

Руководство администратора по использованию программного обеспечения

Гражданский фактор

Содержание

| | | |
|--------|---|----|
| 1.1 | Требования..... | 7 |
| 1.2 | Обязанности | 7 |
| 1.3 | Доступы..... | 7 |
| 2.1 | Типовая схема развёртывания | 8 |
| 2.2 | Компоненты типовой схемы развертывания | 8 |
| 2.3 | Взаимосвязи типовой схемы развертывания | 9 |
| 2.4 | Отказоустойчивое решение (горячий резерв) | 9 |
| 2.4.1 | Работа в режиме горячего резерва | 10 |
| 2.4.2 | Обновление без прерывания обслуживания | 11 |
| 3.1 | Группы задач | 12 |
| 3.1.1 | Особенности..... | 12 |
| 3.1.2 | Результат выполнения | 13 |
| 3.1.3 | Идентификация и слияние дубликатов..... | 13 |
| 3.1.4 | Проверка по черным спискам | 14 |
| 3.1.5 | Массовые атрибуты | 14 |
| 3.1.6 | Миграция..... | 15 |
| 3.1.7 | Повторная обработка контрагентов | 15 |
| 3.1.8 | Подготовка списков для повторной обработки..... | 18 |
| 3.1.9 | Работа с задачами..... | 20 |
| 3.1.10 | Параметры, общие для всех типов задач | 20 |
| 3.2 | Планировщик задач | 20 |
| 3.2.1 | Запуск задачи по расписанию | 21 |
| 3.2.2 | Остановка выполнения задачи | 22 |
| 3.3 | Взаимодействие с JMS-очередями | 23 |
| 3.3.1 | Взаимодействие с JMS-очередями | 23 |
| 3.3.2 | Задача очистки хранилища JMS-сообщений..... | 23 |
| 3.3.3 | Задача переобработки сохраненных сообщений из входной очереди | 24 |
| 3.3.4 | Задача переправки сообщений в выходную очередь | 24 |
| 3.3.5 | Отправка новых JMS сообщений..... | 25 |
| 3.3.6 | Повторная отправка ошибочных JMS сообщений за последние 24 часа | 26 |
| 3.4 | Импорт данных..... | 26 |
| 3.4.1 | Особенности..... | 27 |
| 3.4.2 | Результат выполнения | 27 |
| 3.4.3 | Общие параметры для задач загрузки данных | 28 |
| 3.4.4 | Задача загрузки записей из буферной таблицы БД..... | 28 |
| 3.4.5 | Задача загрузки договоров из буферной таблицы БД..... | 29 |
| 3.4.6 | Задача загрузки связей договоров из буферной таблицы БД | 30 |
| 3.4.7 | Задача загрузки связей | 31 |
| 3.4.8 | Задача загрузки записей из CSV-файла | 31 |
| 3.4.9 | Задача загрузки записей из каталога | 32 |
| 3.4.10 | Задача загрузки записей по SFTP | 33 |
| 3.4.11 | Задача загрузки справочников ЕГРЮЛ / ЕГРИП | 34 |
| 3.4.12 | Трансформация данных | 35 |
| 3.4.13 | Активировать исходные записи | 37 |
| 3.4.14 | Деактивировать исходные записи | 38 |
| 3.4.15 | Задача циклической загрузки данных из БД..... | 39 |
| 3.5 | Идентификация и слияние дубликатов | 40 |
| 3.5.1 | Идентификация и слияние дубликатов | 40 |
| 3.5.2 | duplicateGroupActualize. Актуализация групп дубликатов..... | 41 |
| 3.5.3 | fullRefreshDuplicateGroupTrigger. Триггер для полного перестроения групп дубликатов ... | 41 |
| 3.5.4 | Закрытие дублирующих дубликатов | 42 |
| 3.5.5 | Инкрементальный поиск дубликатов..... | 43 |
| 3.5.6 | Кросс-сверка ФЛ-ЮЛ | 44 |
| 3.5.7 | Кросс-сверка ФЛ и проспектов ФЛ..... | 44 |
| 3.5.8 | Перестроение хэшей полного поиска дубликатов | 45 |
| 3.5.9 | Конфигурация задачи | 45 |
| 3.5.10 | Слияние дубликатов по внешнему списку | 46 |
| 3.6 | Повторная обработка | 47 |
| 3.6.1 | Повторная обработка контрагентов | 47 |
| 3.6.2 | Задача перерасчета hidden-флагов атрибутов по списку записей..... | 50 |
| 3.6.3 | Задача полного переслияния записей | 52 |
| 3.6.4 | Задача переслияния реквизитов контрагентов | 53 |

| | | |
|--------|--|-----|
| 3.6.5 | Разливание контрагентов по списку MergedId..... | 54 |
| 3.6.6 | Полное разливание записей по списку результирующих карточек | 56 |
| 3.6.7 | Задача восстановления атрибутов по списку HIDов..... | 57 |
| 3.6.8 | Задача актуализации адресов по ФИАС..... | 58 |
| 3.6.9 | Исправьте базу данных в случае неконсистентного состояния..... | 60 |
| 3.6.10 | Переоткрытие закрытых по ошибке контрагентов с сохранением истории | 60 |
| 3.6.11 | Актуализация атрибутов на основе данных staging..... | 61 |
| 3.6.12 | Закрытие актуальных (результирующих и всех исходных) атрибутов по списку | 62 |
| 3.6.13 | Задача очистки таблиц параметров для массовой переобработки данных | 63 |
| 3.6.14 | Подготовка списков для повторной обработки..... | 63 |
| 3.6.15 | Таблицы с параметрами задач массовой переобработки..... | 64 |
| 3.6.16 | Актуализация статуса невалидности документов..... | 67 |
| 3.6.17 | Закрытие исходных атрибутов по списку с переслиянием результирующих | 68 |
| 3.6.18 | Переобработка телефонов | 69 |
| 3.6.19 | Переобработка телефонов. Второй вариант..... | 71 |
| 3.6.20 | Задача восстановления записи на версию по списку HID..... | 72 |
| 3.6.21 | Задача переочистки реквизитов | 73 |
| 3.6.22 | Задача переобработки UNDEF-телефонов | 74 |
| 3.6.23 | Задача пересчета скоринговых значений атрибутов..... | 74 |
| 3.6.24 | Переочистка гранулярных кодов качества ФИО | 75 |
| 3.6.25 | Полная переочистка ФИО | 76 |
| 3.6.26 | Заполнение ФИО ИП | 77 |
| 3.6.27 | Заполнение полной ОПФ..... | 78 |
| 3.6.28 | Переочистка кодов качества паспортов..... | 79 |
| 3.6.29 | Переочистка кодов качества дат рождения..... | 81 |
| 3.6.30 | Задача переоткрытия связей | 82 |
| 3.6.31 | Переобработка email | 83 |
| 3.6.32 | Переочистка email..... | 83 |
| 3.6.33 | Задача заполнения технических полей..... | 84 |
| 3.6.34 | Задача поиска массивных карточек | 85 |
| 3.6.35 | Задача обработки массивных карточек контрагентов | 86 |
| 3.6.36 | Задача поиска золотых карточек для переслияния | 87 |
| 3.6.37 | Задача переноса открытых связей закрытых транзитивных карточек на активные золотые 88 | |
| 3.7 | Системные задачи | 89 |
| 3.7.1 | Системные задачи | 89 |
| 3.7.2 | allEnginesRebuildByName. Перестроение указанных поисковых индексов..... | 89 |
| 3.7.3 | Общее описание | 90 |
| 3.7.4 | Валидация поисковых индексов по идентификаторам..... | 90 |
| 3.7.5 | Актуализация поискового индекса по списку контрагентов..... | 92 |
| 3.7.6 | Выполнение вспомогательной хранимой процедуры | 93 |
| 3.7.7 | Удаление старых обработанных конфликтов..... | 93 |
| 3.7.8 | Создание и редактирование системных учетных записей | 94 |
| 3.7.9 | searchEngineOptimize. Оптимизация поисковых индексов | 96 |
| 3.7.10 | graphEngineFullRebuild перестроение графа связей..... | 97 |
| 3.7.11 | enginesFullRebuild. Полное перестроение поискового индекса | 98 |
| 3.7.12 | findHistoricalConflicts. Поиск конфликтных изменений | 98 |
| 3.7.13 | Задача обновления кэшей | 100 |
| 3.7.14 | Очистка кэша инкрементального поиска дубликатов | 100 |
| 3.7.15 | Очистка хэшей поиска дубликатов | 100 |
| 3.8 | Экспорт данных | 101 |
| 3.8.1 | Инкрементальный экспорт эталонных связей | 101 |
| 3.8.2 | Полный экспорт эталонных связей..... | 102 |
| 3.9 | Диагностика | 103 |
| 3.9.1 | Когда использовать..... | 103 |
| 3.9.2 | Как собрать диагностику..... | 103 |
| 3.9.3 | Сбор логов | 104 |
| 3.9.4 | Сбор диагностики по контрагенту | 104 |
| 3.9.5 | Сбор диагностики с указанным профилем | 106 |
| 3.9.6 | Проверка БД на непротиворечивость | 107 |
| 3.9.7 | Профили диагностики..... | 109 |
| 3.9.8 | Проверка слитых контрагентов на непротиворечивость | 112 |
| 3.9.9 | Найти и исправить проблемы в золотых карточках | 112 |
| 3.9.10 | Сбор диагностики в режиме горячего резерва | 112 |
| 3.9.11 | Сбор метрик..... | 113 |
| 3.9.12 | Сбор диагностики по задаче | 114 |
| 3.9.13 | Сбор логов за сегодня | 115 |

| | | |
|--------|--|-----|
| 3.9.14 | Сбор метрик за предыдущий месяц | 116 |
| 3.9.15 | Импорт датасета с контрагентами (xls)..... | 116 |
| 3.10 | Триггеры | 117 |
| 3.10.1 | Триггер | 118 |
| 3.10.2 | Список триггеров..... | 118 |
| 3.10.3 | Работа с триггерами | 119 |
| 3.10.4 | Список триггеров..... | 122 |
| 3.11 | Возобновляемые задачи..... | 124 |
| 3.11.1 | Запуск на ночь | 125 |
| 3.12 | Непрерываемые задачи..... | 125 |
| 3.13 | Проверка по черным спискам..... | 125 |
| 3.13.1 | Проверка по черным спискам | 125 |
| 3.13.2 | Обновление дат актуальности ЧС | 125 |
| 3.14 | Необратимое удаление данных | 126 |
| 3.14.1 | Общее описание | 126 |
| 3.14.2 | Особенности задачи | 126 |
| 3.14.3 | Запуск задачи | 126 |
| 3.14.4 | Обработка ошибок | 127 |
| 3.15 | Заккрытие записей | 127 |
| 3.15.1 | Запуск задачи | 127 |
| 3.15.2 | Обработка ошибок | 128 |
| 3.16 | Порционные задачи | 128 |
| 3.16.1 | Конфигурация задачи | 128 |
| 3.16.2 | Работа задачи | 129 |
| 3.16.3 | Подключение порционности..... | 129 |
| 3.17 | Проверка по ЕГРЮЛ..... | 129 |
| 3.17.1 | Полная проверка по ЕГРЮЛ | 129 |
| 3.17.2 | Актуализация ЮЛ – карточек-копий ЕГРЮЛ..... | 130 |
| 3.18 | Массовые атрибуты | 131 |
| 3.18.1 | Массовые атрибуты | 131 |
| 3.18.2 | Поиск массовых реквизитов и атрибутов..... | 131 |
| 3.18.3 | Проставление популярности по найденным значениям (ФЛ и ЮЛ) | 132 |
| 3.19 | Миграция | 132 |
| 3.19.1 | Автопрогон скриптов выдачи грантов пользователям..... | 132 |
| 3.19.2 | Актуализация реестров | 132 |
| 3.19.3 | Задача переобработки места рождения | 134 |
| 3.19.4 | Инициализация прошлых версий атрибутов | 135 |
| 3.19.5 | Миграция нового поля | 135 |
| 3.20 | Миграция структуры хранения исторических данных | 136 |
| 3.20.1 | Конфигурация задачи | 136 |
| 3.20.2 | Форсирование миграции структуры исторических данных (Фаза 3)..... | 137 |
| 3.21 | Сбор статистики | 138 |
| 3.21.1 | Конфигурация задачи | 138 |
| 3.21.2 | Особенности задачи | 139 |
| 3.21.3 | Настройка сбора статистики | 139 |
| 3.21.4 | DATA_METRIC..... | 140 |
| 3.21.5 | DATA_STATISTIC | 141 |
| 3.21.6 | Добавление новых метрик качества данных | 141 |
| 3.22 | Фоновые задачи | 145 |
| 3.22.1 | Алгоритм мониторинга и перераспределения нагрузки между онлайн-поток и фоновыми задачами | 145 |
| 3.23 | Задача массовой обработки конфликтов | 147 |
| 3.23.1 | Описание задачи..... | 147 |
| 3.23.2 | Запуск задачи | 147 |
| 3.23.3 | Конфигурация задачи | 147 |
| 3.24 | Полное необратимое удаление ВСЕХ данных выбранного типа..... | 148 |
| 3.24.1 | Общее описание | 148 |
| 3.24.2 | Запуск задачи | 148 |
| 3.24.3 | Конфигурация задачи | 149 |
| 3.25 | Общие параметры задач | 149 |
| 3.25.1 | Секретный параметр unsafe.forceExecution | 149 |
| 3.25.2 | Параметр jmsEventFlag | 150 |
| 4.1 | Мэппинг ролей LDAP | 152 |
| 4.1.1 | Список методов, доступных только для чтения/записи | 153 |
| 4.2 | Настройки задач во внешней директории..... | 153 |

| | | |
|--------|--|-----|
| 4.3 | Параметры JMS | 153 |
| 4.3.1 | Общие параметры | 154 |
| 4.3.2 | Специфичные для очередей TIBCO параметры | 155 |
| 4.3.3 | Специфичные для очередей IBM MQ параметры | 155 |
| 4.3.4 | Параметры для передачи по HTTP-протоколу | 155 |
| 4.3.5 | Параметры для отправки в KAFKA | 155 |
| 4.3.6 | Специфичные для очередей RabbitMQ параметры | 156 |
| 4.4 | Параметры LDAP | 157 |
| 4.5 | Параметры SOAP и REST сервисов | 159 |
| 4.6 | Параметры задач | 163 |
| 4.6.1 | Таймауты при запросах к длительным БД процедурам | 168 |
| 4.7 | Параметры задачи дедупликации | 168 |
| 4.8 | Параметры запуска JBOSS | 169 |
| 4.9 | Параметры импорта из буферной области | 170 |
| 4.10 | Параметры импорта справочников | 170 |
| 4.11 | Параметры модуля мониторинга | 170 |
| 4.12 | Параметры отправки email | 173 |
| 4.12.1 | Механизм отправки писем об ошибках на почту | 179 |
| 4.13 | Параметры подключения к Фактору | 180 |
| 4.14 | Параметры поискового индекса | 183 |
| 4.14.1 | Неблокирующее перестроение поискового индекса | 189 |
| 4.15 | Параметры сервисов | 190 |
| 4.15.1 | Стратегия решения конфликтов | 194 |
| 4.16 | Параметры синхронизации нод | 195 |
| 4.17 | Параметры экспорта данных | 196 |
| 5.1 | Добавление в справочник новой записи | 198 |
| 5.2 | Модификация существующей в справочнике записи | 198 |
| 5.3 | Представление с содержимым справочников | 199 |
| 5.4 | Список справочников, которые можно пополнять и редактировать | 199 |
| 6.1 | Журналы системы «Гражданский фактор» | 200 |
| 6.1.1 | Журнал (лог) производительности | 201 |
| 6.1.2 | Журнал (лог) сетевых соединений | 204 |
| 6.1.3 | cdi-distributed-stats.log | 206 |
| 6.1.4 | Журнал аутентификаций | 206 |
| 6.1.5 | Журнал доступа к персональным данным | 208 |
| 6.1.6 | Журнал исключенных ошибок | 208 |
| 6.1.7 | Журнал количества ошибок, исключенных из основного лога | 209 |
| 6.1.8 | Журнал обновлений версий сервера | 209 |
| 6.1.9 | Журнал с персональными данными | 210 |
| 6.1.10 | Логи задачи трансформации | 210 |
| 6.1.11 | Лог триггера | 213 |
| 6.1.12 | Система «Гражданский фактор» (cdi-soap-stats.log) | 213 |
| 6.1.13 | cdi-ui-stats.log | 215 |
| 6.1.14 | Трассировка JMS-сообщений | 217 |
| 6.1.15 | Трассировка SOAP-запросов | 218 |
| 6.1.16 | Трассировка произвольного пакета | 219 |
| 6.2 | Журналы Фактора | 219 |
| 6.3 | Уровни протоколирования | 221 |
| 6.4 | Аудит | 221 |
| 6.4.1 | Поля записи журнала | 222 |
| 6.4.2 | Типы событий | 222 |
| 7.1 | Общее описание | 225 |
| 7.2 | Обновление модулей | 225 |
| 7.2.1 | Для обновления системы необходимо выполнить следующие шаги: | 225 |
| 7.2.2 | Обновление модуля «Фактор» | 226 |
| 7.3 | Ручной запуск и остановка | 226 |
| 7.4 | Для обновления системы необходимо выполнить следующие шаги: | 227 |
| 7.5 | Обновление модуля «Фактор» | 227 |
| 7.6 | Автоматическое создание схемы CDI | 228 |
| 7.6.1 | Настройка WildFly | 228 |
| 7.6.2 | Создание новой схемы | 228 |
| 7.6.3 | Пересоздание существующей схемы | 228 |
| 8.1 | Основное (FAQ) | 230 |
| 8.1.1 | CPU под 100% на задаче | 230 |

| | | |
|--------|--|-----|
| 8.1.2 | Индекс неактуальный (Index may be corrupted) | 230 |
| 8.1.3 | Как переместить прод с одной машины на другую | 230 |
| 8.1.4 | Как поднять бекап базы с PROD на TEST стенде? | 231 |
| 8.1.5 | Что такое "схлопывание" атрибутов? | 232 |
| 8.2 | Менеджер Данных (FAQ) | 233 |
| 8.2.1 | Не могу зайти в АРМ Менеджера данных | 234 |
| 8.2.2 | Отслеживание логинов в CDM | 234 |
| 8.2.3 | Ошибка «Разрешите противоречия между подтвержденными и отклоненными дубликатами, прежде чем выполнять слияние» | 234 |
| 8.3 | АРМ Администратора (FAQ) | 235 |
| 8.3.1 | В IE* нельзя создать триггер - не удается выбрать задачи из списка | 235 |
| 8.3.2 | Изменение параметра oldMaxId | 235 |
| 8.4 | Дополнительно (FAQ) | 236 |
| 8.4.1 | Запросы в БД, соответствующие фильтрам в CDM | 236 |
| 8.4.2 | Как пополнить справочник имен / фамилий? | 237 |
| 8.5 | Инструменты для проверки (FAQ) | 238 |
| 8.5.1 | Как снять Thread Dump | 238 |
| 8.5.2 | Как снять Thread Dump через командную строку в Unix | 240 |
| 8.5.3 | Как можно вызвать метод SOAP-интерфейса? | 241 |
| 8.5.4 | Как можно вызвать REST-метод вручную | 247 |
| 8.5.5 | Как посмотреть JMS, уходящие в RabbitMQ | 251 |
| 8.5.6 | Как посмотреть JMS, уходящие в IBM MQ | 253 |
| 8.5.7 | Как посмотреть JMS, уходящие в Tibco | 259 |
| 8.5.8 | Как послать в Tibco входящее JMS-сообщение | 263 |
| 8.5.9 | Как настроить приветствие на Linux | 270 |
| 8.5.10 | Как по ID в исходной системе понять с кем слит контрагент | 271 |
| 8.6 | Запуск задач / триггеров | 271 |
| 8.6.1 | Фазы миграции на _HIST таблицы | 271 |
| 8.6.2 | Что делать, если инкремент упал с ошибкой | 275 |
| 8.7 | FAQ для сопровождающих | 275 |
| 8.8 | Фактор (FAQ) | 275 |
| 8.8.1 | Есть ли doBatchClean для телефонов? | 275 |
| 8.9 | Интеграция (faq) | 277 |
| 8.9.1 | Как передать две записи с одной связью | 278 |
| 9.1 | Загрузка и обновление ЕГРЮЛ | 279 |
| 9.1.1 | Первичная загрузка справочников ЕГРЮЛ и ЕГРИП в систему «Гражданский фактор» | 279 |
| 9.1.2 | Обновление справочников ЕГРЮЛ | 279 |
| 9.2 | Обновление справочников Подсказок | 280 |
| 9.2.1 | Справочник email | 280 |
| 9.2.2 | Справочник ip-адресов | 281 |
| 9.2.3 | Справочник ФИАС (20.2–20.8) | 281 |
| 9.2.4 | Справочник ФИАС (20.9+) | 282 |
| 9.2.5 | Справочник адресов (20.2–20.8) | 283 |
| 9.2.6 | Справочник адресов (20.9+) | 284 |
| 9.2.7 | Справочник банков | 287 |
| 9.2.8 | Обновление внешних справочников | 288 |
| 9.2.9 | Справочник имен | 289 |
| 9.2.10 | Справочник компаний | 289 |
| 9.2.11 | Справочник координат | 290 |
| 9.3 | Обновление справочников Фактора | 291 |
| 9.3.1 | База данных перенесенных номеров | 291 |
| 9.3.2 | Обновление ЕГРЮЛ и ЕГРИП | 293 |
| 9.3.3 | Обновление ФИАС | 295 |
| 9.3.4 | Обновление доменов первого уровня | 297 |
| 9.3.5 | Перечень недействительных паспортов | 299 |
| 9.3.6 | Настройка подключения Фактора к подсказкам | 301 |
| 9.3.7 | Справочник Россвязи | 302 |
| 9.3.8 | Справочник геокоординат | 303 |
| 9.3.9 | Справочник локальных доменов | 305 |
| 9.3.10 | Справочник одноразовых e-mail | 307 |
| 9.3.11 | Справочник площадей квартир | 309 |
| 9.3.12 | Справочник стоимости квадратного метра | 311 |

1 Требования к администратору CDI

1.1 Требования

- Базовые знания 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).

1.2 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 системы «Гражданский фактор» и отслеживание их выполнения.

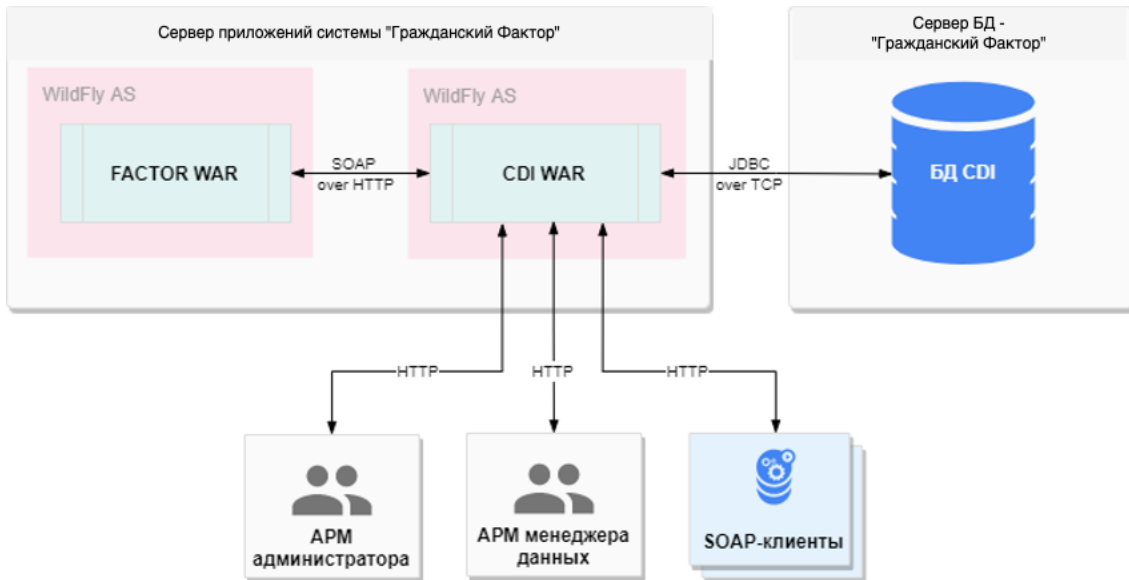
1.3 Доступы

Администратору CDI необходимы права администратора на сервере приложений.

2 Архитектура системы

2.1 Типовая схема развёртывания

На рисунке ниже представлена типовая схема развёртывания системы «Гражданский фактор»:



В разделе описана схема развёртывания одного экземпляра системы. Также возможна реализация отказоустойчивого решения за счет использования [горячего резерва](#).

2.2 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 системы «Гражданский фактор» через предоставляемые системой SOAP-сервисы. |
| WildFly AS | WildFly Application Server, не ниже версии 8.2.0+. Предоставляет контейнер для работы серверов систем «Гражданский фактор» и «Фактора». |
| CDI WAR | Модуль сервера системы «Гражданский фактор». Исполняется в контейнере WildFly AS. Обеспечивает выполнение функций системы «Гражданский фактор». |
| ФАКТОР WAR | Модуль сервера «Фактор». Обеспечивает выполнение функций системы «Фактор». |

| Компонент схемы | Назначение |
|-----------------|---|
| БД CDI | База данных системы «Гражданский фактор». Реализуется на базе PostgreSQL. |

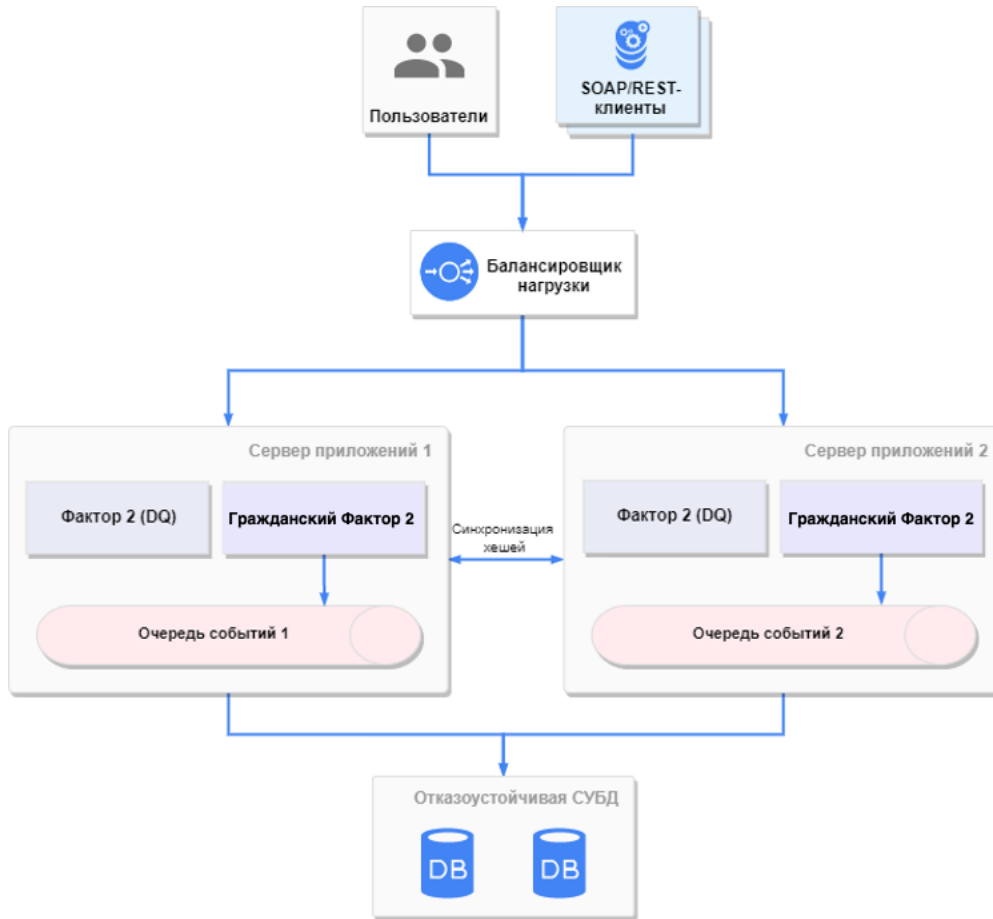
2.3 Взаимосвязи типовой схемы развертывания

| Тип взаимосвязи | Компоненты | Назначение |
|-----------------|------------------------|---|
| JDBC over TCP | CDI WAR – БД CDI | Сервер системы «Гражданский фактор» взаимодействует с базой данных для работы с информацией о физ. лицах, находящейся под управлением системы «Гражданский фактор». |
| 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 «Гражданский фактор» 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 «Гражданский фактор» реализуется с помощью вызова SOAP-сервисов, предоставляемых последней. |

2.4 Отказоустойчивое решение (горячий резерв)

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

Общая схема развертывания системы приведена на рисунке ниже:



2.4.1 Работа в режиме горячего резерва

Сценарий использования выглядит следующим образом:

1. Клиенты (Пользователи, SOAP-клиенты) обращаются к балансировщику нагрузки, который направляет запросы на тот или иной экземпляр системы «Гражданский фактор».
2. При необходимости доступа к данным система «Гражданский фактор» работает с БД.
3. Кэши, необходимые для работы системы «Гражданский фактор» (поисковый индекс и кэш поиска дубликатов) автоматически синхронизируются между экземплярами системы через очередь сообщений.
4. Общие настройки всех экземпляров системы «Гражданский фактор» 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 системы «Гражданский фактор» 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 систем «Гражданский фактор» и ФАКТОР.
2. Предоставить два сервера приложений для системы «Гражданский фактор».

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).

2.4.2 Обновление без прерывания обслуживания

Типичное обновление системы «Гражданский фактор» при переходе на следующую версию может включать в себя следующие шаги:

1. Обновление ПО «Гражданский фактор».
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 системы «Гражданский фактор», 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 системы «Гражданский фактор» для проведения миграции.

3 Вход в систему (администратор)

Страница входа: <http://CEPBEP:8080/cdi/>.

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

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

3.1 Группы задач

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

Группа задач, предназначенных для загрузки данных из внешних источников в систему «Гражданский фактор».

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

Типовые задачи, которые есть всегда:

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

Общий список задач по загрузке данных (задачи подключаются сотрудниками службы сопровождения вендора решения)

3.1.1 Особенности

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

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

Режим запуска определяется значением параметра:

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

3.1.2 Результат выполнения

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

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

- [Общие параметры для задач загрузки данных](#)
- [Задача загрузки записей из буферной таблицы БД](#)
- [Задача загрузки договоров из буферной таблицы БД](#)
- [Задача загрузки связей договоров из буферной таблицы БД](#)
- [Задача загрузки связей](#)
- [Задача загрузки записей из CSV-файла](#)
- [Задача загрузки записей из каталога](#)
- [Задача загрузки записей по SFTP](#)
- [Задача загрузки справочников ЕГРЮЛ / ЕГРИП](#)
- [Трансформация данных](#)
- [Активировать исходные записи](#)
- [Деактивировать исходные записи](#)
- [Задача циклической загрузки данных из БД](#)

3.1.3 Идентификация и слияние дубликатов

Группа задач, предназначенных для идентификации и слияния дубликатов:

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

с последующим слиянием гарантированных дубликатов;

- `importExternalDuplicateWithMerge` — слияние дубликатов по внешнему списку;
- `rebuildFullDedupHashes` — перестроение хэшей полного поиска дубликатов;
- `findGuaranteedDuplicatesAndMerge` — поиск гарантированных дубликатов и слияние;
- `egrCheck` — полная проверка по ЕГРЮЛ;
- `egrCheckActualize` — Проверка по ЕГРЮЛ контрагентов, требующих актуализации.

3.1.4 Проверка по черным спискам

Группа задач, предназначенных для проверки данных по черным спискам:

- `fullBlacklistCheck` — полная проверка сущностей по черным спискам;
- `actualizeBlacklistCheck` — проверки по черным спискам контрагентов, требующих актуализации. Задача проверяет только новые и измененные записи;
- `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.

3.1.5 Массовые атрибуты

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

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

3.1.6 Миграция

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

- `innMigrateRequisiteTask` — Миграция поля ИНН
- `kioMigrateRequisiteTask` — Миграция поля КИО

3.1.7 Повторная обработка контрагентов

- [Задача перерасчета hidden-флагов атрибутов по списку записей](#)
- [Задача полного переслияния записей](#)
- [Задача переслияния реквизитов контрагентов](#)
- [Разлияние контрагентов по списку MergedId](#)
- [Полное разлияние записей по списку результирующих карточек](#)
- [Задача восстановления атрибутов по списку NIDов](#)
- [Задача актуализации адресов по ФИАС](#)
- [Исправьте базу данных в случае неконсистентного состояния](#)
- [Переоткрытие закрытых по ошибке контрагентов с сохранением истории](#)
- [Актуализация атрибутов на основе данных staging](#)
- [Закрытие актуальных \(результирующих и всех исходных\) атрибутов по списку](#)
- [Задача очистки таблиц параметров для массовой переобработки данных](#)
- [Подготовка списков для повторной обработки](#)
- [Таблицы с параметрами задач массовой переобработки](#)
- [Актуализация статуса невалидности документов](#)
- [Закрытие исходных атрибутов по списку с переслиянием результирующих](#)
- [Переобработка телефонов](#)
- [Переобработка телефонов. Второй вариант](#)
- [Задача восстановления записи на версию по списку NID](#)
- [Задача переочистки реквизитов](#)
- [Задача переобработки UNDEF-телефонов](#)
- [Задача пересчета скоринговых значений атрибутов](#)
- [Переочистка гранулярных кодов качества ФИО](#)
- [Полная переочистка ФИО](#)
- [Заполнение ФИО ИП](#)
- [Заполнение полной ОПФ](#)
- [Переочистка кодов качества паспортов](#)
- [Переочистка кодов качества дат рождения](#)
- [Задача переоткрытия связей](#)
- [Переобработка email](#)
- [Переочистка email](#)
- [Задача заполнения технических полей](#)

- [Задача поиска массивных карточек](#)
- [Задача обработки массивных карточек контрагентов](#)
- [Задача поиска золотых карточек для переслияния](#)
- [Задача переноса открытых связей закрытых транзитивных карточек на активные золотые](#)

[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 | hiddenFlagRefineByHids | Задача перерасчета hidden-флагов атрибутов по списку контрагентов | | |
| 2 | reMergeFullByPartyHids | Задача полного переслияния контрагентов | | |
| 3 | reMergeRequisiteByPartyHids | Задача переслияния реквизитов контрагентов | | |
| 4 | unMergeByMergedIds | Разлияние контрагентов по списку MergedId | | |
| 5 | fullUnmergeByResultHids | Полное разлияние контрагентов по списку результирующих контрагентов | | |
| 6 | attributeRecoverByHids | Задача восстановления атрибутов по списку HIDs | | |
| 7 | addressActualization | Задача актуализации адресов по ФИАС | | |
| 8 | mergedPartiesCheckAndRmerge | Исправьте базу данных в случае неконсистентного состояния | | |
| 9 | partyReopenByHids | Переоткрытие закрытых по ошибке | | |

| | Задача | Описание | Действия перед выполнением задачи | Действия после выполнения задачи |
|----|---|--|-----------------------------------|----------------------------------|
| | | контрагентов с сохранением истории | | |
| 10 | attributeActualizeByStaging | Актуализация атрибутов на основе данных staging | | |
| 11 | attributeCloseActualByHids | Закрытие актуальных (результующих и всех исходных) атрибутов по списку | | |
| 12 | cleanTaskParameterTables | Задача очистки таблиц параметров для массовой переобработки данных | | |
| 13 | fillTaskPartyHidsActual fillTaskPartyHidsOriginal fillTaskPartyHidsResulted fillAddressActualizationTaskPartyHids fillTaskPartyHidsBadAdresses fillTaskPartyHidsOriginalWithClosed | Подготовка списков для повторной обработки | | |
| 14 | passportInvalidStatusReCleaner | Актуализация статуса невалидности документов | | |
| 15 | attributeCloseSourceByHids | Закрытие исходных атрибутов по списку с переслиянием результирующих | | |
| 16 | phoneExtraFieldsReCleaner | Переобработка телефонов | | |
| 17 | phoneExtraFieldsActualization.physical phoneExtraFieldsActualization.legal | Переобработка телефонов. Новый вариант | | |

| | Задача | Описание | Действия перед выполнением задачи | Действия после выполнения задачи |
|----|------------------------------|--|-----------------------------------|---------------------------------------|
| 18 | partyRecoverByHids | Задача восстановления КА на версию по списку HID | | |
| 19 | reCleanMockRequisiteTask | Задача переочистки реквизитов | | |
| 20 | phoneUndefReCleaner | Задача переобработки UNDEF-телефонов | | |
| 21 | refreshScoresTask | Задача пересчета скоринговых значений атрибутов | | |
| 22 | physicalFioQcReCleaner | Переочистка гранулярных кодов качества ФИО | Заполнение таблицы актуальными КА | Удаление таблицы по завершении задачи |
| 23 | passportQcReCleaner | Переочистка кодов качества паспортов | | |
| 24 | physicalBirthdateQcReCleaner | Переочистка кодов качества дат рождения | Заполнение таблицы актуальными КА | Удаление таблицы по завершении задачи |
| 25 | relationReopenByIds | Задача переоткрытия связей | | |
| 26 | legalIpReCleaner | Заполнение ФИО ИП | | |

3.1.8 Подготовка списков для повторной обработки

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

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

Данные задачи следует выполнять перед запуском задачи повторной обработки выборочных записей.

В результате работы задачи таблица TASK_PARTY_HIDS будет наполнена идентификаторами записей:

- `fillTaskPartyHidsActual` — все актуальные записи (`merged_status 0` или `2`);
- `fillTaskPartyHidsOriginal` — все оригинальные записи (`merged_status 0` или `1`);
- `fillTaskPartyHidsResulted` — все результирующие записи (`merged_status 2`);
- `fillTaskPartyHidsOriginalWithClosed` — все оригинальные записи (`merged_status 0` или `1`) и закрытые записи (`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) — карточки, адреса из которых должны быть актуализированы по КЛАДР (`mergedStatus: 0,2`);
- `fillTaskPartyHidsBadAdresses` — карточки с адресами, отмеченными на ручную обработку (`mergedStatus 0` или `2`), задача не выбирает карточки, если все адреса в ней, отмеченные на ручную обработку, скрыты.

3.1.8.1 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 карточек

- `searchEngineOptimize` — оптимизация поисковых индексов (граф связей, индексы по записям, 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 системы «Гражданский фактор»:

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

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

3.1.9 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).

3.1.10 Параметры, общие для всех типов задач

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

3.2 Планировщик задач

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

На странице обобразится 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.

3.2.1 Запуск задачи по расписанию

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

Если нужно, чтобы триггер сохранился и после перезапуска системы «Гражданский фактор», необходимо добавить его в конфигурацию:

1. Посмотреть имя интересующей задачи в АРМ Администратора (страница *Задачи*, столбец *Название*).
2. Остановить сервис CDI.
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 сервис CDI.

