте архив.
3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{ "action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["bank"] }'
```

4. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process bank from '/opt/suggestions/dictionaries/bank' (03.12.2018 00:00:00)
```

5. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing bank completed at 2.099 min  
[INFO] (IndexManager) New bank index (disk directory mode) from '/opt/suggestions/indexes/index_bank_***' (03.12.2018 00:00:00) loaded
```

Время обновления справочника — до 5 минут.

# ОБНОВЛЕНИЕ ВНЕШНИХ СПРАВОЧНИКОВ

**i** Требуется лицензия на внешние справочники. Лицензия позволяет подключить любое количество справочников — как готовых из перечисленных ниже, так и собственных справочников заказчика.

Инструкция одна и та же для всех внешних справочников. Отличается только код справочника.

Поддерживаются следующие справочники:

Код	Описание	Версия «Подсказок»
car_brand	Марки автомобилей	20.2+
country	Страны	19.1+
country_history	Страны с историческими названиями	20.3+
currency	Валюты	19.1+
fms_unit	Кем выдан паспорт	19.1+
fns_unit	Налоговые инспекции	19.1+
fts_unit	Таможенные органы	20.11+
metro	Станции метро	20.2+
mktu	Товары и услуги (МКТУ)	21.5+
okpd2	Виды продукции (ОКПД 2)	19.1+
oktmo	Муниципальные образования	20.11+
okved2	Виды деятельности (ОКВЭД 2)	19.1+
postal_office	Почтовые отделения (старый)	19.1+
postal_unit	Почтовые отделения (новый) С координатами, часами работы и признаком «временно закрыто».	19.11+
region_court	Мировые суды	19.1+

## Как подключить справочник

Все команды выполняются на сервере «Подсказок» под пользователем factor.

Установить код справочника (подставьте вместо \*\*\* подходящий код из таблицы выше):

```
DICTIONARY=***
```

Создать каталоги:

```
mkdir -p /opt/suggestions/dictionaries/$DICTIONARY/  
mkdir -p /opt/suggestions/configuration/outward
```

Скачать справочник:

- из репозитория [HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **\$DICTIONARY-ггггммдд.zip**;
- или из корпоративного репозитория, если это корпоративный справочник.

Положить конфигурацию и сам справочник в нужные места:

```
unzip -o $DICTIONARY-*.zip $DICTIONARY.yaml -d /opt/suggestions/configuration/outward/  
mv $DICTIONARY-*.zip /opt/suggestions/dictionaries/$DICTIONARY/
```

```
unzip $DICTIONARY-*.zip  
mv $DICTIONARY-*.csv /opt/suggestions/dictionaries/$DICTIONARY/  
mv $DICTIONARY.yaml /opt/suggestions/configuration/outward/
```

Загрузить справочник в подсказки:

```
curl -X POST \  
  http://localhost:8080/suggestions/api/4_1/rs/manage/outward \  
  -H 'Content-Type: application/json' \  
  -d '{ "action": "RELOAD_SAFE", "names":["$DICTIONARY"] }'
```

После этого он должен появиться в общем списке ([http://localhost:8080/suggestions/api/4\\_1/rs/status](http://localhost:8080/suggestions/api/4_1/rs/status))

# СПРАВОЧНИК ИМЕН

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида `fiо-ггггммдд.zip`. Поместите скачанный файл на сервере в каталог

```
/opt/suggestions/dictionaries/fio
```

2. [для версии 22.7 или раньше] Распакуйте архив.
3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{ "action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["fio"] }'
```

4. В журнале `suggestions.log` появится запись вида:

```
[INFO] (IndexManager) Process fio from '/opt/suggestions/dictionaries/fio' (06.11.2018 00:00:00)
```

5. Дождитесь, пока в журнале `suggestions.log` появится запись вида:

```
[INFO] (IndexManager) Processing fio completed at 25 s  
[INFO] (IndexManager) New fio index (disk directory mode) from '/opt/suggestions/indexes/index_fio_***' (06.11.2018 00:00:00) loaded
```

Время обновления справочника — меньше минуты.

# СПРАВОЧНИК КОМПАНИЙ

1. Скачайте [основной справочник](#) из репозитория HFLabs. Выберите каталог с наиболее свежей датой, в нём файл вида **egr-sgt-gggгммдд-bson.dic**. Поместите скачанный файл на сервере в каталог

```
/opt/suggestions/dictionaries/party
```

2. Скачайте [вспомогательный справочник](#) из репозитория HFLabs. Выберите каталог с наиболее свежей датой, в нём файл вида **party\_ext-gggгммдд.zip**. Поместите скачанный файл на сервере в каталог

```
/opt/suggestions/dictionaries/party
```

3. [для версии 22.7 или раньше] Распакуйте вспомогательный справочник в каталог

```
/opt/suggestions/dictionaries/party/party_ext
```

4. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["party"]}'
```

5. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process egr from '/opt/suggestions/dictionaries/party/egr-sgt-20180823-bson.dic' (23.08.2018 00:00:00)
```

6. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing egr completed at 1.046 h  
[INFO] (IndexManager) New egr index (disk directory mode) from '/opt/suggestions/indexes/index_party_***' (23.08.2018 00:00:00) loaded
```

Время обновления справочника — до 3 часов.

# СПРАВОЧНИК КООРДИНАТ

**i** Начиная с версии подсказок 20.4 обновляется вместе совместно со [справочниками адресов](#)

Для «Подсказок» до версии 20.4

1. Скачайте справочник из [репозитория HFLabs](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида **geo-ггггммдд.zip**
2. Распакуйте архив на сервере в **/opt/suggestions/dictionaries/geo**
3. Запустите обновление справочника:

```
curl -X POST 'http://localhost:8080/suggestions/api/4_1/rs/manage/resource' -H 'Content-Type: application/json' -d '{"action": "RELOAD_SAFE", "type": "DICTIONARY", "names": ["geo"]}'
```

4. В журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Process geo from '/opt/suggestions/dictionaries/geo' (30.01.2019 00:00:00)
```

5. Дождитесь, пока в журнале suggestions.log появится запись вида:

```
[INFO] (IndexManager) Processing geo completed at 50.16 s  
[INFO] (IndexManager) New geo index (disk directory mode) from '/opt/suggestions/indexes/index_geo_***' (30.01.2019 00:00:00) loaded
```

Время обновления справочника — около минуты.

Для «Подсказок» 20.4+ координаты обновляются вместе со [справочниками адресов](#)



# ОБНОВЛЕНИЕ СПРАВОЧНИКОВ ФАКТОРА

# БАЗА ДАННЫХ ПЕРЕНЕСЕННЫХ НОМЕРОВ

1. Перейдите в [репозиторий HFLabs](#) с новыми справочниками.
2. Вы увидите несколько директорий вида **ГГГГММДДЧММ\_NNNN**:

Index of repo/ru/zniis/bdpn/Port_All_New		
Name	Last Modified	Size
<a href="#">202307260000_3535/</a>	27-07-23 08:01:51 +0300	
<a href="#">202307270000_3536/</a>	28-07-23 08:01:20 +0300	
<a href="#">202307280000_3537/</a>	28-07-23 10:01:02 +0300	
<a href="#">202308010000_3541/</a>	02-08-23 08:30:10 +0300	
<a href="#">202308030000_3543/</a>	04-08-23 18:13:05 +0300	
<a href="#">202308070000_3547/</a>	08-08-23 08:01:26 +0300	
<a href="#">202308080000_3548/</a>	08-08-23 12:01:38 +0300	
<a href="#">202308090000_3549/</a>	10-08-23 08:01:47 +0300	
<a href="#">202308100000_3550/</a>	11-08-23 08:01:40 +0300	
<a href="#">202308110000_3551/</a>	12-08-23 08:01:40 +0300	

Первые 8 цифр — это дата в формате **ГГГГММДД**.

Выберите каталог с наиболее свежей датой.

3. Скачайте справочник и его контрольную сумму:
  - Port\_All\_New-ГГГГММДДЧММ\_NNNN.zip — справочник;
  - Port\_All\_New-ГГГГММДДЧММ\_NNNN.zip.sha1 — его контрольная сумма.
4. Поместите справочник и его контрольную сумму на сервер с «Фактором» в каталог **/opt/factor/dictionaries/**
5. Через [API горячих обновлений](#) запустите обновление справочника **changeMobileProvider**:

```
curl -X POST \  
-H ""Content-Type": application/json" \  
-H ""Accept": application/json" \  
-d '{"reloadDictionaryRequest":{"dictionary":{"name":"changeMobileProvider"}}}' \  
http://СЕРВЕР:ПОРТ/factor-service-<customer>/api/services/rs/ManagementDictionaryService/reloadDictionary
```

В журнале factor.log появится запись вида:

```
[INFO] (ResourceChangeListener) Processing '/opt/factor/dictionaries/Port_All_New-202309220000_3593.zip' resource
```

6. Дождитесь, пока в журнале factor.log появится запись вида:

```
[INFO] (ResourceChangeListener) Process '/opt/factor/dictionaries/Port_All_New-202309220000_3593.zip' resource took '22.45 s'
```

# ОБНОВЛЕНИЕ ДОМЕНОВ ПЕРВОГО УРОВНЯ

**i** Горячее обновление — обновление без остановки «Фактора». Доступно для «Фактора» 19.8 и выше.

- Горячее обновление доменов первого уровня
- Как запустить обновление
- Что происходит после завершения обновления
  - Признаки успешного обновления
- Время обновления
- Проверка версии справочников

## Как запустить обновление

Скопировать новый справочник вида *tlds-alpha-by-domain-ГГГГММДД.txt* и его контрольную сумму (один из файлов с его названием и расширением sha1/sha256) из [хранилища](#) на сервер с Фактором.