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

```
<beans>

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

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

</beans>
```

Само расписание запуска настраивается в строке:

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

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

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

```
* * * * *
- - - - -
| | | | |
| | | | | ----- День недели (MON - SUN)
| | | | | ----- Месяц (1 - 12)
| | | | | ----- День (1 - 31)
| | | | | ----- Час (0 - 23)
| | | | | ----- Минута (0 - 59)
| | | | | ----- Секунда (0-59)
```

Примеры:

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

3.2.2 Остановка выполнения задачи

Запущенную задачу можно остановить двумя способами:

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

Действие доступно только для прерываемых задач.

| Запущенные задачи | | | | | |
|-------------------|----------------------|---------|-------------------------------------|-------------|--|
| Группа | Название | Статус | Описание | Комментарий | Действия |
| system | searchEngineOptimize | PENDING | Оптимизация всех поисковых индексов | | Завершить Детали |

- **Убить** - жесткая остановка. Система принудительно завершает все процессы задачи. Несохранившийся 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 | | | | | |
|-------------------|--|---------|---|-------------|--|
| Группа | Название | Статус | Описание | Комментарий | Действия |
| dedup | findDuplicatesAndMergeAndRebuildGroups | RUNNING | Поиск дубликатов, слияние и перестроение групп дубликатов | | Убить Детали |

3.3 Взаимодействие с JMS-очередями

3.3.1 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 переправки сообщений в выходную очередь](#)
- [Отправка новых JMS сообщений](#)
- [Повторная отправка ошибочных JMS сообщений за последние 24 часа](#)

3.3.2 Задача очистки хранилища JMS-сообщений

CleanEventsPersistence

3.3.2.1 Описание задачи

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

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

3.3.2.2 Конфигурация задачи

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

3.3.3 Задача переобработки сохраненных сообщений из входной очереди

| |
|-------------------------|
| ReprocessIncomingEvents |
|-------------------------|

3.3.3.1 Описание задачи

Переобрабатывает сообщения, поступившие во входную очередь и сохраненные в БД системы «Гражданский фактор», 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.

3.3.3.2 Конфигурация задачи

| Параметр | Тип | Описание | Значение по умолчанию |
|---------------|---|--|-----------------------|
| minCreateDate | Дата в формате <code>dd.mm.yyyy hh:mi:ss</code> | Нижняя граница (включительно) даты создания сообщений, которые будут переобработаны | |
| maxCreateDate | Дата в формате <code>dd.mm.yyyy hh:mi:ss</code> | Верхняя граница (включительно) даты создания сообщений, которые будут переобработаны | |

3.3.4 Задача переотправки сообщений в выходную очередь

| |
|-------------------------|
| RepublishOutgoingEvents |
|-------------------------|

3.3.4.1 Описание задачи

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

Предназначена для следующих ситуаций:

- Повторная выгрузка сообщений, которые не удалось записать в выходную очередь ранее (такие сообщения имеют статус ERROR),
- Повторная выгрузка сообщений, которые ранее были переданы в выходную очередь, но по каким-либо причинам не дошли до систем-получателей. Такие сообщения в системе «Гражданский фактор» 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.

3.3.4.2 Конфигурация задачи

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

3.3.5 Отправка новых JMS сообщений

```
publishPendingEvents
```

3.3.5.1 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.

3.3.5.2 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.

3.3.5.3 Конфигурация задачи

| Параметр | Описание | Значение по умолчанию |
|-------------------|--|-----------------------|
| packSize | Размер пакета сообщений для записи в очередь | 1 |
| sleepMilliseconds | Время ожидания (мс) между отправками пакетов сообщений | 50 |
| maxSend | Максимальное количество отправленных сообщений за один запуск задачи (включая ошибки), по умолчанию "-1" (без ограничения) | -1 |
| maxErrors | Максимальное количество ошибок, после которого задача останавливается | 100 |

3.3.5.4 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
```

3.3.6 Повторная отправка ошибочных JMS сообщений за последние 24 часа

```
RepublishOutgoingEventsForTheLastDay
```

3.3.6.1 Описание задачи

Переотправляет в выходную очередь ошибочные события за последние 24 часа.

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

3.3.6.2 Конфигурация задачи

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

3.4 Импорт данных

Группа задач, предназначенных для загрузки данных из внешних источников в систему «Гражданский фактор».

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

Типовые задачи, которые есть всегда:

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

Общий список задач по загрузке данных (задачи подключаются сотрудниками службы сопровождения вендора решения)

3.4.1 Особенности

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

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

Режим запуска определяется значением параметра:

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

3.4.2 Результат выполнения

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

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

- [Общие параметры для задач загрузки данных](#)

- [Задача загрузки записей из буферной таблицы БД](#)
- [Задача загрузки договоров из буферной таблицы БД](#)
- [Задача загрузки связей договоров из буферной таблицы БД](#)
- [Задача загрузки связей](#)
- [Задача загрузки записей из CSV-файла](#)
- [Задача загрузки записей из каталога](#)
- [Задача загрузки записей по SFTP](#)
- [Задача загрузки справочников ЕГРЮЛ / ЕГРИП](#)
- [Трансформация данных](#)
- [Активировать исходные записи](#)
- [Деактивировать исходные записи](#)
- [Задача циклической загрузки данных из БД](#)

3.4.3 Общие параметры для задач загрузки данных

3.4.3.1 Параметры для задач, использующих трансформацию данных
Если при импорте записей требуется обработка данных Фактором, тогда загрузка данных осуществляется в два этапа:

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

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

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

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

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

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

3.4.4 Задача загрузки записей из буферной таблицы БД

3.4.4.1 Описание задачи

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

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

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

3.4.4.2 Выборка данных

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

```

/**
 * Возвращает курсор на записи инкремента (все записи с идентификатором >
 old_max_id)
 * @param party_type тип контрагента
 * @param old_max_id значение идентификатора последней успешно загруженной
 записи на данный момент
 * @return new_max_id максимальное значение идентификатора в инкремента
 * @return lRecordset 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
lrecordset 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);

```

3.4.5 Задача загрузки договоров из буферной таблицы БД

3.4.5.1 Описание задачи

Загружает инкремент договоров из буферной таблицы и опционально удаляет успешно загруженные данные. Чтобы загружать именно инкремент, а не каждый раз всю таблицу целиком, отслеживает и хранит уникальный целочисленный идентификатор последней

успешно загруженной записи (исходя из предположения, что идентификаторы в буферной таблице строго возрастающие).

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

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

3.4.5.2 Выборка данных

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

```

/**
 * Возвращает курсор на записи инкремента (все записи с идентификатором >
 old_max_id)
 * @param essenceType тип загружаемых данных, по умолчанию AGREEMENT
 * @param old_max_id значение идентификатора последней успешно загруженной
 записи на данный момент
 * @return new_max_id максимальное значение идентификатора в инкремента
 * @return lRecordset 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
lrecordset 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);

```

3.4.6 Задача загрузки связей договоров из буферной таблицы БД

3.4.6.1 Описание задачи

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

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

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

3.4.6.2 Выборка данных

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

```

/**
 * Возвращает курсор на записи инкремента (все записи с идентификатором >
 old_max_id)
 * @param party_type тип контрагента
 * @param old_max_id значение идентификатора последней успешно загруженной
 записи на данный момент
 * @return new_max_id максимальное значение идентификатора в инкремента
 * @return lRecordset 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
lrecordset 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);

```

3.4.7 Задача загрузки связей

3.4.8 Задача загрузки записей из CSV-файла

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

| Параметр | Тип | Описание |
|-------------------------|------------------------------|---|
| sourceFileName | Строка | Полный путь к загружаемому файлу. Для Windows путь необходимо указывать с задвоением обратного слэша, например: C:\\upload\\client_data.csv |
| sourceDataEncoding | Строка | Кодировка файла. Примеры: windows-1251, UTF-8. |
| sourceDataSeparator | Символ | Разделитель полей файла. Примеры: ; ¤ |
| 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 SYSTEM системы «Гражданский фактор») |
| essenceType | Тип сущности | Тип загружаемых сущностей |

3.4.9 Задача загрузки записей из каталога

Загружает записи 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 |
|--------------------|--------|---|
| sourceFileName | Строка | Полный путь к загружаемому каталогу. Для Windows путь необходимо указывать с задвоением обратного слэша, например: C:\\upload |
| sourceDataEncoding | Строка | Кодировка файлов в каталоге. Примеры: windows-1251, UTF-8. |

| Параметр | Тип | Описание |
|-------------------------|--------------|---|
| sourceDataSeparator | Символ | Разделитель полей файлов в каталоге. Примеры: ; ✕ |
| 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. |
| essenceType | Тип сущности | Тип загружаемых сущностей |

3.4.10 Задача загрузки записей по SFTP

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

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

| Параметр | Тип | Описание |
|------------------------------|--------|--|
| Параметры доступа по SFTP | | |
| serverHost | Строка | Доменное имя SFTP-сервера |
| serverPort | Строка | Порт для подключения по SFTP |
| username | Строка | Имя пользователя для подключения по SFTP |
| password | Строка | Пароль для подключения по SFTP |
| connectionTimeout | Целое | Таймаут подключения (в миллисекундах) |
| validateLinkInterval | Целое | Интервал проверки соединения с SFTP-сервером после обрыва соединения |
| reconnectCount | Целое | Количество повторных попыток подключения к SFTP-серверу |
| reconnectInterval | Целое | Интервал между повторными попытками подключения (в миллисекундах) |
| Параметры загружаемых файлов | | |
| sourceFileName | Строка | Полный путь к загружаемому каталогу на SFTP-сервере. |
| sourceDataEncoding | Строка | Кодировка файлов в каталоге. Примеры: windows-1251, UTF-8. |
| sourceDataReadBlockSize | Целое | Размер пакета записей для выборки из файла |
| sourceDataSeparator | Символ | Разделитель полей файлов в каталоге. Примеры: ; ✕ |

| Параметр | Тип | Описание |
|-------------|--------------|---|
| rawIdIndex | Целое | Порядковый номер поля, в котором расположен уникальный идентификатор исходной записи. Нумерация начинается с 0. |
| essenceType | Тип сущности | Тип загружаемых сущностей |

3.4.11 Задача загрузки справочников ЕГРЮЛ / ЕГРИП

EgrDictionaryImport

3.4.11.1 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 |
|--------------------------------------|---|----------------------------------|
| <code>sourceFileName</code> | Путь к директории со справочником ЕГР | <code>/path/to/egrul</code> |
| <code>sourceDataEncoding</code> | Кодировка справочника | UTF-8 |
| <code>bufferDataReadBlockSize</code> | Размер пакета записей для выборки | 1000 |
| <code>poolSize</code> | Максимальное количество потоков | 8 |
| <code>essenceType</code> | Тип загружаемых сущностей | EGR |
| <code>cipher</code> | Загрузка зашифрованного справочника, по умолчанию включена (значение <code>true</code>) | <code>true</code> |
| <code>massiveChanging</code> | Оповещение Фактора изменениями, по умолчанию включено (значение <code>false</code>). Если задать <code>true</code> , то Фактор не будет вызываться с инкрементальными <code>populate</code> | <code>false</code> |
| <code>closeAbsentBranches</code> | Закрывать отсутствующие в инкременте филиалы и агентства (по которым нет связей и ручных правок). По умолчанию <code>true</code> . | <code>true</code> |
| <code>sourceSystemsToLoad</code> | Список систем, записи из которых загружаются по умолчанию | FTS_LG, FTS_IN, RAFP, MLTST_EGRN |

3.4.12 Трансформация данных

transform

3.4.12.1 Описание задачи

Задача нужна для принудительной обработки записей из staging-области.

3.4.12.2 Конфигурация задачи

| Параметр | Тип | Описание | Значение по умолчанию |
|--------------------------|---------------------|---|-----------------------|
| terminateOnErrors | Логическое значение | Прервать выполнение при возникновении ошибок. | false |
| massiveChanging | Логическое значение | Оповещение Фактора изменениями. Если задать true, то Фактор не будет вызываться с инкрементальными populate | false |
| notifyListeners | Логическое значение | Оповещение поискового индекса КА о трансформации пачек Если оповещение отключено, после завершения задачи нужно перестроить индекс <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставить в запускаемой задаче</i> | true |
| iteratorRefreshPeriodMin | Числовое значение | Время обновления итератора по staging <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставить в запускаемой задаче</i> | 120 |

3.4.12.3 Трансформация данных с указанными параметрами

transformWithCustomConfig

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

3.4.12.3.1 Описание задачи

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

3.4.12.3.2 Конфигурация задачи

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

| Параметр | Описание | Значение по умолчанию |
|--------------------------|--|-----------------------------------|
| poolSize | Размер пула потоков для стандартизации | кол-во доступных серверов фактора |
| pageSize | Число записей в пачке, отдаваемой одному потоку обработки на Факторе | 1000 |
| maxStoredErrors | Максимальное кол-во ошибок для хранения в памяти | 1000 |
| cleanMapping | Мэппинг для очистки, если надо использовать отдельный (неумолчательный) мэппинг | |
| customOrigin | Перекрытие стандартного поведения, для каких источников запускать задачу трансформации | |
| closeMissedAttributes | Флаг необходимости закрывать отсутствующие атрибуты | |
| cacheSize | Размер кеша страниц | 1 |
| iteratorRefreshPeriodMin | Время обновления итератора по staging <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставлять в запускаемой задаче</i> | 120 |
| notifyListeners | Оповещение поискового индекса КА о трансформации пачек Если оповещение отключено, после завершения задачи нужно перестроить индекс <i>*При запуске трансформации в составе другой задачи, например загрузки данных, параметр нужно выставлять в запускаемой задаче</i> | true |

3.4.13 Активировать исходные записи

enableStagingRecords

3.4.13.1 Описание задачи

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

3.4.13.2 Конфигурация задачи

| Параметр | Описание | Пример |
|-------------|--|-----------|
| stagingHids | Список hid_staging записей, которые необходимо игнорировать. | 1, 4, 100 |

| Параметр | Описание | Пример | | | | | | | | | | | | |
|---|------------------------------------|----------------|--------------------|----------|----------------|-------------|------------|--|--|--|--|--|--|--|
| | Значения указываются через запятую | | | | | | | | | | | | | |
| <p>Группа import Название enableStagingRecords</p> <p>Параметры задачи</p> <table border="1"> <thead> <tr> <th>Название параметра</th> <th>Значение</th> <th>Обязательность</th> </tr> </thead> <tbody> <tr> <td>stagingHids</td> <td>1,2,3,1000</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>Сохранить Отмена</p> | | | Название параметра | Значение | Обязательность | stagingHids | 1,2,3,1000 | | | | | | | |
| Название параметра | Значение | Обязательность | | | | | | | | | | | | |
| stagingHids | 1,2,3,1000 | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |

Для получения списка записей, игнорируемых ранее, необходимо выполнить запрос

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

3.4.13.3 Технические детали

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

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

3.4.14 Деактивировать исходные записи

disableStagingRecords

- [Описание задачи](#)
- [Конфигурация задачи](#)
- [Технические детали](#)

3.4.14.1 Описание задачи

Задача нужна для временной приостановки обработки записей.

Например, в буферной таблице прислали неправильное значение и *система «Гражданский фактор»* 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](#).

3.4.14.2 Конфигурация задачи

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

Группа import
Название disableStagingRecords

Параметры задачи

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

Сохранить Отмена

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

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

3.4.14.3 Технические детали

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

Чтобы посмотреть список деактивированных записей, достаточно выполнить запрос:

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

3.4.15 Задача циклической загрузки данных из БД

Возобновляемая

Конфликтующая

CyclicIncrementLoad

3.4.15.1 Описание задачи

Фактически эта задача являет собой циклическое исполнение следующей последовательности:

1. Отработка 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 |
|--------------------|---------------------|--|
| terminateOnErrors | Логическое значение | Флаг, указывающий что задачу нужно остановить при возникновении ошибок при обработке данных. |
| maxIncrementsCount | Целое | Максимальное количество инкрементов, загружаемых за одно выполнение задачи. |
| executionTimeout | Целое | Максимальное время выполнения задачи в минутах. При превышении задача останавливается. |

3.5 Идентификация и слияние дубликатов

3.5.1 Идентификация и слияние дубликатов

Группа задач, предназначенных для идентификации и слияния дубликатов:

- findDuplicates — поиск дубликатов по всей базе контрагентов;
- merge — слияние гарантированных и подтвержденных дубликатов;
- findDuplicatesAndMerge — поиск дубликатов по всей базе контрагентов с последующим слиянием гарантированных дубликатов;
- findDuplicatesAndMergeAndRebuildGroups — поиск дубликатов по всей базе контрагентов, слияние гарантированных и перестроение индекса негарантированных групп дублей;
- incrementDuplicatesAndMerge — инкрементальный поиск дубликатов (дубликаты ищутся среди контрагентов, по которым были изменения после предыдущего поиска дубликатов) с последующим слиянием гарантированных дубликатов;
- importExternalDuplicateWithMerge — слияние дубликатов по внешнему списку;
- rebuildFullDedupHashes — перестроение хэшей полного поиска дубликатов;
- findGuaranteedDuplicatesAndMerge — поиск гарантированных дубликатов и слияние;
- egrCheck — полная проверка по ЕГРЮЛ;

- `egrCheckActualize` — Проверка по ЕГРЮЛ контрагентов, требующих актуализации.

3.5.2 `duplicateGroupActualize`. Актуализация групп дубликатов

Задача прерываемая, конфликтная, возобновляемая

3.5.2.1 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.

3.5.2.1.1 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 записей в блоке "похожих". Для перестроения всех групп дубликатов без ограничений, необходимо выполнить триггер `fullRefreshDuplicateGroupTrigger`.

3.5.3 `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 |
|--|---------------------------------------|
| duplicateGroupFullMarkup | Полная разметка групп дубликатов |
| duplicateGroupIndexRebuild | Перестроение индекса групп дубликатов |

3.5.3.1 `duplicateGroupFullMarkup`. Полная разметка групп дубликатов

Невозобновляемая, конфликтная.

3.5.3.1.1 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.

3.5.3.1.1.1 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.

3.5.3.1.1.2 Логирование процесса выполнения задачи

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

Сохранение данных в результирующие таблицы пачками с промежуточными коммитами. В одной пачке `pageSize` групп. Также логгируется, 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...
```

3.5.3.2 `duplicateGroupIndexRebuild`. Перестроение индекса групп дубликатов

Задача прерываемая, конфликтная, возобновляемая

3.5.3.2.1 Перестроение индекса групп дубликатов — задача `duplicateGroupIndexRebuild`
Для полного перестроения индекса групп дубликатов необходимо в АРМ выполнить задачу `duplicateGroupIndexRebuild`.

3.5.3.2.1.1 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.

3.5.4 Закрытие дублирующих дубликатов

`closeDuplicatedDuplicates`

Возобновляемая, неконфликтная

3.5.4.1 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

3.5.4.2 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 |

3.5.4.3 Конфигурация задачи

Конфигурировать задачу не надо.

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

3.5.5 Инкрементальный поиск дубликатов

incrementDuplicates

Невозобновляемая, конфликтная, прерываемая

3.5.5.1 Описание задачи

Задача инициирует инкрементальный поиск дубликатов на Факторе.

3.5.5.2 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|-----------|--|-----------------------|
| partyType | Тип сущности, для которого будет запущена задача (указывать несколько нельзя). | |

| Параметр | Описание | Значение по умолчанию |
|------------------------|---|-----------------------|
| | Если не задан, запускается для всех типов. | |
| keepDeDuplicationFiles | <p>Флаг, указывающий на необходимость сохранить (в рабочую папку dedup) временные файлы:</p> <ul style="list-style-type: none"> • out_PHYSICAL — найденные инкрементальные дубликаты; • ids_PHYSICAL — хиды проверенных карточек. | false |

3.5.6 Кросс-сверка ФЛ-ЮЛ

findCrossRelationsPhysicalAndLegal

3.5.6.1 Описание задачи

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

Если совпадение найдено — между записями проставляется связь.

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

3.5.6.2 Конфигурация задачи

Конфигурировать задачу не надо.

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

3.5.6.3 Перестроение хэшей кросс-сверки ФЛ-ЮЛ

rebuildCrossRelationsPhysicalAndLegalHashes

3.5.6.3.1 Описание задачи

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

3.5.6.3.2 Конфигурация задачи

Конфигурировать задачу не надо.

3.5.7 Кросс-сверка ФЛ и проспектов ФЛ

findCrossRelationsPhysicalAndProspects

[Непрерываемая](#)

3.5.7.1 Описание задачи

Задача ищет дубликаты между ФЛ и проспектами по набору правил.

Если совпадение найдено — между записями проставляется связь.

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

3.5.7.2 Конфигурация задачи

Конфигурировать задачу не надо.

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

3.5.8 Перестроение хэшей полного поиска дубликатов

```
rebuildFullDedupHashes
```

3.5.8.1 Описание задачи

Техническая задача, используется по инструкции сотрудников службы сопровождения вендора решения или после того, как развернут бэкап с прода.

Технические подробности, не видимые заказчику

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

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

Полученные хэши используются для инкрементального дедапа и запросов fuzzySearch

3.5.8.2 Конфигурация задачи

Настраиваемые параметры:

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

3.5.9 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|-----------|---------------------------------|-----------------------|
| partyType | Тип контрагента | PHYSICAL_PARTY |

3.5.10 Слияние дубликатов по внешнему списку

3.5.10.1 Общее описание

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

3.5.10.2 Механизм

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

Функция возвращает курсор на выборку в следующем формате:

| Поле | Описание |
|--------------------------|---|
| <code>hid_2</code> | HID второй PARTY |
| <code>hid_1</code> | HID 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.
- Сущности 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.

3.5.10.3 Примечание

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

Если задачу выполняете в первый раз, то перед запуском обратитесь в службу сопровождения вендора решения.

3.5.10.4 Механизм загрузки дубликатов

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

Функция возвращает курсор на выборку в следующем формате:

| Поле | Описание |
|--------------------------|---|
| <code>hid_2</code> | HID второй PARTY |
| <code>hid_1</code> | HID 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.
- Сущности 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.

3.6 Повторная обработка

3.6.1 Повторная обработка контрагентов

- [Задача перерасчета hidden-флагов атрибутов по списку записей](#)
- [Задача полного переслияния записей](#)
- [Задача переслияния реквизитов контрагентов](#)
- [Разлияние контрагентов по списку MergedId](#)
- [Полное разлияние записей по списку результирующих карточек](#)
- [Задача восстановления атрибутов по списку HIDов](#)
- [Задача актуализации адресов по ФИАС](#)
- [Исправьте базу данных в случае неконсистентного состояния](#)
- [Переоткрытие закрытых по ошибке контрагентов с сохранением истории](#)
- [Актуализация атрибутов на основе данных staging](#)
- [Закрытие актуальных \(результирующих и всех исходных\) атрибутов по списку](#)
- [Задача очистки таблиц параметров для массовой переобработки данных](#)
- [Подготовка списков для повторной обработки](#)
- [Таблицы с параметрами задач массовой переобработки](#)
- [Актуализация статуса невалидности документов](#)
- [Закрытие исходных атрибутов по списку с переслиянием результирующих](#)
- [Переобработка телефонов](#)
- [Переобработка телефонов. Второй вариант](#)
- [Задача восстановления записи на версию по списку HID](#)
- [Задача переочистки реквизитов](#)
- [Задача переобработки UNDEF-телефонов](#)
- [Задача пересчета скоринговых значений атрибутов](#)
- [Переочистка гранулярных кодов качества ФИО](#)
- [Полная переочистка ФИО](#)
- [Заполнение ФИО ИП](#)
- [Заполнение полной ОПФ](#)
- [Переочистка кодов качества паспортов](#)
- [Переочистка кодов качества дат рождения](#)

- [Задача переоткрытия связей](#)
- [Переобработка email](#)
- [Переочистка email](#)
- [Задача заполнения технических полей](#)
- [Задача поиска массивных карточек](#)
- [Задача обработки массивных карточек контрагентов](#)
- [Задача поиска золотых карточек для переслияния](#)
- [Задача переноса открытых связей закрытых транзитивных карточек на активные золотые](#)

[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 hiddenFlagRefineByHids | Задача перерасчета hidden-флагов атрибутов по списку контрагентов | | |
| 2 reMergeFullByPartyHids | Задача полного переслияния контрагентов | | |
| 3 reMergeRequisiteByPartyHids | Задача переслияния реквизитов контрагентов | | |
| 4 unMergeByMergedIds | Разлияние контрагентов по списку MergedId | | |
| 5 fullUnmergeByResultHids | Полное разлияние контрагентов по списку результирующих контрагентов | | |
| 6 attributeRecoverByHids | Задача восстановления атрибутов по списку HIDs | | |
| 7 addressActualization | Задача актуализации адресов по ФИАС | | |
| 8 mergedPartiesCheckAndRmerge | Исправьте базу данных в | | |

| Задача | Описание | Действия перед выполнением задачи | Действия после выполнения задачи |
|--|--|-----------------------------------|----------------------------------|
| | случае неконсистентного состояния | | |
| 9 partyReopenByHids | Переоткрытие закрытых по ошибке контрагентов с сохранением истории | | |
| 10 attributeActualizeByStaging | Актуализация атрибутов на основе данных staging | | |
| 11 attributeCloseActualByHids | Закрытие актуальных (результатирующих и всех исходных) атрибутов по списку | | |
| 12 cleanTaskParameterTables | Задача очистки таблиц параметров для массовой переобработки данных | | |
| 13 fillTaskPartyHidsActual fillTaskPartyHidsOriginal fillTaskPartyHidsResulted fillAddressActualizationTaskPartyHids fillTaskPartyHidsBadAdresses fillTaskPartyHidsOriginalWithClosed | Подготовка списков для повторной обработки | | |
| 14 passportInvalidStatusReCleaner | Актуализация статуса невалидности документов | | |
| 15 attributeCloseSourceByHids | Закрытие исходных атрибутов по списку с переслиянием результирующих | | |

| | Задача | Описание | Действия перед выполнением задачи | Действия после выполнения задачи |
|----|---|---|--|---|
| 16 | phoneExtraFieldsReCleaner | Переобработка телефонов | | |
| 17 | phoneExtraFieldsActualization.physical phoneExtraFieldsActualization.legal | Переобработка телефонов. Новый вариант | | |
| 18 | partyRecoverByHids | Задача восстановления КА на версию по списку HID | | |
| 19 | reCleanMockRequisiteTask | Задача переочистки реквизитов | | |
| 20 | phoneUndefReCleaner | Задача переобработки UNDEF-телефонов | | |
| 21 | refreshScoresTask | Задача пересчета скоринговых значений атрибутов | | |
| 22 | physicalFioQcReCleaner | Переочистка гранулярных кодов качества ФИО | Заполнение таблицы актуальными КА | Удаление таблицы по завершению задачи |
| 23 | passportQcReCleaner | Переочистка кодов качества паспортов | | |
| 24 | physicalBirthdateQcReCleaner | Переочистка кодов качества дат рождения | Заполнение таблицы актуальными КА | Удаление таблицы по завершению задачи |
| 25 | relationReopenByIds | Задача переоткрытия связей | | |
| 26 | legalIpReCleaner | Заполнение ФИО ИП | | |

3.6.2 Задача перерасчета hidden-флагов атрибутов по списку записей

| |
|------------------------|
| hiddenFlagRefineByHids |
|------------------------|

Возобновляемая

3.6.2.1 Описание задачи

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

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

Работает по оригинальным карточкам.

3.6.2.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу `cleanTaskParameterTables` для очистки таблицы `task_party_hids`;
2. Выяснить идентификаторы записей (`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
;
```

3.6.2.3 Конфигурация задачи

Настраиваемые параметры:

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

Группа reprocess
Название hiddenFlagRefineByHids

Параметры задачи

| Название параметра | Значение | Обязательность |
|--|---------------------------------------|----------------|
| <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"/> | |

3.6.3 Задача полного переслияния записей

reMergeFullByPartyHids

[Возобновляемая](#), Конфликтная

3.6.3.1 Описание задачи

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

Для атрибутов выбираются все атрибуты исходных записей 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.

3.6.3.2 Запуск задачи

Просто запустить задачу reMergeFullByPartyHids, 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 для полного переслияния объединенных записей:

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

commit;
```

3.6.3.3 Конфигурация задачи
 Настраиваемые параметры:

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

Группа reprocess
 Название reMergeFullByPartyHids

Параметры задачи

| Название параметра | Значение | Обязательность |
|--------------------|----------|----------------|
| poolSize | 8 | |
| pageSize | 1000 | |
| partyType | PHYSICAL | |
| | | |
| | | |

Сохранить Отмена

3.6.4 Задача переслияния реквизитов контрагентов

reMergeRequisiteByPartyHids

[Возобновляемая](#)

3.6.4.1 Описание задачи

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

Задача включает в себя переслияние только реквизитов контрагентов.

3.6.4.2 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 ...
;
```

3.6.4.3 Конфигурация задачи

Настраиваемые параметры:

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

Группа reprocess
Название reMergeRequisiteByPartyHids

Параметры задачи

| Название параметра | Значение | Обязательность |
|--------------------|----------|----------------|
| poolSize | 8 | |
| pageSize | 1000 | |
| partyType | PHYSICAL | |
| | | |
| | | |

Сохранить Отмена

3.6.5 Разливание контрагентов по списку MergedIds

unMergeByMergedIds

[Возобновляемая](#)

3.6.5.1 Описание задачи

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

3.6.5.2 Запуск задачи

Перед запуском задачи необходимо:

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

Пример:

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

3.6.5.3 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|------------------------------------|---|-----------------------|
| <code>partyType</code> | Тип контрагента , для которых выполняется разлияние | PHYSICAL |
| <code>unmergeDescription</code> | Комментарий к задаче (например, причина, по которой выполняется разлияние) | |
| <code>poolSize</code> | Максимальное количество потоков | 1 |
| <code>pageSize</code> | Размер пачки | 1000 |
| <code>massiveChanging</code> | Оповещение Фактора изменениями. Если задать <code>true</code> , то Фактор не будет вызываться с инкрементальными <code>populate</code> | false |
| <code>preventFurtherMerging</code> | Запретить разлитым контрагентам сливаться в дальнейшем (проставить им Confirmation = 4) | false |

Группа reprocess
Название unMergeByMergedIds

Параметры задачи

| Название параметра | Значение | Обязательность |
|---|--|----------------|
| <input type="text" value="poolSize"/> | <input type="text" value="1"/> | |
| <input type="text" value="pageSize"/> | <input type="text" value="1000"/> | |
| <input type="text" value="unmergeDescription"/> | <input type="text" value="Unmerge description"/> | |
| <input type="text" value="partyType"/> | <input type="text" value="PHYSICAL"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

3.6.5.4 Выполнение задачи

Задача запускается из АРМ Администратора.

3.6.6 Полное разливание записей по списку результирующих карточек

fullUnmergeByResultHids

3.6.6.1 Описание задачи

Золотые карточки разливаются целиком, без промежуточных версий — экономит место в БД.

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

3.6.6.2 Запуск задачи

Перед запуском задачи необходимо:

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

Пример:

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

3.6.6.3 Конфигурация задачи

Настраиваемые параметры:

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

Группа reprocess

Название fullUnmergeByResultHids

Параметры задачи

| Название параметра | Значение | Обязательность |
|---|--|----------------|
| <input type="text" value="unmergePackLimit"/> | <input type="text" value="1000"/> | |
| <input type="text" value="poolSize"/> | <input type="text" value="4"/> | |
| <input type="text" value="pageSize"/> | <input type="text" value="50"/> | |
| <input type="text" value="unmergeDescription"/> | <input type="text" value="Unmerge description"/> | |
| <input type="text" value="partyType"/> | <input type="text" value="PHYSICAL"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

3.6.7 Задача восстановления атрибутов по списку HIDов

attributeRecoverByHids

3.6.7.1 Описание задачи

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

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

3.6.7.2 Особенности задачи

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

Изменений в связях атрибутов не происходит.

3.6.7.3 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 системе «Гражданский фактор»;
 - b. тип и идентификатор контрагента в системе «Гражданский фактор»;
 - 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 ...
```

3.6.8 Задача актуализации адресов по ФИАС

addressActualization

3.6.8.1 Описание задачи

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

3.6.8.2 Алгоритм работы задачи

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

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

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

3.6.8.2.1 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.

3.6.8.2.2 Повторное распознавание

На стандартизацию отправляется содержимое служебного поля «исходный адрес после препроцессинга».

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

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

3.6.8.3 Особенности задачи

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

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

Для инициализации списка контрагентов на обработку необходимо выполнить задачу fillAddressActualizationTaskPartyHids.

Для обработки своего списка заполнить таблицу TASK_ACT_PARTY_HIDS вручную.

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

3.6.8.4 Конфигурация задачи

Настраиваемые параметры:

| Имя | Описание | Значение по умолчанию |
|--------------------------|--|-----------------------|
| executionTimeout | Максимальное время выполнения задачи в минутах. При превышении задача останавливается. | 0 |
| partyType | Тип контрагента | PHYSICAL |
| pageSize | Размер одной страницы данных (увеличивает время блокировок БД) | 0 |
| poolSize | Количество потоков, которые будут обрабатывать страницы с данными | 0 |
| maxStoredErrors | Количество ошибок, которое можно накапливать в памяти | 0 |
| iteratorRefreshPeriodMin | Как часто (в минутах) переоткрывать итератор по данным, чтобы избежать переполнения UNDO | 0 |
| terminateOnErrors | Параметр, указывающий что задачу нужно остановить при возникновении N ошибок | false |
| massiveChanging | Признак массового изменения - сигнал к отключению оповещения | false |

| Имя | Описание | Значение по умолчанию |
|-----------------|--|-----------------------|
| | инкрементального дедапа и, возможно, др. слушателей | |
| cacheSize | Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память) | 0 |
| notifyListeners | Флаг необходимости оповещать слушателей | false |

3.6.9 Исправьте базу данных в случае неконсистентного состояния

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

3.6.10 Переоткрытие закрытых по ошибке контрагентов с сохранением истории

`partyReopenByHids`

[Возобновляемая](#)

3.6.10.1 Особенности работы

Контрагент восстанавливается на последнюю перед закрытием версию, при этом:

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

3.6.10.2 Запуск задачи

Перед запуском задачи необходимо:

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

Пример:

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

3.6.10.3 Конфигурация задачи

Настраиваемые параметры:

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

Группа reprocess
Название partyReopenByHids

Параметры задачи

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

Сохранить Отмена

3.6.11 Актуализация атрибутов на основе данных staging

attributeActualizeByStaging

3.6.11.1 Описание задачи

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

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

3.6.11.2 Алгоритм работы задачи

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

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

3.6.11.3 Конфигурация задачи

Настраиваемые параметры:

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

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

3.6.11.4.1 1. Заполнить список к переобработке HIDами контрагентов

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

3.6.11.4.2 2. Запустить переочистку адресов

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

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

3.6.11.5.1 1. Заполнить список к переобработке HIDами контрагентов

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

3.6.11.5.2 2. Запустить переочистку адресов

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

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

| |
|----------------------------|
| attributeCloseActualByHids |
|----------------------------|

3.6.12.1 Описание задачи

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

3.6.12.2 Запуск задачи

1. Выполнить задачу `cleanTaskParameterTables`. Она очистит таблицу `task_attribute_hids`.
2. Подготовить список атрибутов для закрытия. Для каждого атрибута выяснить:
 - a. Тип и идентификатор атрибута в системе «Гражданский фактор»;
 - b. Тип и идентификатор сущности в системе «Гражданский фактор».
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`.

3.6.12.3 Особенности задачи

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

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.

3.6.13 Задача очистки таблиц параметров для массовой переобработки данных

`cleanTaskParameterTables`

3.6.13.1 Описание задачи

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

В результате работы задачи будут очищены [таблицы с параметрами задач массовой переобработки](#) (кроме `task_remerge_hids` и `task_engine_actualize_hids`).

3.6.14 Подготовка списков для повторной обработки

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

fillTaskPartyHidsActual, fillTaskPartyHidsOriginal, fillTaskPartyHidsResulted,
fillAddressActualizationTaskPartyHids, fillTaskPartyHidsBadAdresses,
cleanTaskParameterTables, fillTaskPartyHidsOriginalWithClosed

Данные задачи следует выполнять перед запуском задачи повторной обработки выборочных записей.

В результате работы задачи таблица TASK_PARTY_HIDS будет наполнена идентификаторами записей:

- fillTaskPartyHidsActual — все актуальные записи (merged_status 0 или 2);
- fillTaskPartyHidsOriginal — все оригинальные записи (merged_status 0 или 1);
- fillTaskPartyHidsResulted — все результирующие записи (merged_status 2);
- fillTaskPartyHidsOriginalWithClosed — все оригинальные записи (merged_status 0 или 1) и закрытые записи (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) — карточки, адреса из которых должны быть актуализированы по КЛАДР (mergedStatus: 0,2);
- fillTaskPartyHidsBadAdresses — карточки с адресами, отмеченными на ручную обработку (mergedStatus 0 или 2), задача не выбирает карточки, если все адреса в ней, отмеченные на ручную обработку, скрыты.

3.6.15 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](#)

3.6.15.1 Описание таблиц

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

3.6.15.1.1 Таблица task_party_hids

| № | Поле | Описание |
|---|------------------|--|
| 1 | HID_PARTY | Уникальный идентификатор записи в системе «Гражданский фактор» |
| 2 | PARTY_TYPE | Тип контрагента (литера) |
| 3 | HID_RESULT_PARTY | Уникальный идентификатор результирующего контрагента (для более эффективной группировки при обработке) |

3.6.15.1.2 Таблица task_merged_ids

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

| № | Поле | Описание |
|---|-----------|--------------------------|
| 1 | ID_MERGED | Идентификатор id_merged. |

3.6.15.1.3 Таблица task_act_party_hids

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

| № | Поле | Описание |
|---|-----------|--|
| 1 | HID_PARTY | Уникальный идентификатор записи в системе «Гражданский фактор» |

3.6.15.1.4 Таблица task_attribute_hids

| № | Поле | Описание |
|---|----------------|---|
| 1 | ATTRIBUTE_TYPE | Тип атрибута (литера) |
| 2 | HID_ATTRIBUTE | Уникальный идентификатор атрибута в системе «Гражданский фактор» |
| 3 | PARTY_TYPE | Тип контрагента (литера) |
| 4 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 5 | TARGET_DATE | Дата и время, на которое нужно восстановить атрибут |

3.6.15.1.5 Таблица delete_party_hids

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

| № | Поле | Описание |
|---|------------|---|
| 1 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 2 | PARTY_TYPE | Тип контрагента (литера) |

3.6.15.1.6 Таблица close_party_hids

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

| № | Поле | Описание |
|---|------------|---|
| 1 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 2 | PARTY_TYPE | Тип контрагента (литера) |

3.6.15.1.7 Таблица task_doubtful_reclean_hids

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

| № | Поле | Описание |
|---|------------|---|
| 1 | PARTY_TYPE | Тип контрагента (литера) |
| 2 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 3 | TYPES | Тип атрибута (литера) (через запятую когда несколько) |

3.6.15.1.8 Таблица task_phone_party_hids

Таблица, используемая задачей [phoneExtraFieldsActualization](#) для актуализации дополнительных полей телефонов.

| № | Поле | Описание |
|---|------------|---|
| 1 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 2 | PARTY_TYPE | Тип контрагента (литера) |

3.6.15.1.9 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 переслияния](#).

| № | Поле | Описание |
|---|------------|---|
| 1 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 2 | PARTY_TYPE | Тип контрагента (литера) |

3.6.15.1.10 Таблица task_engine_actualize_hids

Используется задачей [searchEngineActualizeByParties](#) для актуализации поискового индекса по списку физ. лиц.

Заполняется задачей [searchEngineValidateByIds](#) при выявлении записи с расхождениями между БД и индексом.

| | | |
|---|------------|---|
| 1 | HID_PARTY | Уникальный идентификатор контрагента в системе «Гражданский фактор» |
| 2 | PARTY_TYPE | Тип контрагента (литера) |

3.6.16 Актуализация статуса невалидности документов

passportInvalidStatusReCleaner

[Возобновляемая](#)

3.6.16.1 Описание задачи

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

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

3.6.16.1.1 Какие проверки для каких типов документов проводятся

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

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

Если поле `EXPIRY_DATE` не заполнено производится проверка по дате выдачи:

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

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

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

3.6.16.1.2 Актуализация признака недействительности у документов

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

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

3.6.16.1.3 Перепроверка всех документов с постановкой 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.

3.6.16.1.4 Настройка периодического запуска для задачи актуализации признака недействительности

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

Задача переобработки может выполняться одновременно с другими задачами.

3.6.16.2 Запуск задачи по отдельным записям

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

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

3.6.16.3 Конфигурация задачи

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

3.6.17 Закрытие исходных атрибутов по списку с переслиянием результирующих

`attributeCloseSourceByHids`

3.6.17.1 Описание задачи

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

3.6.17.2 Запуск задачи

1. Выполнить задачу `cleanTaskParameterTables`. Она очистит таблицу `task_attribute_hids`.
2. Подготовить список атрибутов для закрытия. Для каждого атрибута выяснить:
 - a. Тип и идентификатор атрибута в системе «Гражданский фактор»;
 - b. Тип и идентификатор контрагента в системе «Гражданский фактор».
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`.

3.6.17.3 Особенности задачи

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

Игнорирует атрибуты, которые:

- закрыты;
- не найдены по идентификатору.

Проигнорированные атрибуты задача пишет в лог-файл.

Если атрибут находится в закрытой записи или относится к «золотым» карточкам, задача упадет с ошибкой

3.6.18 Переобработка телефонов

`phoneExtraFieldsReCleaner`

[Возобновляемая](#)

3.6.18.1 Описание задачи

Задача используется для обновления дополнительной информации по телефонам.

Например:

- изменились часовые пояса и нужно переобработать текущие;
- физ. лицо сменило оператора связи, а в системе «Гражданский фактор» остался старый.

Можно выборочно переобработать одну или несколько записей, а также создать триггер на периодическую актуализацию всей базы данных.

3.6.18.2 Выборочная переобработка

Перед запуском задачи необходимо:

1. Выполнить задачу `cleanTaskParameterTables` для очистки таблицы `task_party_hids` от существующих записей;
2. Выяснить идентификаторы исходных контрагентов (`hid_party`), для которых необходимо выполнить актуализацию данных;
3. Добавить список данных идентификаторов в таблицу `task_party_hids` вручную или с помощью вызова одной из [задач](#).
Пример ручного наполнения таблицы:

```

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

-- Нужно переобработать набор записей. 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.

3.6.18.3 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.

3.6.18.4 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|-----------------|---|-----------------------|
| partyType | Тип сущностей, для которых выполняется переслияние | PHYSICAL |
| massiveChanging | Оповещение Фактора инкрементальными изменениями. При массовой переобработке нужно задать true, а потом запустить полный поиск дубликатов | false |
| hidsTableName | Название таблицы для заполнения идентификаторами сущностей, по умолчанию пусто | |

3.6.18.5 Особенности задачи

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

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

- относятся к влитым записям;
- 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.

3.6.19 Переобработка телефонов. Второй вариант

```
phoneExtraFieldsActualization.physical  
phoneExtraFieldsActualization.legal
```

[Возобновляемая](#)

3.6.19.1 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;
- физ. лицо сменило оператора связи, а в системе «Гражданский фактор» 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 системе «Гражданский фактор» (в таком случае изменение доп. полей роли не играет)

3.6.19.2 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|------------------------|--|-----------------------|
| <code>partyType</code> | Тип сущностей, для которых выполняется переслияние | PHYSICAL |

3.6.19.3 Особенности задачи

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

Игнорирует идентификаторы (без ошибки в лог), которые:

- относятся к «золотым» записям;
- 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.

3.6.20 Задача восстановления записи на версию по списку HID

partyRecoverByHids

[Возобновляемая](#)

3.6.20.1 Описание задачи

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

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

3.6.20.2 Особенности работы

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

- если запись не найдена, идентификатор удаляется из таблицы `task_party_hids`, задача продолжает работу;
- если в указанное время запись не существовала, идентификатор удаляется из таблицы `task_party_hids`, задача продолжает работу;
- если запись закрыта или влита в золотую, задача прервется с ошибкой;
- изменений в связях не происходит;
- атрибуты восстанавливаются.

3.6.20.3 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу `cleanTaskParameterTables` для очистки таблицы `task_party_hids`;
2. Добавить идентификаторы записей `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

3.6.20.4 Конфигурация задачи

Настраиваемые параметры:

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

Формат даты — 10.06.2018 00:00:03.

3.6.21 Задача переочистки реквизитов

reCleanMockRequisiteTask

[Возобновляемая](#)

3.6.21.1 Описание задачи

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

3.6.21.2 Особенности задачи

Работает через провайдер для получения всех оригинальных контрагентов.

Переочищенные реквизиты обновляются принудительно.

3.6.21.3 Конфигурация задачи

Настраиваемые параметры:

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

3.6.21.4 Логирование

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

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

В итоговой статистике выводится:

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

3.6.22 Задача переобработки UNDEF-телефонов

phoneUndefReCleaner

[Возобновляемая](#)

3.6.22.1 Описание задачи

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

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

3.6.22.2 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.

3.6.22.3 Конфигурация задачи

Настраиваемые параметры:

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

3.6.23 Задача пересчета скоринговых значений атрибутов

refreshScoresTask

3.6.23.1 Запуск задачи

1. Выполнить задачу `fillTaskPartyHidsResulted` для наполнения таблицы `task_party_hids` идентификаторами КА для переобработки.
2. Запустить задачу `refreshScoresTask` для пересчета скоринга.

3.6.23.2 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;
```

3.6.24 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 переобработке основного кода качества ФИО: если изначальный основной код качества ФИО «Подтвержден контролером», задача его не меняет.

3.6.24.1 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.

3.6.24.2 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|-----------------------|---------------------------------|-----------------------|
| <code>poolSize</code> | Максимальное количество потоков | 4 |
| <code>pageSize</code> | Размер пачки | 1000 |

| Параметр | Описание | Значение по умолчанию |
|-------------------|--|-----------------------|
| fillTaskPartyHids | Нужно ли предварительно заполнить таблицу hid-ами контрагентов. Первичное заполнение не выполняется, если в таблице что-то есть. | false |
| hidsTableName | Название таблицы для заполнения | task_party_hids |

3.6.24.3 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
```

3.6.25 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.

3.6.25.1 Запуск задачи

3.6.25.1.1 Переобработка всех актуальных ФИО

Запустите задачу с параметрами:

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

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

3.6.25.1.2 Переобработка заданных ФИО

Наполните заранее таблицу хидами записей, ФИО из которых необходимо переобработать. Можно использовать, например, таблицу TASK_ACT_PARTY_HIDS.

Запустите задачу с параметрами:

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

Задача переобработает только ФИО из указанных записей.

В режиме горячего резерва запускать только на одной ноде.

3.6.25.2 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
```

3.6.26 Заполнение ФИО ИП

```
legalIpReCleaner
```

3.6.26.1 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).

3.6.26.2 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:
 - o 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)
```

- o 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 `legalIpReCleaner`.

3.6.27 Заполнение полной ОПФ

```
legalOpfFullCleaner
```

Возобновляемая, неконфликтная

3.6.27.1 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.

3.6.27.2 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.

3.6.27.3 Параметры задачи

| Параметр | Описание | Значение по умолчанию |
|----------------------|--|-----------------------|
| skipJmsEventsPublish | Не отправлять события в обратный поток | true |

3.6.28 Переочистка кодов качества паспортов

passportQCReCleaner

[Возобновляемая](#), неконфликтная

3.6.28.1 Описание задачи

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

3.6.28.2 Особенности задачи

Задача работает по списку `task_party_hids`, обновление атрибутов принудительное.

Документы подтвержденные контролером на переочистку не отправляются.

3.6.28.3 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу `cleanTaskParameterTables` для очистки таблицы `task_party_hids`;
2. Выполнить задачу `fillTaskPartyHidsActual`, указав в параметре `procedureParams = PHYSICAL`;

После этого запустить задачу `passportQCReCleaner`.

В режиме горячего резерва запускать только на одной ноде.

3.6.28.4 Конфигурация задачи

Настраиваемые параметры:

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

3.6.28.5 Переочистка кодов качества EDITED паспортов

3.6.28.5.1 Запуск задачи

Задача работает по списку записей, 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 записей;

После этого можно выполнять задачу `passportQCRCleaner.edited`

3.6.28.5.2 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|------------------------|--|-----------------------|
| <code>partyType</code> | Тип сущности , для которой выполняется переобработка | PHYSICAL |

3.6.28.6 Переочистка кодов качества EMPTY_DATE паспортов

3.6.28.6.1 Запуск задачи

Задача работает по списку записей, 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 записей;

После этого можно выполнять задачу `passportQCRCleaner.emptyDate`

3.6.28.6.2 Конфигурация задачи

Настраиваемые параметры:

| | | |
|------------------------|--|----------|
| <code>partyType</code> | Типы сущностей , для которых выполняется переобработка | PHYSICAL |
|------------------------|--|----------|

3.6.28.7 Переочистка кодов качества NOT_SURE паспортов

3.6.28.7.1 Запуск задачи

Задача работает по списку записей, 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 записей;

После этого можно выполнять задачу `passportQCRCleaner.notSure`

3.6.28.7.2 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|------------------------|---|-----------------------|
| <code>partyType</code> | Тип сущности, для которых выполняется переобработка | PHYSICAL |

3.6.29 Переочистка кодов качества дат рождения

physicalBirthdateQcReCleaner

[Возобновляемая](#), неконфликтная

3.6.29.1 Описание задачи

В системе «Гражданский фактор» предусмотрены [коды 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.

3.6.29.2 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.

3.6.29.3 Конфигурация задачи

| Параметр | Описание | Значение по умолчанию |
|-------------------|---|-----------------------|
| partyType | Тип сущности , для которого выполняется переобработка | PHYSICAL |
| fillTaskPartyHids | Автоматическое заполнение временной таблицы хидами сущностей | true |

3.6.30 Задача переоткрытия связей

relationReopenByIds

[Возобновляемая](#)

3.6.30.1 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;
```

3.6.30.2 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.

3.6.30.3 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|----------|---------------------------------|-----------------------|
| poolSize | Максимальное количество потоков | 1 |

3.6.31 Переобработка email

emailNotValidReCleaner

[Возобновляемая](#), неконфликтная

3.6.31.1 Описание задачи

Для email в системе «Гражданский фактор» проставляются маркеры некорректности. Задача создана для того, чтобы переобработать существующий объём базы емейлов с кодом качества NOT_VALID:

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

3.6.31.2 Особенности задачи

1. Задача очищает только емейлы с кодом качества NOT_VALID.
2. Сама создаёт себе таблицу tmp\$rcln_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.

3.6.31.3 Запуск задачи

Нужно запустить задачу emailNotValidReCleaner, дополнительных параметров указывать не надо.

В режиме горячего резерва запускать только на одной ноде.

3.6.31.4 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|-----------------------|--|-----------------------|
| partyType | Тип контрагента , для которого выполняется переобработка | PHYSICAL |
| unsafe.forceExecution | | true |

3.6.32 Переочистка email

emailAllReCleaner

[Возобновляемая](#), неконфликтная

3.6.32.1 Описание задачи

Система «Гражданский фактор» умеет 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

3.6.32.2 Особенности задачи

emailForDedup используется для подсчёта популярности, если заполнено.

3.6.32.3 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_not_valid_hids и заполнит актуальными хидами при запуске. Далее обработка будет выполняться по этой таблице.

3.6.32.4 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|-----------|--|-----------------------|
| partyType | Тип контрагента , для которого выполняется переобработка | PHYSICAL |

3.6.33 Задача заполнения технических полей

extraUtilityFieldsFill

[Возобновляемая](#), конфликтная

3.6.33.1 Описание задачи

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

3.6.33.2 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.

3.6.33.3 Конфигурация задачи

Настраиваемые параметры:

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

3.6.33.4 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;
```

3.6.34 Задача поиска массивных карточек

```
findMassiveParties
```

неконфликтная, непрерываемая

Предназначена для поиска массивных карточек.

Внутри задачи есть параметр `services.party.massive.sourceCountThreshold` (по умолчанию равен 1000 записей).

Этот параметр настраивается в [админке](#), требует рестарта системы «Гражданский фактор».

3.6.34.1 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.

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.

3.6.34.2 Таблица MASSIVE_PARTY_REGISTRY — реестр массивных карточек для отложенной обработки

| Имя | Тип | Описание |
|------------|-------------|--|
| hid_party | bigint | Исторический идентификатор контрагента |
| party_type | varchar(50) | Тип контрагента |
| created | timestamp | Дата слияния |

Обработка массивных карточек делается задачей massivePartyReMergeTask.

3.6.35 Задача обработки массивных карточек контрагентов

massivePartyReMergeTask

конфликтная, прерываемая

Предназначена для обработки массивных контрагентов.

Пересливает все карточки полностью (реквизиты и атрибуты) по хидам из таблиц massive_party_remerge_hids и massive_party_unmerge_ids.

massive_party_remerge_hids — реестр массивных КА для переслияния

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

massive_party_unmerge_ids — реестр массивных КА для разлияния

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

Задача обработки массивных карточек добавлена в ежедневный и еженедельный триггеры.

Данные в таблицы попадают при следующих действиях над массивными карточками:

- разлияние (при стратегии разрешения конфликтов UNMERGE_AND_APPLY)
- закрытие исходной
- обновление исходной

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

3.6.36 Задача поиска золотых карточек для переслияния

```
findPartiesToRemerge
```

[Возобновляемая](#), неконфликтная

3.6.36.1 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`.

3.6.36.2 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;
```

3.6.36.3 Конфигурация задачи

Настраиваемые параметры:

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

3.6.37 Задача переноса открытых связей закрытых транзитивных карточек на активные золотые

`moveTransitionalPartyRelations`

Возобновляемая, неконфликтная

3.6.37.1 Особенности задачи

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

3.6.37.2 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.

3.6.37.3 Параметры задачи

| Параметр | Описание | Значение по умолчанию |
|----------------------|--|-----------------------|
| skipJmsEventsPublish | Не отправлять события в обратный поток | true |

3.7 Системные задачи

3.7.1 Системные задачи

- `changeJdbcUser` — изменение пользователя безопасности

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

- `deleteOutdatedConflicts` — удаление старых обработанных конфликтов
- `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 карточек
- `searchEngineOptimize` — оптимизация поисковых индексов (граф связей, индексы по записям, конфликты), группы дубликатов не оптимизируются
- `searchEngineOptimizeByName` — оптимизация указанных поисковых индексов;
- `searchEngineValidate` — валидация всех поисковых индексов
- `executeDBProcedure` — выполнение хранимой процедуры
- `findHistoricalConflicts` — поиск конфликтных изменений

3.7.2 `allEnginesRebuildByName`. Перестроение указанных поисковых индексов

3.7.2.1 Общее описание

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

3.7.2.2 Конфигурация задачи

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

3.7.2.3 Допустимые значения параметра `enginesToRebuild`

- `PHYSICAL` — физические лица;

- GRAPH — граф связей;
- MANUAL_DUPLICATE_GROUP — группы для ручной проверки дубликатов;
- INDIVIDUAL — индивидуальные предприниматели;
- LEGAL — юридические лица;
- AGREEMENT — договоры;
- EGR — ЕГРЮЛ;
- ALL — все индексы.

3.7.3 Общее описание

Поисковый индекс может быть испорчен (если во время его перестроения возникла ошибка или был остановлен сервер системы «Гражданский фактор»). 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 системы «Гражданский фактор».
- 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`.

3.7.4 Валидация поисковых индексов по идентификаторам

```
searchEngineValidateByIds
```

[Возобновляемая](#)

Конфликтующая

3.7.4.1 Общее описание

Задача валидации поисковых индексов по идентификаторам.

3.7.4.2 Алгоритм работы

1. Идентификаторы из базы данных и индекса собираются в сортированные файлы на диске.

2. Определяется разница — что есть в БД и нет в индексе, и наоборот.
3. 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».
4. Все «кривые» идентификаторы записываются в таблицу `engine_validation_error` — в колонку `id_essence` (отрицательные `id` у графа — это ребра `merged`, положительные — связи).
5. Также хиды «кривых» КА из индексов записываются в таблицу `task_engine_actualize_hids`, по которой потом работает задача актуализации поискового индекса `searchEngineActualizeByParties`.

3.7.4.2.1 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"> • NOT_FOUND_IN_ENGINE • NOT_FOUND_IN_DB |
| 5 | CREATED | Дата создания |
| 6 | NODE_ID | Идентификатор ноды |

3.7.4.3 Конфигурация задачи

| Имя | Тип | Описание | Обязательное | Значение по умолчанию |
|-------------------------------|----------------------|--|--------------|-----------------------|
| <code>executionTimeout</code> | <code>int</code> | Максимальное время выполнения задачи в минутах. При превышении задача останавливается. | | 0 |
| <code>cacheSize</code> | <code>int</code> | Сколько страниц вычитывать в кэш заранее (увеличивает потребляемую память) | | 0 |
| <code>terminateOnError</code> | <code>boolean</code> | Прекратить выполнение при возникновении ошибки? | | <code>false</code> |
| <code>poolSize</code> | <code>int</code> | Количество потоков, которые будут обрабатывать страницы с данными | | 0 |

| Имя | Тип | Описание | Обязательное | Значение по умолчанию |
|--------------------------|---------|---|--------------|-----------------------|
| 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 |

3.7.5 Актуализация поискового индекса по списку контрагентов

searchEngineActualizeByParties

[Возобновляемая](#)

3.7.5.1 Общее описание

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

3.7.5.2 Запуск задачи

1. Очистить таблицу `task_engine_actualize_hids`, выполнив запрос:

```
truncate table task_engine_actualize_hids;
```

2. Добавить идентификаторы контрагентов `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`.

3.7.5.3 Конфигурация задачи

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

Группа `system`
 Название `searchEngineActualizeByParties`

Параметры задачи

| Название параметра | Значение | Обязательность |
|--|---------------------------------------|----------------|
| <input type="text" value="partyType"/> | <input type="text" value="PHYSICAL"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

3.7.6 Выполнение вспомогательной хранимой процедуры

3.7.6.1 Общее описание

Задача `executeDBProcedure` предназначена для запуска произвольной хранимой процедуры.

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

3.7.6.2 Конфигурация задачи

| Параметр | Тип | Описание |
|------------------------------|--------|---|
| <code>procedureName</code> | Строка | Название процедуры |
| <code>procedureParams</code> | Строка | Параметры одной строкой в формате "param1;param2;..." |

3.7.7 Удаление старых обработанных конфликтов

`deleteOutdatedConflicts`

3.7.7.1 Общее описание

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

3.7.7.2 Конфигурация задачи

| Параметр | По умолчанию | Описание |
|------------------------------|--------------|--|
| <code>deleteAfterDays</code> | 30 | Количество дней, которые конфликты хранятся в базе |

3.7.8 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

3.7.8.1 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.

3.7.8.2 Конфигурация задачи

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

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

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

3.7.8.3 Примеры

3.7.8.3.1 Создать нового пользователя

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

3.7.8.3.2 Изменить пароль

| Параметр | Значение |
|----------|----------|
| username | user1 |

| Параметр | Значение |
|--------------|---|
| passwordhash | \$s0\$e0801\$SKg6OZwd84heZhCMn7nvcA==\$MixhOphIPBsG1Wgy0sWyLhIVsga082LCL0GEVhRNKWM= |
| delete | false |

3.7.8.3.3 Изменить роли

| Параметр | Значение |
|----------|---------------|
| username | user1 |
| roles | ROLE_OPERATOR |
| delete | false |

3.7.8.3.4 Удалить пользователя

| Параметр | Значение |
|----------|----------|
| username | user1 |
| delete | true |

3.7.8.4 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 password1 password1
```

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

```
$s0$e0801$jyKC7AAbZsu8XBcuRHP1pw==$60d4+3vRC05ZOa5l6x0bw+EK3F1JzfET+wCgU33
LCZc=
$s0$e0801$fqDFHtqwp7MUEFKkZJCUUw==$VBNVjgjqy2QdzZYUwfGPTwywlbeB197M0Xykr7
1Z30=
$s0$e0801$9IqwdWeg2R5exDQui5cHDQ==$cSr1W0H30cmRNf0jeSDExser43b4vc1RZbIqAqB
6wxw=
```

3.7.8.4.1 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`.

3.7.8.4.1.1 Пример:

На вход:

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

Code Block 1 `scryptpass\data.txt`

На выходе:

```
# Файл в кодировке UTF-8 (без BOM)
# Комментарии копируются как есть
# Можно задавать несколько колонок, разделитель табуляция. Кодировается
# первая колонка. Ниже строки для примера.
$s0$e0801$naKBj92C00i11ERiN1XNfQ==$DgJGHcoeGJYA6b6qoR/PdYTnkWSVfHwfziKcWgl
Vo+0=
$s0$e0801$iehrFvIwQ9FNi0574WwkbQ==$OSfM7rTReZjIgfM6zK3UUo7e1igsSzMnKqRXj14
CQHg= неcodируемое_значение_1 неcodируемое_значение_2
```

Code Block 2 `rypted.txt`

3.7.9 searchEngineOptimize. Оптимизация поисковых индексов

3.7.9.1 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`.

3.7.9.2 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 (держатся люсиновский метод и он не говорит пустой индекс или нет, и что именно с ним сделано).

3.7.9.3 Конфигурация задачи

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

3.7.9.4 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
    
```

3.7.10 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.

3.7.10.1 Конфигурация задачи

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

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

3.7.11 enginesFullRebuild. Полное перестроение поискового индекса

3.7.11.1 Общее описание

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

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

Без инструкций от службу сопровождения вендора решения запускать не надо, так как может занимать продолжительное время, если в системе много типов сущностей: физические лица, юридические лица, ЕГРЮЛ... Вместо нее обычно используется дочерняя задача [allEnginesRebuildByName](#), чтобы не тратить время на перестроение ненужных индексов. При работе в параллель с другими задачами скорость перестроения может заметно падать.

3.7.11.2 Конфигурация задачи

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

3.7.12 findHistoricalConflicts. Поиск конфликтных изменений

findHistoricalConflicts

[Возобновляемая](#)

Неконфликтная

3.7.12.1 Описание задачи

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

3.7.12.2 Запуск задачи

Задача работает по списку записей, 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 заполнить таблицу `TASK_PARTY_HIDS` идентификаторами актуальных записей 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.

3.7.12.3 Алгоритм

1. Для актуальных записей 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`.

3.7.12.4 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|------------------------|---|-----------------------|
| <code>partyType</code> | Тип записей, для которых выполняется поиск конфликтов | PHYSICAL |

3.7.12.5 Логирование

После выполнения задачи в лог пишется статистика:

- `conflicted` — сколько всего нашли конфликтов (= сколько записей попало в `conflicted_party_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.

3.7.13 Задача обновления кэшей

```
clearCache
```

3.7.13.1 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.

3.7.13.2 Запуск задачи

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

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

3.7.14 Очистка кэша инкрементального поиска дубликатов

```
clearIncDeDuplicationCache
```

3.7.14.1 Общее описание

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

3.7.14.2 Конфигурация задачи

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

3.7.15 Очистка хэшей поиска дубликатов

```
truncateDeDuplicationData
```

3.7.15.1 Общее описание

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

3.7.15.2 Конфигурация задачи

| Параметр | По умолчанию | Описание |
|-----------------|--------------|--|
| clearFullCaches | false | Нужно ли помимо инкрементальных удалить полные данные дедупликации |

3.8 Экспорт данных

Группа задач, предназначенных для экспорта данных из системы «Гражданский фактор»:

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

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

3.8.1 Инкрементальный экспорт эталонных связей

`incrementalExportRelations`

Непрерываемая

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

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

3.8.1.1 Концепция задачи

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

3.8.1.2 Правила работы

1. Удаляет закрытые связи из таблицы и связи, которые стали указывать на одну сущность*.
2. Выгружает новые актуальные связи с уникальным сочетанием типа связи и типов связанных объектов и исторических идентификаторов и порядка объектов во взаимосвязи.

Концы связи всегда представляют актуальные объединенные сущности, если связь указывает на сущность, которая была объединена.

3. Обновляет связи, один или оба конца которых стали указывать на новую объединенную сущность.

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

3.8.1.3 Конфигурация задачи

С релиза 20.18 у задачи появился настраиваемый параметр:

| Название параметра | Тип | Описание | Значение по умолчанию |
|--------------------------------------|---------------------------|--|-----------------------|
| <code>updateEndDateForDeleted</code> | <code>true / false</code> | Параметр определяет, удалять ли запись о закрытой связи или проставлять ей дату закрытия. Если задать <code>true</code> , запись не удалится из таблицы, а закрывается. | <code>false</code> |

3.8.2 Полный экспорт эталонных связей

`fullExportRelations`

[Непрерываемая](#)

Полностью очищает таблицу [RELATION_EXPORT](#) и потом заполняет ее актуальными связями с учетом объединения сущностей.

Используется для первичного заполнения таблицы [RELATION_EXPORT](#) или ее полного перестроения. Для актуализации связей после загрузки инкремента используйте задачу [инкрементального экспорта связей](#).

3.8.2.1 Правила заполнения:

1. Выгружаются только актуальные связи с уникальным сочетанием типа связи и типов связанных объектов и исторических идентификаторов и порядка объектов во взаимосвязи.
2. Если любая из сторон связи — контрагент, который был влит, то вернуть объединенного контрагента.
3. Не выгружать связи, оба конца которых указывают на одну сущность.
4. Если связь ранее подтверждалась Фактором, а сейчас нет, то такая связь выгрузится закрытой.

3.8.2.2 Задача прошла, таблица пустая

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

1. Выполнить задачу `graphEngineFullRebuild`
2. Выполнить задачу `fullExportRelations`

3.9 Диагностика

3.9.1 Когда использовать

По просьбе сотрудника техподдержки.

3.9.2 Как собрать диагностику

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

| 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 система «Гражданский фактор» 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 система «Гражданский фактор» 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} система «Гражданский фактор» 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@cleandata.ru
```

3.9.3 Сбор логов

3.9.3.1 Общее описание

Задача `diagnosticLogs` предназначена для сбора логов систем «Гражданский фактор» и Фактор в диапазоне дат [`lowerBound`, `upperBound`].

3.9.3.2 Конфигурация задачи

Группа `diagnostic`
 Название `diagnosticLogs`

Параметры задачи

| Название параметра | Значение | Обязательность |
|---------------------------------------|---|----------------|
| <input type="text" value="profiles"/> | <input type="text" value="logs"/> | |
| <input type="text" value="params"/> | <input "="" type="text" value="lowerBound=;upperBound="/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

Настраиваемые параметры:

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

3.9.3.3 Пример

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

3.9.4 Сбор диагностики по контрагенту

3.9.4.1 Общее описание

Задача `diagnosticParty` предназначена для сбора информации по конкретной сущности или по группе сущностей. Также можно собирать в диагностике информацию по связанным сущностями и потенциальным дубликатам.

3.9.4.2 Конфигурация задачи

Задача: diagnosticParty
Группа: Диагностика

| Название параметра | Значение |
|--|--|
| <input type="text" value="profiles"/> | <input type="text" value="sql.party,mapdb-state"/> |
| <input type="text" value="hid"/> | <input type="text" value="1024"/> |
| <input type="text" value="sendEmail"/> | <input type="text" value="true"/> |
| <input type="text" value="sendTo"/> | <input type="text" value="email@hflabs.ru"/> |

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию | Пример |
|--------------------|---|-----------------------|------------------------|
| sendEmail | Признак необходимости отправки диагностики по почте | true | |
| sendTo | Список получателей письма с диагностикой через запятую | | |
| hid | НID конкретной сущности или группы сущностей, указанных через ", " | 1024 | |
| level | Уровень сбора диагностики, можно указывать несколько уровней через ", " | | BASIC, INCLUDE_RELATED |
| includeSearchIndex | Признак необходимости собирать данные из поискового индекса | false | |

3.9.4.3 Уровни сбора диагностики

- BASIC – в диагностику собираются только запрошенные записи (по умолчанию всегда подразумевается этот уровень, его можно не указывать);
- INCLUDE_RELATED – в диагностику включаются связанные с заданными записями сущности;

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

- o 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.

3.9.4.4 Данные из поискового индекса

Данные собираются по всем типам сущностей, 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.

3.9.5 Сбор диагностики с указанным профилем

3.9.5.1 Общее описание

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

3.9.5.2 Конфигурация задачи

Группа `diagnostic`
 Название `diagnosticProfiles`

Параметры задачи

| Название параметра | Значение | Обязательность |
|---|--|----------------|
| <input type="text" value="runInCluster"/> | <input type="text" value="true"/> | |
| <input type="text" value="profiles"/> | <input type="text" value="config,versions"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

| Параметр | Описание | Значение по умолчанию |
|------------------------------|--|--|
| runInCluster | Сбор диагностики в режиме горячего резерва | true |
| profiles | Список профилей , указанных через <code>" , "</code> | config,db.connection,logs,search,sql.counts,sql.db,sql.metadata,versions |

3.9.5.3 Несколько профилей

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

3.9.5.4 Пример

| Параметр | Описание | Значение по умолчанию |
|-----------------------|--|--|
| <code>profiles</code> | Список профилей , указанных через ", " | <code>config, db.connection, logs, search, sql.counts, sql.db, sql.metadata, versions</code> |
| <code>params</code> | 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. | <code>lowerBound=2020-01-10;hid=1024;task=findDuplicates</code> |

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

3.9.6 Проверка БД на непротиворечивость

3.9.6.1 Общее описание

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

Найденные проблемы записывает в таблицу `tmp$PROBLEMS` и отправляет по электронной почте администратору.

3.9.6.2 Конфигурация задачи

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

Группа `diagnostic`
 Название `runDBCheckers`

Параметры задачи

| Название параметра | Значение | Обязательность |
|--|--|----------------|
| <input style="width: 95%;" type="text"/> | <input style="width: 95%;" type="text"/> | |
| <input style="width: 95%;" type="text"/> | <input style="width: 95%;" type="text"/> | |

Если нужно запустить определенную проверку (по просьбе сотрудника из службы сопровождения вендора решения), настройте параметр `procedureParams`.

Группа diagnostic
Название runDBCheckers

Параметры задачи

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

| Параметр | Описание | Пример |
|-----------------|---------------------------------------|-------------|
| procedureParams | Список проверок, указанных через ", " | 1, 2, 10, 5 |

3.9.6.3 Проверка БД на непротиворечивость (техподдержка, checker)

3.9.6.3.1 Проверки

Проверяется наличие проблем:

| Тип проблемы | Описание |
|--------------|---|
| 0 | Атрибут есть в исходном, нет в МА |
| 1 | Атрибут есть в слитом, нет в МА Если он что-то вернет, проанализировать и тогда применять 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 связи MERGED нет, или она закрыта |
| 10 | Результирующий атрибут состоит из одного исходного, но их RAW_ID не совпадают |

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

| HID_PARTY | PROBLEM_TYPE | HID_ATTR |
|-----------|--------------|----------|
| 1 | PHONE_0 | 11 |

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

3.9.6.3.2 Дополнительная информация о проблеме

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

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

3.9.7 Профили диагностики

Задачи диагностики предоставляют возможность сбора следующей информации:

| Профиль | Параметры | Какую информацию собирает |
|---------|--|---|
| logs | <code>lowerBound</code> <code>upperBound</code> | Журналы систем «Гражданский фактор» и Фактора в диапазоне дат |

| Профиль | Параметры | Какую информацию собирает |
|---------------|--------------------------|--|
| | | [lowerBound, upperBound]. Оба параметра опциональные, задаются в формате ГГГГ-ММ-ДД. |
| logs-full | lowerBound upperBound | Полные журналы систем «Гражданский фактор» и Фактора в диапазоне дат [lowerBound, upperBound]. Оба параметра опциональные, задаются в формате ГГГГ-ММ-ДД. |
| sql.party | hid | Информация по конкретной записи или по группе записей, указанных через ", " |
| sql.party.old | hid | Устаревший профиль сбора диагностики по сущностям. Работает аналогично sql.party, но использует другой механизм. |
| mapdb-state | hid | Состояние mapdb-индекса по сущности. Обычно собирается в паре с sql.party |
| graph-state | hid | Состояние графа связей по сущности. |
| cdi.jvm | | Информация о JVM системы «Гражданский фактор»: дампы потоков, параметры обработки HTTP-запросов, использование ресурсов ОС |
| config | | Конфигурационные файлы |
| db.connection | | Информация о соединении с БД |
| factor.jvm | | Информация о JVM Фактора: дампы потоков, параметры обработки HTTP-запросов, использование ресурсов ОС. |
| search | | Статистика по поисковому индексу |
| sql.common | | Системные таблицы |

| Профиль | Параметры | Какую информацию собирает |
|------------------------------------|--------------------------|--|
| sql.counts | | Количество записей |
| sql.db | | Физические характеристики БД (нужны права GRANT SELECT_CATALOG_ROLE TO <user>;) |
| sql.duplicate | | Общая статистика по дубликатам |
| sql.duplicate.physical | | Статистика по дубликата (физические лица) |
| sql.metadata | | Метаданные БД |
| sql.quality.physical | | Статистика по качеству данных (физические лица) |
| sql.quality.phone | | Статистика по качеству данных (телефоны) |
| sql.data.statistics.previous.month | | Статистика по качеству данных за прошлый месяц (общая) |
| versions | | Версии систем «Гражданский фактор», Фактора и справочников Фактора |
| sql.checker | | Проверка целостности и согласованности бизнес-объектов системы |
| sql.stat.task | task=task_key | Информация обо всех запусках указанной задачи. task_key - ключ задачи как в админке, например, findDuplicates |
| user.env | | Информации о браузере |
| user.event | lowerBound upperBound | Информации о событиях в веб-интерфейсе в диапазоне дат [lowerBound, upperBound]. Оба параметра опциональные, задаются в формате ГГГГ-ММ-ДД. |
| metadata | | Метаинформация о модели данных для автоматической генерации документации |

3.9.8 Проверка слитых контрагентов на непротиворечивость

3.9.8.1 Общее описание

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

Найденные проблемы записывает в таблицу `tmp$PROBLEMS` и отправляет по электронной почте администратору.

3.9.8.2 Конфигурация задачи

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

3.9.8.3 Особенности работы

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

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

3.9.8.4 Поиск неконсистентности "Запись закрыта, связь — нет"

3.9.8.4.1 Общее описание

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

Запись закрыта, но ее связи открыты

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

3.9.8.4.2 Конфигурация задачи

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

3.9.9 Найти и исправить проблемы в золотых карточках

3.9.9.1 Общее описание

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

3.9.9.2 Конфигурация задачи

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

3.9.10 Сбор диагностики в режиме горячего резерва

Задачи диагностики содержат параметр **runInCluster**, который отвечает за одновременный сбор диагностической информации с разных экземпляров системы «Гражданский фактор».

3.9.10.1 Значения параметра runInCluster :

| Значение параметра | Описание |
|--------------------|---------------------------|
| false | Сбор диагностики локально |

| Значение параметра | Описание |
|--------------------|---|
| true | Сбор диагностики с разных экземпляров системы «Гражданский фактор». |

3.9.10.2 Значения параметра runInCluster в задачах диагностики по умолчанию:

| Задача | Значение по умолчанию | Описание |
|--------------------|-----------------------|--|
| diagnosticParty | false | Задача предназначена для сбора информации по конкретной сущности или по группе сущностей. Значение параметра runInCluster настраивается в task-performer-diagnostic.xml. |
| diagnosticLogs | true | Задача предназначена для сбора логов систем «Гражданский фактор» и Фактор в диапазоне дат. Значение параметра runInCluster настраивается в task-performer-diagnostic.xml. |
| diagnosticProfiles | true | Задача предназначена для сбора информации по любому профилю диагностики . Значение параметра runInCluster настраивается в jobs_system.xml или через интерфейс Администратора. |

При запуске задачи с runInCluster = true формируется событие, которое уходит на вторую ноду и запускает там сбор диагностики. В истории это будет один запуск задачи. Задача не дожидается завершения сбора информации на второй ноде, она только отгружает событие.

3.9.11 Сбор метрик

3.9.11.1 Общее описание

Задача diagnosticMetrics предназначена для сбора анонимной статистики по использованию системы «Гражданский фактор».

Не всегда система «Гражданский фактор» 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 систему «Гражданский фактор» сбор статистики использования мелких «фич» вроде поиска на списке записей, фильтрации на столбцах и тому подобного.

Статистика полностью анонимная (не записываем даже имя пользователя) и деперсонализованная (не сохраняем никакие данные о физ. лице)

3.9.11.2 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 |
|------------|---|--------------------------|
| profiles | Профиль диагностики | user.event,user.env. |
| params | Диапазон дат, задается в формате ггг-мм-дд. По умолчанию даты не заданы. | lowerBound=; upperBound= |
| attachLogs | Нужно ли собирать вместе с метриками логи | false |

3.9.11.3 Пример

| params | Что собирает диагностика |
|---|---------------------------------------|
| 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 |

3.9.12 Сбор диагностики по задаче

3.9.12.1 Общее описание

Задача diagnosticTaskStat предназначена для сбора информации по задаче:

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

3.9.12.2 Конфигурация задачи

Создать/Редактировать задачу

Группа diagnostic
Название diagnosticTaskStat

Параметры задачи

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

Настраиваемые параметры:

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

3.9.13 Сбор логов за сегодня

Задача недоступна для ручного редактирования и запуска

Общее описание

Задача diagnosticLogsForToday аналогична задаче diagnosticLogs и собирает логи систем «Гражданский фактор» и Фактор только за сегодня.

3.9.13.1 Конфигурация задачи

Создать/Редактировать задачу

Группа diagnostic
Название diagnosticLogsForToday

Параметры задачи

| Название параметра | Значение | Обязательность |
|--|-----------------------------------|----------------|
| <input type="text" value="sendEmail"/> | <input type="text" value="true"/> | |
| <input type="text" value="profiles"/> | <input type="text" value="logs"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

Настраиваемые параметры:

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

3.9.13.2 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.

3.9.14 Сбор метрик за предыдущий месяц

3.9.14.1 Общее описание

Задача `diagnosticMetricsForPreviousMonth` аналогична задаче `diagnosticMetrics` и предназначена для сбора анонимной статистики по использованию системы «Гражданский фактор» только за предыдущий месяц.

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

3.9.14.2 Конфигурация задачи

Группа `diagnostic`
 Название `diagnosticMetricsForPreviousMonth`

Параметры задачи

| Название параметра | Значение | Обязательность |
|---------------------------------------|---|----------------|
| <input type="text" value="profiles"/> | <input type="text" value="user.event.previous.month,user.env"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

Настраиваемые параметры:

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

3.9.15 Импорт датасета с контрагентами (xls)

`importDataSetTask`

3.9.15.1 Общее описание

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

3.9.15.2 Особенности

Показывается в разделе диагностики, если

в standalone.conf/standalone.conf.bat указать секретный ключ:

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

Code Block 3 Windows: standalone.conf.bat

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

Code Block 4 Linux: standalone.conf

! 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.

3.9.15.3 Конфигурация задачи

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

3.10 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 | Срон выражение |
|---|--------------------------------|-------------------------|----------------|
| Редактировать Удалить Выполнить | afterDatabaseBackupTrigger | | |
| Редактировать Удалить Выполнить Выключить | deleteOutdatedConflictsTrigger | 24.03.2018 21:00:00 | 0 0 21 ? * SAT |
| Редактировать Удалить Выполнить Включить | detectPopularAttributes | | |
| Редактировать Удалить Выполнить | 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 |

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

3.10.1 Триггер

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

В триггере есть возможность указать цепочки последовательно запускающихся задач.

3.10.1.1 Пример

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

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

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

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

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

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

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

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

3.10.2 Список триггеров

На данной странице можно увидеть:

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

- *Срон-выражение* - срон-выражение, по которому настроен автоматический запуск триггера

3.10.3 Работа с триггерами

Ниже перечислены возможные операции по работе с триггерами.

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

3.10.3.1 Запуск триггеров

Триггеры стартуют автоматически, время запуска настраивают сотрудники службы сопровождения вендора решения.

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 | Срон выражение |
|--|--------------------------------|-------------------------|----------------|
| Редактировать Удалить Выполнить | afterDatabaseBackupTrigger | | |
| Редактировать Удалить Выполнить Включить | deleteOutdatedConflictsTrigger | 24.03.2018 21:00:00 | 0 0 21 ? * SAT |
| Редактировать Удалить Выполнить Включить | detectPopularAttributes | | |
| Редактировать Удалить Выполнить | rebuildAllIndexesTrigger | | |
| Редактировать Удалить Выполнить Включить | refreshDataStatisticTrigger | 23.03.2018 2:00:00 | 0 0 2 * * ? |
| Редактировать Удалить Выполнить Включить | searchEngineOptimizeTrigger | 24.03.2018 22:00:00 | 0 0 22 ? * SAT |

Создать триггер

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

Список триггеров

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

Создать триггер


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

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

Список триггеров

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

Создать триггер



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

3.10.3.2 Создание триггеров


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

Для создания нового триггера нажимаем кнопку *Создать триггер*.

Список триггеров

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

Срон триггер

Название Уникальный код триггера (только латинские символы и цифры)

Срон выражение секунда, минута, час, день, месяц, день недели

Продолжать при ошибках Продолжать выполнение следующей задачи при наличии ошибок

Продолжать при конфликтах Продолжать выполнение следующей задачи при наличии конфликтов выполнения

Выберите список задач к выполнению:

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

После задания параметров, нажимаем кнопку *Сохранить*.

Срон триггер

Название Уникальный код триггера (только латинские символы и цифры)

Срон выражение секунда, минута, час, день, месяц, день недели

Продолжать при ошибках Продолжать выполнение следующей задачи при наличии ошибок

Продолжать при конфликтах Продолжать выполнение следующей задачи при наличии конфликтов выполнения

Выберите список задач к выполнению:

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

Новый триггер появится в списке.

Список триггеров

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

[Создать триггер](#)

3.10.3.3 Редактирование триггеров

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

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

3.10.3.4 Удаление триггеров

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

3.10.3.5 Настройка времени выполнения задачи в триггере

Создать триггер и настроить срон-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 | addressActualization | Актуализация адресов |
| Детали Редактировать Выполнить | archivePartyClose | Закрытие устаревших карточек |
| Детали Редактировать Выполнить | archivePartyClosePrepare | Отбор устаревших карточек 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 герprocess
Название archivePartyClose

Параметры задачи

| Название параметра | Значение | Обязательность |
|--------------------|----------|----------------|
| executionTimeout | 1 | |
| | | |

Нажать кнопку *Сохранить*.

3.10.4 Список триггеров

По умолчанию в системе подключены триггеры:

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

| Название | Список задач | Описание | Расписание работы |
|--------------------------------|--|--|--|
| | <p><i>черным спискам).</i></p> <p>5. Актуализация групп дубликатов (раздел <i>Системные</i>).</p> <p>6. Полное перестроение индекса конфликтов обновления (раздел <i>Системные</i>).</p> <p>7. Перестроение хэшей полной проверки по ЕГРЮЛ (раздел <i>Дедупликация и слияние</i>).</p> | | |
| deleteOutdatedConflictsTrigger | Удаление неактуальных конфликтов | Удаление обработанных конфликтов старше, чем 30 дней назад | Каждую субботу в 21:00 |
| detectPopularAttributesTrigger | <p>1. Найти массовые атрибуты (раздел <i>Массовые атрибуты</i>).</p> <p>2. Проставить популярность по найденным значениям (ФЛ) (раздел <i>Массовые атрибуты</i>).</p> <p>3. Проставить популярность по найденным значениям (другие типы контрагентов)</p> | Выявление и проставление популярности массовым атрибутам. | Каждое первое воскресенье месяца в 12:00 |

| Название | Список задач | Описание | Расписание работы |
|--|---|--|------------------------------------|
| diagnosticLogsForTodayTrigger | Отправить логи за сегодня на указанную в настройках почту. | | Каждый день в 23:55 и в 7:55 |
| diagnosticMetricsForPreviousMonthTrigger | Отправить веб-метрики и общую статистику по качеству данных за предыдущий месяц на указанную в настройках почту. | | Каждое первое число месяца в 02:40 |
| rebuildAllIndexesTrigger | <ol style="list-style-type: none"> 1. Полное перестроение графа связей (раздел <i>Системные</i>). 2. Полное перестроение поискового индекса (раздел <i>Системные</i>). 3. Актуализация групп дубликатов (раздел <i>Системные</i>). | Перестроение поисковых индексов. Запускается вручную. | |
| searchEngineOptimizeTrigger | Оптимизация поисковых индексов | Оптимизирует поисковый индекс. Запускается автоматически раз в неделю. | Каждую субботу в 22:00 |
| refreshDataStatisticTrigger | Сбор метрик по задаче refreshDataStatistic | | Каждый день в 02:00 |

3.11 Возобновляемые задачи

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

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

3.11.1 Запуск на ночь

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

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

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

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

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

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

3.12 Непрерываемые задачи

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

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

3.13 Проверка по черным спискам

3.13.1 Проверка по черным спискам

Группа задач, предназначенных для проверки данных по черным спискам:

- `fullBlacklistCheck` — полная проверка сущностей по черным спискам;
- `actualizeBlacklistCheck` — проверки по черным спискам контрагентов, требующих актуализации. Задача проверяет только новые и измененные записи;
- `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.

3.13.2 Обновление дат актуальности ЧС