При необходимости справочник и его контрольную сумму можно расположить в пользовательской директории. Для этого нужно задать переменную `FACTOR_EMAIL_TLDS_PATH` и указать путь.

`FACTOR_EMAIL_TLDS_PATH` можно задать как переменную окружению или как параметр в *stand alone.conf*:

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_EMAIL_TLDS_PATH=/data/dict/email/"
```

Если переменная не задана, то используется общая директория для обновления справочников (настраивается через переменную `FACTOR_DICTIONARIES_PATH`). Если она тоже не задана, то используется директория по умолчанию "**`$(factor.userLibPath)/email/resources/user`**".

**1. После завершения копирования** справочника и его контрольной суммы в той же директории создать файл *<название справочника>.ready*.

Например, если справочник называется *tlds-alpha-by-domain-2019092100.txt*, то необходимо создать файл *tlds-alpha-by-domain-2019092100.txt.ready*.

После этого начнётся загрузка нового справочника.

«Фактор» проверяет новый справочник на корректность.

- Если справочник корректный, то «Фактор»:
  - a. распаковывает справочник в локальную директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
  - b. переключается на работу с новым справочником.

- Если справочник некорректный, то Фактор продолжит работать со старым справочником.

Если нужно обновить справочник на более старый, создайте файл с названием справочника и расширением **.force** (.ready создавать не нужно). Будет выполнено обновление вне зависимости от даты прежней версии справочника.

## Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

### Признаки успешного обновления

1. Файл <название справочника>.ready «Фактор» переименует в файл со статусом SUCCESS, названием справочника, ip-адресом, датой и временем обновления в названии. Дата и время выводятся в формате ууууммдд\_ххммсс.

Например,

*tlds-alpha-by-domain-2019080600.txt\_10.0.63.16\_20190807\_161926.success*

2. В логе Фактора *factor.log* появятся строки:

```
07.08.2019/16:19:25 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (ResourceChangeListener) Processing 'file:///Z:/jboss/wildfly-16.0.0.Final/user_lib
/email/resources/user/tlds-alpha-by-domain-2019080600.txt' resource
07.08.2019/16:19:25 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (LuceneIndexBuilder) Build emailFirstLevelDomains index from 'CryptByteSource
[PathByteSource[Z:/jboss/wildfly-16.0.0.Final/user_lib/email/resources/user/tlds-alpha-by-
domain-2019080600.txt]].asCharSource(UTF-8)' in 12 threads
07.08.2019/16:19:26 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (LuceneIndexBuilder) -- optimize emailFirstLevelDomains index
07.08.2019/16:19:26 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (LuceneIndexBuilder) Indexing 1527 entities of emailFirstLevelDomains took
598,1 ms
07.08.2019/16:19:26 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (LuceneIndexManager) emailFirstLevelDomains resource from 'file:///Z:/jboss
/wildfly-16.0.0.Final/standalone/data/index_emailFirstLevelDomains_47f22729-f83e-489b-
a413-75209eb7b815/' (06.08.2019 00:00:00) loaded
07.08.2019/16:19:26 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (ResourceHolder) Obsolete emailFirstLevelDomains resource from 'Z:
\jboss\wildfly-16.0.0.Final\standalone\data\index_emailFirstLevelDomains_48b50545-
22db-491b-a8c6-f18a5f61f082' removed in 3,453 ms
07.08.2019/16:19:26 127.0.0.1#anonymous (ResourceReload-emailFirstLevelDomains-1)
[INFO] (ResourceChangeListener) Process 'file:///Z:/jboss/wildfly-16.0.0.Final/user_lib
/email/resources/user/tlds-alpha-by-domain-2019080600.txt' resource took '690,2 ms'
```

## Время обновления

Зависит от:

- скорости распаковки нового справочника в локальную директорию;
- скорости проверки нового справочника.

Время обоих пунктов зависит от технических особенностей сервера.

Переключение на работу с новым справочником происходит моментально.

## **Проверка версии справочников**

Проверить версию справочников можно по [инструкции](#).

# ОБНОВЛЕНИЕ ЕГРЮЛ И ЕГРИП

## Средства обновления:

Доступно из коробки	Необходима лицензия
через ready-файлы	через апдейтер
через API	

## Обновление через ready-файлы

### Предусловие:

«Фактор» запущен.

### Шаги для запуска обновления:

1. Скачайте из хранилища [NFLabs](#) новый справочник и его контрольную сумму (один из файлов с его названием и расширением sha1/sha256) в remote-директорию, не меняя название.
2. Создать в этой же директории пустой файл с названием файла нового справочника и расширением **.ready** (например, `egr-15082016.v2.mv.db.ready`).

#### Совместимость ЕГРЮЛ и ЕГРИП с «Фактором»

С версии «Фактора» 22.2 справочники ЕГРЮЛ и ЕГРИП выпускаем в новом формате. Причина: обновили драйвер h2 до версии 2.1.210. Поводом для обновления драйвера послужили уязвимости.

Новый формат не совместим с более ранними версиями «Фактора».

Для «Фактора» 22.2+ выбирайте справочник с классификатором v2 (пример наименования — `egrul2-20220216-v2.mv.db`).

Для «Фактора» 22.1 и ниже — без классификатора v2 (пример наименования — `egrul2-20220216.mv.db`).

**!** Добавлять файл **.ready** нужно обязательно после завершения копирования нового справочника и его контрольной суммы в remote-директорию. Иначе копирование в директорию временного хранения начнётся преждевременно и будет нарушена целостность справочника — обновление завершится ошибкой.

После запуска обновления «Фактор» проверяет новый справочник на корректность.

- Если справочник корректный, то «Фактор»:
  1. копирует справочник в директорию, заданную в переменной `FACTOR_EGR_CURRENT_PATH`;
  2. переключается на работу с новым справочником.

- Если справочник некорректный, то «Фактор» продолжит работать со старым справочником.

Если remote-директория сетевая (примонтирована как локальная), то перед проверкой нового справочника «Фактор» копирует его в директорию временного хранения;

После завершения обновления из директории временного хранения новый справочник удаляется.

После окончания обновления рекомендуем удалить файл .ready. Если не удалить файл .ready, то при подключении нового экземпляра «Фактора» он поочередно обновится на справочники, для которых есть .ready-файлы.

### Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

### Признаки успешного обновления

1. В remote-директории «Фактор» создаст пустой файл с названием справочника, IP сервера, статусом SUCCESS, датой и временем обновления в названии. Дата и время выводятся в формате ууууммдд\_ххммсс.

Например, `egrul2-20170723.v2.mv.db_192.168.0.156_20170725_155631.success`

Если работает 5 экземпляров «Фактора», то должно появиться 5 SUCCESS файлов.

2. В логе «Фактора» `factor.log` появятся строки:

```
[INFO] (ResourceChangeListener) Processing 'file:///data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/share/egrul2-20170723.v2.mv.db' resource
[INFO] (EgrulDataSourceManager) egrul2 resource from '/data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/current/egrul2-20170723.v2.mv.db' (23.07.2017 04:34:06) loaded
[INFO] (ResourceHolder) Obsolete egrul2 resource from '/data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/current/egrul2-20170724.v2.mv.db' removed in 361,0 ms
[INFO] (ResourceChangeListener) Process 'file:///data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/share/egrul2-20170723.v2.mv.db' resource took '1,286 min'
```

### Если что-то пошло не так

В remote-директории «Фактор» создаст пустой файл с названием справочника, IP сервера, датой и временем обновления, статусом FAILED или SKIPPED:

- SKIPPED — при попытке обновления на справочник с датой актуальности меньше/равной, чем у локального.

В логе «Фактора» `factor.log` появится сообщение:

```
[INFO] (ResourceChangeListener) Processing 'file:///Z:/wildfly-10.1.0.Final/user_lib/egrul2/share/egrul2-20181224.v2.mv.db' resource
[INFO] (ResourceChangeListener) Resource 'file:///Z:/wildfly-10.1.0.Final/user_lib/egrul2'
```

```
/share/egrul2-20181224.v2.mv.db' skipped. Cause by: egrul2 from 'Z:\wildfly-10.1.0.Final\user_lib\egrul2\share\egrul2-20181224.v2.mv.db' (24.12.2018 05:52:39) ignored because it older/equals than the current (24.12.2018 05:52:39)
```

- FAILED — в остальных случаях, например, справочник битый, контрольная сумма не соответствует справочнику.

В логе «Фактора» *factor.log* появится сообщение об ошибке:

```
[ERROR] (ResourceChangeListener) Can't process 'file:///data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/share/egrul2-20170725.v2.mv.db'  
ru.hflabs.cleaner.exception.FactorDBException: Datasource egrul2 by path '/data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/share/egrul2-20170725.v2.mv.db' corrupted.  
Cause by: Fail-fast during pool initialization  
at ru.hflabs.cleaner.egrul2.storage.H2EgrDirectory.detect(H2EgrDirectory.java:114
```

## Обновление через API

Доступно с версии «Фактора» 9.3+.

Описание в разделе [обновления через API](#).

## Обновление через Апдейтер

Описание в разделе [обновления через Апдейтер](#).



# ОБНОВЛЕНИЕ ФИАС

❗ Шаги необходимо выполнять в строгой последовательности!

Файл **.ready** создавать/копировать только после завершения копирования файла справочника и его контрольной суммы.

Для версии «Фактора» 19.10+ наличие контрольной суммы рядом со справочником обязательно.

1. Скачать ФИАС нужной версии и его контрольную сумму — один из файлов с его названием и расширением md5/sha1/sha256.

#### Пример ФИАС и его контрольной суммы

```
fias-19.10-20191014.zip  
fias-19.10-20191014.zip.md5
```

2. Скопировать скачанный справочник и его контрольную сумму в директорию *FACTOR\_FIAS\_SHARE\_PATH*.
3. Только после окончания копирования файла справочника и его контрольной суммы в директории для ФИАС создать пустой файл с названием файла с ФИАС и расширением **.ready**:

#### Пример имени файла .ready для файла с ФИАС «fias-19.4-20190328.zip»

```
fias-19.4-20190328.zip.ready
```

4. Дождаться окончания обновления (до 10 минут).

Если нужно обновить справочник на более старый, создайте файл с названием справочника и расширением **.force** (.ready создавать не нужно). Будет выполнено обновление вне зависимости от даты прежней версии справочника.

## Что происходит после завершения обновления

Адреса, которые начали обрабатываться до обновления ФИАС, обрабатываются по старой версии ФИАС. После успешного обновления — по новому ФИАС.

При успешном обновлении «Фактор»:

- помещает новый справочник и его контрольную сумму в директорию  $\${factor.userLibPath}/address2/resources/current$  либо в *FACTOR\_INDEXES\_PATH*, если нет другой переменной
- удаляет из *current* прежнюю версию.

Если обновление не удалось выполнить, «Фактор» продолжает работать со старым справочником.

## Как понять, завершено ли обновление ФИАС

В логе «Фактора» *factor.log* должны появиться записи вида:

«Фактор» 19.4+

### Шаблон сообщения в factor.log

```
[INFO] (AddressClassifierManager) fias resource from 'file:///Z:/factor_indexes/fias-19.4-20190328.zip' (28.03.2019 00:00:00) loaded
[INFO] (ResourceHolder) Obsolete fias resource from 'Z:/factor_indexes/fias-19.4-20190325.zip' removed in 166,3 ms
[INFO] (ResourceChangeListener) Process 'file:///Z:/wildfly-10.1.0.Final/user_lib/address2/resources/share/fias-19.4-20190328.zip' resource took '7,095 min'
```

«Фактор» до 19.3 включительно

### Шаблон сообщения в factor.log

```
[INFO] (DefaultAddressClassifierProvider) Загружен ФИАС [date]
```

## Как понять, если что-то пошло не так

В логе «Фактора» *factor.log* есть ошибки (*ERROR*).

Например:

«Фактор» 19.4+

- Версия нового ФИАС не соответствует версии «Фактора». Например, установлен «Фактор» 19.10, а новый ФИАС для «Фактора» 19.3 — *fias-19.3-20190408.zip*.

### Шаблон сообщения в factor.log

```
[ERROR] (ResourceReloadLockWatcher) fias by path 'file:///Z:/jboss/wildfly-16.0.0.Final/user_lib/address2/resources/share/fias-19.3-20190408.zip' corrupted. Cause by: Serialized FIAS structure corrupted
```

- Контрольная сумма не соответствует скачанному ФИАС.

```
[ERROR] (ResourceReloadLockWatcher) Wrong SHA1 checksum for 'file:///Z:/jboss/wildfly-16.0.0.Final/user_lib/address2/resources/share/fias-19.6-20190620.zip' (actual: '3f7766710c907e6d25c97f0f758ee7a46eb099f2', expected: '69bc8b891eeb06a3155dfe50d4db9e10c3137cb7')
```

«Фактор» до 19.3 включительно

### Шаблон сообщения в factor.log

```
04.10.2017/15:55:12 (Camel (switchAddressClassifierContext) thread #2 - file:///data/Data_doc/wildfly-9.0.2.Final/user_lib/address2/resources/update/) [INFO] (ClassifierLoaderProcessor) Deserialize addresses from '/data/Data_doc/wildfly-9.0.2.Final/user_lib/address2/resources'
```

```
'/update/fias-8.3-20170921.zip'  
04.10.2017/15:55:12 (Camel (switchAddressClassifierContext) thread #2 - file:///data/Data_doc  
/wildfly-9.0.2.Final/user_lib/address2/resources/update/) [ERROR] (DefaultErrorHandler)  
Failed delivery for (MessageId: ID-valentina-desktop-38455-1507116478426-0-1 on  
ExchangeId: ID-valentina-desktop-38455-1507116478426-0-6). Exhausted after delivery  
attempt: 1 caught: java.lang.RuntimeException: ru.hflabs.cleaner.exception.  
FactorConfigurationException: FIAS file corrupted/has wrong version
```

В этом случае версия нового ФИАС не соответствует версии «Фактора». Например, установлен «Фактор» 8.7, а новый ФИАС для «Фактора» 8.3 — *fias-8.3-20170921.zip*.

Версию ФИАС проверяет «Фактор» версии 8.6 и выше.

# ПЕРЕЧЕНЬ НЕДЕЙСТВИТЕЛЬНЫХ ПАСПОРТОВ (ПНП)

**i** Если серия или номер в справочнике пустая, то при построении индексов такие строки игнорируем, т.к. считаем серию и номер обязательными атрибутами.

## Для одного «Фактора»

1. Скачать из хранилища [HFLabs](#) новый справочник `passports-expired-ГГГГММДД.csv.bz2` и его контрольную сумму (один из файлов с его названием и расширением `sha1/sha256`).
2. Положить скачанный справочник и его контрольную сумму в директорию для горячего обновления.
3. Обновить справочник в «Факторе»:

### *С остановкой «Фактора»*

Остановить и запустить «Фактор».  
Из нескольких ПНП «Фактор» выберет наиболее новый по дате модификации.

### *Без остановки «Фактора»*

Создайте в этой же директории (из п.1) пустой файл с названием файла с ПНП и расширением `.ready`.

Например: для `passports-expired-20160725.csv.bz2` создать пустой `passports-expired-20160725.csv.bz2.ready`.

Если нужно обновить справочник на более старый, создайте файл с названием файла с ПНП и расширением `.force` (`.ready` создавать не нужно). Будет выполнено обновление вне зависимости от даты и существования прежней версии справочника.

По завершении обновления справочника «Фактор» переименует файл `ready/force` в файл со статусом `SUCCESS`, названием справочника, датой и временем обновления в названии. Дата и время выводятся в формате `ууууммдд_ххммсс`.

Если «Фактор» на Linux-системе игнорирует `ready` файл, возможно, у него нет прав на перечень. Проверьте, что файлы подложены от имени пользователя «Фактора».

Паспорта, которые начали обрабатываться до обновления ПНП, заканчивают обрабатываться на старом ПНП. После успешного завершения обновления паспорта начинают обрабатываться с использованием нового ПНП.

## Для нескольких «Факторов» без остановки

Обратитесь к сотруднику HFLabs для настройки обновления на нескольких «Факторах».

1. Положить скачанный справочник и его контрольную сумму в в директорию для горячего обновления.

Это должен быть локальный относительно «Факторов» ресурс. Директория в сети должна быть примонтирована к серверу «Фактора» средствами ОС.

2. Создайте в этой же директории (из п.1) пустой файл с названием файла с ПНП и расширением **.ready**.

Например: для справочника *passports-expired-20160725.csv.bz2* создать пустой *passports-expired-20160725.csv.bz2.ready*.

По завершении обновления «Фактор» в директории со справочником создаст пустой файл со статусом SUCCESS, IP машины с «Фактором», датой и временем обновления в названии. Дата и время выводятся в формате уууу-мм-дд-hh-мм-ss.

Если работает 5 экземпляров «Фактора», то должно появиться 5 SUCCESS файлов.

Паспорта, которые начали обрабатываться до обновления ПНП, заканчивают обрабатываться на старом ПНП. После успешного завершения обновления паспорта начинают обрабатываться с использованием нового ПНП.

## Время подключения (обновления)

Для работы со справочником «Фактор» формирует lucene-индекс. Формирует 10-30 минут, в зависимости от мощности сервера. В это время «Фактор» работает со старым индексом.

## Сколько занимают рабочие файлы

Файлы с индексом занимают около 1 Гб.

## Где находятся рабочие файлы

По умолчанию «Фактор» формирует индекс в директории *factor\_jboss\_home/standalone/data*.

Чтобы сменить директорию, в *standalone.conf.bat* (Windows) / *standalone.conf* (Linux) перед запуском «Фактора» задать параметр `FACTOR_INDEXES_PATH`, например:

```
set "JAVA_OPTS=%JAVA_OPTS% -DFACTOR_INDEXES_PATH=D:\invalidpassport\indexes"
```

Путь можно указать в переменной окружения `FACTOR_INDEXES_PATH`. В этом случае параметр в конфигурационном файле указывать не нужно.

## Как понять, что подключение завершено

В логе «Фактора» *factor.log* должны появиться записи вида:

```
Шаблон сообщения в factor.log
08.11.2019/11:52:52 127.0.0.1#anonymous (LuceneIndexer-invalidPassports-3) [INFO]
(LuceneIndexBuilder) -- processed 124000000 invalidPassports. Average speed: 223021 rec
/sec
08.11.2019/11:52:54 127.0.0.1#anonymous (ResourceReload-invalidPassports-1) [INFO]
(LuceneIndexBuilder) -- optimize invalidPassports index
08.11.2019/11:57:53 127.0.0.1#anonymous (ResourceReload-invalidPassports-1) [INFO]
(LuceneIndexBuilder) Indexing 124090652 entities of invalidPassports took 14,29 min
08.11.2019/11:57:53 127.0.0.1#anonymous (ResourceReload-invalidPassports-1) [INFO]
(LuceneIndexManager) invalidPassports resource from 'file:///c:/wildfly-16.0.0.Final-18080-
factor/standalone/data/index_invalidPassports_231bfb31-6ee4-4573-b6ae-936da7449e05/'
(07.11.2019 00:00:00) loaded
08.11.2019/11:57:53 127.0.0.1#anonymous (ResourceReload-invalidPassports-1) [INFO]
(ResourceChangeListener) Process 'c:\wildfly-16.0.0.Final-18080-
factor\user_lib\passport\resources\user\passports-expired-20191107.csv.bz2' resource took
'14,30 min'
```

## Ошибка подключения

Если файл с ПНП некорректный или отсутствует контрольная сумма, то «Фактор» продолжит работать с прежним справочником.

«Фактор» переименует файл **ready/force** в файл с названием справочника, IP сервера, датой и временем обновления в названии и статусом FAILED или SKIPPED:

- SKIPPED — при попытке обновления на более старый справочник;
- FAILED — в остальных случаях, например, справочник битый, контрольная сумма не соответствует справочнику.

# ПОДКЛЮЧЕНИЕ ФАКТОРА К ПОДСКАЗКАМ

## Настройка подключения Фактора к подсказкам

Необходимо убедиться, что в **suggestions/bin/standalone.conf** Подсказок добавлена ссылка на сервис стандартизации (Фактор) и указаны корректные значения **HOST** и **PORT**. При отсутствии записи ее нужно добавить:

```
# Параметр для обогащения Подсказок через Фактор
JAVA_OPTS="$JAVA_OPTS -Denrich.url=http://HOST:PORT/factor-service-<customer>"
```

# СПРАВОЧНИК ГЕОКООРДИНАТ

**i** Работает, если выполняются условия:

- версия «Фактора» 20.02+
- ОС — любая из Linux-систем, см. [Требования к платформе для «Фактора»](#)
- директория для новых справочников — **/opt/factor/dictionaries**

**i** Шаги нужно выполнять из-под пользователя factor:  
**sudo -u factor <команда>**

1. Узнайте версию Фактора. Например, 23.7.
2. Перейдите в [репозиторий HFLabs](#) с новыми справочниками.
3. Найдите директории вида `index_fiasIdGeoCodes-<версия Фактора>` — они содержат справочник геокоординат для указанной версии «Фактора».



Зайдите в директорию для нужной версии «Фактора». Например, `index_fiasIdGeoCodes-23.7`.

4. Вы увидите несколько директорий с датой формата **ГГГГММДД** в наименовании. Выбирайте каталог с наиболее свежей датой.
5. Скачайте справочник и его контрольную сумму:
  - **index\_fiasIdGeoCodes-<версия Фактора>-ГГГГММДД.idx** — справочник;
  - **index\_fiasIdGeoCodes-<версия Фактора>-ГГГГММДД.idx.md5** — его контрольная сумма.
6. Поместите справочник и его контрольную сумму в каталог **/opt/factor/dictionaries/**
7. Через [API горячих обновлений](#) запустите обновление справочника **fiasIdGeoCodes**:

```
curl -X POST \  
-H ""Content-Type": application/json" \  
-H ""Accept": application/json" \  
-d '{"reloadDictionaryRequest":{"dictionary":{"name":"fiasIdGeoCodes"}}}' \  
http://СЕРВЕР:ПОРТ/factor-service-<customer>/api/services/rs/ManagementDictionaryService/reloadDictionary
```



---

В журнале factor.log появится запись вида:

```
[INFO] (ResourceChangeListener) Processing '/opt/factor/dictionaries/index_fiasIdGeoCodes-23.7-20230801.idx' resource
```

8. Дождитесь, пока в журнале factor.log появится запись вида:

```
[INFO] (ResourceChangeListener) Process '/opt/factor/dictionaries/index_fiasIdGeoCodes-23.7-20230801.idx' resource took '1.745 min'
```

# СПРАВОЧНИК ЛОКАЛЬНЫХ ДОМЕНОВ

**i** Горячее обновление — обновление без остановки «Фактора». Доступно для Фактора 19.12 и выше.

- Как запустить обновление
- Что происходит после завершения обновления
  - Признаки успешного обновления
- Время обновления
- Проверка версии справочников

## Как запустить обновление

Создать файл/справочник локальных доменов (одна строка — одно полное доменное имя) вида `local-domains.txt` на сервере с «Фактором».

При необходимости справочник можно расположить в пользовательской директории. Для этого нужно задать переменную `FACTOR_EMAIL_LOCAL_PATH` и указать путь.

`FACTOR_EMAIL_LOCAL_PATH` можно задать как переменную окружению или как параметр в `standalone.conf`:

### Пример указания пути в `standalone.conf`

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_EMAIL_LOCAL_PATH=/data/dict/email"
```

Если переменная не задана, то используется директория (по снижению приоритета):

`FACTOR_DICTIONARIES_PATH`

Директория по умолчанию `${factor.userLibPath}/email/resources/user`

Если переменная не задана, то используется общая директория для обновления справочников (настраивается через переменную `FACTOR_DICTIONARIES_PATH`). Если она тоже не задана, то используется директория по умолчанию "**`${factor.userLibPath}/email/resources/user`**".

**После завершения копирования** справочника в той же директории создать файл `local-domains.txt.ready`

После этого начнётся загрузка нового справочника, «Фактор»:

1. распаковывает справочник в локальную директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
2. переключается на работу с новым справочником.

Если нужно обновить справочник на более старый, создать файл с названием справочника и расширением **.force** (.ready создавать не нужно). Будет выполнено обновление вне зависимости от даты прежней версии справочника.

## Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

### Признаки успешного обновления

1. Файл <название справочника>.ready Фактор переименует в файл со статусом SUCCESS, названием справочника, ip-адресом, датой и временем обновления в названии. Дата и время выводятся в формате ууууммдд\_ххммсс.  
Например,  
`local-domains.txt_10.0.63.16_20191227_203927.success`
2. В логе Фактора `factor.log` появятся строки:

```
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (ResourceChangeListener) Processing 'Z:\jboss\wildfly-16.0.0.
Finaluser_lib\email\resources\user\local-domains.txt' resource
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (LuceneIndexBuilder) Build emailLocalDomains index from 'CryptByteSource
[PathByteSource[Z:\jboss\wildfly-16.0.0.Finaluser_lib\email\resources\user\local-
domains.txt]].asCharSource(UTF-8)' in 12 threads
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (LuceneIndexBuilder) -- optimize emailLocalDomains index
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (LuceneIndexBuilder) Indexing 5 entities of emailLocalDomains took 129.2 ms
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (LuceneIndexManager) emailLocalDomains resource from 'file:///Z:\jboss\wildfly-
16.0.0.Final\standalone\data\index_emailLocalDomains_648fbc9-b1d5-496b-bbd4-
df6456c74b4a/' (29.12.2019 20:38:39) loaded
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (ResourceHolder) Obsolete emailLocalDomains resource from 'Z:\jboss\wildfly-
16.0.0.Final\standalone\data\index_emailLocalDomains_866dd2cb-5880-4b71-b1d0-
17f5cda2c3d9' removed in 3.227 ms
27.12.2019/20:39:27 127.0.0.1#anonymous (ResourceReload-emailLocalDomains-2)
[INFO] (ResourceChangeListener) Process 'Z:\jboss\wildfly-16.0.0.
Finaluser_lib\email\resources\user\local-domains.txt' resource took '161.5 ms'
```

## Время обновления

Зависит от:

- скорости распаковки нового справочника в локальную директорию;
- скорости проверки нового справочника.