```
grabBlacklistActualDate
```

Непрерываемая

3.13.2.1 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](#).

3.13.2.2 Алгоритм задачи

Система «Гражданский фактор» через веб-сервиса 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.

3.14 Необратимое удаление данных

3.14.1 Общее описание

Задача `PartyDelete` полностью удаляет данные из системы «Гражданский фактор».

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

Данные удаляются полностью, восстановить их будет нельзя!

3.14.2 Особенности задачи

1. Задача работает только по закрытым оригинальным контрагентам — открытые игнорируются.
2. Задача удаляет всех закрытых слитых контрагентов, которые относятся к удаляемым оригинальным.
3. Если закрываемая карточка была разлита из открытой результирующей, то в версиях результирующей все равно останется информация о реквизитах/атрибутах удаляемой записи.
4. Удаление дубликатов выполняется следующим образом: задача ищет актуальные дубликаты, содержащие HID-ы удаляемых сущностей 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 записью сущности (MSISDN, договоры, машины и т.п.).

3.14.3 Запуск задачи

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

Пример ручного заполнения 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 |

3.14.4 Обработка ошибок

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

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

Не считаются ошибками и игнорируются следующие записи в таблице:

- НID-ы открытых записей. При обнаружении НID'ы записываются в лог, удаляются из delete_party_hids.
- НID-ы «золотых» записей. 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 записи (например, 'PH' вместо 'PHYSICAL'). Не удаляются из delete_party_hids, в лог не записываются.

3.15 Закрытие записей

| |
|------------|
| PartyClose |
|------------|

3.15.1 Запуск задачи

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

Пример ручного заполнения таблицы:

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

3.15.2 Обработка ошибок

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

Не считаются ошибками и игнорируются следующие записи в таблице:

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

3.16 Порционные задачи

Иногда исходные системы выгружают в систему «Гражданский фактор» миллионы записей за раз. Например, когда добавляется новое поле в модель данных объекта. Тогда система «Гражданский фактор» несколько дней обрабатывает инкремент, а затем еще несколько дней отдает данные системам-потребителям. Получается, что информация сильно запаздывает.

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

3.16.1 Конфигурация задачи

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

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

Параметры применяются сразу, без рестарта сервера

Создать/Редактировать задачу

Группа import
 Название BufferDDWH

Параметры задачи

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

Сохранить Отмена

3.16.2 Работа задачи

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

В нем записывается импортированное "сегодня" количество. Значение свойства имеет формат "dd.mm.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

3.16.3 Подключение порционности

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

3.17 Проверка по ЕГРЮЛ

Группа задач, предназначенных для проверки по ЕГРЮЛ данных из системы «Гражданский фактор»:

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

3.17.1 Полная проверка по ЕГРЮЛ

egrCheck

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

Не считаются дублями:

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

3.17.2 Актуализация ЮЛ – карточек-копий ЕГРЮЛ

```
egrMirrorPartiesActualize
```

[Возобновляемая](#), конфликтная

3.17.2.1 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.

3.17.2.2 Запуск задачи

Перед запуском задачи необходимо:

1. Выполнить задачу `cleanTaskParameterTables` для очистки таблицы `task_party_hids`;
2. Выяснить идентификаторы контрагентов (`hid_party`), для которых необходимо выполнить перерасчет;
3. Добавить список данных идентификаторов в таблицу `task_party_hids` вручную или выбрать для обновления все актуальные карточки ЕГРЮЛ, для которых есть карточка-копия в ЮЛ запросом:

```
insert into task_party_hids
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;
```

3.17.2.3 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|------------------|--|-----------------------|
| poolSize | Максимальное количество потоков | 8 |
| pageSize | Размер пачки | 1000 |
| executionTimeout | Максимальное время выполнения задачи в минутах. При превышении задача останавливается. | Без ограничения |

3.18 Массовые атрибуты

3.18.1 Массовые атрибуты

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

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

3.18.2 Поиск массовых реквизитов и атрибутов

`findPopularValues`

[Непрерываемая](#)

3.18.2.1 Описание задачи

Система «Гражданский фактор» 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.

3.18.3 Проставление популярности по найденным значениям (ФЛ и ЮЛ)

markPopularPhysicalAttributes

markPopularLegalAttributes

[Возобновляемая](#)

3.18.3.1 Описание задачи

Система «Гражданский фактор» забирает записи из таблицы task_doubtful_reclean_hids и отправляет в Фактор для проставления [кода популярности](#).

3.19 Миграция

- [Автопрогон скриптов выдачи грантов пользователям](#)
- [Актуализация реестров](#)
- [Задача переобработки места рождения](#)
- [Инициализация прошлых версий атрибутов](#)
- [Миграция нового поля](#)

3.19.1 Автопрогон скриптов выдачи грантов пользователям

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

Для этого в системе «Гражданский фактор» 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».

3.19.2 Актуализация реестров

3.19.2.1 Добавление нового реестра

Количество записей в одном реестре не должно превышать 100 000.

Загрузка реестра большого 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-записей', '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 записей (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 запускать задачу actualizeExternalListIds.

В режиме горячего резерва запускать только на одной ноде.

3.19.2.2 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.

3.19.2.3 Конфигурация задачи

| Параметр | Описание | Значение по умолчанию |
|-----------|-----------------|-----------------------|
| partyType | Тип контрагента | |

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

3.19.2.4 Логирование

Задача стандартно логирует обработку каждые 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;
```

3.19.2.5 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.

3.19.3 Задача переобработки места рождения

```
birthPlaceInitTask
```

[Возобновляемая](#)

Конфликтующая

3.19.3.1 Описание задачи

Задача стандартизирует реквизит «Место рождения».

3.19.3.2 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 birthPlaceInitTask.

В режиме горячего резерва запускать только на одной ноде.

3.19.3.3 Конфигурация задачи

| Параметр | Описание | Значение по умолчанию |
|-----------|---------------------------------|-----------------------|
| partyType | Тип контрагента | PHYSICAL |

3.19.4 Инициализация прошлых версий атрибутов

3.19.4.1 Запуск задачи

Задача работает по списку записей, 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 записей;

После этого можно выполнять задачу `pastValuesInit`.

В режиме горячего резерва запускать только на одной ноде.

3.19.4.2 Конфигурация задачи

Настраиваемые параметры:

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

3.19.4.3 После выполнения задачи

3.19.4.3.1 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.

3.19.5 Миграция нового поля

| |
|--------------------------------------|
| <code>fooMigrateRequisiteTask</code> |
|--------------------------------------|

3.19.5.1 Описание задачи

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

3.19.5.2 Особенности задачи

Перед запуском задачи необходимо:

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

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

2. Добавить в систему «Гражданский фактор» задачу миграции (*выполняется специалистами службы сопровождения вендора решения*)

3.19.5.3 Конфигурация задачи

Если первоначальная загрузка выполняется без очистки данных, в задаче конфигурируется только параметр `partyType` — тип записи,

Создать/Редактировать задачу

Группа `reprocess`
 Название `innMigrateRequisiteTask`

Параметры задачи

| Название параметра | Значение | Обязательность |
|--|---------------------------------------|----------------|
| <input type="text" value="essenceType"/> | <input type="text" value="PHYSICAL"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

Сохранить Отмена

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

- `partyType` — тип записи;
- `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 |
|--|--|----------------|
| <input type="text" value="essenceType"/> | <input type="text" value="LEGAL"/> | |
| <input type="text" value="mappingName"/> | <input type="text" value="demo-legal-soap"/> | |
| <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | |

Сохранить Отмена

3.20 Миграция структуры хранения исторических данных

- [Форсирование миграции структуры исторических данных \(Фаза 1\)](#)
- [Форсирование миграции структуры исторических данных \(Фаза 3\)](#)

3.20.1 Конфигурация задачи

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

Но есть параметры, которые можно добавить по необходимости (если порекомендовал сотрудник службы сопровождения вендора решения)

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

3.20.2 Форсирование миграции структуры исторических данных (Фаза 3)

migratePhase3

[Возобновляемая](#)

3.20.2.1 Описание задачи

Задача удаляет нулевые версии данных в `_HIST` таблицах.

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

3.20.2.2 Запуск задачи

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

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

3.20.2.3 Алгоритм работы

Задача проверяет таблицу `migration_phase`, какая фаза у таблицы:

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

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

3.20.2.3.1 Было до миграции

`STAGING_HIST`

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

3.20.2.3.2 Стало после STAGING_HIST (удалили нулевую версию)

| ID_STAGING | HID_STAGING | VERSION |
|------------|-------------|---------|
| 1024 | 1024 | 1 |
| 1025 | 1024 | 2 |

Если таблица уже отмигрирована, больше ее не трогаем.

3.20.2.4 Конфигурация задачи

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

Но есть параметры, которые можно добавить по необходимости (если порекомендовал сотрудник службы сопровождения вендора решения):

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

3.21 Сбор статистики

3.21.1 Конфигурация задачи

Для стандартного запуска задачу конфигурировать не нужно.

При специальном запросе поддержки, можно настраивать следующие параметры:

| Параметр | Описание | Значение по умолчанию |
|----------|---------------------------------|-----------------------|
| poolSize | Максимальное количество потоков | 1 |

3.21.2 Особенности задачи

Задача не конфликтующая.

3.21.3 Настройка сбора статистики

3.21.3.1 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.cleandata.cdi.task.performer.statistics.parser.ParserChainDataSt
aticRequestParser">
  <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>
```

```
<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>
```

Code Block 5 allSystemAliasDataStatisticRequestParser

3.21.3.2 Пример 1

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

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

3.21.3.3 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%)
```

3.21.3.4 Пример 3

Для добавления ложного 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. '<=' это '<='. Будут отобраны все правила из таблицы duplicate с коэффициентом <= 96.

2. Наполнить таблицу data_metric запросом для подсчета негарантированных дубликатов.

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

3.21.3.5 Пример 4

Итераторы по типам контрагентов и атрибутов

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

Аналогично можно использовать итератор по типам атрибутов %allAttributeTypes%.

3.21.4 DATA_METRIC

3.21.4.1 Столбцы

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|-------------|---------------|----------|--------------|-----------|---|
| ID_METRIC | NUMBER(*) | No | null | 1 | Идентификатор записи |
| DESCRIPTION | VARCHAR2(300) | No | null | 2 | Описание метрики |
| PARTY_TYPE | VARCHAR2(50) | No | null | 3 | Поисковый индекс для осуществления запроса. Всегда указывается только одно значение, несколько значений указывать нельзя. Может принимать следующие значения: |

| COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|-------------|----------------|----------|--------------|-----------|---|
| | | | | | <ul style="list-style-type: none"> Тип контрагента. Например, PHYSICAL или LEGAL; Идентификатор поиска по группам дубликатов. Значение MANUAL_DUPLICATE_GROUP |
| QUERY | VARCHAR2(1500) | No | null | 4 | Поисковый запрос |

3.21.4.2 Индексы

| INDEX_NAME | UNIQUENESS | COLUMNS |
|----------------|------------|-----------|
| PK_DATA_METRIC | UNIQUE | ID_METRIC |

3.21.5 DATA_STATISTIC

3.21.5.1 Столбцы

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

3.21.5.2 Индексы

| INDEX_NAME | UNIQUENESS | COLUMNS |
|-------------------|------------|---------------------|
| UK_DATA_STATISTIC | - | ID_METRIC, COUNT_TS |

3.21.6 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](#)

3.21.6.1 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 системы «Гражданский фактор».

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

3.21.6.2 Особенности формирования поисковых запросов для сбора метрик данных

3.21.6.2.1 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.

3.21.6.2.1.1 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.

3.21.6.2.1.1.1 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 пола  
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 |

3.21.6.3 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.

3.21.6.3.1 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.

3.21.6.4 Примеры метрик

3.21.6.4.1 Для контрагентов

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

3.21.6.4.1.1 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;
```

3.21.6.4.1.2 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;
```

3.21.6.4.2 Для групп дубликатов

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

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

3.21.6.4.2.1 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, 'Duplicates group >=3 and < 100', 'MANUAL_DUPLICATE_GROUP', '?  
ManualDuplicateGroup.partyType = PHYSICAL and  
ManualDuplicateGroup.partyCount >= 3 and ManualDuplicateGroup.progress <  
100');  
COMMIT;
```

3.21.6.4.3 SQL-метрики

3.21.6.4.3.1 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;
```


3.21.6.4.3.2 Пример 6. Добавление sql-метрики, возвращающей два значения

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

3.22 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.

3.22.1 Алгоритм мониторинга и перераспределения нагрузки между онлайн-поток и фоновыми задачами

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

3.22.1.1 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-перцентили заданных в конфигурации методов с заданным минимальным количеством вызовов за интервал времени (по умолчанию 5 минут).
5. Если показания 90-перцентили 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-перцентили 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-перцентили 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-перцентили 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-перцентили 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-перцентили 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.

3.22.1.2 Алгоритм при отсутствующем или незначительном онлайн-потоке

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

Если статистика по показаниям 90-перцентили 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-перцентили находятся в норме, то продолжаем обрабатывать записи максимальным количеством потоков, пока записи не закончатся.
3. Если показания 90-перцентили начинают превышать норму, то уменьшаем число потоков на дельту (по умолчанию 4 потока). Далее переходим к п.6 основного алгоритма.

3.22.1.3 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.

3.23 Задача массовой обработки конфликтов

massConflictResolution

[Возобновляемая](#)

Для корректного выполнения задачи УЗ, от имени которой запускается задача, нужна роль `ROLE_CONFLICT_MANAGER`

3.23.1 Описание задачи

Задачу следует использовать для автоматического разрешения/отклонения конфликтов.