Время обоих пунктов зависит от технических особенностей сервера.

Переключение на работу с новым справочником происходит моментально.

## **Проверка версии справочников**

Проверить версию справочников можно по [инструкции](#).

# СПРАВОЧНИК ОДНОРАЗОВЫХ E-MAIL

**i** Горячее обновление — обновление без остановки «Фактора». Доступно для «Фактора» 19.4 и выше.

## Как запустить обновление

Скачать из хранилища [HFLabs](#) новый справочник вида *email-delivery-ГГГГММДД.txt* и его контрольную сумму (один из файлов с его названием и расширением *sha1/sha256*) на сервер с «Фактором».

При необходимости справочник можно расположить в пользовательской директории. Для этого нужно задать переменную `FACTOR_EMAIL_DISPOSABLE_PATH` и указать путь.

`FACTOR_EMAIL_DISPOSABLE_PATH` можно задать как переменную окружению или как параметр в *standalone.conf*:

**Пример указания пути в *standalone.conf***

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_EMAIL_DISPOSABLE_PATH=/data/dict/email/"
```

Если переменная не задана, то используется общая директория для обновления справочников (настраивается через переменную `FACTOR_DICTIONARIES_PATH`). Если она тоже не задана, то используется директория по умолчанию "**`${factor.userLibPath}/email/resources/user`**".

**1. После завершения копирования** справочника и его контрольной суммы в той же директории создать файл *<название справочника>.ready*.

Например, если справочник называется *email-delivery-20190425.txt*, то необходимо создать файл *email-delivery-20190425.txt.ready*.

После этого начнётся загрузка нового справочника.

«Фактор» проверяет новый справочник на корректность.

- Если справочник корректный, то «Фактор»:
  - a. распаковывает справочник в локальную директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
  - b. переключается на работу с новым справочником.
- Если справочник некорректный, то «Фактор» продолжит работать со старым справочником.

## Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

## Признаки успешного обновления

1. Файл <название справочника>.ready «Фактор» переименует в файл со статусом SUCCESS, названием справочника, ip-адресом, датой и временем обновления в названии. Дата и время выводятся в формате ууууммдд\_ххммсс.  
Например,  
`email-delivery-20190425.txt_172.18.73.81_20190502_154120.success`
2. В логе «Фактора» `factor.log` появятся строки:

```
02.05.2019/15:41:19 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (ResourceChangeListener) Processing 'file:///C:/wildfly-8.2.1.Final/user_lib/email
/resources/user/email-delivery-20190425.txt' resource
02.05.2019/15:41:19 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (LuceneIndexBuilder) Build emailDisposableDomains index from
'CryptByteSource[PathByteSource[C:\wildfly-8.2.1.
Final\user_lib\email\resources\user\email-delivery-20190425.txt]].asCharSource(UTF-8)'
in 12 threads
02.05.2019/15:41:20 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (LuceneIndexBuilder) - optimize emailDisposableDomains index
02.05.2019/15:41:20 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (LuceneIndexBuilder) Indexing 19739 entities of emailDisposableDomains took
758,0 ms
02.05.2019/15:41:20 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (LuceneIndexManager) emailDisposableDomains resource from 'file:///C:/wildfly-
8.2.1.Final/standalone/data/index_emailDisposableDomains_5f4452b3-b181-4dc5-aa5d-
a259d77dbd4a/' (25.04.2019 00:00:00) loaded
02.05.2019/15:41:20 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (ResourceHolder) Obsolete emailDisposableDomains resource from 'C:\wildfly-
8.2.1.Final\standalone\data\index_emailDisposableDomains_c42d07dd-4940-441d-bea6-
e6d877679d69' removed in 2,918 ms
02.05.2019/15:41:20 127.0.0.1#anonymous (ResourceReload-emailDisposableDomains-1)
[INFO] (ResourceChangeListener) Process 'file:///C:/wildfly-8.2.1.Final/user_lib/email
/resources/user/email-delivery-20190425.txt' resource took '842,4 ms'
```

## Время обновления

Зависит от:

- скорости распаковки нового справочника в локальную директорию;
- скорости проверки нового справочника.

Время обоих пунктов зависит от технических особенностей сервера.

Переключение на работу с новым справочником происходит моментально.

## Проверка версии справочников


Проверить версию справочников можно по [инструкции](#).



# СПРАВОЧНИК ПЛОЩАДЕЙ КВАРТИР

## Как запустить обновление

1. Скачайте справочник и его контрольную сумму из [хранилища HFLabs](#). Выбирайте каталог с наиболее свежей датой, а в нём файлы вида:
  - **index\_flatArea-ГГГГММДД.idx** — справочник;
  - **index\_flatArea-ГГГГММДД.idx.md5** — его контрольная сумма.
2. Поместите справочник и его контрольную сумму в remote-директорию;
3. Создайте в этой же директории пустой файл с названием нового справочника и расширением **.ready** (например, `index_flatArea-20161121.idx.ready`).

 Добавлять файл **.ready** нужно обязательно после завершения копирования нового справочника и его контрольной суммы в remote-директорию. Иначе проверка нового справочника начнётся преждевременно — обновление завершится ошибкой.

После запуска обновления «Фактор» проверяет новый справочник на корректность.

- Если справочник корректный, то «Фактор»:
  1. распаковывает справочник в директорию, заданную в переменной **FACTOR\_INDEXES\_PATH**;
  2. проверяет на корректность распакованный справочник;
  3. переключается на работу с новым справочником.
- Если справочник некорректный, то «Фактор» продолжит работать со старым справочником.

Если нужно обновить справочник на более старый, создайте файл с названием справочника и расширением **.force** (.ready создавать не нужно). Будет выполнено обновление вне зависимости от даты и существования прежней версии справочника.

## Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника. Старый справочник удаляется.



## Признаки успешного обновления

1. В remote-директории «Фактор» переименует файл ready в файл со статусом SUCCESS, названием справочника, датой и временем обновления в названии. Дата и время выводятся в формате ууууммдд\_ххммсс.

Например,

index\_flatArea-20161122.idx\_20170110\_153157.success

Если работает 5 экземпляров «Фактора», то должно появиться 5 SUCCESS файлов.

2. В логе Фактора *factor.log* появятся строки:

```
Успешное обновление  
(ResourceChangeListener) Processing 'file:///Z:/Dictionaries/flatarea/index_flatArea-20161220.idx' resource  
(ResourceReload-flatArea-1) [INFO] (ResourceChangeListener) Resource 'file:///Z:/Dictionaries/flatarea/index_flatArea-20161220.idx' skipped. Cause by: flatArea from 'Z:\Dictionaries\flatarea\index_flatArea-20161220.idx' (20.12.2016 00:00:00) ignored because it older/equals than the current (12.04.2019 00:00:00)  
(ResourceReload-flatArea-2) [INFO] (ResourceChangeListener) Processing 'file:///Z:/Dictionaries/flatarea/index_flatArea-20161220.idx' resource  
(ResourceReload-flatArea-2) [INFO] (LuceneIndexAccessor) Unpack flatArea index from 'Z:\Dictionaries\flatarea\index_flatArea-20161220.idx'  
(ResourceReload-flatArea-2) [INFO] (LuceneIndexAccessor) Unpack flatArea index took 15,74 min  
(ResourceReload-flatArea-2) [INFO] (LuceneIndexManager) flatArea resource from 'file:///Z:/factor_indexes/index_flatArea_c169d538-2fb0-480f-8713-fef918593517' (20.12.2016 00:00:00) loaded  
(ResourceReload-flatArea-2) [INFO] (ResourceHolder) Obsolete flatArea resource from 'Z:\factor_indexes\index_flatArea_8a54d3e5-1188-40eb-911c-fdd8fcd6c41' removed in 28,28 ms  
(ResourceReload-flatArea-2) [INFO] (ResourceChangeListener) Process 'file:///Z:/Dictionaries/flatarea/index_flatArea-20161220.idx' resource took '15,75 min'
```

## Если что-то пошло не так

1. В remote-директории «Фактор» переименует файл ready в файл со статусом обновления FAILED, названием справочника, датой и временем обновления.

Например,

index\_flatArea-20161123.idx\_20170110\_130052.failed

2. В логе *factor.log* «Фактор» сообщит об ошибке:

```
Попытка обновления на более старый справочник  
(ResourceReload-flatArea-2) [ERROR] Can't process 'file://overmind/public/elizavetam/flatarea/index_flatArea-20160909.idx'  
ru.hflabs.cleaner.io.hotupdate.ResourceReloadException: flatArea from '/opt/wildfly-8.2.1.Final/user_lib/tmp/flatareaReloader/index_flatArea-20160909.idx' (09.09.2016 00:00:00) ignored because it older/equals than the current (20.12.2016 00:00:00)
```

**Попытка обновления на битый справочник**

**(ResourceReload-flatArea-2) [ERROR] Can't process 'file://overmind/public/elizavetam  
/flat\_area/index\_flatArea-20170701.idx'  
java.io.IOException: unexpected end of stream**

# СПРАВОЧНИК РОССВЯЗИ

## Средства обновления:

- через ready-файлы
- через API


## Обновление через ready-файлы

### *Предусловие:*

«Фактор» запущен.

### *Шаги для запуска обновления:*

1. Скачайте из хранилища HFLabs и его контрольную сумму (один из файлов с его названием и расширением md5/sha1/sha256) с нашего ресурса и поместите в FACTOR\_ROSSVYAZ\_PATH, не меняя название.
2. Создайте в этой же директории пустой файл с названием файла нового справочника и расширением **.ready** (например, *rossvyaz-20180417.zip.ready*).

 Добавлять файл **.ready** нужно обязательно после завершения копирования нового справочника и его контрольной суммы в remote-директорию. Иначе копирование в директорию временного хранения начнётся преждевременно и будет нарушена целостность справочника — обновление завершится ошибкой.

После запуска обновления «Фактор» проверяет новый справочник на корректность.

- Если справочник корректный, то «Фактор»:
  1. строит индексы в FACTOR\_INDEXES\_PATH;
  2. переключается на работу с новым справочником.
- Если справочник некорректный, то «Фактор» продолжит работать со старым справочником.

Если FACTOR\_ROSSVYAZ\_PATH сетевая (примонтирована как локальная), то перед проверкой нового справочника «Фактор» копирует его в директорию временного хранения — FACTOR\_TMP\_DIR;

После завершения обновления из директории временного хранения новый справочник удаляется.

После окончания обновления рекомендуем удалить файл **.ready**. Если не удалить файл **.ready**, то при подключении нового экземпляра «Фактора» он поочередно обновится на справочники, для которых есть **.ready**-файлы.

## Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

### Признаки успешного обновления

1. В `FACTOR_ROSSVYAZ_PATH` «Фактор» создаст пустой файл с названием справочника, IP сервера, статусом `SUCCESS`, датой и временем обновления в названии. Дата и время выводятся в формате `ууууммдд_ххммсс`.

Например, `rossvyaz-20180417.zip_192.168.0.156_20170725_155631.success`

2. В логе «Фактора» `factor.log` появятся строки:

```
[INFO] (ResourceChangeListener) Processing 'file:///C:/wildfly-10.1.0.Final/user_lib/phone/rossvyaz/rossvyaz-20180418.zip' resource  
[INFO] (ResourceChangeListener) Process 'file:///C:/wildfly-10.1.0.Final/user_lib/phone/rossvyaz/rossvyaz-20180418.zip' resource took '1,059 s'
```

### Если что-то пошло не так

В `FACTOR_ROSSVYAZ_PATH` «Фактор» создаст пустой файл с названием справочника, IP сервера, датой и временем обновления, статусом `FAILED` или `SKIPPED`:

- `SKIPPED` — при попытке обновления на справочник с датой актуальности меньше/равной, чем у локального.

В логе «Фактора» `factor.log` появится сообщение:

```
31.07.2019/16:04:33 127.0.0.1#anonymous (ResourceReload-rossvyaz-2) [INFO]  
(ResourceChangeListener) Processing 'file:///C:/wildfly-8.2.1.Final/user_lib/phone/rossvyaz/rossvyaz-20190726.zip' resource  
31.07.2019/16:04:33 127.0.0.1#anonymous (ResourceReload-rossvyaz-2) [INFO]  
(ResourceChangeListener) Resource 'file:///C:/wildfly-8.2.1.Final/user_lib/phone/rossvyaz/rossvyaz-20190726.zip' skipped. Cause by: rossvyaz from 'C:\wildfly-8.2.1.Final\user_lib\phone\rossvyaz\rossvyaz-20190726.zip' (26.07.2019 00:00:00) ignored because it older/equals than the current (26.07.2019 00:00:00)
```

- `FAILED` — в остальных случаях, например, справочник битый, контрольная сумма не соответствует справочнику.

В логе «Фактора» `factor.log` появится сообщение об ошибке

## Обновление через API

Доступно с версии «Фактора» 19.7+.

Описание в разделе [обновления через API](#).

# СПРАВОЧНИК СТОИМОСТИ КВАДРАТНОГО МЕТРА

- Описание
- Подключение справочника на основе поискового индекса
- Горячее обновление справочника стоимостей

## Описание

Возвращает стоимость квадратного метра в доме из справочника недвижимости.

- Возвращает пустое значение:
  - если нет данных о стоимости в справочнике;
  - дома нет в адресе. То есть если добавить среднюю стоимость квадратного метра по городу или посёлку – вернётся пустое значение.
- Если известна и стоимость квадратного метра, и **площадь**, то будет рассчитана **общая стоимость**.

Имя компонента: `HOUSE_PRICE_PER_METER`.

Поставляется отдельно от дистрибутива «Фактора» в виде поискового индекса.

## Горячее обновление справочника стоимостей

1. Скачать из хранилища [HFLabs](#) новый справочник (`index_realtyPrice-VERSION.idx`) и его контрольную сумму (один из файлов с его названием и расширением md5/sha1/sha256) на сервер с Фактором.

При необходимости справочник может быть расположен в пользовательской директории.

Для этого нужно задать переменную `FACTOR_REALTY_PRICE` и указать путь.

`FACTOR_REALTY_PRICE` может быть задана как переменная окружения или как параметр в `standalone.conf`:

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_REALTY_PRICE=/data/Data_doc/realty/
```

Если переменная не задана, то используется общая директория для обновления справочников (настраивается через переменную `FACTOR_DICTIONARIES_PATH`). Если она тоже не задана, то используется директория по умолчанию "**`factor.userLibPath`** /**`address2/resources/realty_price`**".

2. После завершения копирования нового справочника и его контрольной суммы в той же директории создать файл `<название справочника>.ready`.

Например, если справочник называется *index\_realtyPrice-20170607.idx*, то необходимо создать файл *index\_realtyPrice-20170607.idx.ready*.

После этого начнётся загрузка нового справочника.

По завершении файл *index\_realtyPrice-VERSION.idx.ready* будет переименован в *index\_realtyPrice-VERSION.idx.success* (либо в *index\_realtyPrice-VERSION.idx.failed*, если обновление завершилось с ошибками)

# ШПАРГАЛКА ПО НАСТРОЙКЕ АПДЕЙТЕРА

**i** Подробнее про утилиту обновления справочников и доступные опции можно [почитать по ссылке](#).

## Доступы между серверами

### Открыть необходимые порты

№	Источник запроса			Получатель запроса			Проток	Назначение и описание информационного потока (какой тип данных передается)
	Имя (DNS-имя)	IP-адрес (а)	Порт (ы)	Имя (DNS-имя)	IP-адрес (а)	Порт(ы)		
<b>1 Взаимодействие между различными сегментами безопасности</b>								
1.	Сервер с апдейтером	XXX.XXX.XXX	>= 10	Сервер с Подсказками	XXX.XXX.XXX	22 или 3389	TCP P	Для копирования справочников по ssh
1.	Сервер с апдейтером	XXX.XXX.XXX	>= 10	Сервер с Подсказками	XXX.XXX.XXX	8080 (или другой порт, если Подсказки развернуты не на умолчательном порту)	TCP	Вызов API для запуска обновления справочника
1.	Сервер с апдейтером	XXX.XXX.XXX	>= 10	Сервер с ЕК	XXX.XXX.XXX	22 или 3389	TCP P	Для копирования справочников по ssh
1.	Сервер с апдейтером	XXX.XXX.XXX	>= 10	Сервер с ЕК	XXX.XXX.XXX	18080, 8080	TCP	Вызов API для запуска обновления справочника
<b>Пункт 2 необходим только, если уже подключены или планируются подключаться в дальнейшем внешние справочники для Подсказок</b>								
2	Сервер с Подсказками	XXX.XXX.XXX	>= 10	Сервер с апдейтером	XXX.XXX.XXX	22 или 3389	TCP P	Для подключения директории по sshfs

### Беспарольная авторизация

Для работы утилиты создать пользователя updater на всех серверах, с которыми работает утилита:

1. На сервере с апдейтером.
2. На всех серверах с Подсказками

### 3. На всех серверах с ЕК

Пользователя создаем с паролем

Дополнительно между сервером с апдейтом и серверами с Подсказками, ЕК необходимо настроить **беспарольную авторизацию** (авторизацию по ключу).