3.23.2 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;
```

3.23.3 Конфигурация задачи

Настраиваемые параметры:

| Параметр | Описание | Значение по умолчанию |
|---------------|---|-----------------------|
| partyType | Тип контрагента , для которых выполняется разрешение конфликта | PHYSICAL |
| poolSize | Максимальное количество потоков | 4 |
| resolveStatus | Статус конфликта после обработки <ul style="list-style-type: none"> REJECTED — Отклонен. UPDATED_AFTER_UNMERGE — Применен (обновлен) после предварительного разливания. | REJECTED |

Создать/Редактировать задачу

Группа gerprocess
Название massConflictResolution

Параметры задачи

| Название параметра | Значение |
|--|---------------------------------------|
| <input type="text" value="poolSize"/> | <input type="text" value="4"/> |
| <input type="text" value="resolveStatus"/> | <input type="text" value="REJECTED"/> |
| <input type="text" value="partyType"/> | <input type="text" value="PHYSICAL"/> |
| <input type="text"/> | <input type="text"/> |
| <input type="text"/> | <input type="text"/> |

3.24 Полное необратимое удаление ВСЕХ данных выбранного типа

PartyTruncate

[Возобновляемая](#), неконфликтующая

3.24.1 Общее описание

Задача **необратимо удаляет ВСЕ** данные указанного типа сущности из системы «Гражданский фактор».

Данные удаляются полностью, восстановить их будет нельзя!

3.24.2 Запуск задачи

1. Запустить задачу, указав в параметре partyType нужный тип сущности.

2. Перестроить индексы (поисковый, граф связей, конфликты, дубли), в [режиме горячего резерва](#) на каждой ноде.

3.24.3 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 записей (включая исторические).

3.25 Общие параметры задач

- [Секретный параметр unsafe.forceExecution](#)
- [Параметр jmsEventFlag](#)

3.25.1 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 |
|-----------------------|----------|
| unsafe.forceExecution | true |

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!

3.25.2 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.

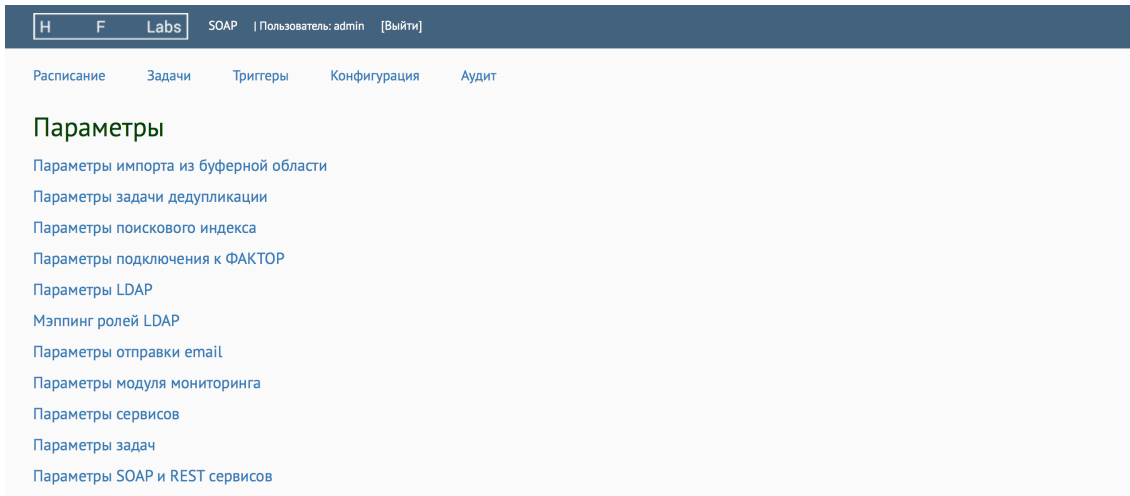
3.25.2.1 Пример

Размечаем в обратном потоке флагом массовости MASSIVE:

| Параметр | Значение |
|---------------------------|----------|
| <code>jmsEventFlag</code> | MASSIVE |

4 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 сервисов](#)
- [Параметры синхронизации нод](#)
- [Параметры экспорта данных](#)

Вне интерфейса администратора настраиваются:

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

4.1 Мэппинг ролей LDAP

Изменения настроек вступают в силу только после рестарта сервиса CDI!

Мэппинг ролей LDAP

| Название параметра | Значение |
|------------------------|--|
| ROLE_ADMINISTRATOR | <input type="text" value="GLO-SEC-RLE-CDI_ADMINISTRATOR"/> |
| ROLE_CONSENT_MANAGER | <input type="text" value="CHANGE_LDAP_CONSENT_MANAGER"/> |
| ROLE_DUPLICATE_MANAGER | <input type="text" value="CHANGE_LDAP_DUPLICATE_MANAGER"/> |
| ROLE_EXTERNAL_SYSTEM | <input type="text" value="GLO-SEC-RLE-CDI_EXTERNAL_SYSTEM"/> |
| ROLE_GUARD | <input type="text" value="GLO-SEC-RLE-CDI_GUARD"/> |
| ROLE_OPERATOR | <input type="text" value="GLO-SEC-RLE-CDI_OPERATOR"/> |
| ROLE_PERFORMER | <input type="text" value="GLO-SEC-RLE-CDI_PERFORMER"/> |
| ROLE_RELATION_MANAGER | <input type="text" value="CHANGE_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 |
|----------------------------|---|
| ROLE_ADMINISTRATOR | администратор |
| ROLE_GUARD | офицер информационной безопасности |
| ROLE_OPERATOR | оператор (read-only) |
| ROLE_CONSENT_MANAGER | оператор согласий (read-only) |
| ROLE_PERFORMER | операционист |
| ROLE_DUPLICATE_MANAGER | менеджер дубликатов |
| ROLE_EXTERNAL_SYSTEM | исходная система с полным доступом к онлайн-интерфейсам |
| ROLE_EXTERNAL_SYSTEM_READ | исходная система с доступом только к методам на чтение |
| ROLE_EXTERNAL_SYSTEM_WRITE | исходная система с доступом только к методам на запись |
| ROLE_RELATION_MANAGER | верификатор связей |
| ROLE_CONFLICT_MANAGER | менеджер конфликтов |

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

4.1.1 Список методов, доступных только для чтения/записи

4.1.1.1 Методы, доступные для чтения

1. search
2. fuzzySearch
3. checkBlakList
4. getByHID
5. getByRawID
6. getRelations
7. getSourceAttribute
8. getReference

4.1.1.2 Методы, доступные для записи

1. save
2. saveAndMerge
3. saveRelations
4. close
5. closeAttribute
6. closeAttributeByRawId

4.2 Настройки задач во внешней директории

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

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

4.3 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 CDI!

4.3.1 Общие параметры

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| <code>jms.connectionFactory.name</code> | - | Название <code>connectionFactory</code> , если используется. |
| <code>jms.events.destination.isTopic</code> | <code>false</code> | Выходная очередь — <code>topic</code> ? |
| <code>jms.events.destination.name</code> | - | Название выходной очереди. |
| <code>jms.events.enabled</code> | <code>false</code> | <p>Рассылка сообщений в выходную очередь включена?</p> <p>✔ После изменения значения данного параметра рестарт CDI не требуется.</p> |
| <code>jms.soap.destination.isTopic</code> | <code>false</code> | <p>Входная очередь — <code>topic</code>?</p> <p>Если <code>false</code>, используется очередь. (Очередь — один получатель, <code>топик</code> — много)</p> |
| <code>jms.soap.destination.name</code> | - | Название входной очереди. |
| <code>jms.soap.enabled</code> | <code>false</code> | Получение сообщений из входной очереди включено? |
| <code>jms.events.attributes.defaultValues</code> | <code>false</code> | Возвращать в обратный поток согласия по умолчанию? |
| <code>jms.events.changeDetailsEnabled</code> | <code>false</code> | Возвращать ли результат в детализированном виде |
| <code>jms.send.slowSendThresholdSec</code> | 10 | <p>Время в секундах работы с очередью, при превышении которого в лог записывается ошибка следующего вида: ERROR</p> <p><code>JmsEventPublishHelper - Sending JMS message took 14.575 s! Event: 1040</code></p> |
| <code>jms.events.filter.enabled</code> | <code>false</code> | <p>Включить фильтры при формировании событий?</p> <p>Набор фильтров уникален для каждого заказчика. Изменение применяется без рестарта.</p> |

4.3.2 Специфичные для очередей TIBCO параметры

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

4.3.3 Специфичные для очередей IBM MQ параметры

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

4.3.4 Параметры для передачи по HTTP-протоколу

| Название параметра | Значение по умолчанию | Описание |
|---------------------------------------|-------------------------------|--|
| <code>jms.events.rest.enabled</code> | <code>false</code> | Флаг для отключения отправителя |
| <code>jms.events.rest.login</code> | | Логин для авторизации |
| <code>jms.events.rest.password</code> | | Пароль для авторизации |
| <code>jms.events.rest.poolSize</code> | 10 | Во сколько потоков посылать данные |
| <code>jms.events.rest.timeout</code> | 60000 | Максимальное время ожидания ответа от сервиса-получателя |
| <code>jms.events.rest.url</code> | <code>http://host:8888</code> | Строка для подключения к сервису-получателю |

4.3.5 Параметры для отправки в KAFKA

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------------|------------------------------|
| <code>jms.events.kafka.enabled</code> | <code>false</code> | Отправить сообщения в кафку? |
| <code>jms.events.kafka.bootstrapServers</code> | <code>localhost:9092</code> | Список серверов кафки |
| <code>jms.events.kafka.topic</code> | <code>cdi_events</code> | Название топика |

| Название параметра | Значение по умолчанию | Описание |
|---|-----------------------|---|
| <code>jms.events.kafka.timeoutInSec</code> | 5 | Таймаут на отправку одного сообщения |
| <code>jms.events.kafka.transactionId</code> | <code>cdi</code> | Идентификатор ноды системы «Гражданский фактор» в транзакционном координаторе |

4.3.6 Специфичные для очередей RabbitMQ параметры

| Название параметра | Описание |
|---|--|
| <code>jms.events.rabbitmq.enabled</code> | Включена ли отправка сообщений в шину. |
| <code>jms.events.rabbitmq.destination.name</code> | Название выходной очереди. При отправке сообщения значение данного параметра указывается в <code>Routing Key</code> . |
| <code>jms.events.rabbitmq.destination.exchange</code> | Место назначения. При отправке сообщения значение данного параметра указывается в <code>Exchange</code> . Может использоваться в RabbitMQ для маршрутизации сообщений. По умолчанию: - не заполнено. RabbitMQ пустое значение данного параметра воспринимает как — <code>AMQP default</code> . |
| <code>jms.events.rabbitmq.host</code> | Хост шины / несколько хостов через запятую. Важно: Все указанные хосты должны быть настроены на порт, указанный в <code>jms.events.rabbitmq.port</code> . Система выбирает хост из указанных и отправляет сообщения на него. Если хост перестает отвечать, сообщения отправляются на другой доступный хост из списка. |
| <code>jms.events.rabbitmq.port</code> | Порт шины |
| <code>jms.events.rabbitmq.username</code> | Логин для авторизации. |
| <code>jms.events.rabbitmq.password</code> | Пароль для авторизации. |
| <code>jms.events.rabbitmq.connectionTimeout</code> | Максимальное время ожидания (в мс) ответа от JMS. Значение по умолчанию: 5000 |

4.4 Параметры LDAP

Изменения настроек вступают в силу только после рестарта сервиса CDI!

Параметры LDAP

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

[Сохранить](#)

| Название параметра | Значение по умолчанию | Описание |
|----------------------|--------------------------------------|--|
| ldap.url | ldap://hostname:port | Hostname и порт сервера LDAP |
| ldap.account | - | Логин системной учетной записи |
| ldap.domain | - | Имя домена локальной сети |
| ldap.groupSearchBase | - | Ветка, в которой хранятся учетные записи для ролей |
| ldap.maxActive | 50 | Максимальное количество активных соединений одного вида (чтение/запись) |
| ldap.maxIdle | 10 | Максимальное количество бездействующих соединений одного типа (чтение/запись) в пуле |
| ldap.maxWait | -1 | Максимальная длительность (миллисек) ожидания свободного |

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

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

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

(2) Ldap.account + Ldap.domain + Ldap.password определяют служебного системного пользователя «Гражданский фактор» 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).

Система «Гражданский фактор», 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 системы «Гражданский фактор» (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.

4.5 Параметры SOAP и REST сервисов

Изменения настроек вступают в силу только после рестарта сервиса CDI!

| Название параметра | Значение по умолчанию | Описание |
|---|-----------------------|--|
| <code>webservice.error.logMode</code> | FULL | Если FULL, то в <code>cdi.log</code> записывается текст ошибочного запроса и вызванная им ошибка. Если SHORT и включена трассировка запросов к веб-сервисам, то в <code>cdi.log</code> записывается ошибка и идентификатор полного сообщения в <code>soap-trace.log</code> . Если SHORT и не включена трассировка запросов к веб-сервисам, то в <code>cdi.log</code> записывается только ошибка. |
| <code>webservice.stats.periodInSecs</code> | 1800 | Периодичность вывода в лог статистики по вызову веб-сервисов (в секундах). |
| <code>webservice.response.attributes.defaultValues</code> | false | Возвращать в ответах согласия по умолчанию? Если true, <code>search</code> начинает использовать БД для загрузки согласий |
| <code>webservice.response.attributes.pastValues</code> | false | Если true, то прошлые значения возвращаются в веб-сервисах Исключением является метод <code>search</code> , он не зависит от значения параметра |

| Название параметра | Значение по умолчанию | Описание |
|--|--|--|
| <code>webservice.concurrent.numberOfRetries</code> | 3 | Количество попыток повторной обработки контрагента при получении ошибки о параллельной модификации |
| <code>webservice.concurrent.delayBetweenRetries</code> | 1000 | Задержка между попытками повторной обработки при получении ошибки о параллельной модификации |
| <code>webservice.response.relations.maxCountByParty</code> | 2000 | Ограничение на количество возвращаемых связей в <u>SOAP-методах</u> (<code>getByHid</code> , <code>getByRawId</code> , <code>getRelations</code>) |
| <code>webservice.maxPartiesCount</code> | 2000 | Ограничение на количество возвращаемых записей для методов <code>search</code> и <code>fuzzySearch</code> |
| <code>webservice.mergeEnabled</code> | true | Разрешено ли слияния и дедубликация в soap методах? Если false — то метод работает как простой <code>save</code> |
| <code>webservice.logging.filters</code> | <pre>search=WARN:1s,CRITICAL:2s; getByHID=WARN:1s,CRITICAL:2s; getByRawID=WARN:1s,CRITICAL:2s; fuzzySearch=WARN:5s,CRITICAL:10s; saveAndMerge=WARN:2s,CRITICAL:5s; save=WARN:2s,CRITICAL:5s; saveRelations=WARN:1s,CRITICAL:2s; other=WARN:5s,CRITICAL:15s</pre> <p>С условием на количество:</p> <pre>search=WARN:1s,CRITICAL:2s,COUNT:45;</pre> | <p>Настройки перекрывают умолчательное поведение, заданное в <code>webservice.logging.defaultBehaviour</code>.</p> <p>Для каждого вида запросов можно конкретно указать таймаут и тип реакции.</p> <p>Пример: <code>search=WARN:1s,CRITICAL:5s; other=WARN:5s</code></p> <p>Расшифровывается так:</p> <ul style="list-style-type: none"> • если вызвали "search", то <ul style="list-style-type: none"> ○ если он длился больше 5 секунд — CRITICAL ○ если больше 1 секунды — WARN ○ иначе умолчательная реакция (обычно SKIP) • для остальных методов <ul style="list-style-type: none"> ○ если метод длился больше 5 секунд — WARN |

| Название параметра | Значение по умолчанию | Описание |
|--------------------|-----------------------|--|
| | | <ul style="list-style-type: none"> ○ иначе умолчательная реакция <p>На данный момент при превышении значения CRITICAL особое поведение не предусмотрено. Однако, если среднее значение выполнения запроса превышает CRITICAL, в лог выводится ошибка</p> <pre>[SoapLongopsAnalyzer# Flusher] ERROR SoapLongopsAnalyzer - Average processing time of 'name' is x.xx sec (critical=x sec)</pre> <p>Иногда запросы очень редкие и система может срабатывать на один долгий запрос — это часто некорректно, поэтому используется усреднение за период, указанный в <code>webservice.stats.periodInSeconds</code>.</p> <p>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.</p> <p>Пример: <code>search=WARN:1s,CRITICAL:2s,COUNT:45;</code></p> <p>Расшифровывается так:</p> <ul style="list-style-type: none"> • если вызвали "search", то <ul style="list-style-type: none"> ○ если за период было меньше 45 вызовов — ничего не делать, ○ если среднее больше 2 секунд — CRITICAL |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| | | <ul style="list-style-type: none"> ○ если среднее больше 1 секунды — WARN <p>Изменения данных параметров вступает в силу без рестарта системы «Гражданский фактор».</p> |
| <code>webservice.logging.sampling.periodInMs</code> | 100 | <p>Период сбора статистики при автоматической jvmt-трассировке долгих запросов для выявления причины тормозов. Чем меньше период, тем точнее трассировка, но при малых значениях возможна избыточная нагрузка на приложение — не рекомендуем изменять дефолтное значение.</p> <p>Чтобы выключить jvmt-трассировку долгих запросов, установите отрицательное значение (например, -1).</p> <p>Изменение данного параметра вступает в силу без рестарта системы «Гражданский фактор».</p> |
| <code>webservice.searchRelatedParties.response.maxEntries</code> | 20 | <p>Ограничение на количество возвращаемых пар связанных записей в методе <code>searchRelatedParties</code>.</p> <p>Возможен вариант без ограничений, для этого установите значение -1.</p> <p>Изменение данного параметра вступает в силу без рестарта системы «Гражданский фактор».</p> |
| <code>webservice.saveAndMerge.dedupAndMergeRelatedParties</code> | false | <p>Если установлено в true:</p> <ul style="list-style-type: none"> • из указанных в SaveAndMerge связей отбираем записи с сильной связью (<code>strong=true</code> в таблице <code>relation_type</code> БД); • группируем записи по типу и отправляем на |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| | | дедупликацию и слияние. |
| <code>webservice.saveAndMerge.dedupAndMergeRelatedPartiesMaximum</code> | 50 | <p>Действует только при <code>webservice.saveAndMerge.dedupAndMergeRelatedParties=true</code>.</p> <p>Если количество связанных записей одного типа больше указанного значения, то они не будут отправлены на дедупликацию и слияние.</p> |
| <code>webservice.logging.longops.ui.warnTimeThresholdInMilliseconds</code> | 1000 | <p>При достижении warn порога начинается сбор информации об использовании CPU во время выполнения запроса веб-интерфейса.</p> <div style="border: 1px solid black; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> Изменение данного параметра вступает в силу без рестарта системы «Гражданский фактор». </div> |
| <code>webservice.logging.longops.ui.criticalTimeThresholdInMilliseconds</code> | 3000 | <p>При достижении critical порога вся информация об использовании CPU во время выполнения запроса веб-интерфейса записывается в лог <code>cdi-ui-longops.log</code>.</p> <div style="border: 1px solid black; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> Изменение данного параметра вступает в силу без рестарта системы «Гражданский фактор». </div> |

4.6 Параметры задач

1. Изменения настроек вступают в силу только после рестарта сервиса CDI!
2. Настройки [фоновых задач](#) не требуют рестарта сервиса.

Параметры задач


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

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

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

| Название параметра | Значение по умолчанию | Описание |
|---|-----------------------|---|
| <code>task.physical.household.relationTypeId</code> | 8 | Идентификатор типа связи для задачи поиска "Родственник" физ.лиц (при подключении поиска домохозяйств) |
| <code>task.physical.household.hhRelationTypeId</code> | 500 | Идентификатор типа связи для задачи поиска "Домохозяйств" физ.лиц (при подключении поиска домохозяйств) |
| <code>task.sql.dbProcedure.timeout</code> | 24h | Таймаут для выполнения хранимых процедур и sql-файлов, варианты записи значения см в дочерней странице |
| <code>task.sql.connector.timeout</code> | 2h | Таймаут для коннекторов, которые возвращают данные (допустим, не удалось подключиться к БД — коннектор отваливается), варианты записи значения см в дочерней странице |
| <code>task.silent.poolSize.delta</code> | 4 | Дельта наращивания/уменьшения количества потоков для работы фоновых задач. По умолчанию — 4 потока. |
| <code>task.silent.poolSize.max</code> | -1 | Ограничение по числу используемых ядер для работы фоновых задач. По умолчанию, используется 2/3 от всех ядер сервера. |
| <code>task.silent.soapStats.minRequests</code> | other: 50 | <p>Минимальное количество вызовов для мониторинга 90-процентилей. Если количество запросов меньше указанного, то даже если они будут долгими, этот факт будет проигнорирован.</p> <p>По умолчанию — 50 вызовов для всех методов.</p> <p>Можно указывать для каждого метода свое значение (через разделитель ";").</p> |

Пример

| Название параметра | Значение по умолчанию | Описание |
|-------------------------------------|-----------------------|--|
| | | <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <pre>getByHID:40; other:21; customSave:25</pre> </div> <p>(предупреждение) Указывать только название метода, вариант "PartyWS.search" не будет работать, также обязательно указывать дефолтное ограничение "other", без него загружается.</p> |
| task.silent.soapStats.percentile90 | other:5s | <p>Показания 90-перцентиля 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 black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">Пример</p> <pre>getByHID:10s; other:5s</pre> </div> <p> Указывать только название метода, вариант "PartyWS.search" не будет работать, также <u>обязательно</u> указывать дефолтное ограничение "other", без него загружается.</p> |
| task.silent.soapStats.periodSec | 300 | Частота снятия статистики по онлайн нагрузки. По умолчанию снимается 1 раз в 5 минут. |
| task.silent.soapStats.seriesLength | 5 | Количество временных интервалов по которому принимается решение об онлайн нагрузке. |
| task.silent.nodeSync.queueSizeLimit | 1000 | Максимальный размер очереди синхронизации между нодами горячего резерва, после превышения которого фоновые задачи |

| Название параметра | Значение по умолчанию | Описание |
|--------------------|-----------------------|---|
| | | начинают сбрасывать количество потоков. |

4.6.1 Таймауты при запросах к длительным БД процедурам

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

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

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

В обоих случаях ввели таймауты. Дефолтные значения:

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

4.6.1.1 Изменение таймаута

Изменять значения можно через [параметры задачи](#)

- task.sql.dbProcedure.timeout - таймаут для выполнения хранимых процедур и sql-файлов
- task.sql.connector.timeout - таймаут для коннекторов, которые возвращают данные

где значение может выглядеть, например, так:

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

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

4.7 Параметры задачи дедубликации

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

Напр. для физических лиц:

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

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

4.8 Параметры запуска JBOSS

При старте система проверяет наличие следующих параметров запуска для JVM:

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

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

Параметры настраиваются в:

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

У пользователя 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 |
|--|----------------|-------------------|
| <code>cdi.root.folder</code> | полный | не требуется |
| <code>cdi.properties.folder</code> | полный | не требуется |
| <code>cdi.dedup.folder</code> | полный | полный |
| <code>cdi.search.folder</code> | полный | не требуется |
| <code>cdi.services.state.mapdb.folder</code> | полный | не требуется |

4.9 Параметры импорта из буферной области

Изменения настроек вступают в силу только после рестарта сервиса CDI!

Параметры импорта из буферной области

| Название параметра | Значение |
|---------------------------------------|----------|
| <code>oldMaxId.buffer.physical</code> | 21839 |

| Название параметра | Значение по умолчанию | Описание |
|-----------------------|-----------------------|--|
| <code>oldMaxId</code> | 0 | Последний (максимальный) идентификатор |

4.10 Параметры импорта справочников

| Название параметра | Значение по умолчанию | Описание |
|---------------------------------------|-----------------------|--|
| <code>dictionary.conf.egr.date</code> | | Дата актуальности загруженного справочника ЕГРЮЛ |
| <code>dictionary.conf.egr.hash</code> | | Контрольная сумма загруженного справочника ЕГРЮЛ |

4.11 Параметры модуля мониторинга

Изменения настроек вступают в силу только после рестарта сервиса CDI!

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|---|
| monitoring.applicationHealth.badHealthRefreshPeriodInSeconds | 600 | Как часто (в секундах) логировать состояние приложения, находящегося в критическом состоянии. |
| monitoring.applicationHealth.badHealthThreshold | 0.95 | Если отношение занятой памяти к общему её объему больше чем установленное значение, состояние приложения считается критическим. |
| monitoring.applicationHealth.goodHealthThreshold | 0.85 | Если отношение занятой памяти к общему её объему меньше чем установленное значение, состояние приложения считается хорошим. |
| monitoring.applicationHealth.refreshPeriodInSeconds | 10 | Как часто (в секундах) проверять состояние приложения. |
| monitoring.applicationHealth.http.httpPort | 8080 | Http-порт, на котором ведется статистика активных подключений. |
| monitoring.applicationHealth.http.httpsPort | 8443 | Https-порт, на котором ведется статистика |

| Название параметра | Значение по умолчанию | Описание |
|---|-----------------------|---|
| | | активных подключений . |
| monitoring.applicationHealth.http.connectionsToLogThreshold | 100 | Число активных подключений , при превышении которого в лог будет записываться информация о размере входящей очереди сообщений. |
| monitoring.applicationHealth.http.workerQueueSizeToLogThreshold | 10 | Число очередей приложения, при превышении которого в лог будет записываться информация о размере входящей очереди сообщений. |
| monitoring.configuration.getLocalHostTimeout | 1s | Время идентификации имени машины, на которой разворачивается система «Гражданский фактор». При превышении в лог пишется сообщение уровня ERROR. |
| monitoring.nodeSync.jmsLatencyThresholdMs | 5000 | Время (в миллисекундах) задержки сообщений в очереди . |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| | | При превышении в лог пишется сообщение уровня |
| monitoring.nodeSync.loggingPeriodSec | 300 | Как часто (в секундах) логировать размер очереди сообщений между нодами в ГР и время задержки сообщений в очереди. |
| monitoring.applicationHealth.queueSizeAlert.alertPeriodSec | 86400 (24 часа) | Периодичность логирования в лог предупреждения о размере очереди |
| monitoring.applicationHealth.queueSizeAlert.queueSizeThreshold | 30000 | Порог отправки почтового уведомления о превышении размера очереди синхронизации нод |

4.12 Параметры отправки email

Изменения настроек вступают в силу только после рестарта сервиса системы «Гражданский фактор»!

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

| Название параметра | Значение по умолчанию / Пример | Описание |
|----------------------------|--------------------------------|---|
| Общие параметры | | |
| mail.from | automation@cleandata.ru | Имя почтового ящика, с которого будет производиться рассылка |
| mail.host | smtp.gmail.com | Адрес smtp сервера, с которого будет производиться рассылка |
| mail.debug | false | Флаг, отвечающий за отправку сообщений DEBUG-уровня |
| mail.user | automation@cleandata.ru | Логин для подключения к серверу рассылки |
| mail.password | *** | Пароль для подключения к серверу рассылки |
| mail.port | -1 | Порт для отправки почты (если отличается от умолчательного для протокола) |
| mail.protocol | smtps | Протокол для отправки почты (smtp / smtps) |
| Диагностическая информация | | |

| Название параметра | Значение по умолчанию / Пример | Описание |
|---|---|--|
| mail.diagnostic.enabled | false | Флаг, отвечающий за отправку диагностик и на почту |
| mail.diagnostic.to | log.catcher@cleandata.ru | Список получателей (разделитель - ", ") собранной диагностик и |
| mail.diagnostic.subject | «Гражданский фактор» {nodeId} - диагностика | Формат заголовка сообщения о собранной диагностик и |
| mail.diagnostic.body | Профили: {profiles}. Параметры: {parameters}. | Содержание сообщения о собранной диагностик и |
| mail.diagnostic.defaultProfiles | config, db.connection, logs, search, sql.counts, sql.metadata, versions | Умолчательный набор профилей |
| Автоматическая миграция приложения | | |
| mail.migration.enabled | false | Флаг, отвечающий за отправку оповещений миграции на почту |
| mail.migration.to | log.catcher@cleandata.ru | Список получателей (разделитель - ", ") оповещений миграции |
| mail.migration.subject.rebuildIndexes | «Гражданский фактор» {nodeId} обновляется на версию {newVersion} | Формат заголовка оповещения о перестроен |

| Название параметра | Значение по умолчанию / Пример | Описание |
|--|---|--|
| | | ии индексов |
| mail.migration.body.rebuildIndexes | Обновление с версии {oldVersion} на {newVersion}. Будут перестроены индексы ({indexes}). | Содержание оповещения о перестроении индексов |
| mail.migration.subject.diagnostic | «Гражданский фактор» {nodeId} обновился на версию {newVersion} | Формат заголовка оповещения об окончании миграции |
| mail.migration.body.diagnostic | Обновление с версии {oldVersion} на {newVersion}. Диагностика во вложении ({profiles}). | Содержание оповещения об окончании миграции |
| Автоматический мониторинг логов | | |
| mail.logging.enabled | false | Флаг, отвечающий за отправку сообщений об ошибках на почту |
| mail.logging.to | log.catcher@cleandata.ru | Список получателей (разделитель - ", ") сообщений об ошибках |
| mail.logging.subject.single | CDI Error Occurred @ \${cdi.customer.name} ({errorName}) | Формат заголовка сообщения об одиночной ошибке |
| mail.logging.subject.multiple | CDI Errors Occurred @ \${cdi.customer.name} during {reportDate} ({lastTime} - {reportTime}) | Формат заголовка агрегированного сообщения об ошибках, произошедших за определен |

| Название параметра | Значение по умолчанию / Пример | Описание |
|---------------------------------------|--|---|
| | | ный период |
| mail.logging.bufferSize | 10 | Количество блоков информации, предшествующих ошибке |
| mail.logging.accumulatePeriod | 3600 | Частота оповещения в секундах |
| mail.logging.skipEventPredicate1(2-5) | errorClass.equals('IncorrectVersionException') | <p><u>Фильтр событий ошибок</u> к отправке по почте. Если ошибка попадает под шаблон, то оповещение о ней не будет выслано.</p> <p>Изменения данных параметров вступает в силу без рестарта системы «Гражданский фактор».</p> |

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

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

Пример:

- `mail.diagnostic.subject = «Гражданский фактор» {nodeId}` - диагностика.
- В `standalone.conf.bat` задан параметр `-Dcdi.app.nodeId=MY_NODE_1`.

- Тема отправленного письма — «Гражданский фактор» MY_NODE_1 - диагностика».

4.12.1 Механизм отправки писем об ошибках на почту

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

При возникновении ошибки система «Гражданский фактор» 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.

4.12.1.2 Пример 1

В течении часа произошло:

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

На почту придет:

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

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

4.12.1.3 Пример 2

В течении часа произошло:

- 10:05 — 2 ошибки `NotAuthorizedLocalUserAlert`
- 10:05 — ошибка `IllegalArgumentException`
- 10:05 — 2 ошибки `AccessDeniedException`

- 10:36 — ошибка `AccessDeniedException`

На почту придет:

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

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

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

| | |
|--|---|
| <code>NotAuthorizedLocalUserAlert</code> | 1 |
| <code>AccessDeniedException</code> | 2 |

4.13 Параметры подключения к Фактору

Изменения настроек вступают в силу только после рестарта сервиса CDI!

| Название параметра | Значение по умолчанию | Описание |
|--|--|---|
| <code>factor.blacklist.cdi.context</code> | <code>/factor-service-{customer}/services/CdiBlackListService</code> | WSDL сервиса поиска по Черным спискам |
| <code>factor.blacklist.cdi.timeout</code> | 600000 | Тайм-аут ожидания ответа от Фактора во время поиска по Черным спискам, в миллисекундах |
| <code>factor.clean.context</code> | <code>/factor-service-{customer}/services/CleanService</code> | WSDL CleanService |
| <code>factor.clean.packetSize</code> | 200 | Размер пакета на стандартизацию / трансформацию |
| <code>factor.clean.timeout</code> | 1200000 | Тайм-аут ожидания ответа от Фактора во время стандартизации / трансформации (для каждого пакета), в миллисекундах |
| <code>factor.dedup.context</code> | <code>/factor-service-{customer}/services/DedupService</code> | WSDL сервиса поиска дубликатов |
| <code>factor.dedup.full.threadCount</code> | -1 | Количество потоков для выполнения полной дедубликации в Факторе |

| Название параметра | Значение по умолчанию | Описание |
|------------------------------------|---|---|
| | | (автоматически, если указан "-1"). Эти же настройки используются для оффлайн проверки по ЧС / ЕГРЮЛ. |
| factor.dedup.full.threadsPriority | -1 | 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 |
| factor.dedup.incremental.queueSize | 10000 | 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) |
| factor.dedup.timeout | 600000 | Тайм-аут ожидания ответа от Фактора во время поиска дубликатов, в миллисекундах |
| factor.factor.context | /factor-service-{customer}/services/FactorService | WSDL Фактора |
| factor.factor.timeout | 600000 | Тайм-аут ожидания ответа от Фактора |

| Название параметра | Значение по умолчанию | Описание |
|-------------------------------|-----------------------|--|
| factor.health.timeoutMs | 1000 | <p>Время ожидания ответа на проверку хэлс-чека Фактора</p> <p>✓ Изменение данного параметра применяется без рестарта модуля CDI</p> |
| factor.host | http://host:port | <p>Базовый URL Фактора в формате http://host:port (используется для поиска дубликатов)</p> |
| factor.hosts | http://host:port | <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 "[количество_потоков]".</p> <p>Например:</p> <pre>http://host1:18080[2]; http://host2:18080</pre> <p>означает, что 2 потока будут посылать запросы на http://host1:18080 и 1 на http://host2:18080.</p> |
| factor.ws.numberOfRetries | 1 | <p>Количество попыток повторного вызова Фактора при наличии сетевой ошибки</p> |
| factor.ws.delayBetweenRetries | 1000 | <p>Задержка в мс между попытками повторного вызова Фактора при наличии сетевой ошибки</p> |

| Название параметра | Значение по умолчанию | Описание |
|-------------------------------------|-----------------------|---|
| <code>factor.ws.chunkingMode</code> | ALWAYS | <p>Режим использования Chunked Encoding при обращении к серверу:</p> <ul style="list-style-type: none"> • ALWAYS — всегда использовать; • REMOTE — использовать только для удаленных серверов; • NEVER — никогда не использовать. <p>При возникновении ошибок "<i>Marshalling Error: Error writing request body to server</i>" рекомендуется переключить в REMOTE.</p> |

4.14 Параметры поискового индекса

Изменения настроек вступают в силу только после рестарта сервиса CDI!

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|---|
| <code>engine.rebuild.freeSpaceRatio</code> | 1.5 | <p>Коэффициент свободного места, которое должно быть при очередном запуске задачи перестроения индекса</p> <p>Если значение <1, то проверка свободного места отключена</p> |
| <code>engine.rebuild.pageSize</code> | 10000 | Размер страницы (количество) |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| | | контрагентов) при перестроении индекса |
| <code>engine.rebuild.poolSize</code> | -1 | Количество потоков для перестроения индекса Если равно -1, устанавливается по формуле: $\min(\text{количество процессоров (ядер)} * 2, 8)$ |
| <code>engine.rebuild.skipOptimize</code> | false | Пропускать ли оптимизацию при перестроении |
| <code>engine.rebuild.sortedIterator</code> | true | Добавлять ли сортировку в итератор по актуальным контрагентам. По умолчанию true, поменять можно без ребута систем «Гражданский фактор». 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 записей делать это быстрее, потому что попадаем в соседние листы индекса. |
| <code>engine.rebuild.strategy</code> | BLOCKING | Стратегия перестроения индекса: <ul style="list-style-type: none"> • BLOCKING — во время перестроения блокируются все модификации индекса, рядом строится новый свежий индекс, которым потом |

| Название параметра | Значение по умолчанию | Описание |
|--------------------|-----------------------|--|
| | | <p>подменяем старый;</p> <ul style="list-style-type: none"> • CONCURRENT — состоит из двух шагов (применима только для индексов контрагентов; граф / группы дублей / конфликты перестраиваются по BLOCKING алгоритму): • в существующем индексе обновляются все документы, индекс оптимизируется; • поиск неконсистентных документов, которых в индекс быть не должно: проверяются все документы старше, чем начало полного перестроения — по ним отдельно проходит проверка актуальности и те, которых быть не должно, удаляются. <p>Если у вас нет проблем с онлайн-потоками, то эффективнее использовать BLOCKING . Преимущества — она быстрее и требует меньше места. Минусы — запоздания в попадании данных в индекс + ограничение очереди (пока строит индекс, все изменения по записям</p> |

| Название параметра | Значение по умолчанию | Описание |
|------------------------------|-----------------------|--|
| | | <p>складываются в очередь. В search ответе вернутся только после окончания перестроения)</p> <p>Если онлайн идет постоянно, используйте CONCURRENT</p> |
| engine.search.bufferSize | 1000 | <p>Размер буфера коммита изменений поискового индекса на диск.</p> <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> |
| engine.search.commitInterval | 5000 | <p>Частота коммита изменений поискового индекса на диск.</p> <p>Все операции из очереди группируются в памяти, пока не сработает таймер коммита.</p> <p>Измеряется в миллисекундах. Не рекомендуется</p> |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|---|
| | | устанавливать менее 1000 мс. |
| <code>engine.search.maxBufferedDocs</code> | 1000 | Максимальное количество документов в памяти до сброса на диск |
| <code>engine.search.needOptimizeWhenOpen</code> | false | Проводить оптимизацию индекса при старте системы? |
| <code>engine.search.queueSize</code> | 20000 | Размер очереди операций по модификации индекса. Если очередь полностью заполняется, на все последующие операции модификации индекса возвращается ошибка. |
| <code>engine.search.indexDirectoryType</code> | AUTO | Способ чтения файлов индекса (тип директории — AUTO / MMAP / SIMPLE / NIO / RAM) для всех индексов, кроме групп дубликатов. Можно переключать с AUTO (который на современных серверах всегда MMAP) на NIO. Это замедлит скорость отдельно взятых процессов, но в сумме дает большую стабильность сервера и прогнозируемость. |
| <code>engine.search.graph.indexDirectoryType</code> | AUTO | Способ чтения файлов индекса (тип директории — AUTO / MMAP / SIMPLE / NIO / RAM) для графа связей. |
| <code>engine.search.groups.indexDirectoryType</code> | AUTO | Способ чтения файлов индекса (тип директории — AUTO / MMAP / SIMPLE / NIO / RAM) для групп дубликатов. |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| | | Если есть проблемы с ММАР, то в первую очередь стоит поменять доступ на NIO для групп дубликатов. |
| engine.search.groups.maxFilteredPartyCount | 100000 | Максимальное допустимое количество контрагентов, которое может вернуть поиск по хидам записей, инициированный фильтрацией на группах дубликатов. Если будет найдено больше, то ошибка: Ваш запрос находит слишком много записей. Уточните условие запроса. |
| engine.search.supportLegacyNumericFields | true | Флаг для переключения формата полей, которые пишутся в индекс на новый Lucene 6. Если false, то все запросы начинают работать по новым полям Lucene 6, старый формат полей не пишется в индекс. Параметр удален в релизе 18.15. |
| engine.search.useNIOForIterateIndex | false | Флаг использования доступов через NIO для итераторов по всему индексу, в частности, при выгрузке кандидатов для поиска дубликатов. |
| engine.validation.maxAcceptableMismatch | -1 | Максимально допустимое расхождение при валидации между индексом и БД. По умолчанию -1, на любое расхождение будет ERROR-сообщение в лог. |

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|--|
| <code>engine.search.queueWorkersCount</code> | 1 | 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. |

4.14.1 Неблокирующее перестроение поискового индекса

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

4.14.1.1 Стратегия перестроения

Состоит из двух шагов:

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

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

4.14.1.2 Возобновляемость

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

4.14.1.3 Требования к месту

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

4.14.1.4 Рекомендации по запуску

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

Для быстрой смены в параметрах задачи можно напрямую задать `poolSize` без перегрузки системы «Гражданский фактор», который будет использоваться вместо умолчательного значения `engine.rebuild.poolSize`.


4.15 Параметры сервисов

Изменения настроек вступают в силу только после рестарта сервиса CDI!

| Название параметра | Значение по умолчанию | Описание |
|--|-----------------------|---|
| <code>services.actualityDate.truncateTime</code> | true | Если true, то для даты актуальности используется тип данных DATE, иначе — TIMESTAMP (для переключения на TIMESTAMP нужна перепоставка релиза). Важно: Перед включением настройки необходимо обратиться в поддержку системы «Гражданский фактор». |
| <code>services.applicationMigration.enabled</code> | false | Флаг включения миграции версий приложения |
| <code>services.applicationMigration.rebuildInvalidIndexes</code> | false | Флаг автоматического перестроения невалидных индексов на миграции приложения |
| <code>services.cache.refreshIntervalInMin</code> | 60 | Периодичность сбора кэша справочников (в минутах). |
| <code>services.dao.versioning.defaultPhase</code> | — | Умолчательная фаза миграции для таблиц, |

| Название параметра | Значение по умолчанию | Описание |
|---|-----------------------|---|
| | | отсутствующих в MIGRATION_PHASE |
| services.deduplication.autoMatchScope | 98 | Пороговое значения коэффициента схожести для гарантированных дубликатов по умолчанию. |
| services.deduplication.autoMatchScopeByPartyType | — | Пороговое значения коэффициента схожести для гарантированных дубликатов для конкретных типов PARTY (заполняется только в случае, если значения разные, например - PHYSICAL : 85 ; LENDER : 95). |
| services.deduplication.byPartyHitsFullScanIteratorThreshold | 2000000 | Максимальный размер контрагентов для итерирования фулсканом по дубликатам при инкрементальной загрузке |
| services.deduplication.matchScopeThreshold | 0 | Минимальный порог схожести дубликатов для создания дубликата (все, что ниже — уже не дубли, а домохозяйства) |
| services.deduplication.matchScopeThresholdByPartyType | PHYSICAL : 50 | Минимальный порог схожести дубликатов для создания дубликата по типам контрагентов |
| services.duplicateGroup.bigGroup.duplicatesCountToLoad | 100000000 | Максимальное количество дубликатов, загружаемое для большой группы. |
| services.duplicateGroup.bigGroup.partyCountThreshold | 200 | Количество контрагентов, начиная с которого группа считается большой и обрабатывается особым образом. |
| services.deduplication.maxDuplicatesForGroupsCalculation | 2000000 | Максимальный размер множества дублей, загружаемых в память, для построения группы дубликатов. |
| services.lifecycleEvents.persistent | true | Сохранять ли в таблице logging события старта-остановки приложения |
| services.merge.attribute.mergePackSize | 50 | Допустимое количество атрибутов, которые можно сливать "каждый с каждым" (при превышении делить на группы) |

| Название параметра | Значение по умолчанию | Описание |
|---|-----------------------|--|
| <code>services.merge.findCycles</code> | <code>true</code> | Если <code>false</code> , то поиск циклов не производится. Группа дубликатов отмечается как ошибочная при подозрении на цикл. Если <code>true</code> , то определяются атрибуты-победители без поиска циклов. Если в результате есть несовпадение ожидаемого и фактического количества атрибутов, происходит поиск циклов. |
| <code>services.party.includeSourceToOriginal</code> | <code>false</code> | Возвращать в ответах SOAP-методов для исходной карточки тег <code>source</code> ? |
| <code>services.pastValues.enabled</code> | <code>false</code> | Включить сервис использования прошлых атрибутов для поиска дублей |
| <code>services.personalData.mode</code> | <code>DISABLED</code> | <ul style="list-style-type: none"> • <code>DISABLED</code> – логируем все в основной лог, ничего не скрываем • <code>SEPARATE</code> – логируем перс.данные в отдельный лог <code>cdi-personal-data.log</code>, а в основном ставим ссылку, чтобы можно было найти • <code>HIDDEN</code> – в основном лог перс.данные скрываем, и никуда их не пишем (<code>paranoid mode</code>) |
| <code>services.relation.enabled</code> | <code>false</code> | Включить сервис расчета графа взаимосвязей. |
| <code>services.relations.fillSourceIdByDefault</code> | <code>false</code> | Флаг включения расширенного оповещения JMS очереди о создании связей. Если <code>true</code> , то в сообщении о создании/удалении связи для исходных карточек помимо <code>hid</code> будет указываться <code>rawId</code> и <code>sourceSystem</code> . Для золотых карточек и вспомогательных сущностей (например, Сегментов) всегда будет указан только <code>hid</code> . Важно: Включение |

| Название параметра | Значение по умолчанию | Описание |
|--|--|--|
| | | данной настройки может увеличить скорость выполнения запросов. |
| <code>services.relation.maxRelatedEntitiesCount</code> | 100 | Максимальное число связанных сущностей, которое уходит в индекс |
| <code>services.sql.paramsLoggingEnabled</code> | true | Флаг записи параметров sql-запросов в <code>cdi.log</code> . |
| <code>services.sqlProfiler.thresholdInSeconds</code> | 60 | Пороговое значение для логирования sql-запросов. Запрос, выполняющийся дольше установленного значения, будет сохранен в логе.  Рестарт CDI не требуется |
| <code>services.state.mapdb.useMmap</code> | true | Использовать mmap для работы с локальной базой (файловая), в которой хранятся даты последнего обновления индекса. Если <code>false</code> , то mmap не используется — используется обычный доступ к файлам |
| <code>services.suggestions.enabled</code> | false | Флаг включения подсказок для адресов |
| <code>services.suggestions.token</code> | {api-ключ} | Ключ доступа к облачным подсказкам |
| <code>services.suggestions.upperCase</code> | true | Флаг перевода значений из подсказок в верхний регистр при подстановке |
| <code>services.suggestions.url</code> | <code>https://dadat.a.ru/api/v2</code> | URL сервиса подсказок. При использовании <code>standalone-подсказок</code> заменить на: <code>http://{ip}:{port}/suggestions/api/4_1/rs.</code> |
| <code>services.transform.closeMissedAttributes</code> | false | Закрывать ли при обновлении атрибуты, которые отсутствуют в обновлении? |
| <code>services.transform.updateConflictStrategy</code> | DISABLED | Стратегия мониторинга конфликтующих изменений. Для подключения выставить в значение <code>SKIP_UPDATE_AND_SAVE</code> . |

| Название параметра | Значение по умолчанию | Описание |
|---|---------------------------------|---|
| <code>services.transform.updateConflictStrategyByPartyType</code> | | Стратегия мониторинга конфликтующих изменений по типам контрагентов. Формат: «PARTY_TYPE: STRATEGY » |
| <code>services.rules.update.log</code> | false | Флаг включения сработавших правил обновления ✔ Рестарт CDI не требуется |
| <code>services.rules.merge.log</code> | false | Флаг включения сработавших правил слияния ✔ Рестарт CDI не требуется |
| <code>services.rules.skipLoggingForRules</code> | oldWins, reqIfEqualsThenOldWins | Перечень событий (правил), которые не нужно логировать ✔ Рестарт CDI не требуется |
| <code>services.duplicateGroup.actualFilteringMode</code> | DAO | Режим фильтрации при отборе дубликатов: <ul style="list-style-type: none"> • DAO — через БД. • GRAPH — через граф связей. Перестроение групп дублей работает быстрее, чем при фильтрации через БД. • DISABLED — фильтрация отключена. Есть вероятность отобразить неактуальный дубликат. |
| <code>services.party.massive.sourceCountThreshold</code> | 1000 | Параметр, определяющий, что карточка большая |

4.15.1 Стратегия решения конфликтов

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

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

После смены стратегии перезапустите систему «Гражданский фактор».

4.16 Параметры синхронизации нод

Изменения настроек вступают в силу только после рестарта сервиса CDI!

Параметры синхронизации нод

| Название параметра | Значение |
|--|--|
| nodeSync.jms.enabled | <input type="text" value="true"/> |
| nodeSync.jms.queue.name | <input type="text" value="cdi.nodeSync"/> |
| nodeSync.jms.receive.connectionFactoryjndiName | <input type="text" value="nodeSyncRemoteConnectionFactory"/> |
| nodeSync.jms.receive.maxInterval | <input type="text" value="300000"/> |
| nodeSync.jms.receive.password | <input type="text" value="cdi"/> |
| nodeSync.jms.receive.recoveryInterval | <input type="text" value="30000"/> |
| nodeSync.jms.receive.username | <input type="text" value="sync"/> |
| nodeSync.jms.send.connectionFactoryjndiName | <input type="text" value="jmsConnectionFactory"/> |

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

| Название параметра | Значение по умолчанию | Описание |
|---------------------------------------|-----------------------|---|
| nodeSync.jms.receive.password | cdi | Пароль авторизации для соединения с очередью второй ноды |
| nodeSync.jms.receive.recoveryInterval | 30000 | Интервал ожидания между попытками возобновить соединение (в миллисекундах). Значение увеличивается экспоненциально с коэффициентом 1.5 (т.е. проверки будут 30 сек → 45 сек → 68 сек и т.д. до 5 мин). |
| nodeSync.jms.receive.maxInterval | 300000 | Максимальный интервал ожидания между попытками возобновить соединение (в миллисекундах). |

4.17 Параметры экспорта данных

Параметры экспорта данных

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

Сохранить

| Название параметра | Значение по умолчанию | Описание |
|-------------------------------------|-----------------------|---|
| incrementalExportRelations.lastDate | 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.

5 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](#)

5.1 Добавление в справочник новой записи

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

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

5.2 Модификация существующей в справочнике записи

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

| Параметр | Обязательность | Значение по умолчанию | Описание |
|---------------|----------------|---|---|
| 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 сущностей (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 |

5.3 Представление с содержимым справочников

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

5.4 Список справочников, которые можно пополнять и редактировать

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

6 Журналирование и аудит

- [Журналы системы «Гражданский фактор»](#)
- [Журналы Фактора](#)
- [Уровни протоколирования](#)
- [Аудит](#)

6.1 Журналы системы «Гражданский фактор»

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 |
|---|---|---------------------------------------|
| <code>console.log</code> | Консольный вывод джавы. Перезатируется при рестарте приложения. | всегда при старте jboss |
| <code>server.log</code> | Системные сообщения и сообщения от сервера JBOSS. | всегда при старте jboss |
| <code>cdi.log</code> | Основной журнал работы системы «Гражданский фактор». Не содержит стек-трейсов. | всегда при старте cdi |
| <code>cdi-full.log</code> | Основной журнал работы системы «Гражданский фактор». Содержит стек-трейсы. Доступен по адресу <code>СЕРВЕР_СИСТЕМЫ:ПОРТ_СИСТЕМЫ/cdi/api/manage/logfile</code> | 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 |
| <code>cdi-health.log</code> | Журнал состояния памяти приложения. Содержит <code>treaddump</code> , собранный при превышении минимального порога доступной приложению памяти. | всегда при старте cdi |
| <code>cdi-jms.log</code> | 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 |

| Журнал | Описание | условие создания |
|---|--|-----------------------|
| 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 — webservice.logging.filters | всегда при старте cdi |
| cdi-lifecycle.log | Обновления версий сервиса и события старта/стопа приложения. Лог не пересоздается. | При первом старте cdi |
| cdi-personal-data.log | 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 |
| cdi-task.log | Журнал, в который копируется вся информация о работе задач из cdi.log | всегда при старте cdi |
| cdi-distributed-stats.log | Журнал метрик размера очереди и задержки сообщений в очереди между нодами в горячем резерве. | всегда при старте cdi |
| cdi-stats-avg.log | Журнал производительности | всегда при старте cdi |
| cdi-distributed-state.log | Журнал информационных сообщений, относящихся к индексам. Сообщения вынесены из основного лога. | всегда при старте cdi |
| cdi-ui-longops.log | Журнал трассировки долгих запросов веб-интерфейса. Настройка критичного времени ответа — webservice.logging.longops.ui.criticalTimeThresholdInMillis | всегда при старте cdi |
| cdi-ui-stats.log | Журнал статистики запросов веб-интерфейса | всегда при старте cdi |

6.1.1 Журнал (лог) производительности

Данный лог доступен в версии системы «Гражданский фактор» 20.9 и выше.

6.1.1.1 cdi-stats-avg.log

Статистика как общей, так и создаваемой системой «Гражданский фактор» 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.

6.1.1.2 Описание параметров

| название параметра | единица измерения | описание | возможные проблемы/рекомендации |
|--------------------|-----------------------------|--|---|
| | | Дата и время в формате: <i>yyyy-MM-dd 'HH:mm:ss'</i> | |
| cpu_proc | от 0 до 1, где 1 — это 100% | нагрузка на cpu от системы «Гражданский фактор» | |
| cpu_user | от 0 до 1, где 1 — это 100% | Нагрузка на cpu от всех приложений, работающих на сервере | Если на сервере стоит только система «Гражданский фактор» и показатель > 1%, проблемы с нагрузкой от других приложений, подключать админов со стороны заказчика. Если стоят системы «Гражданский фактор» + 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 система «Гражданский фактор» | |

| название параметра | единица измерения | описание | возможные проблемы/рекомендации |
|--------------------|--|---|---|
| jvm_eden | ГБ | Динамическая память, которая используется системой «Гражданский фактор» | 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 | |

| название параметра | единица измерения | описание | возможные проблемы/рекомендации |
|--------------------|-------------------|--|---|
| 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. |

6.1.1.3 Как настроить

По умолчанию статистика собирается каждые 10 секунд.

```
monitoring.applicationHealth.refreshPeriodInSeconds
```

Code Block 6 Настройка в панели администратора

6.1.1.4 Пример журнала

```

cpu_steal  gc_pause  cpu_proc  cpu_user  cpu_sys  cpu_io  cpu_irq
mem_vss    mem_swap  files     threads  jvm_eden jvm_old  mem_rss
disk_tmp   net_rb    disk_q    disk_bi   disk_bo   disk_root
net_re     net_sb    net_se
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
441.94/468.361396169 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
441.94/468.361457102 0      1325446 0

```

6.1.2 Журнал (лог) сетевых соединений

Данный лог доступен в версии системы «Гражданский фактор» 21.12 и выше.

6.1.2.1 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 запущена система «Гражданский фактор».

Для диагностики проблем, связанных с сетью:

- поиск сетевых проблем;
- определение производительности сети.

6.1.2.2 Описание параметров

| значения состояния сокета | единица измерения | описание |
|---------------------------|--------------------|--|
| | | Дата и время в формате: <i>yyyy-MM-dd 'HH:mm:ss'</i> |
| listen | количество сокетов | Ожидает входящих соединений. |
| syn_sent | количество сокетов | Активно пытается установить соединение. |
| syn_received | количество сокетов | Идет начальная синхронизация соединения. |
| established | количество сокетов | Соединение установлено. |
| fin_wait_1 | количество сокетов | Сокет закрыт; отключение соединения. |
| fin_wait_2 | количество сокетов | Сокет закрыт; ожидание отключения удаленной стороны. |
| close_wait | количество сокетов | Удаленная сторона отключилась; ожидание закрытия сокета. |
| closing | количество сокетов | Сокет закрыт, затем удаленная сторона отключилась; ожидание подтверждения. |
| last_ack | количество сокетов | Удаленная сторона отключилась, затем сокет закрыт; ожидание подтверждения. |
| time_wait | количество сокетов | Сокет закрыт, но ожидает пакеты, ещё находящиеся в сети для обработки. |
| closed | количество сокетов | Закрыт. Сокет не используется. |

6.1.2.3 Как настроить

По умолчанию статистика собирается каждые 10 секунд.

Порт для мониторинга 8080:

```
monitoring.netstat.port=8080
```

Code Block 7 Настройка в панели администратора

6.1.2.4 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' | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 4 | 0 | | | | | 0 | | | | |
| 2021-04-01 | | | | | | | | | | | |
| '00:07:10' | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 4 | 0 | | | | | 0 | | | | |

6.1.3 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.

6.1.3.1 Формат

```
{datetime} {max_queue_size} {max_latency, ms}
```

6.1.3.2 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> | <code>ms</code> | Максимальное время задержки применения сообщения за период (разница между временем применения и временем создания сообщения) |

6.1.3.3 Пример

```
datetime                               max_queue_size
max_latency, ms
2020-03-12 '21:45'                      34                122
2020-03-12 '21:50'                      2                 45
```

6.1.3.4 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`

6.1.4 Журнал аутентификаций

```
cdi-security.log
```

6.1.4.1 Стандартное журналирование

По умолчанию в `cdi-security.log` записываются:

1. Неудачные попытки авторизации:
`{Timestamp} {Имя потока} WARN authentication - Authentication failure. Principal: {логин пользователя}. Cause by: {причина ошибки} (Source: {UI - при авторизации в веб-интерфейсе CDI, API - при авторизации на веб-сервисах} ;RemoteAddress: {IP-адрес, с которого производилась попытка логина})`

2. **Сообщения о нехватке прав для выполнения действия:**
{Timestamp} {Имя потока} WARN authorization - Недостаточно прав на {Запрошенное действие}. Требуются права: {Необходимые права}. Текущие роли: {Роли пользователя} (Source: {UI - при авторизации в веб-интерфейсе CDI, API - при авторизации на веб-сервисах} ;RemoteAddress: {IP-адрес, с которого производилась попытка логина})
3. **Долгие запросы к Active Directory:**
{Timestamp} {Имя потока} WARN LongopsLdapBindAuthenticator - Long running LDAP request {длительность запроса}. Username: {логин пользователя}, WS-Operation: {название метода веб-интерфейса}, WS-MessageID: {ссылка на журнал cdi-soap-trace.log}
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.cleandata.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.cleandata.cdi.exception.security.ConflictingRolesAuthenticationException: Found conflicting roles [ROLE_OPERATOR, ROLE_PERFORMER] for user 'superuser'
```

6.1.4.2 Расширенное журналирование

Для журналирования всех попыток аутентификации поменяйте уровень журналирования с INFO на TRACE для AuthEventListener в файле

JBOSS_CDI_HOME\standalone\tmp\vfs\temp*** (с наибольшей датой актуальности)\cdi-web-*.war\WEB-INF\classes\logback.xml

```
<logger
name="ru.cleandata.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 системы «Гражданский фактор».

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.

6.1.5 Журнал доступа к персональным данным

```
cdi-data-access.log
```

6.1.5.1 Доступ к персональным данным

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

6.1.5.2 Примеры

Пример 1. Вызов метода save через API (веб-сервис PartyWS) вернул информацию по записи 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 записям с идентификаторами 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 записи через веб-интерфейс:

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

6.1.6 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`.

Система «Гражданский фактор» умеет фильтровать в лог `cdi-minor-errors` предупреждения – сообщения с типом `WARN`.

6.1.6.1 Особенности

Лог создается всегда, даже если условия фильтрации не заданы.

Включен в сбор диагностики.

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

6.1.7 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.

6.1.8 Журнал обновлений версий сервера

Журнал отслеживает:

- миграции БД системы «Гражданский фактор» 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

6.1.9 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)
```

6.1.10 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](#)

6.1.10.1 Логирувание 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
```

6.1.10.2 Задача дошла до конца без ошибок

- APM Администратора, статус - *FINISHED*
- APM 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
```

6.1.10.3 Задача дошла до конца, но были ошибки

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

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

- Логи

```
INFO interceptor.TaskPerformerStatsInterceptor - ERROR task
'PhysicalBufferIPC [bufferUploadTaskPerformer]';
BUFFER: processed 1 entities (create: 1); 0 with errors;
took 1 seconds or 19,26 ms; speed 1.0 records/sec
TRANSFORM: processed 1 entities (error: 1); 1 with errors;
took 1 seconds or 192,6 ms; speed 1.0 records/sec
ERROR manager.TaskWorker - Task 'bufferUploadTaskPerformer'
finished with error: Task finished with 1 errors
```

6.1.10.4 Задача остановилась из-за ошибок

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

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

- Логи

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

6.1.10.5 Подсчет ошибок

Кол-во ошибок считается:

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

6.1.11 Лог триггера

- [Все задачи прошли успешно](#)
- [Задача упала, триггер остановился](#)
- [Задача упала, триггер продолжил работу](#)

6.1.11.1 Все задачи прошли успешно

В логе есть старт и финиш триггера, а также его шаги в виде [n/m]

```
INFO task.TaskTriggerExecutor - Trigger 'rebuildAllIndexesTrigger'
STARTED manually
INFO task.TaskTriggerExecutor - [1/3] step 'graphEngineFullRebuild' of
trigger 'rebuildAllIndexesTrigger' STARTED
INFO task.TaskLauncherTemplate - Task graphEngineFullRebuild FINISHED
INFO task.TaskTriggerExecutor - [2/3] step 'enginesFullRebuild' of
trigger 'rebuildAllIndexesTrigger' STARTED
INFO task.TaskLauncherTemplate - Task enginesFullRebuild FINISHED
INFO task.TaskTriggerExecutor - [3/3] step 'duplicateGroupFullRebuild'
of trigger 'rebuildAllIndexesTrigger' STARTED
INFO task.TaskLauncherTemplate - Task duplicateGroupFullRebuild FINISHED
INFO task.TaskTriggerExecutor - Trigger 'rebuildAllIndexesTrigger'
finished. All task executed successfully.
```

6.1.11.2 Задача упала, триггер остановился

В логе мы видим количество не запущенных задач:

```
INFO task.TaskTriggerExecutor - Trigger
'dailyIncrementLoadWithDedupTrigger' STOPPED on errors in task
'bufferUploadTaskPerformer'
INFO task.TaskTriggerExecutor - Trigger
'dailyIncrementLoadWithDedupTrigger' finished. 1 tasks successful, 1
tasks failed, 4 tasks not started.
```

6.1.11.3 Задача упала, триггер продолжил работу

В логе можно увидеть, что триггер прошел успешно, но N задач прошло успешно, а M упали с ошибкой

```
INFO task.TaskTriggerExecutor - Trigger
'dailyIncrementLoadWithDedupTrigger' finished. 5 tasks successful, 1
tasks failed.
```

6.1.12 Система «Гражданский фактор» (cdi-soap-stats.log)

В журнале содержится агрегированная информация о вызовах SOAP-методов (об IP, авторе запроса и системе).