Источник запроса	Получатель запроса
Сервер с апдейтером	Сервер с ЕК
Сервер с апдейтером	Сервер с Подсказками
Сервер с Подсказками	Сервер с апдейтером

## Пользователи и права

### *На сервере с утилитой*

#### 1. Создать директории для копирования справочников и выдать на них права

##### a. Подсказок:

```
sudo mkdir /opt/suggestions/  
sudo chown -R updater:updater /opt/suggestions/  
sudo chmod -R 777 /opt/suggestions/
```

##### b. Фактора

```
sudo mkdir /opt/factor/  
sudo chown -R updater:updater /opt/factor/  
sudo chmod -R 777 /opt/factor/
```

##### c. Единый клиент

```
sudo mkdir /opt/cdi/  
sudo chown -R updater:updater /opt/cdi/  
sudo chmod -R 777 /opt/cdi/
```

### *На сервере с "Единым клиентом"*

#### 1. Создать пользователя updater для запуска утилиты

#### 2. Выдать права всем пользователям для директории со справочниками factor (до директории user\_lib).

```
sudo chmod -R 777 /opt/factor/jboss/user_lib
```

#### 3. Создать директорию для справочника egrul и выдать на нее права (путь может быть задан другой)

```
sudo chmod -R 777 /opt/cdi/dictionaries/  
sudo mkdir /opt/cdi/dictionaries/
```



## На сервере с "Подсказками"

1. Создать пользователя updater для запуска утилиты
2. Выдать права всем пользователям для директории со справочниками Подсказок  
**sudo chmod -R 777 /opt/suggestions/dictionaries/**

## Выдать права sudo для команд пользователю updater

**Создать файл:** visudo -f /etc/sudoers.d/updater

И прописать в нём:

```
updater ALL=(ALL) NOPASSWD: /usr/bin/chown
updater ALL=(ALL) NOPASSWD: /usr/bin/touch
```

## Внешние справочники

Из-за особенности реализации внешних справочников для их обновления необходимо настроить соединение между директориями для справочников по SSHFS/ NFS

## На сервере с "Подсказками"

1. На сервере с Подсказками устанавливаем sshfs  
**sudo yum install sshfs**
2. Подготавливаем директории для внешних справочников.

Необходимо создать директории, указанные ниже и выдать на эти директории права пользователю, под которым запущены подсказки, и полные права 777

```
mkdir /opt/suggestions/dictionaries/car_brand
mkdir /opt/suggestions/dictionaries/country
mkdir /opt/suggestions/dictionaries/country_history
mkdir /opt/suggestions/dictionaries/currency
mkdir /opt/suggestions/dictionaries/fms_unit
mkdir /opt/suggestions/dictionaries/fns_unit
mkdir /opt/suggestions/dictionaries/fts_unit
mkdir /opt/suggestions/dictionaries/metro
mkdir /opt/suggestions/dictionaries/okpd2
mkdir /opt/suggestions/dictionaries/oktmo
mkdir /opt/suggestions/dictionaries/okved2
mkdir /opt/suggestions/dictionaries/postal_office
mkdir /opt/suggestions/dictionaries/postal_unit
mkdir /opt/suggestions/dictionaries/region_court
```

3. Прописываем в /etc/fstab директории, которые необходимо примонтировать с сервера с апдейтером.

В данном примере подключаются директории для пользователя updater и сервера 10.10.10.10 - сервер, где установлен updater.

Важно для авто-монтирования директорий (чтобы не надо было монтировать после каждого рестарта сервера вручную, указываем пользователя, для которого настроена беспарольная авторизация).

#### Добавляем в конец файла

```
updater@10.10.10.10:/opt/suggestions/dictionaries/car_brand /opt/suggestions
/dictionaries/car_brand fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,
defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/country /opt/suggestions/dictionaries
/country fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/country_history /opt/suggestions
/dictionaries/country_history fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,
defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/currency /opt/suggestions/dictionaries
/currency fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/fms_unit /opt/suggestions/dictionaries
/fms_unit fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/fns_unit /opt/suggestions/dictionaries
/fns_unit fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/fts_unit /opt/suggestions/dictionaries
/fts_unit fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/metro /opt/suggestions/dictionaries
/metro fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/okpd2 /opt/suggestions/dictionaries
/okpd2 fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/oktmo /opt/suggestions/dictionaries
/oktmo fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/okved2 /opt/suggestions/dictionaries
/okved2 fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/postal_office /opt/suggestions
/dictionaries/postal_office fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,
defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/postal_unit /opt/suggestions
/dictionaries/postal_unit fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,
defaults 0 0
updater@10.10.10.10:/opt/suggestions/dictionaries/region_court /opt/suggestions
/dictionaries/region_court fuse.sshfs IdentityFile=/home/.ssh/updater/rsa,allow_other,
defaults 0 0
```

4. Если не было настроено ранее, то настраиваем беспарольную авторизацию по ключу с сервера с Подсказками до сервера с апдейтером.
5. Запускаем автоматическое перемонтирование директории явно вручную.  
**sudo mount -a**

## Пример файлов для запуска утилиты

Ниже примеры файлов по основным этапам обновления справочников.

Файлы могут быть как сгруппированы в один, так и компоноваться под задачи нужного обновления. Приведены как примеры.

В файлах настроены все справочники доступные на текущий момент (ненужные можно просто исключить).

1. Загрузить все справочники на сервер с апдейтером.  
**download-dictionaries.sh**
2. Скопировать все справочники CDI на соответствующий сервер и запустить обновление справочников  
[test-cdi-update.sh](#)
3. Скопировать все справочники Factor на соответствующий сервер и запустить обновление справочников  
[test-factor-update.sh](#)
4. Скопировать все справочники Подсказок на соответствующий сервер и запустить обновление справочников  
[test-sugg-update.sh](#)

# БЕСПАРОЛЬНАЯ АВТОРИЗАЦИЯ МЕЖДУ СЕРВЕРАМИ

Если утилиту автоматического обновления справочников разворачивают на удаленном сервере необходимо дополнительно настроить правильно пользователей, права, доступы.

Данный доступы могут быть созданы как админами на стороне заказчика, а также могут быть настроены самостоятельно, если есть пользователь с правами sudo.

## ***Необходимо на всех серверах с приложениями куда будут подкладываться справочники:***

1. Создать пользователя updater. Пользователя создаем с паролем.

```
useradd -p <пароль> updater
```

2. Проверить, что на сервере установлен и запущен ssh (при необходимости установить и запустить)

```
sudo service sshd status
```

3. Открыт доступ по ssh до данного сервера при подключении с сервера с утилитой.

## ***На сервере с андейтемом***

- ✓ Если какие-либо шаги были выполнены ранее, то просто их пропустить.

1. Создать пользователя **updater**. Пользователя создаем с паролем.

```
useradd -p <пароль> updater
```

2. Проверить, что на сервере установлен и запущен ssh

```
sudo service sshd status
```

3. Авторизуемся под пользователем updater на сервере где развернута утилита и переходим к выполнению пунктов, указанных ниже.

4. Создаем файл с конфигурацией ssh для данного пользователя, чтобы не перекрывать общие настройки: **touch ~/.ssh/config**.

В файлике указываем как в инструкции updater следующее:

```
Host *
User updater
IdentityFile ~/.ssh/rsa
```

Где Host — сервера при подключении, к которым необходимо применять настройки указанные ниже;

User — пользователь, под которым будет выполняться подключение и авторизация на удаленном сервере;

IdentityFile — файл с ключом для беспарольной авторизации по rsa

5. Выдаем правильные права на файл `chmod 600 ~/.ssh/config`

6. Создаем ключ для этого пользователя `ssh-keygen -f ~/.ssh/rsa`.

При создании можно будет создать доп.пароль, но нам он не нужен, поэтому на оба вопроса про пароль просто нажимаем enter.

7. Выполняем команду `ssh-copy-id -i /home/updater/rsa updater@remote_host`, заменяя хост на нужный.

Для копирования может не хватить прав до файла, тогда делаем следующие шаги:

При первом подключении к серверу система может его не распознать, поэтому при вопросе нужно ввести yes. Затем пароль для подключения по ssh к серверу. Утилита подключится к удаленному серверу, а затем использует содержимое ключа `rsa.pub` для загрузки его на сервер в файл `~/.ssh/authorized_keys`.

На удаленном сервере в директории `/home/.ssh` будет создан файл `authorized_keys`.

⚠ Автоматическое копирование командой не всегда срабатывает нормально (видимо из-за особенностей сети и безопасности или если на сервере приложений запрещена авторизация с паролем). Легко можно проверить: ключ на удаленном сервере должен совпадать с тем что задан в `rsa.pub`. Если не совпал, тогда просто копируем содержимое файла `rsa.pub` и заменяем соответствующую строку с ключом в файле `authorized_keys` на скопированную.

**Дальше с сервера с утилитой до удаленного сервера будет выполнять аутентификацию с помощью этого ключа, без ввода пароля.**

Проверяем `ssh updater@remote_host`.

Если все сделано верно, то подключение будет выполнено без ввода пароля.

## ***На сервере с Подсказками***

Настраиваем все аналогичным образом как и с апдейтера. Авторизация нужна с сервера подсказок до сервера с утилитой.