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.

6.1.12.1 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}
```

6.1.12.2 Описание колонок:

| Колонка | Значение | Описание |
|---------|----------|--|
| 1 | date | Дата и время сбора статистики с точностью до минут. Время отображается в одинарных кавычках. |
| 2 | endpoint | Название endpoint'а. Присутствует только если подключен входящий JMS. По нему определяется тип потока: SOAP/JMS или SOAP/HTTP. Если SOAP/JMS, тогда наименование endpoint'а дополняется информацией о потоке (в примере ниже — JMS-поток от системы CFT, диапазон clientId [80-89]). |
| 3 | method | Название метода. |
| 4 | ip | IP-адрес, с которого пришел запрос. |
| 5 | login | Учетная запись, от которой пришел запрос. |
| 6 | system | Наименование исходной системы. |
| 8 | call_qty | Число вызовов. |
| 10 | fail_qty | Число вызовов, завершившихся с ошибкой. |
| 12 | avg_qty | Среднее время обработки успешных вызовов, в мс. |
| 14 | p90_qty | Время обработки, в которое укладывается 90% успешных вызовов, в мс. |
| 16 | p95_qty | Время обработки, в которое укладывается 95% успешных вызовов, в мс. |
| 18 | p99_qty | Время обработки, в которое укладывается 99% успешных вызовов, в мс. |
| 20 | max_qty | Максимальное время обработки успешных вызовов, в мс. |
| 22 | min_qty | Минимальное время обработки успешных вызовов, в мс. |
| 24 | qps_qty | 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).

6.1.12.3 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> call 2 fail 0 p90
3807 p95 3807 p99 3807 avg 2606 max 3807 min 1404
qps 0,03
2019-08-15 '13:51'
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> 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)

```

6.1.12.4 Настройки:

По умолчанию собирается раз в 15 минут.

Для изменения частоты записи требуется изменить параметр `webservice.stats.periodInSeconds` в *APM Администратора — Конфигурация — Параметры SOAP и REST сервисов*.

6.1.13 **cdi-ui-stats.log**

В журнале содержится агрегированная информация о вызовах веб-интерфейса.

Статистика по запросам в рамках периода отделена пустой строкой.

6.1.13.1 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}
```

6.1.13.2 Описание колонок:

| Колонка | Значение | Описание |
|---------|-------------|--|
| 1 | date | Дата и время сбора статистики с точностью до минут. Время отображается в одинарных кавычках. |
| 2 | controller | Название название контроллера. Необходимо для обеспечения уникальности записей, т.к. могут быть одинаковые методы для разных контроллеров. |
| 3 | method | Название метода. |
| 4 | protocol | Протокол (для ui всегда rest) |
| 6 | call_qty | Число вызовов. |
| 8 | fail_qty | Число вызовов, завершившихся с ошибкой. |
| 10 | cancel_qty | Число отменённых вызовов. |
| 12 | timeout_qty | Число вызовов, завершившихся с ошибкой по таймауту. |
| 14 | p90_qty | Время обработки, в которое укладывается 90% успешных вызовов, в мс. |
| 16 | p95_qty | Время обработки, в которое укладывается 95% успешных вызовов, в мс. |
| 18 | p99_qty | Время обработки, в которое укладывается 99% успешных вызовов, в мс. |
| 20 | avg_qty | Среднее время обработки успешных вызовов, в мс. |
| 22 | max_qty | Максимальное время обработки успешных вызовов, в мс. |
| 24 | qps_qty | Количество вызовов в секунду. |

Колонки 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 содержат фиксированный текст.

6.1.13.3 Пример:

```
2021-08-23 '08:47:54' ConflictController
getSummary                                rest
call 8      fail 0      cancel 0      timeout 0      p90 15.73
p95 15.73   p99 15.73   avg 4        max 16        qps 0.01
2021-08-23 '08:47:54' ControllerTemplate
initBinder                                rest
call 25     fail 0      cancel 0      timeout 0      p90 0
p95 15.73   p99 17.3    avg 1.32     max 17        qps 0.03
2021-08-23 '08:47:54' LocalizationController
getLocalization                            rest
call 2      fail 0      cancel 0      timeout 0      p90 125.83
p95 125.83 p99 125.83 avg 64.5     max 129       qps 0
2021-08-23 '08:47:54' MetricsController
trackEnv                                    rest
```


| | | | | | | | | | |
|---|--------|------|--------|--------|-------|---------|-----|------|--------|
| call | 2 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 64.49 |
| p95 | 64.49 | p99 | 64.49 | avg | 40.5 | max | 65 | qps | 0 |
| 2021-08-23 '08:47:54' MetricsController | | | | | | | | | |
| trackEvents | | | | | | | | | |
| | | | | | | | | rest | |
| call | 1 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 46.14 |
| p95 | 46.14 | p99 | 46.14 | avg | 48 | max | 48 | qps | 0 |
| 2021-08-23 '08:47:54' PartyController | | | | | | | | | |
| fetchPartiesPage | | | | | | | | | |
| | | | | | | | | rest | |
| call | 4 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 150.47 |
| p95 | 150.47 | p99 | 150.47 | avg | 80.75 | max | 146 | qps | 0 |
| 2021-08-23 '08:47:54' PartyController | | | | | | | | | |
| getExtendedFilters | | | | | | | | | |
| | | | | | | | | rest | |
| call | 2 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 570.43 |
| p95 | 570.43 | p99 | 570.43 | avg | 291 | max | 582 | qps | 0 |
| 2021-08-23 '08:47:54' PartyController | | | | | | | | | |
| getPartyFilters | | | | | | | | | |
| | | | | | | | | rest | |
| call | 2 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 0 |
| p95 | 0 | p99 | 0 | avg | 0 | max | 0 | qps | 0 |
| 2021-08-23 '08:47:54' PartyController | | | | | | | | | |
| getPartyGridMeta | | | | | | | | | |
| | | | | | | | | rest | |
| call | 2 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 31.46 |
| p95 | 31.46 | p99 | 31.46 | avg | 16 | max | 32 | qps | 0 |
| 2021-08-23 '08:47:54' PartyController | | | | | | | | | |
| getTypes | | | | | | | | | |
| | | | | | | | | rest | |
| call | 2 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 0 |
| p95 | 0 | p99 | 0 | avg | 0 | max | 0 | qps | 0 |
| 2021-08-23 '08:47:54' SimultaneousWorkingController | | | | | | | | | |
| getLockedEssences | | | | | | | | | |
| | | | | | | | | rest | |
| call | 8 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 31.46 |
| p95 | 31.46 | p99 | 31.46 | avg | 4 | max | 32 | qps | 0.01 |
| 2021-08-23 '08:47:54' UIRootController | | | | | | | | | |
| getMainUiPage | | | | | | | | | |
| | | | | | | | | rest | |
| call | 2 | fail | 0 | cancel | 0 | timeout | 0 | p90 | 16.78 |
| p95 | 16.78 | p99 | 16.78 | avg | 8.5 | max | 17 | qps | 0 |

6.1.13.4 Настройки:

По умолчанию собирается раз в 15 минут.

Для изменения частоты записи требуется изменить параметр `monitoring.ui.time.loggingStepInSeconds` в *АРМ Администратора — Конфигурация — Параметры модуля мониторинга*.

6.1.14 Трассировка JMS-сообщений

Временно включить журналирование JMS-сообщений:

1. Найти файл `logback.xml` текущей сборки системы «Гражданский фактор». Местоположение:

- `JBOSS_CDI_HOME\standalone\tmp\vfs\tmp\tmp***` (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.cleandata.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 CDI.

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.

6.1.15 Трассировка SOAP-запросов

Временно включить журналирование SOAP-запросов:

1. Найти файл logback.xml текущей сборки системы «Гражданский фактор».

Местоположение:

- JBoss: JBOSS_CDI_HOME\standalone\tmp\vfs\temp*** (с наибольшей датой актуальности) \cdi-web-*.war\WEB-INF\classes\logback.xml
- WildFly 8: WILDFLY_CDI_HOME\standalone\tmp\vfs\temp\temp*** (с наибольшей датой актуальности) \cdi-web-*.war-*\WEB-INF\classes\logback.xml
- WildFly 10: WILDFLY_CDI_HOME\standalone\tmp\vfs\temp\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.

! Журналирование будет автоматически выключено при перезапуске сервиса CDI.

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.

6.1.16 Трассировка произвольного пакета

Результат команд применяется сразу, рестарт системы «Гражданский фактор» не требуется. После рестарта системы «Гражданский фактор» 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 '{}'
```

6.2 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.

| Журнал | Описание | Условие создания |
|----------------------|--|--------------------------|
| console.log | 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 |

| Журнал | Описание | Условие создания |
|--------------------------------------|---|--|
| 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"> factor-dedup-offline-physical.log factor-dedup-offline-egr_check.log | всегда при старте jboss |
| factor-dedup-online-<deduprule>.log | Журнал онлайн поиска дубликатов для конкретного дедап процессора, напр.: <ul style="list-style-type: none"> factor-dedup-online-physical.log factor-dedup-online-egr_check.log | 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. Пример сообщения | если подключены перекодирующие фильтры |
| | <pre> 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) </pre> | |

| Журнал | Описание | Условие создания |
|--------|--|------------------|
| | <pre> 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 </pre> | |

6.3 Уровни протоколирования

В системе используется 4 уровня протоколирования.

ERROR — сообщения об ошибках;

WARN — предупреждения и сообщения об ошибках;

INFO — ошибки, предупреждения и информационные сообщения;

DEBUG — детальная информация о ходе процесса обработки данных, в т.ч. информационные сообщения, предупреждения и сообщения об ошибках.

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**.

6.4 Аудит

Экран аудита предназначен для мониторинга действий, производимых пользователями в системе «Гражданский фактор».

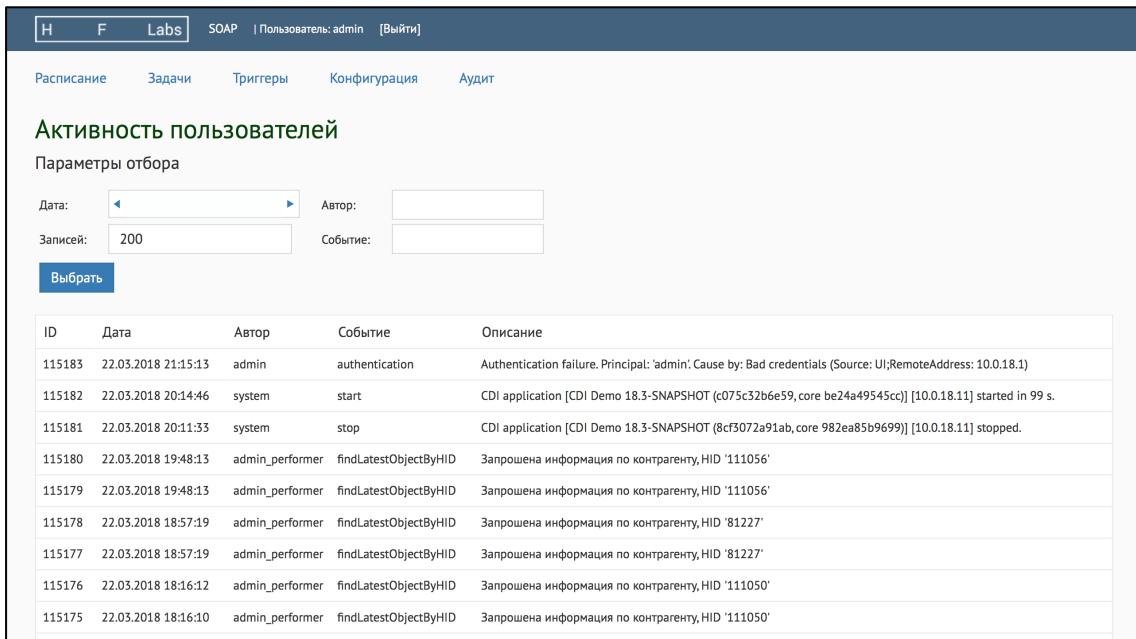


Figure 1 admin_audit.png

Администратор может фильтровать события по следующим параметрам:

- **Дата** - выбор по дате наступления события либо по диапазону дат;
- **Автор** - пользователь системы «Гражданский фактор», произведший действие;
- **Событие** - системное имя произошедшего события.

Также можно указать параметр **Записей** — количество отображаемых событий.

6.4.1 Поля записи журнала

Каждая запись журнала содержит следующие поля:

- **ID** - идентификатор события;
- **Дата** - дата и время наступления события
- **Автор** - пользователь системы «Гражданский фактор», 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.

6.4.2 Типы событий

В системе определены следующие типы событий:

| Код события | Событие | Комментарий |
|-------------|-------------------|------------------------------------|
| start | Запуск приложения | CDI application [version] started. |

| Код события | Событие | Комментарий |
|----------------------------|--|--|
| 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}' |
| getOriginalWithWinnerFlags | Просмотр исходной карточки контрагента из золотой карточки | Запрошена информация по контрагенту, HID '{идентификатор_контрагента}' |

| Код события | Событие | Комментарий |
|-------------------------|---|---|
| fetchParties | Просмотр карточек контрагентов из группы дубликатов | Запрос информации по контрагентам из дубликата, HID дубликата '{идентификатор_контрагента}' |
| mergeGroup | Запуск слияния группы дубликатов | Запуск слияния группы дубликатов, HIDы контрагентов '{идентификатор_контрагента}' |
| performDuplicate | Работа с парами дубликатов | Подтверждение/отклонение/откладывание пары дубликатов, HID '{идентификатор_контрагента}'- '{идентификатор_контрагента}' |
| performDuplicates | Работа с парами транзитивных дубликатов | Подтверждение/отклонение/откладывание пары дубликатов, HID '{идентификатор_контрагента}'- '{идентификатор_контрагента}', в случае присутствия транзитивного дубля |
| performUnmerge | Разъединение контрагентов | Разъединение контрагентов, объединенный HID '{идентификатор_контрагента}', отделяемый HID '{идентификатор_контрагента}' |
| getDiff | Сравнение контрагентов из групп дубликатов | Запрошена информация по контрагенту, HID '{идентификатор_контрагента}', '{идентификатор_контрагента}' |
| performRecover | Восстановление контрагента на дату | Восстановление контрагента, HID '{идентификатор_контрагента}' на дату '{дата}' |
| performRecoverRequisite | Восстановление реквизитов контрагента на дату | Восстановление реквизитов контрагента, HID '{идентификатор_контрагента}' на дату '{дата}' |
| performRecoverAttribute | Восстановление атрибута контрагента на дату | Восстановление атрибута '{атрибут}', HID '{идентификатор_контрагента}' на дату '{дата}' |
| propertiesChange | Изменение свойства конфигурации | Добавлено/изменено/удалено свойство '{свойство}' |
| search | Поиск контрагента | Поиск контрагента ['{запрос}'] - Результат(Всего '{количество}' КА): ['{идентификатор_контрагента}'] |

7 Установка обновлений

Данная инструкция содержит стандартный порядок действий, выполняемый при обновлении системы «Гражданский фактор». При необходимости выполнения дополнительных действий предоставляется отдельная подробная инструкция по обновлению.

7.1 Общее описание

Перед началом обновления системы необходимо сделать резервную копию. В случае сбоя следует восстановить систему.

Система, как видно из схемы в разделе [Архитектура системы](#), состоит из двух независимых модулей («Гражданский фактор», ФАКТОР), работающих под управлением сервера приложений JBOSS AS (JBOSS Application Server), каждый в своём инстансе.

Серверы приложений JBOSS сконфигурированы как системные службы Windows, настроен автоматический старт при загрузке операционной системы.

Модули могут быть обновлены независимо один от другого.

Обновление системы «Гражданский фактор» может повлечь за собой обновление схемы базы данных. В таком случае будет предоставлен набор миграционных скриптов.

Типичный состав обновления:

1. Файл модуля «Гражданский фактор»:
 - o `cdi-web-{CUSTOMER}-{VERSION}.war`
2. Файл модуля ФАКТОР:
 - o `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.

7.2 Обновление модулей

7.2.1 Для обновления системы необходимо выполнить следующие шаги:

1. Сделать резервные копии предыдущей сборки системы «Гражданский фактор» `cdi-web-{customer}-{version}-SNAPSHOT-{data}__{time}-{revision}.war` и БД системы «Гражданский фактор». Резервная копия должна располагаться **вне** директории `{JBOSS_CDI_HOME}\standalone\deployments` и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис CDI

- a. linux: service cdi stop
- b. win: net stop cdi
4. Удалить старый war-файл системы «Гражданский фактор» из директории {JBASS_CDI_HOME}\standalone\deployments.
5. Скопировать новую сборку системы «Гражданский фактор» в директорию deployments.
6. Запустить сервис CDI.
 - a. linux: service cdi start
 - b. win: net start cdi
7. Проверить, что в APM Администратора отображается корректная версия продукта (номер версии указан в Release Notes).

7.2.2 Обновление модуля «Фактор»

1. Сделать резервную копию предыдущей сборки «Фактора» factor-service-{CUSTOMER}.war. Резервная копия должна располагаться **вне** директории {JBASS_HOME_FACTOR}\standalone\deployments и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис FACTOR
 - a. linux: service factor stop
 - b. win: net stop factor
4. Удалить старый war-файл «Фактора» из директории {JBASS_HOME_FACTOR}\standalone\deployments.
5. Скопировать новую сборку «Фактора» в директорию deployments.
6. Запустить сервис FACTOR
 - a. linux: service factor start
 - b. win: net start factor
7. Проверить, что в APM Администратора отображается корректная версия Фактора (номер версии указан в Release Notes).

7.3 Ручной запуск и остановка

Если запуск обновлённого модуля в режиме веб-службы завершился неудачно, может понадобиться запустить JBOSS вручную. При этом необходимо удостовериться, что пользователь, с правами которого производится запуск JBOSS, имеет доступ на запись в файлы лога JBOSS.

Запуск осуществляется с помощью bat-файла {JBASS_HOME}\bin\run.bat (или {JBASS_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 {JBASS_HOME}\bin\shutdown.bat (или аналогичный sh-файл для Linux) с параметром -s. Пример:

```
C:\server\jboss-cdi\bin>shutdown.bat -S
```

7.4 Для обновления системы необходимо выполнить следующие шаги:

1. Сделать резервные копии предыдущей сборки системы «Гражданский фактор» `cdi-web-{customer}-{version}-SNAPSHOT-{data}__{time}-{revision}.war` и БД системы «Гражданский фактор». Резервная копия должна располагаться **вне** директории `{JBOSS_CDI_HOME}\standalone\deployments` и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис CDI
 - a. linux: `service cdi stop`
 - b. win: `net stop cdi`
4. Удалить старый war-файл системы «Гражданский фактор» из директории `{JBOSS_CDI_HOME}\standalone\deployments`.
5. Скопировать новую сборку системы «Гражданский фактор» в директорию `deployments`.
6. Запустить сервис 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 Release Notes).

7.5 Обновление модуля «Фактор»

1. Сделать резервную копию предыдущей сборки «Фактора» `factor-service-{CUSTOMER}.war`. Резервная копия должна располагаться **вне** директории `{JBOSS_HOME_FACTOR}\standalone\deployments` и её поддиректорий, иначе нормальная работа сервера не гарантируется.
2. Проверить MD-5 сумму сборки.
3. Остановить сервис 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 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 Release Notes).

7.6 Автоматическое создание схемы CDI

7.6.1 Настройка WildFly

В конфигурационном файле `standalone.conf` для Linux-серверов или `standalone.conf.bat` (Windows-сервера) указать параметры, задающие логин и пароль пользователя с правами `SYSDBA`, а также `connection-string` БД.

Code Block 8 standalone.conf.bat

Code Block 9 standalone.conf

7.6.2 Создание новой схемы

Запустить WildFly с параметром `-Djdbc.create`.

Windows:

```
standalone.bat -Djdbc.create
```

Linux:

```
sh standalone.sh -Djdbc.create
```

Схема создается с учетом скриптов миграции. Например, «17.12.1.demo». Это позволяет добавлять скрипты автомиграции в текущую сборку (see [CDI-4417](#))

7.6.3 Пересоздание существующей схемы

Запустить WildFly с параметром `-Djdbc.recreate`.

Windows:

```
standalone.bat -Djdbc.recreate
```

Linux:

```
sh standalone.sh -Djdbc.recreate
```

8 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?](#)
 - [Что такое "схлопывание" атрибутов?](#)
- [Менеджер Данных \(FAQ\)](#)
 - [Не могу зайти в APM Менеджера данных](#)
 - [Отслеживание логинов в CDM](#)
 - [Ошибка «Разрешите противоречия между подтвержденными и отклоненными дубликатами, прежде чем выполнять слияние»](#)
- [APM 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?](#)
 - [Авторизация запроса в SoapUI](#)
 - [Создание проекта в SoapUI](#)
 - [Как можно вызвать REST-метод вручную](#)
 - [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 сопровождающих](#)
- [Фактор \(FAQ\)](#)
 - [Есть ли doBatchClean для телефонов?](#)
- [Интеграция \(faq\)](#)

- [Как передать две записи с одной связью](#)

8.1 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?](#)
- [Что такое "схлопывание" атрибутов?](#)

8.1.1 CPU под 100% на задаче

Q: От мониторинга пришло сообщение, что CPU выжрано под 100%. А через час-другой его отпускает (когда завершается задача поиска дублей, например)

A: Можно попробовать изменить параметр `poolSize`:

1. Узнать сколько ядер на сервере (обычно количество потоков выставляется по количеству ядер.).
2. Узнать, сколько тредов работает (можно посмотреть по логам. В `sdi.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

8.1.2 Индекс неактуальный (Index may be corrupted)

Q: Автоматический мониторинг логов сообщает об ошибке:

```
... Invalid 'PhysicalParty' index document count (expected 2, got 0).  
Index may be corrupted. Please rebuild ASAP.
```

A: Поисковый индекс неактуален. Пользователи могут видеть неактуальные данные, во время слияния возможны ошибки.

В АРМ Администратора обновите индексы:

1. В системе «Гражданский фактор» — [rebuildAllIndexesTrigger](#) на закладке [Триггеры](#).
2. В Факторе — [Поиск дубликатов](#) или [Поиск дубликатов и слияние](#) на закладке [Задачи](#).

8.1.3 Как переместить прод с одной машины на другую

1. Запустить машину, выделенную для переноса прод-сервера.
2. Настроить ОС по [инструкции](#).

3. Выполнить перезагрузку после изменения параметров ОС.
4. Остановить сервисы CDI и Factor.
5. Скопировать файл `factor-service-{customer}.war` со старой машины из папки `{JBOSS_FACTOR_HOME}standalone/deployments` на новую в соответствующую папку.
6. Скопировать `standalone.conf` системы «Гражданский фактор» и Фактора со старой машины на новую.
7. Удалить индекс и кэш фактора с новой машины: удалить папку `CDI_SEARCH_DIR` и `{JBOSS_FACTOR_HOME}jboss/standalone/data/dedup-data.ws`.
8. Остановить онлайн поток запросов `save`, `saveAndMerge`, `close`, `closeAttribute` в систему «Гражданский фактор» на проде (дождаться, пока в очереди не останется сообщений).
9. Выполнить команду "sudo fsync".
10. Скопировать индексы и кеши 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`.
11. Запустить службы CDI и Factor на новой машине.
12. Подождать 5 минут, чтобы службы CDI и Factor полностью запустились и могли принимать онлайн-поток.
13. Остановить продуктовую машину.
14. Подменить имя и IP адреса машины.
15. Выполнить рестарт новой машины.

8.1.4 Как поднять бекап базы с PROD на TEST стенде?

Q: Хотим развернуть бекап базы с PROD на тестовом стенде. Как правильно сделать восстановление из бекапа, чтобы система корректно работала?

A: Для этого необходимо:

1. Остановить сервис CDI.
2. Сделать бэкап таблиц с настройками для TEST, чтобы не настраивать повторно после накатывания дампа (см. примечание).
3. Развернуть бекап, **исключив указанные ниже таблицы, либо выполнить их очистку в ручную после того как накатали дампы.**
4. **Удалить старый поисковый индекс.**
5. Запустить сервис CDI.
6. В АРМ 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, очистите 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 заблокировать запуск задач/триггеров, хотя ничего не запущено. Смело чистим эту табличку после того как накатыли дампы.

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 накатыть его поверх дампа, чтобы не настраивать вручную.
- JMS_EVENT – события, отправляемые в обратный 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 исключить данную таблицу из дампа, чтобы не переносить лишнее. Если ее накатывать дампы 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 накатыть 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.

8.1.5 Что такое "схлопывание" атрибутов?

Q: В правилах слияния атрибутов написано, что они могут "схлопываться" во время создания и обновления записи. Что это значит?



A: Когда запись 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 APM Менеджера данных;
- не передается через SOAP;
- передается в представления с признаком `is_deleted=1`.


Например, создается запись с двумя одинаковыми телефонами. Если бы схлопывания не было, то в APM Менеджера данных были бы видны дубли

| Автор | Тип | Информация | Код качества |
|--------|----------|-------------|---|
| GM:547 | Домашний | 74957896655 |  |
| GM:547 | Домашний | 74957896655 |  |

Но сейчас мы увидим только один телефон

| Автор | Тип | Информация | Код качества |
|--------|----------|-------------|---|
| GM:547 | Домашний | 74957896655 |  |

Второй все равно будет в системе, не закрыт, но отмечен флагом `hidden`.

На скрытый атрибут можно присылать обновление. В таком случае атрибуты снова пересливаются и флаг `hidden` перерасчитывается, что дает возможность снова сделать атрибут *видимым* при его изменении.

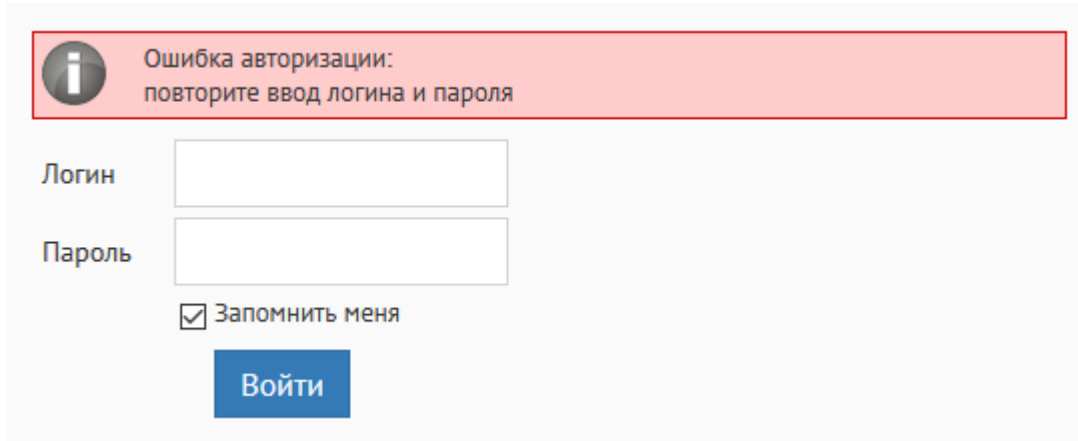
Например, была запись с двумя одинаковыми телефонами, первый из которых оказался скрыт по правилам слияния. На скрытый телефон приходит обновление, и он изменяется на другой. Таким образом, после обновления в карточке оказывается два разных телефона с точки зрения правил слияния, и ранее скрытый телефон становится видимым (флаг `hidden=0`).

8.2 Менеджер Данных (FAQ)

- [Не могу зайти в APM Менеджера данных](#)
- [Отслеживание логинов в CDM](#)
- [Ошибка «Разрешите противоречия между подтвержденными и отклоненными дубликатами, прежде чем выполнять слияние»](#)

8.2.1 Не могу зайти в АРМ Менеджера данных

8.2.1.1 Ошибка



Ошибка авторизации:
повторите ввод логина и пароля

Логин

Пароль

Запомнить меня

Войти

8.2.1.2 Возможные причины

- Неверный пароль.
- Недостаточно прав.
- Пользователь заблокирован.
- Истек срок действия пароля и т.д.

8.2.1.3 Что делать

Прочитать текст сообщения об ошибке.

8.2.2 Отслеживание логинов в CDM

Успешные логины выводятся в cdi-security.log

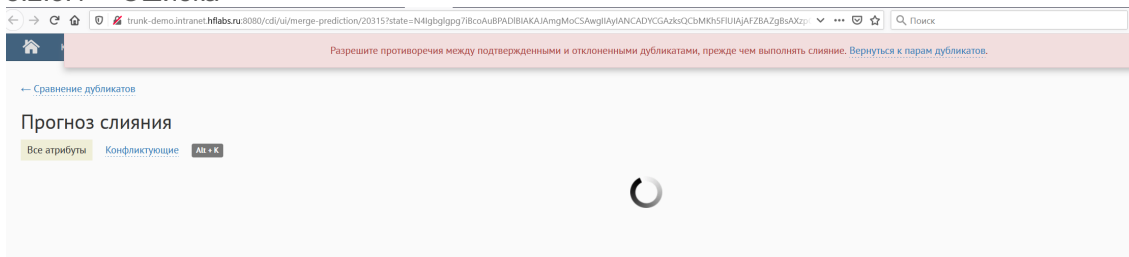
Пример:

```
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.

8.2.3 Ошибка «Разрешите противоречия между подтвержденными и отклоненными дубликатами, прежде чем выполнять слияние»

8.2.3.1 Ошибка



8.2.3.2 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.

8.2.3.3 Что делать

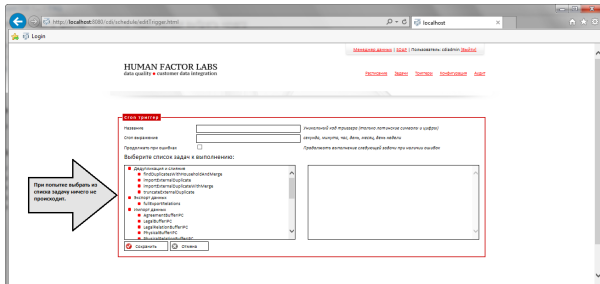
- **Лечится перестроение групп дубликатов, задача `duplicateGroupActualize`.**
-

8.3 АРМ Администратора (FAQ)

- [В IE* нельзя создать триггер - не удается выбрать задачи из списка](#)
- [Изменение параметра `oldMaxId`](#)

8.3.1 В IE* нельзя создать триггер - не удается выбрать задачи из списка

8.3.1.1 Ошибка



8.3.1.2 Возможные причины

- Не включен JavaScript.

8.3.1.3 Что делать

1. В настройках открыть "Свойства браузера" (Internet Options).
2. Выбрать вкладку "Безопасность" (Security) и нажать на ней кнопку "Другой..." (Custom level...).
3. После открытия диалогового окна "Параметры безопасности - зона местной 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.

8.3.2 Изменение параметра `oldMaxId`

Q: Если остановить службы CDI и Factor исправить в конфигурации значение параметра `oldMaxId` в параметрах импорта из буферной области и заново запустить службы, как себя поведет приложение?

A: В зависимости от того, увеличите или уменьшите вы его значение, получите разные последствия:

1. Уменьшите - система пропустит уже загруженные ранее записи и выполнит загрузку только изменившихся записей.
2. Увеличите - система выполнит загрузку только изменившихся записей начиная с указанного oldMaxId.

В обоих случаях ошибок быть не должно.

8.4 Дополнительно (FAQ)

- [Запросы в БД, соответствующие фильтрам в CDM](#)
- [Как пополнить справочник имен / фамилий?](#)

8.4.1 Запросы в БД, соответствующие фильтрам в CDM

Q: Как выполнить выборки из базы, соответствующие фильтрам в интерфейсе CDM?

A: При фильтрации записей в CDM в базе данных выполняются следующие запросы (запросы выводят количество записей, подходящих под установленный фильтр):

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
WHERE (
  (PP.fullname_qc = 'NOT_SURE')
  OR (PP.inn_qc = 'NOT_SURE')
  OR (PP.birthdate_qc = 'NOT_SURE')
)
AND PP.version = 0
AND PP.enddate is null
AND PP.merged_status IN ( 0, 2 );
```

Code Block 10 Ручная обработка: реквизиты

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN address A on PP.hid_party = A.hid_party
WHERE (
  (A.validationCode NOT IN ('VALIDATED', 'NOT_VALIDATED_EMPTY_SOURCE',
  'NOT_VALIDATED_FOREIGN', 'CONFIRMED_MANUALLY'))
  OR (A.qualityCode NOT IN ('GOOD', 'UNDEF_05'))
)
AND PP.version = 0
AND PP.enddate is null
AND A.version = 0
AND A.enddate is null
AND PP.merged_status IN ( 0, 2 );
```

Code Block 11 Ручная обработка: адреса

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN phone P on PP.hid_party = P.hid_party
WHERE (
  (P.qualityCode NOT IN ('GOOD', 'GOOD_CITY_CONFLICT',
'GOOD_REGION_CONFLICT',
'GOOD_REPLACED_CODE', 'GOOD_REPLACED_NUMBER',
'GOOD_REPLACED_CODE_NUMBER', 'CONFIRMED_MANUALLY'))
)
AND PP.version = 0
AND PP.enddate is null
AND P.version = 0
AND P.enddate is null
AND PP.merged_status IN ( 0, 2 );
```

Code Block 12 Ручная обработка: телефоны

```
SELECT count (distinct PP.hid_party)
FROM physical_party PP
JOIN email E on PP.hid_party = E.hid_party
WHERE (
  (E.qualityCode = 'NOT_SURE')
)
AND PP.version = 0
AND PP.enddate is null
AND E.version = 0
AND E.enddate is null
AND PP.merged_status IN ( 0, 2 );
```

Code Block 13 Ручная обработка: email

```
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 );
```

Code Block 14 Ручная обработка: документы

8.4.2 Как пополнить справочник имен / фамилий?

В сложных ФИО возможен неправильный разбор Фактором, так как в наших справочниках может не быть одного из компонентов. Чтобы исправить ситуацию, есть два варианта:

1. Поправить вручную в АРМ Менеджера данных. Если считаете ФИО корректными и часто встречающимися, пришлите его нам, добавим разбор в следующих релизах.
2. Пополнить справочник в Факторе.

Предпочтительно использовать первый вариант.

Но, если данные проходят через Фактор и уходят сразу в несколько систем, используйте пользовательские справочники.

8.4.2.1 Как добавить имя в справочник Фактора

Допустим, мы хотим добавить в справочник женское имя «АЙЫР-СААНА»

Необходимо пополнить [справочник имен](#)

```
АЙЫР-СААНА;1;;
```

После чего рестартануть фактор (займет минуту-две).

Для горячего резерва повторить на обеих нодах.

По аналогии пополняются:

- [справочник фамилий](#);
- [справочник отчеств](#).

8.5 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 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](#)

8.5.1 Как снять Thread Dump

Чтобы снять thread dump, необходимо:

- Запустить [Java VisualVM](#) (находится в папке bin)
- Распаковать архив и запустить приложение
- В левой области под названием “Applications” выбрать jboss. Их там 2 – cdi и factor. Понять, где какой, можно в описании приложения

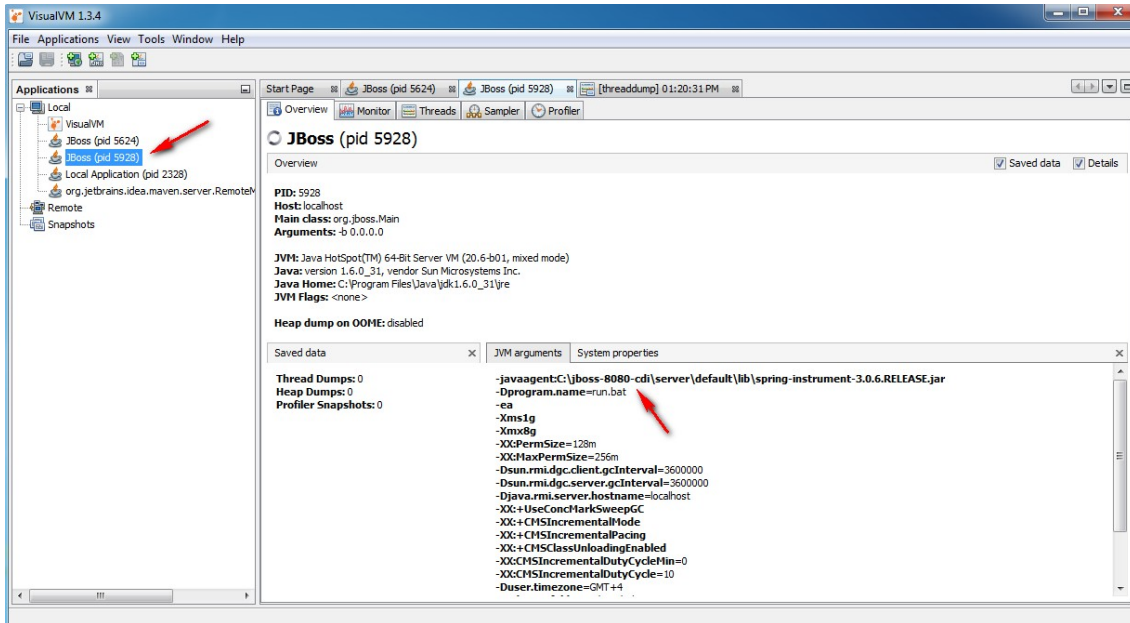


Figure 2 Visual VM выбор jboss-a.jpg

- Перейти на закладку “Threads” и нажать на кнопку “Thread Dump”.

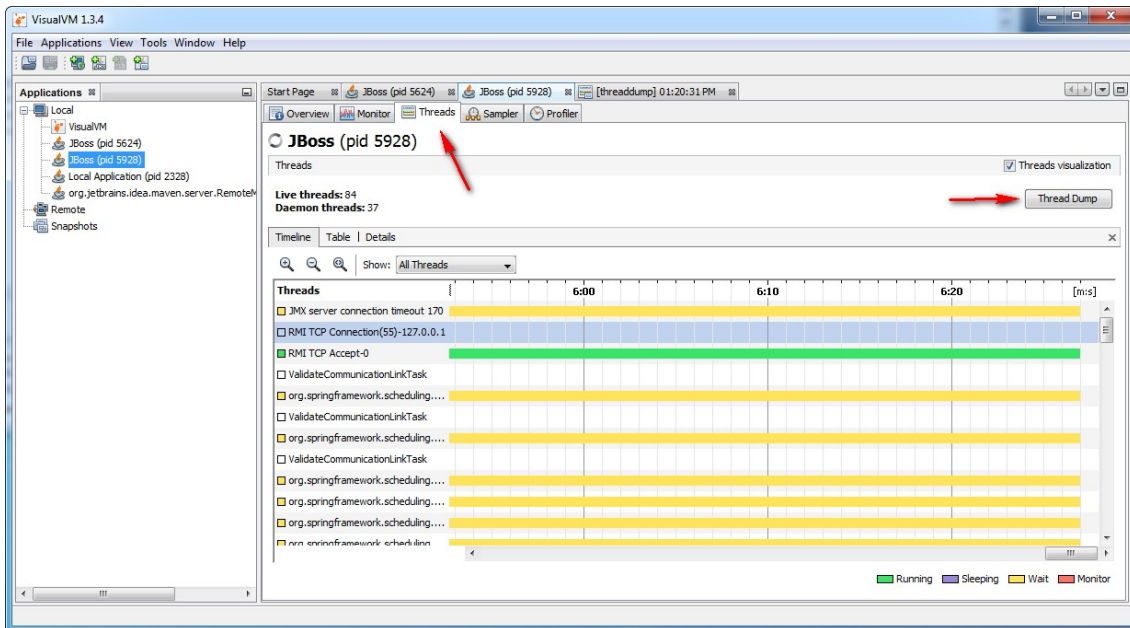


Figure 3 Thread Dump.jpg

- 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».

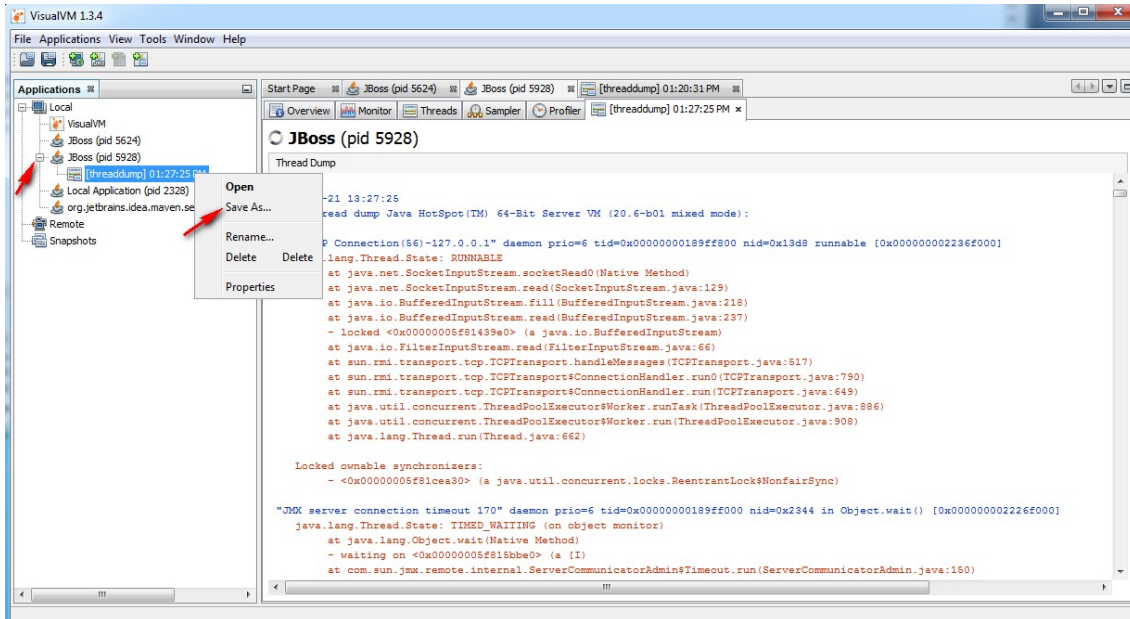


Figure 4 Сохранение дампа.jpg

8.5.2 Как снять Thread Dump через командную строку в Unix

Если нет возможности использовать [Java VisualVM](#), то для снятия дампа потоков необходимо:

- Найти id процесса. Для этого необходимо выполнить команду

```
ps aux | grep java
```

- В результате выполнения этой команды вы увидите запущенные java процессы в виде:

```
cdi      29271  0.4 50.6 10181244 8501280 ?    Sl   Jun19  10:52
/usr/java/jdk1.6.0_29/bin/java -Dprogram.name=run.sh -server -
javaagent:/home/cdi/jboss/server/default/lib/spring-instrument-
3.0.5.RELEASE.jar -ea -XX:PermSize=256m -XX:MaxPermSize=256m -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 -
Xrunjdwptransport=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
```

8.5.2.1 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.

8.5.2.2 Второй вариант Командой вида

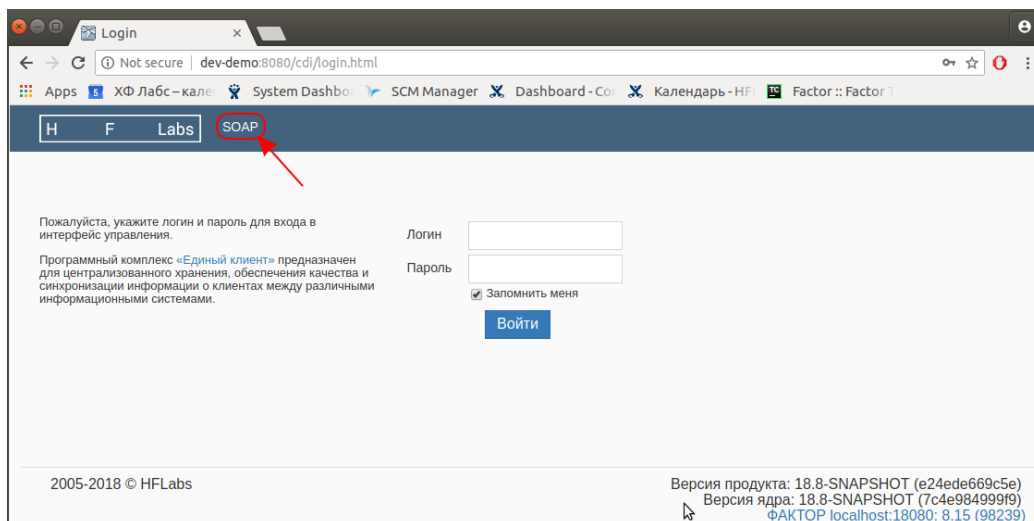
```
jstack -F 29271 > /opt/cdi/jstack.log
```

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.

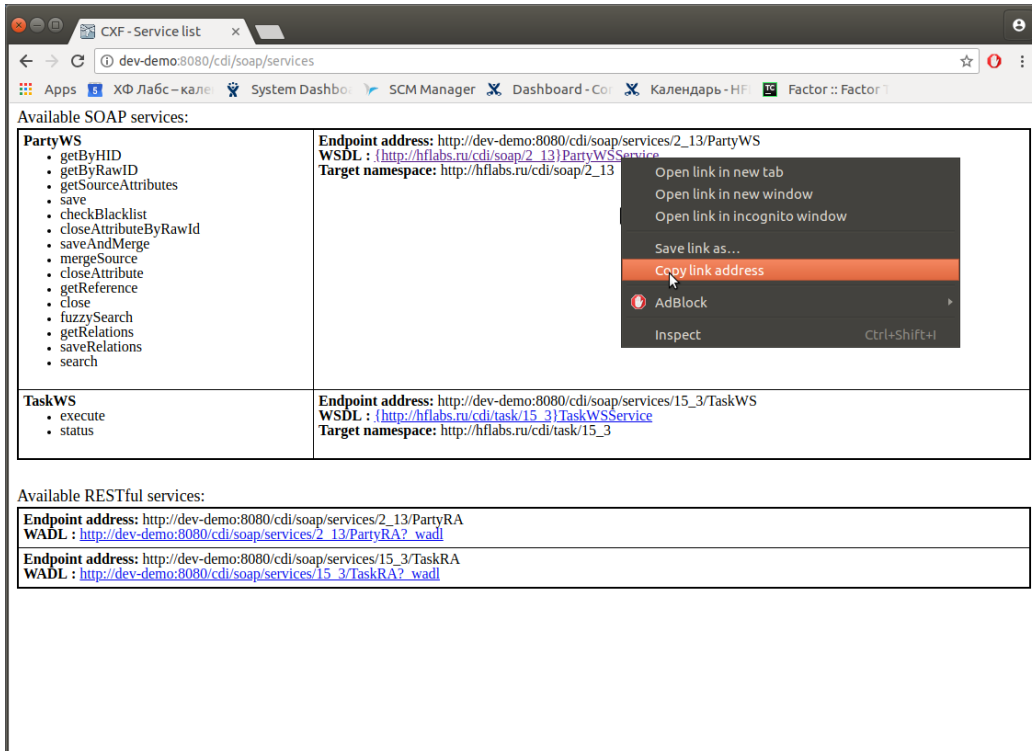
8.5.3 Как можно вызвать метод SOAP-интерфейса?

Чтобы проверить работу [онлайн-интерфейсов](#) вручную, можно использовать инструмент [SOAP UI](#):

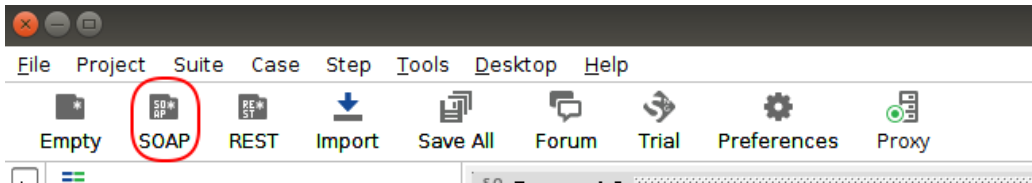
1. Открыть APM Администратора, нажать на ссылку SOAP.



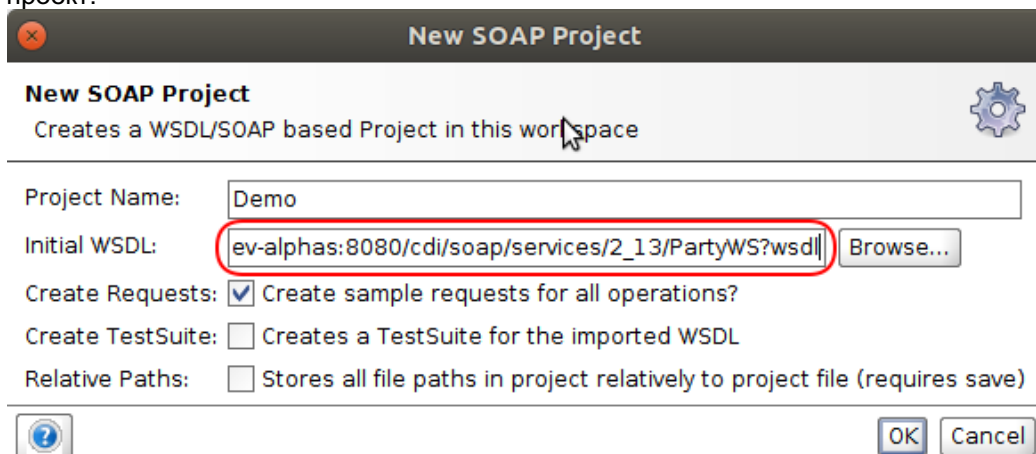
2. Система отображает все доступные пользователю сервисы. Нажать на ссылку WSDL у необходимого сервиса правой кнопкой мыши и выбрать copy link address или перейти по этой ссылке и скопировать адрес:



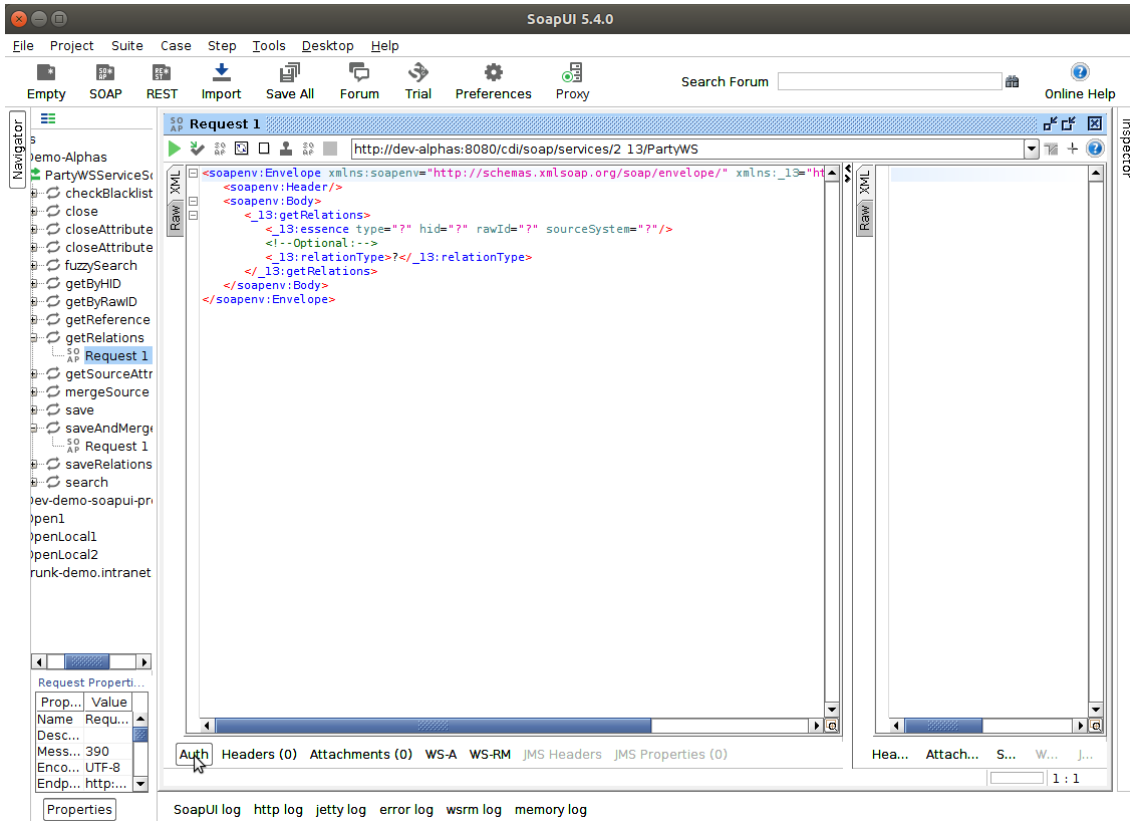
3. В SOAP UI нажать на кнопку Create a new soapUi Project



4. В открывшемся окне необходимо задать имя новому проекту и ввести ссылку на WSDL, полученную на шаге 2. После чего нажать "Ок". Будет создан новый проект.

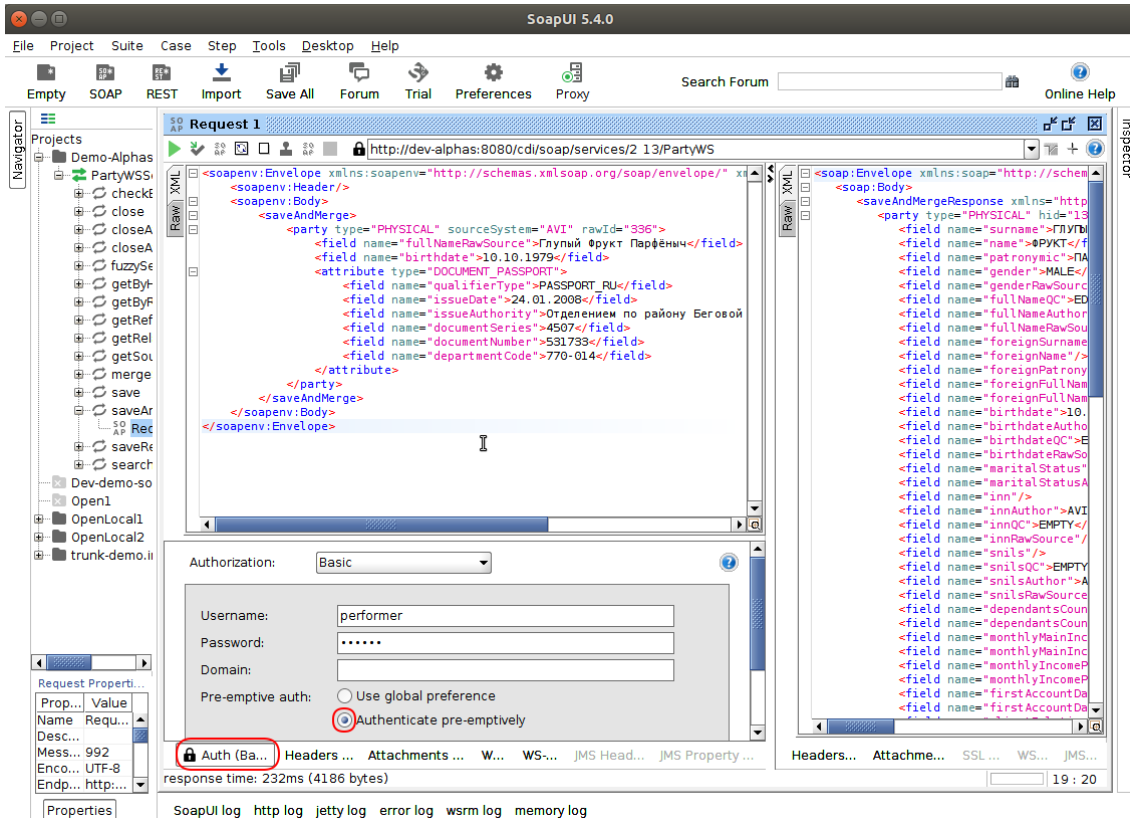


После выбора запроса нажмите на кнопку Authorization слева внизу и авторизуйтесь



Как правило у учётной записи [операциониста](#) есть необходимые права (ROLE_EXTERNAL_SYSTEM), однако в отдельных случаях потребуется специальная учётная запись. Подробнее смотрите в спэйсе заказчика.

Необходимо также выбрать предварительную авторизацию (authenticate preemptively). Если авторизация не выбрана, в доступе будет отказано.

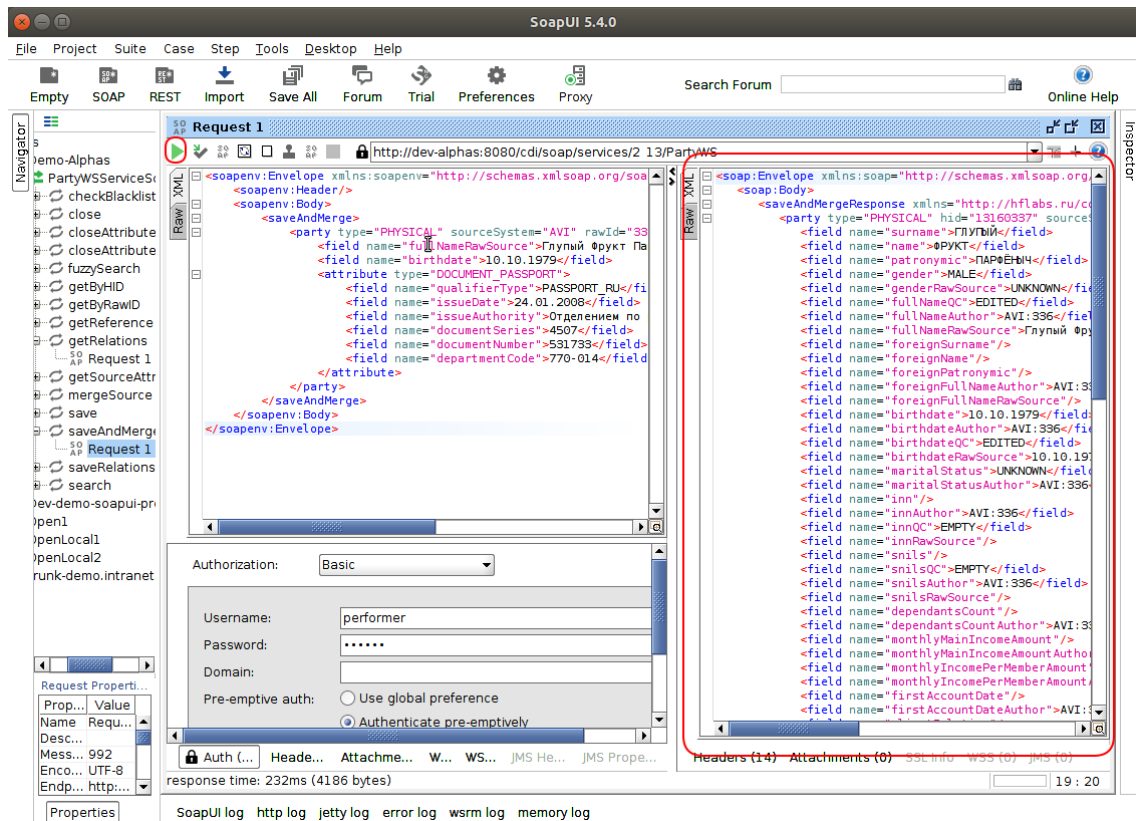


Теперь необходимо отредактировать шаблон запроса (в данном случае [search](#))



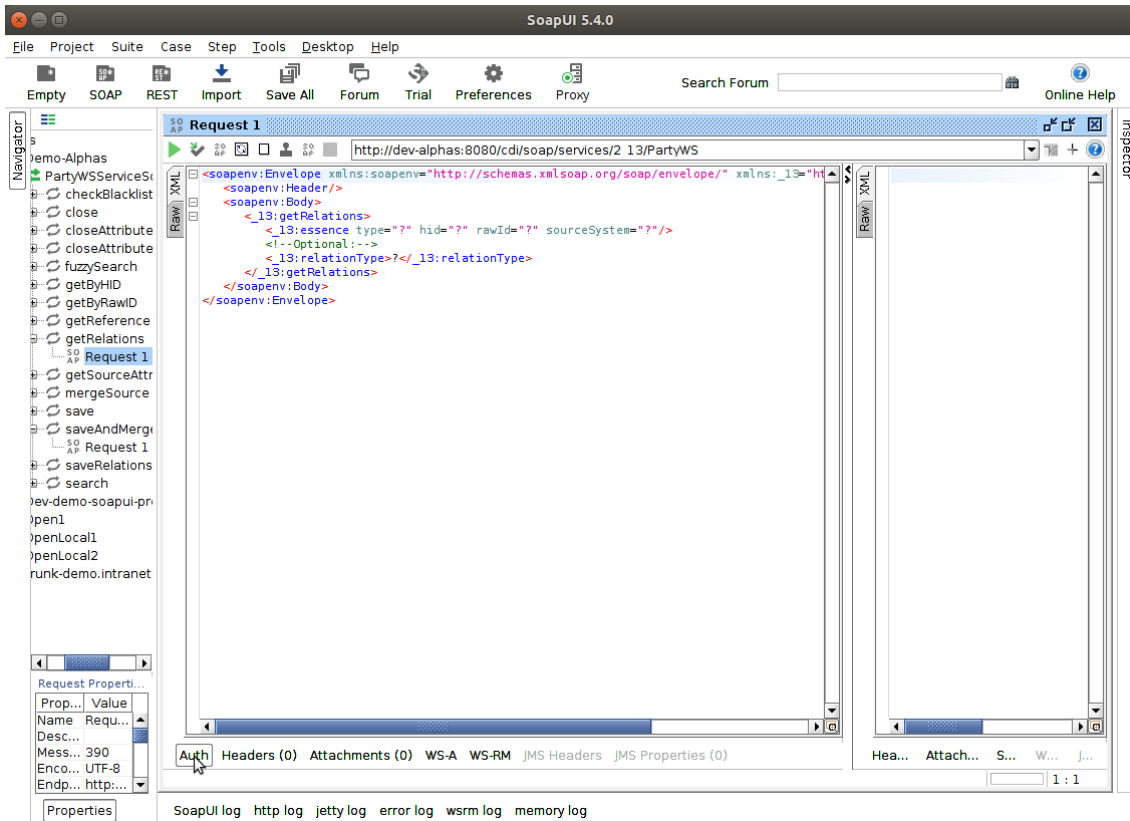
Его необходимо отредактировать согласно [документации](#) (выше приведена ссылка на пример запроса search) и нажать кнопку submit request

В правой части мы получим ответ



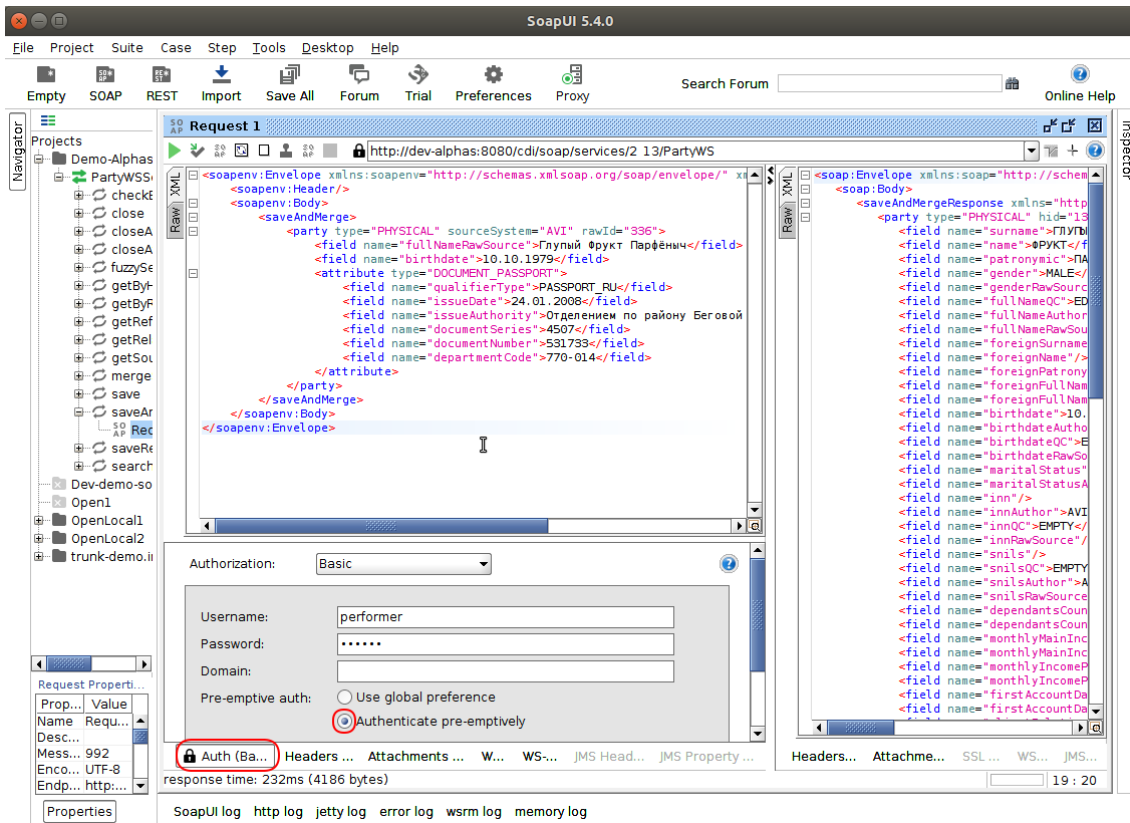
8.5.3.1 Авторизация запроса в SoapUI

После выбора запроса нажмите на кнопку Authorization слева внизу и авторизуйтесь



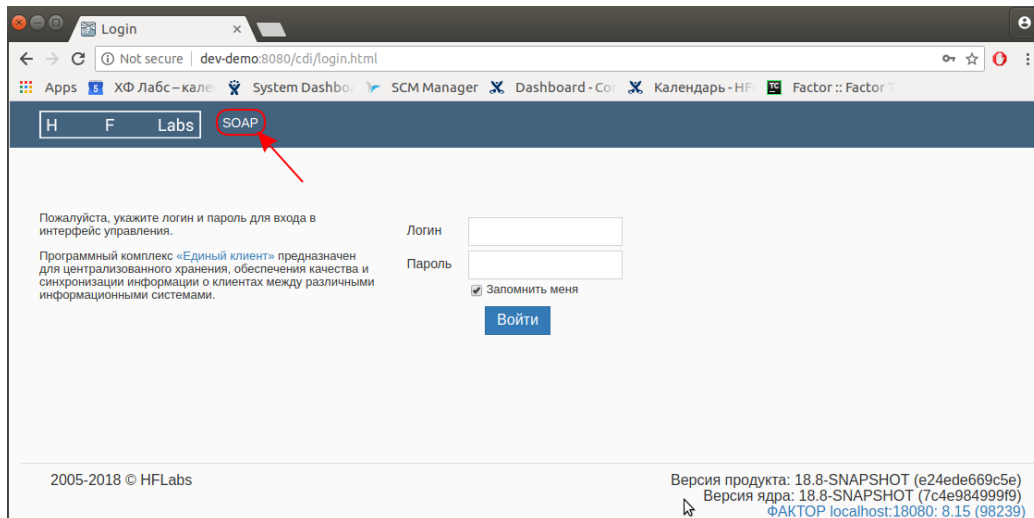
Как правило у учётной записи [операциониста](#) есть необходимые права (ROLE_EXTERNAL_SYSTEM), однако в отдельных случаях потребуется специальная учётная запись. Подробнее смотрите в спэйсе заказчика.

Необходимо также выбрать предварительную авторизацию (authenticate preemptively). Если авторизация не выбрана, в доступе будет отказано.

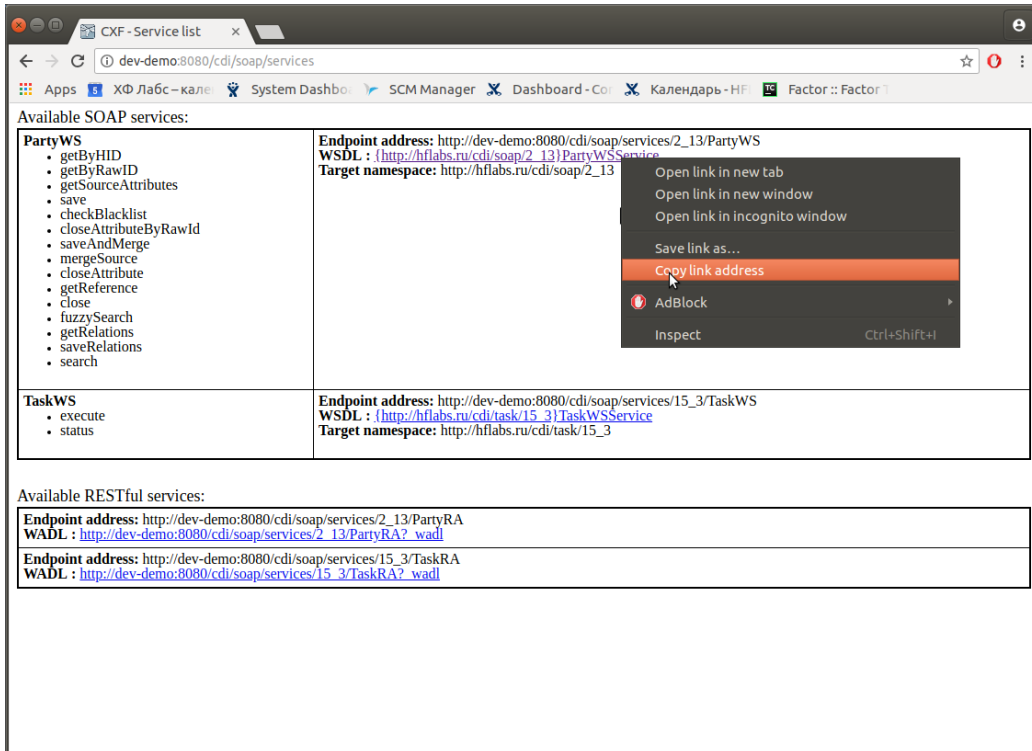


8.5.3.2 Создание проекта в SoapUI

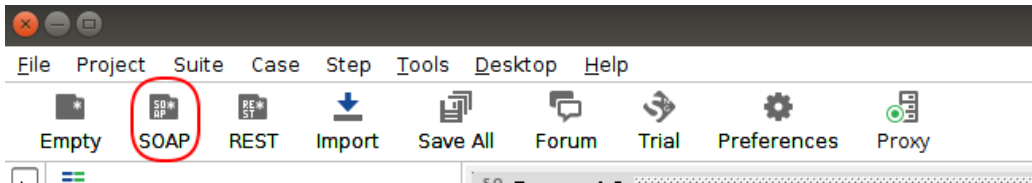
1. Открыть APM Администратора, нажать на ссылку SOAP.



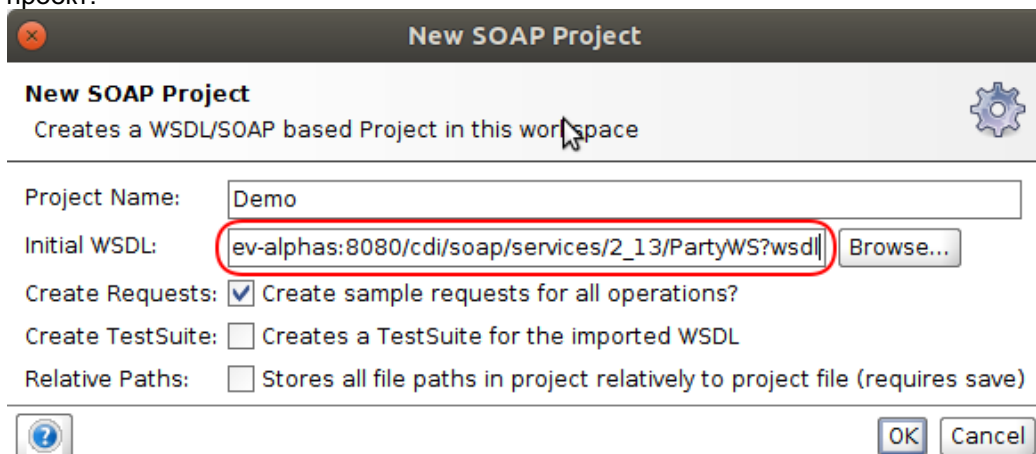
2. Система отображает все доступные пользователю сервисы. Нажать на ссылку WSDL у необходимого сервиса правой кнопкой мыши и выбрать copy link address или перейти по этой ссылке и скопировать адрес:



3. В SOAP UI нажать на кнопку Create a new soapUi Project



4. В открывшемся окне необходимо задать имя новому проекту и ввести ссылку на WSDL, полученную на шаге 2. После чего нажать "Ок". Будет создан новый проект.

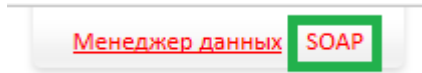


8.5.4 Как можно вызвать REST-метод вручную

Для ручного вызова [REST-методов](#) различных интерфейсов систем «Гражданский фактор» и Фактора можно использовать SoapUI. Описание на примере стенда dev-demo.

8.5.4.1 Методы системы «Гражданский фактор»

1. Открыть APM Администратора, нажать на ссылку SOAP.

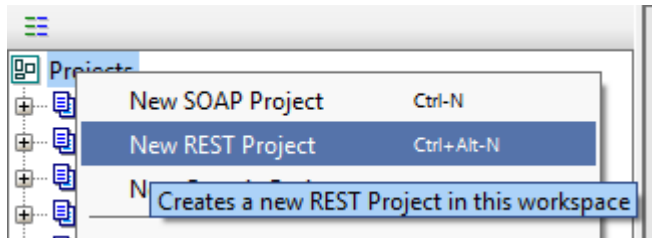


2. Скопировать ссылку на WADL нужного REST-интерфейс.

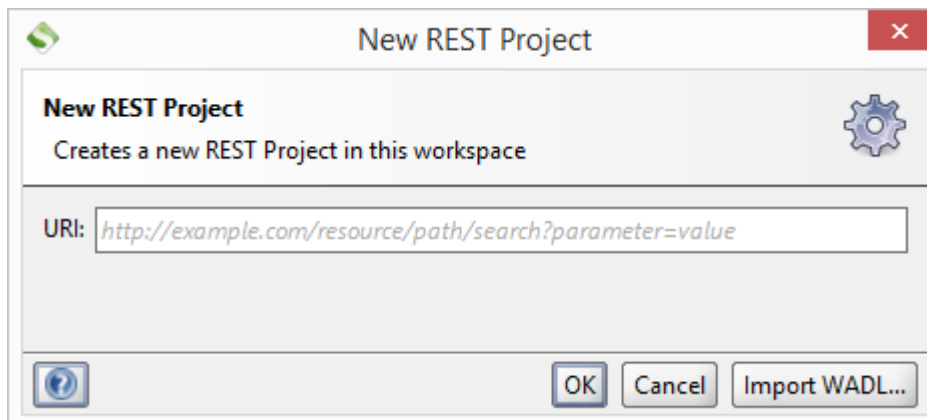
Available RESTful services:

| |
|---|
| Endpoint address: http://dev-demo:8080/cdi/soap/services/15_3/TaskRA WADL : http://dev-demo:8080/cdi/soap/services/15_3/TaskRA? wadl |
| Endpoint address: http://dev-demo:8080/cdi/soap/services/2_13/PartyRA WADL : http://dev-demo:8080/cdi/soap/services/2_13/PartyRA? wadl |

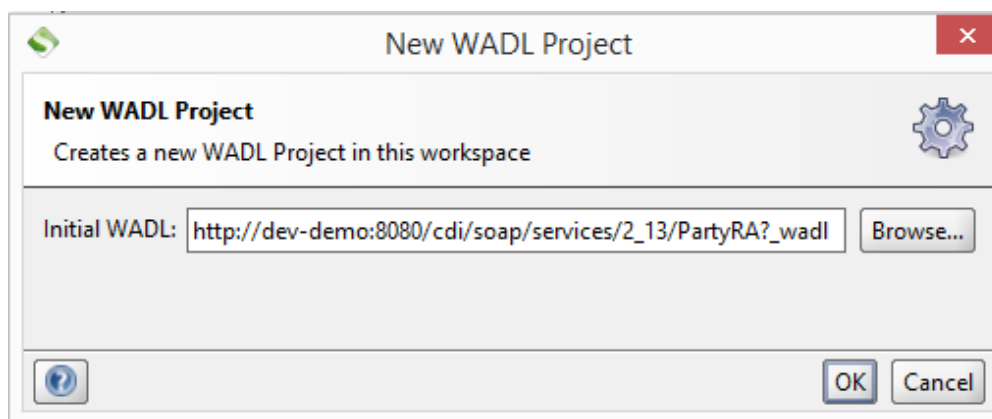
3. В SoapUI создать новый REST-проект.



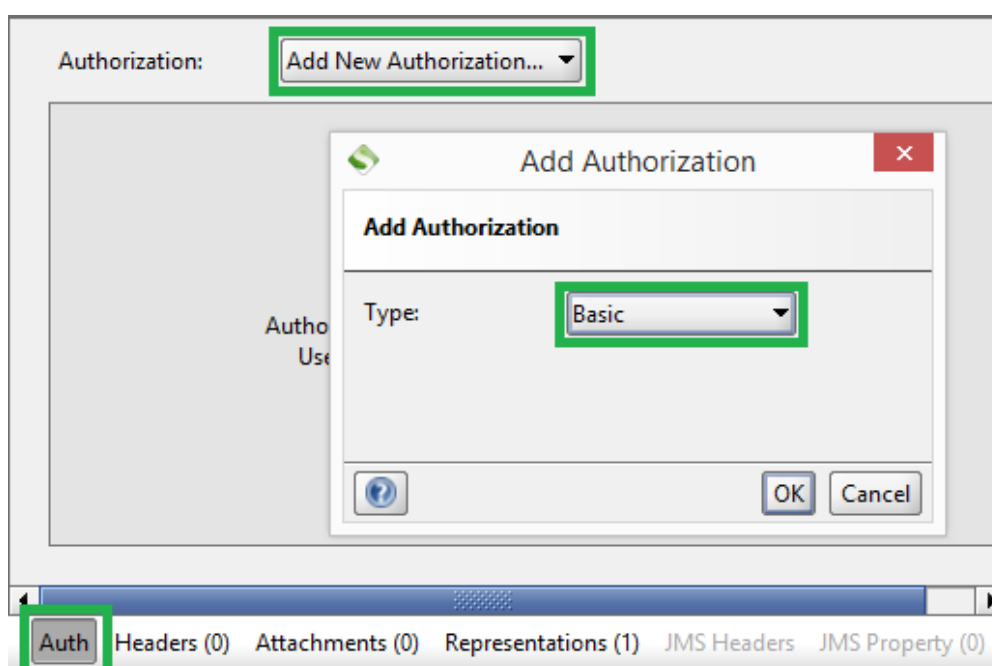
4. В появившемся диалоге нажать на кнопку «Import WADL...».