## ***Что еще может быть полезно знать при настройке подключения***

1. Команда для вывода лога подключения **`ssh -v user@host`**
2. Запуск и останов ssh: **`sudo service stop | start | restart`**

### 3. Включение дэбага ssh.

Если хотим получить лог подключения с подключаемой стороны, то можно включить логирование.

Например так:

- **sudo service stop**
- **sudo /usr/sbin/sshd -d**

Будет включен дэбаг ssh.

Подключаемся по ssh с другого сервера и смотрим на этом лог. Лог выводится прямо в консоли.

4. При проблемах с подключением без пароля стоит проверить права на всех файлах и директориях. Права 777, в данном случае могут дать обратный результат и неудачные попытки авторизации, поэтому нужно важно выдать правильные права.

- На всех серверах: **chmod 700 /home/updater**
- На сервере с утилитой: **chmod 700 /home/updater/.ssh**
- В .ssh права на файликах **chmod 600 rsa | rsa.pub | config**
- Что в ssh разрешено подключение по rsa. Настройки зафиксированы в etc/ssh/sshd\_config. Необходимо в этом файле указать RSAAuthentication yes или в файле ~/.ssh/config. После изменения лучше перезапустить ssh.
- Также в etc/ssh/sshd\_config может быть разрешено подключение только определенным пользователям или группам (параметры AllowUsers и AllowGroups). Если так, нужно добавить пользователя updater в список AllowUsers или добавить группу updater в список AllowGroups или включить пользователя updater в одну из групп из списка AllowGroups.

# ИНСТРУКЦИИ

- Подсказки. Подключение почтовых оповещений (Linux)
- Инструкции по сбору и проверке информации в рамках подготовки к переходу на Spring Boot

# ПОДСКАЗКИ. ПОДКЛЮЧЕНИЕ ПОЧТОВЫХ ОПОВЕЩЕНИЙ (LINUX)

## Оповещение об ошибках по электронной почте

**i** Если подключение почтовых оповещений невозможно, напишите нам в поддержку

Если произошла ошибка в работе «Подсказок», они могут сообщать об этом по электронной почте. При первом возникновении ошибки сообщение отправляется сразу же, если ошибка повторяется, то в течение часа данные накапливаются и отправляется второе сообщение со всеми случаями возникновения этой ошибки.

Чтобы включить почтовые оповещения необходимо:

1. Открыть доступ с сервера с «Подсказками» до SMTP-сервера.
2. В директории **/opt/suggestions/configuration** создать файл **mail.env**.  
В данном файле [указать настройки](#), как в примере, предварительно заменив значения:

- a. **MAIL\_HOST, MAIL\_PORT, MAIL\_USER, MAIL\_PASSWORD** на корректные параметры SMTP-сервера.
- b. **MAIL\_LOGGING\_TO** — список адресатов, указываются через запятую. Обязательно укажите нашу группу рассылки, другие адреса по мере необходимости.
- c. **MAIL\_FROM** — Имя почтового ящика, с которого будет производиться рассылка, можно оставить как в примере без изменения
- d. **MAIL\_LOGGING\_ENABLED** указывается как в примере без изменений.

### /opt/suggestions/configuration/mail.env

```
export MAIL_LOGGING_ENABLED="true"
export MAIL_LOGGING_TO="cdi@hflabs.ru, test@mail.ru"
export MAIL_FROM="sug_logs@hflabs.ru"
export MAIL_HOST="smtp.mycompany.ru"
export MAIL_PORT="25"
export MAIL_USER="logs@mycompany.ru"
export MAIL_PASSWORD="123456"
```

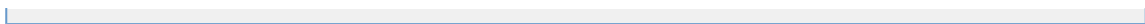
3. Добавить настройки отправки почтовых оповещений в файл **SUGG\_JBOSS/bin/standalone.conf**.

Строки без изменений добавляем в конец файла.

### SUGG\_JBOSS/bin/standalone.conf

```
source /opt/suggestions/configuration/mail.env
JAVA_OPTS="$JAVA_OPTS -Dmail.logging.enabled=$MAIL_LOGGING_ENABLED -Dmail.logging.to=$MAIL_LOGGING_TO"
JAVA_OPTS="$JAVA_OPTS -Dmail.from=$MAIL_FROM -Dmail.host=$MAIL_HOST -Dmail.port=$MAIL_PORT"
JAVA_OPTS="$JAVA_OPTS -Dmail.user=$MAIL_USER -Dmail.password=$MAIL_PASSWORD"
```





# ИНСТРУКЦИИ ПО СБОРУ И ПРОВЕРКЕ ИНФОРМАЦИИ В РАМКАХ ПОДГОТОВКИ К ПЕРЕХОДУ НА SPRING BOOT

- Linux (1 нода)
- Linux (2 ноды)
- Windows (1 нода)
- Windows (2 ноды)

# LINUX (1 НОДА)

1) Написать, по какому пути лежат приложения cdi и factor.

Примеры:

- /opt/cdi/jboss, opt/factor/jboss
- /opt/cdi/appserver, /opt/factor/appserver

2) Скопировать и прислать нам следующие фай?лы:

- /opt/cdi/.../standalone.conf
- /opt/factor/.../bin/standalone.conf
- /opt/cdi/.../standalone/configuration/standalone.xml
- /opt/factor/.../standalone/configuration/standalone.xml
- /opt/cdi/.../standalone/deployments/cdi-{oracle/mariadb/postgresql}-ds.xml

3) Скопировать и прислать фай?лы:

- /etc/systemd/system/cdi.service
- /etc/systemd/system/factor.service

Дополнительно проверить, какой? user указан в каждом из фай?лов: в командной? строке сервера ввести `id {user}`, выписать все группы, в которые входит каждый? user.

**i** Желательно разбить все фай?лы по папкам отдельно для CDI и для Фактора, и все папки сложить в один архив, а все пометки по выше указанным пунктам можно прислать в любом текстовом фай?ле.

# LINUX (2 НОДЫ)

1) Написать, по какому пути лежат приложения cdi и factor.

Примеры:

- /opt/cdi/jboss, opt/factor/jboss
- /opt/cdi/appserver, /opt/factor/appserver

2) Скопировать и прислать нам следующие фай?лы:

- /opt/cdi/.../standalone.conf
- /opt/factor/.../bin/standalone.conf
- /opt/cdi/.../standalone/configuration/standalone.xml
- /opt/factor/.../standalone/configuration/standalone.xml
- /opt/cdi/.../standalone/deployments/cdi-{oracle/mariadb/postgresql}-ds.xml

3) Скопировать и прислать фай?лы:

- /etc/systemd/system/cdi.service
- /etc/systemd/system/factor.service

Дополнительно проверить, какой? user указан в каждом из фай?лов.

В командной? строке сервера ввести id {user}, выписать все группы, в которые входит кажды? user.

4) Прислать результат выполнения запроса:

```
SELECT * FROM APPLICATION_NODE
```

Проделать те же самые шаги на второй? ноде.

**i** Желательно разбить все фай?лы по папкам с разных нод, отдельно для CDI и для Фактора, и все папки сложить в один архив.

Все пометки по выше указанным пунктам можно прислать в любом текстовом фай?ле.

# WINDOWS (1 НОДА)

1) Написать, по какому пути лежат приложения cdi и factor.

Примеры:

- /opt/cdi/jboss, opt/factor/jboss
- /opt/cdi/appserver, /opt/factor/appserver

2) Скопировать и прислать нам следующие фай?лы:

- /opt/cdi/.../standalone.conf.bat
- /opt/factor/.../bin/standalone.conf.bat
- /opt/cdi/.../standalone/configuration/standalone.xml
- /opt/factor/.../standalone/configuration/standalone.xml
- /opt/cdi/.../standalone/deployments/cdi-{oracle/mariadb/postgresql}-ds.xml



Желательно разбить все фай?лы по папкам отдельно для CDI и для Фактора, и все папки сложить в один архив, а все пометки по выше указанным пунктам можно прислать в любом текстовом фай?ле.

# WINDOWS (2 НОДЫ)

1) Написать, по какому пути лежат приложения cdi и factor.

Примеры:

- /opt/cdi/jboss, opt/factor/jboss
- /opt/cdi/appserver, /opt/factor/appserver

2) Скопировать и прислать нам следующие фай?лы:

- /opt/cdi/.../standalone.conf.bat
- /opt/factor/.../bin/standalone.conf.bat
- /opt/cdi/.../standalone/configuration/standalone.xml
- /opt/factor/.../standalone/configuration/standalone.xml
- /opt/cdi/.../standalone/deployments/cdi-{oracle/mariadb/postgresql}-ds.xml

3) Прислать результат выполнения запроса:

```
SELECT * FROM APPLICATION_NODE
```

Проделать те же самые шаги на второй? ноде.

**i** Желательно разбить все фай?лы по папкам с разных нод, отдельно для CDI и для Фактора, и все папки сложить в один архив.

Все пометки по выше указанным пунктам можно прислать в любом текстовом фай?ле.