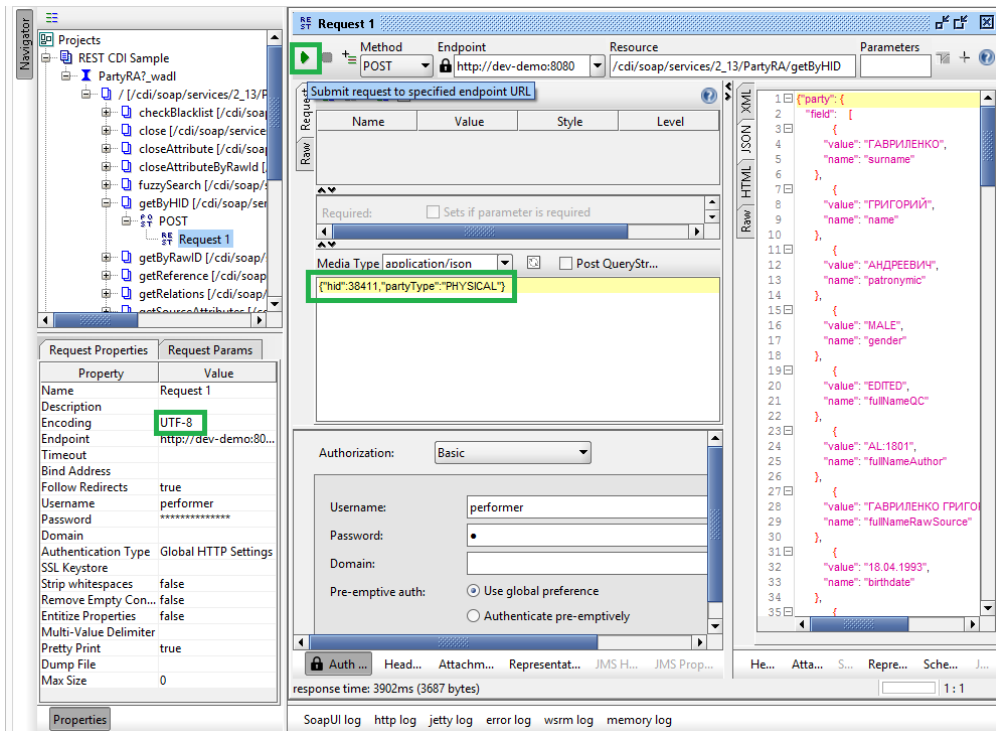
5. Вставить скопированную ранее ссылку и нажать «OK».



6. Указать данные пользователя с правами оператора (или вебсервиса, если эта роль настроена отдельно) для использования в HTTP Basic Auth.



7. Выбрать нужную кодировку, заполнить данные запроса, нажать Submit.



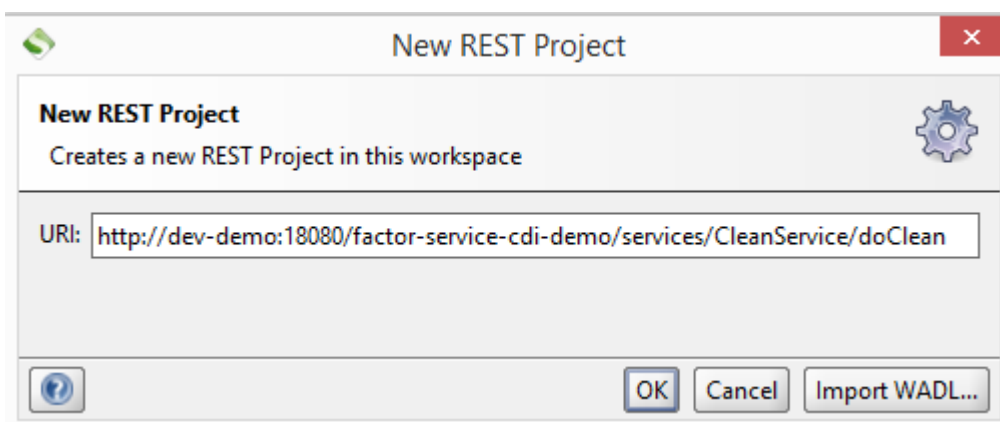
8.5.4.2 Методы Фактор

У Фактора нет WADL и авторизации, поэтому чуть по-другому. Зато можно отправлять как POST-, так и GET-запросы.

1. В SoapUI создать новый REST-проект.
2. В появившемся окне указать ссылку REST-сервиса Фактора.

```
-- Шаблон
http://{ip}:{port}/factor-service-
{customer}/services/CleanService/{metod}

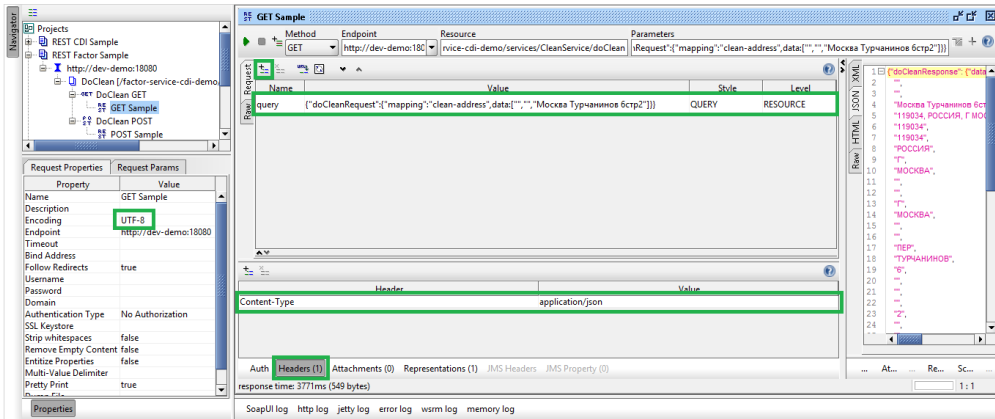
-- Например, для заказчика cdi-demo и метода doClean
http://dev-demo:18080/factor-service-cdi-
demo/services/CleanService/doClean
```



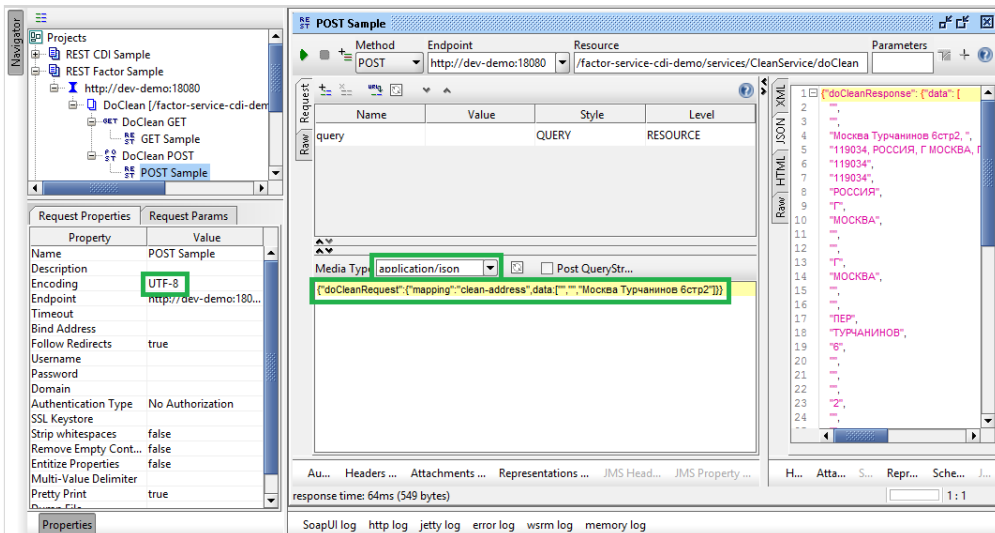
3. Для GET-запроса надо указать кодировку, добавить заголовок Content-Type=application/json, указать параметр query (можно делать через таблицу параметров, можно указывать в Parameters справа сверху). В

приведённом примере значение параметра query такое:

«{"doCleanRequest":{"mapping":"clean-address",data:["","","Москва Турчанинов бстр2"]}}». После нажатия Submit должен прийти ответ от Фактора.



4. Для POST-запроса надо указать те же параметры, но немного иначе.



8.5.5 Как посмотреть JMS, уходящие в RabbitMQ

Под рестриками, чтобы ссылки давать

8.5.5.1 Подготовка

Установить RabbitMq. У нас он развернут тут — <http://dev-vota.intranet.cleandata.ru:15672> (guest/guest)

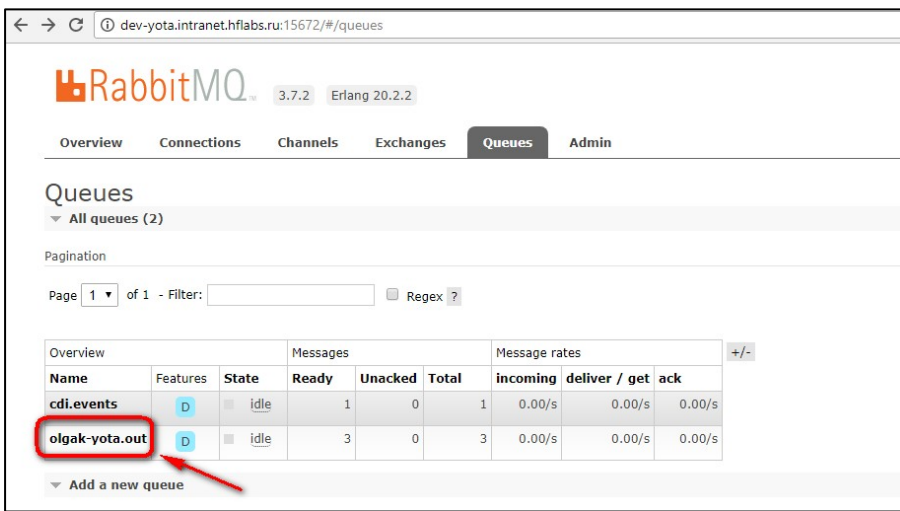
Для настройки использовать параметры:

```
jms.events.rabbitmq.enabled = false
jms.events.rabbitmq.destination.name = cdi.events
jms.events.rabbitmq.destination.isTopic = false
jms.events.rabbitmq.host = rabbitmq-host
jms.events.rabbitmq.port = 5672
jms.events.rabbitmq.username = guest
jms.events.rabbitmq.password = guest
```

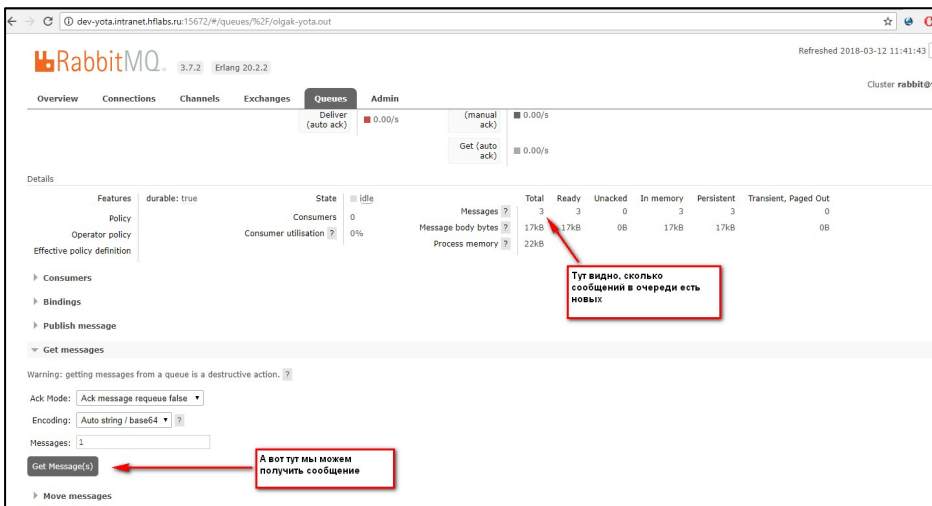
Заметьте, что в настройках порт 5672, а ссылка в админку — 15672. Это нормально, это административный порт

8.5.5.2 Проверка сообщений

1. Открыть очереди — <http://dev-yota.intranet.cleandata.ru:15672/#/queues>
2. Выбрать ту очередь, что настроили в destination.name:



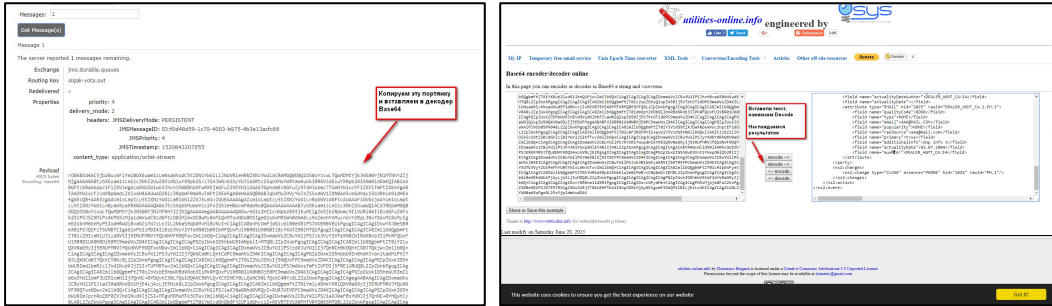
3. Нажать «Get Message» — получите сообщения или ошибку «очередь пуста». Количество сообщений видно сверху.



Как только вы получили сообщение, оно пропадает из очереди → это нормально. Это в тибке сообщения остаются, а тут если получил, то второй раз не посмотришь из интерфейса.

4. Скопировать сообщение (оно в кодировке Base64) и декодировать его. Сервисы декодирования:

- <http://www.utilities-online.info/base64/#.WqZNgGrFK70> → более удобный
- <https://www.base64decode.org/>



8.5.6 Как посмотреть JMS, уходящие в IBM MQ

8.5.6.1 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.cleandata.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
```

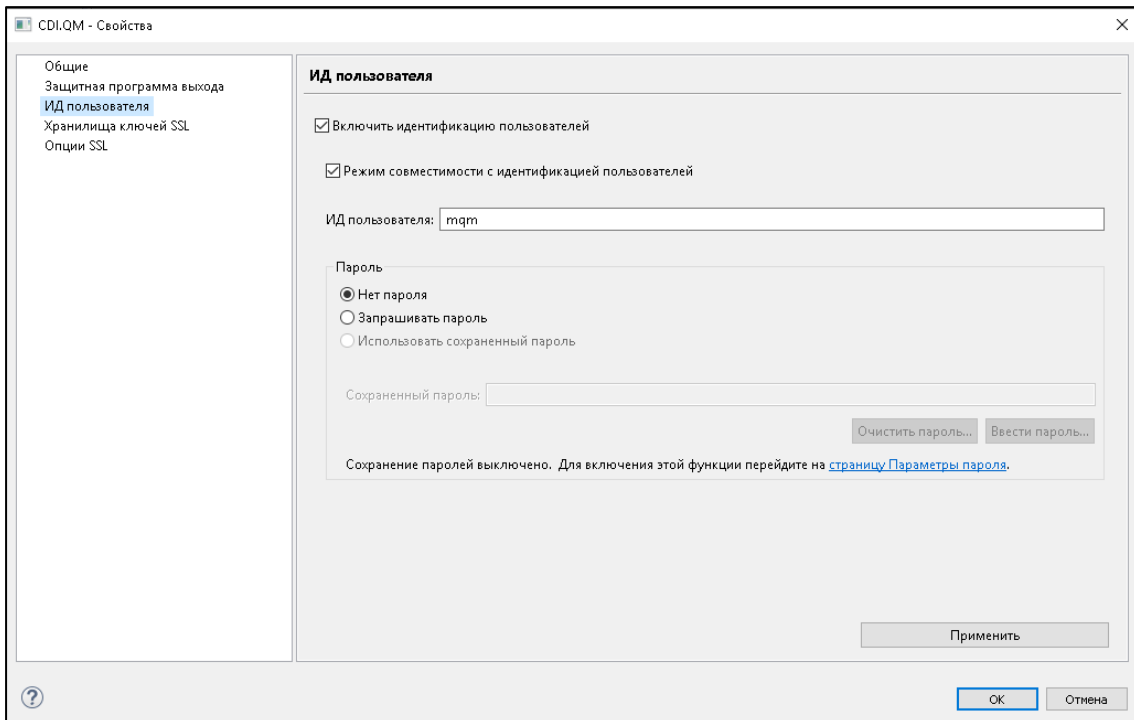
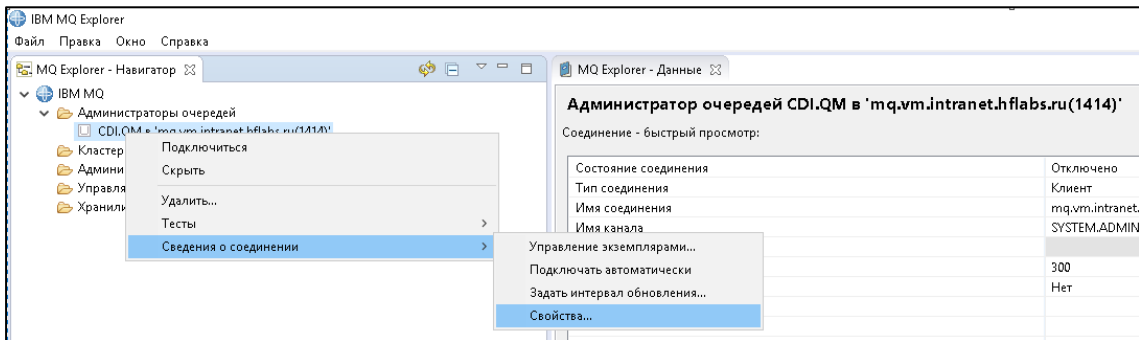
8.5.6.2 Просмотр через MQ Explorer

Запуск и мониторинг в MQ, следовать инструкции, но использовать параметры системы «Гражданский фактор»:

- Имя администратора очередей — `CDI.QM`
- Имя хоста — `mq.vm.intranet.cleandata.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.

8.5.6.3 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

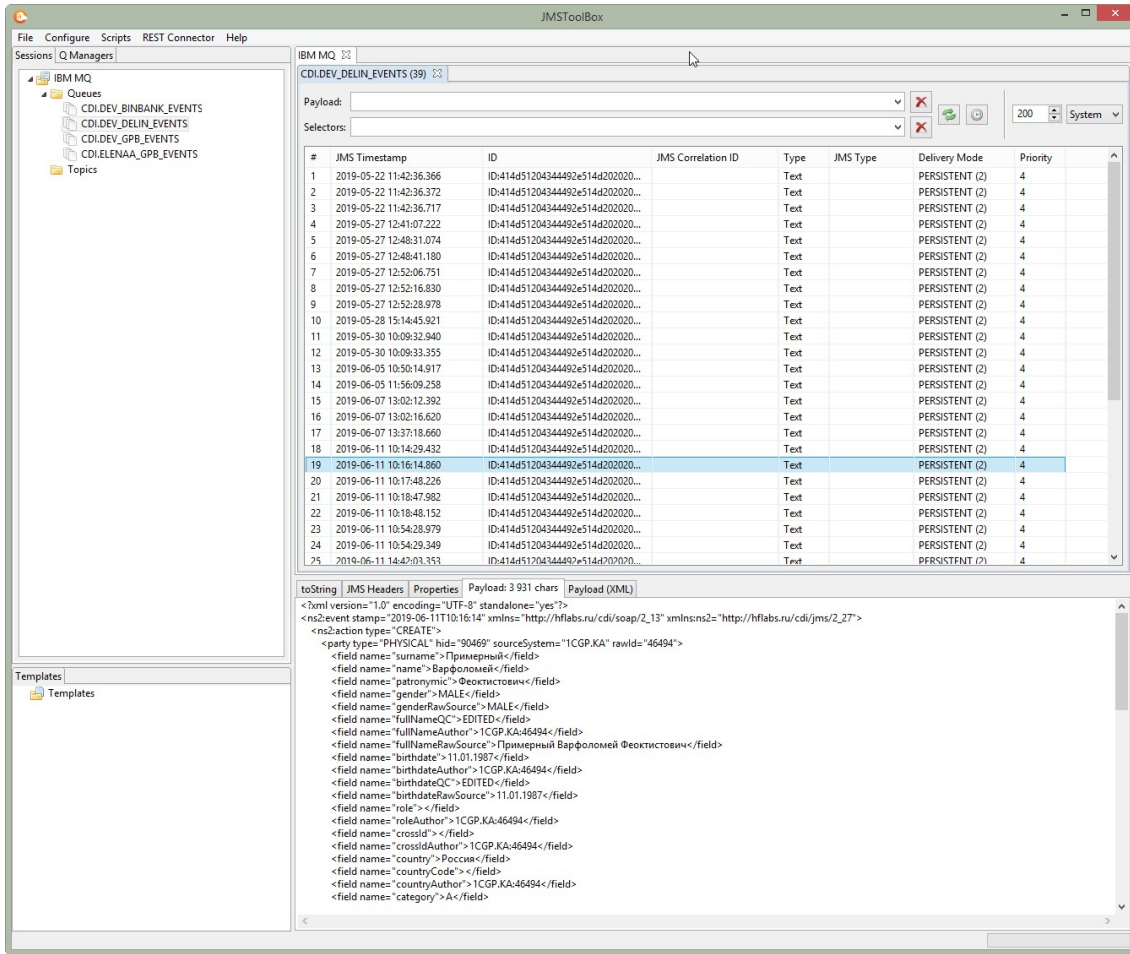
 If the userid/password are NOT saved in the session, the 'REST' and 'scripts' features of JMSToolBox will not work

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:



8.5.6.4 Запуск и мониторинг в MQ

8.5.6.4.1 Установка IBM WebSphere 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.

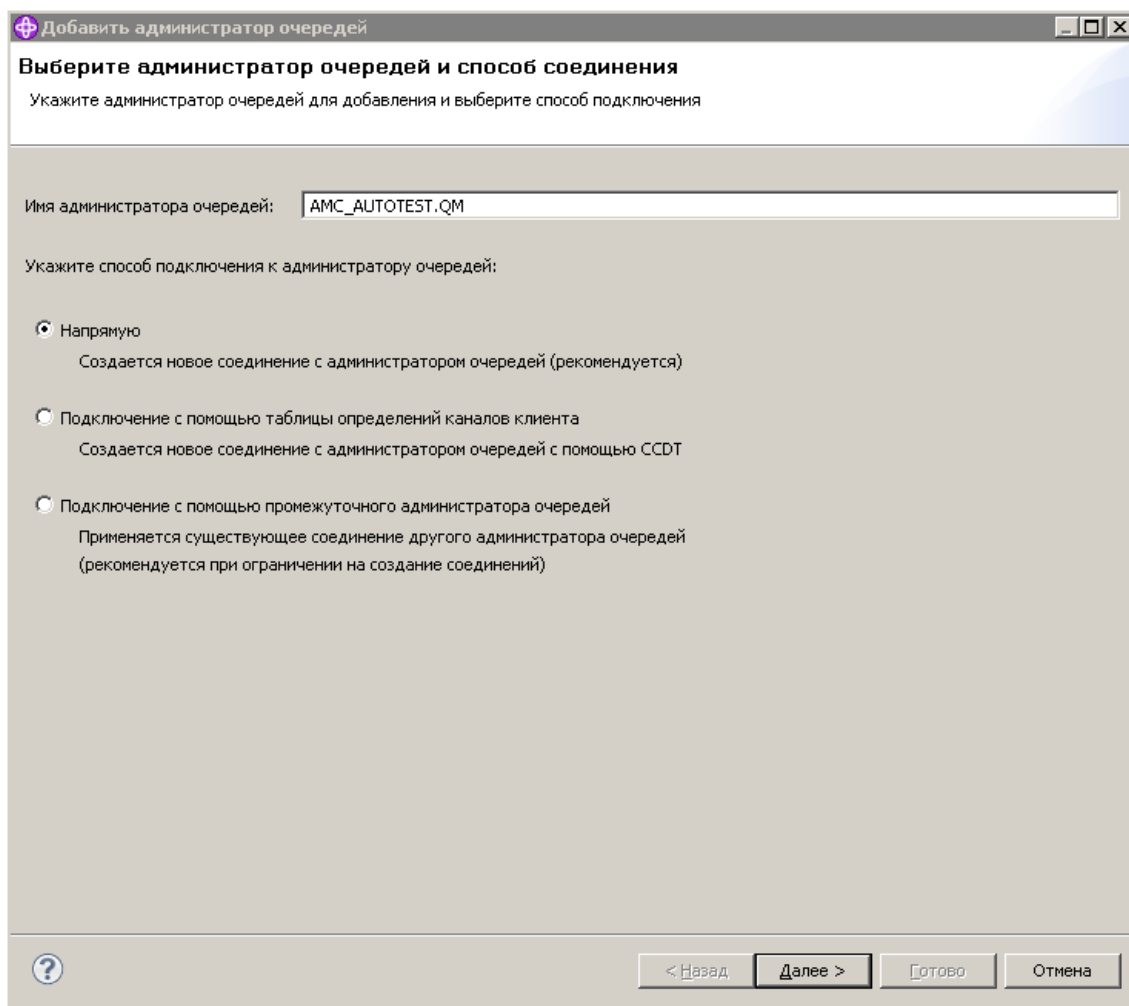
Дистрибутив под Windows здесь:

```

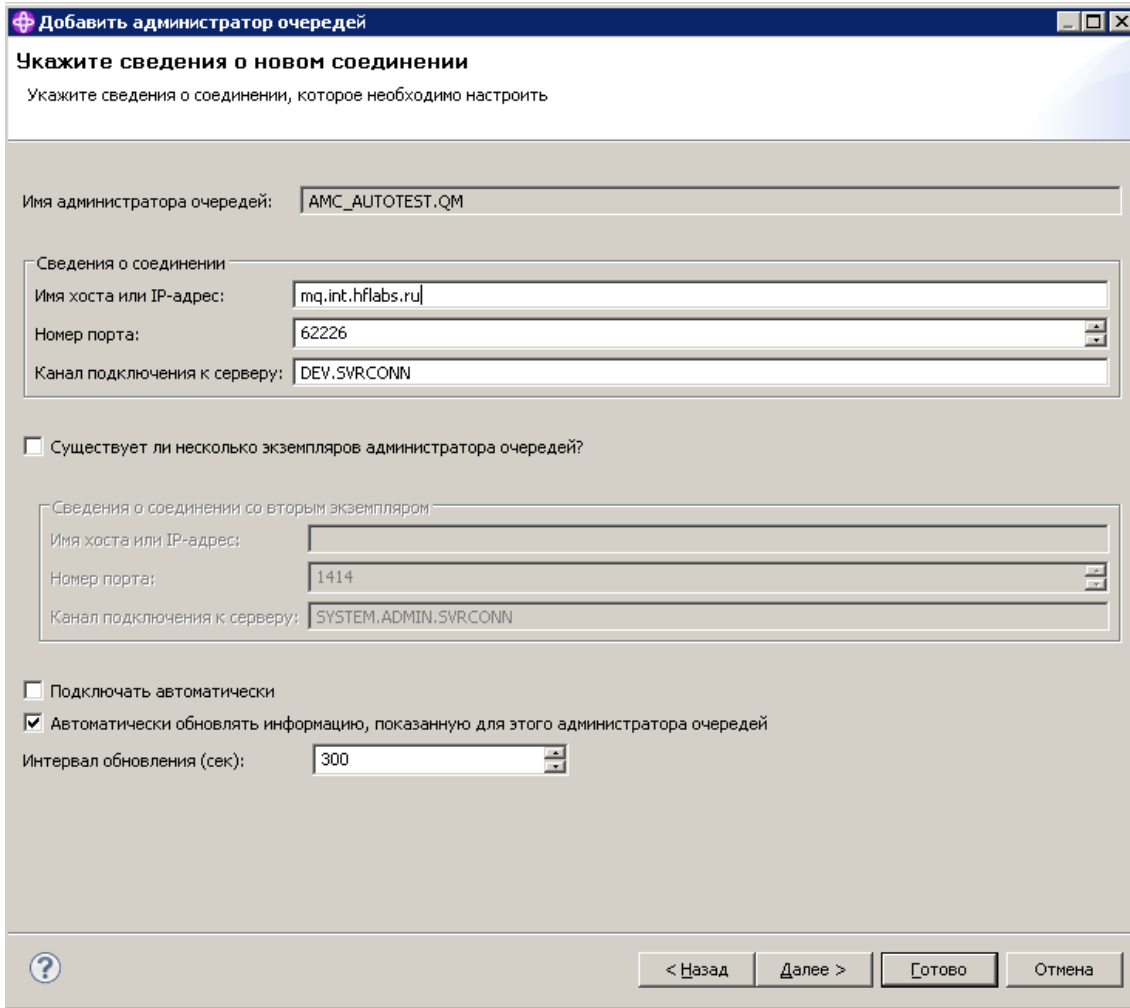
\\10.0.61.20\share\distrib\ms0t_mqexplorer_75_windows.zip
ИЛИ
http://cloud.cleandata.ru/index.php/apps/files/?dir=/others/mq
    
```

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:

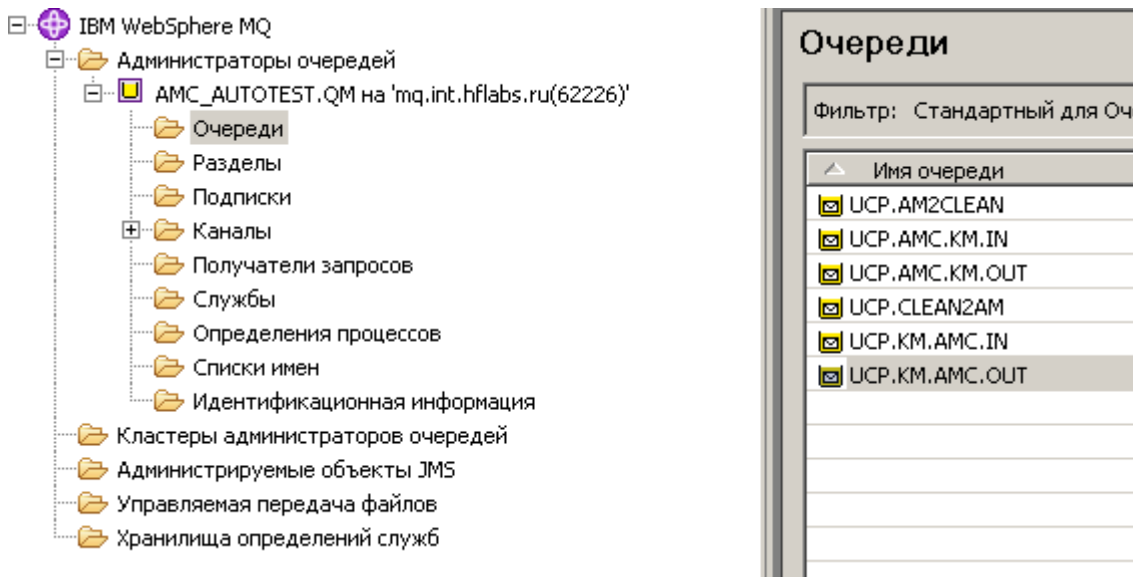


и параметры подключения к нему (хост, порт, канал):

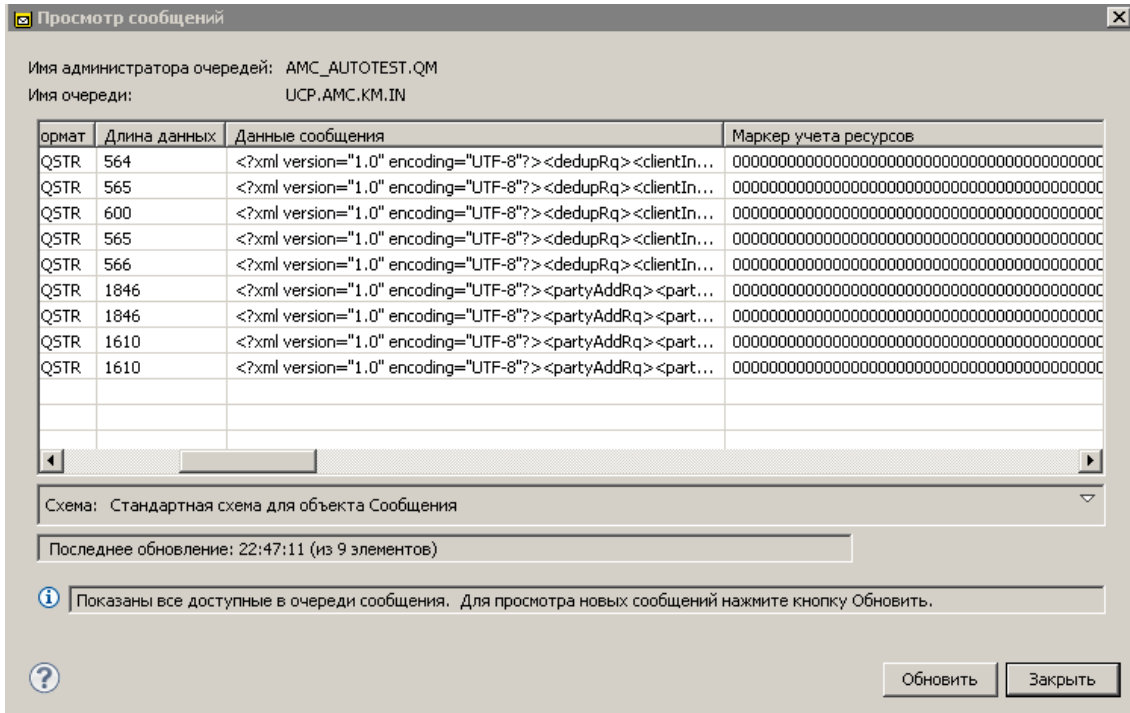


Если для подключения к каналу не используется SSL, то жмем "Готово".

Открывается окошко с очередями на этом менеджере:



Все, работаем. Например, смотрим, какие сообщения есть в очереди:



8.5.6.4.2 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.

8.5.7 Как посмотреть JMS, уходящие в Tibco

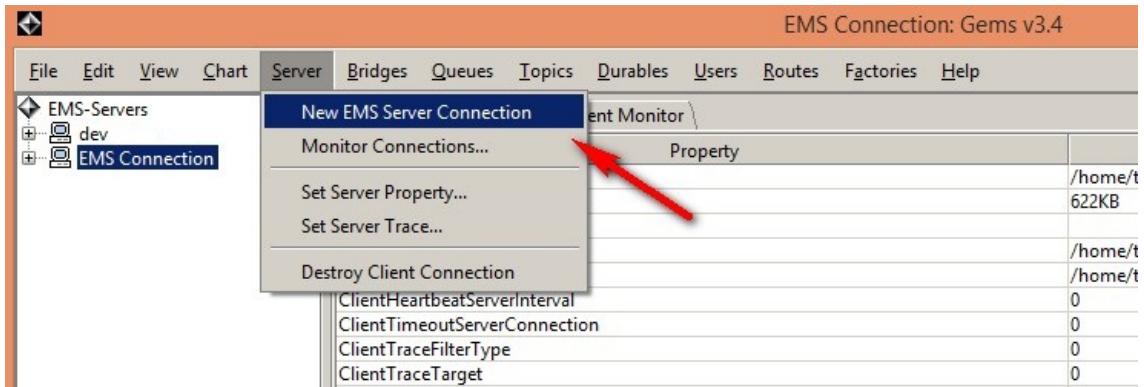
8.5.7.1 Подготовка

Скачать утилиту Gems : распаковать из [архива](#). Утилита в архиве уже настроена, содержит библиотеки нужной версии под нашу внутреннюю tibco, а также уже содержит параметры подключения до тибко на dev-орен.

8.5.7.2 Проверка сообщений

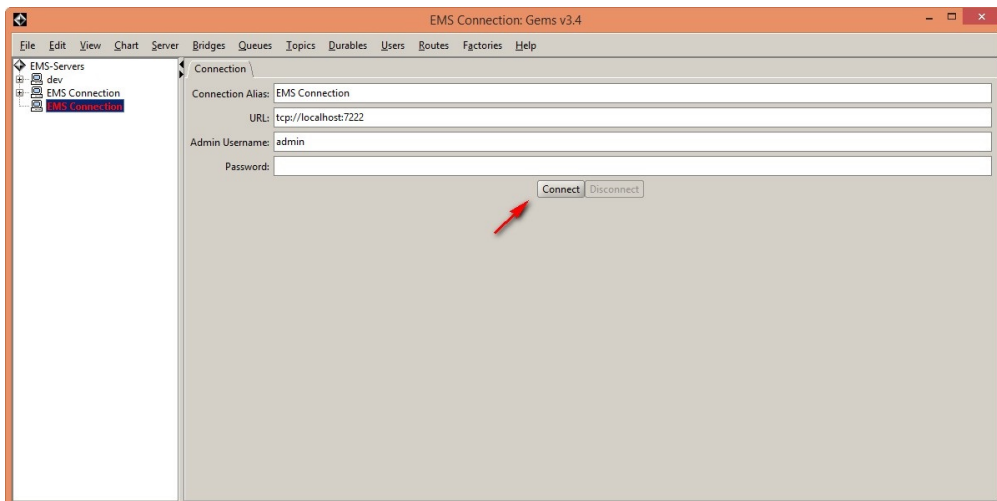
1. Запустить Gems — `rungems.cmd`

2. Server – New EMS Server Connection

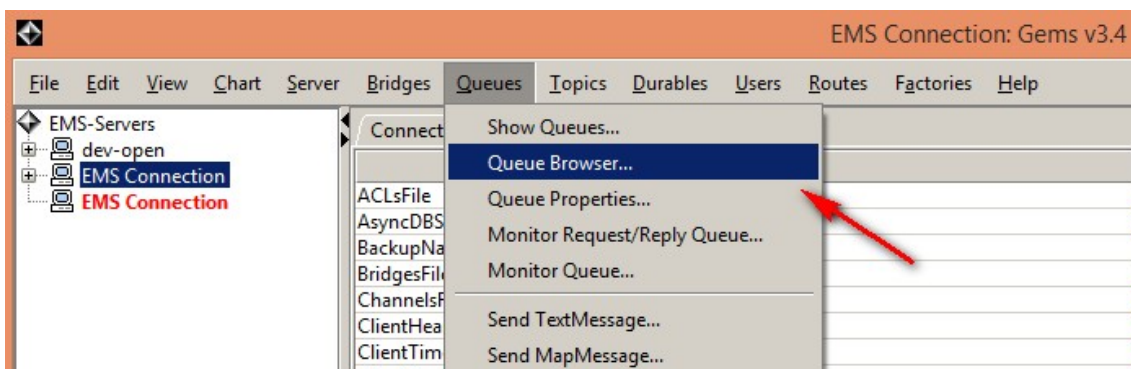


3. Указать настройки подключения, нажать Connect :

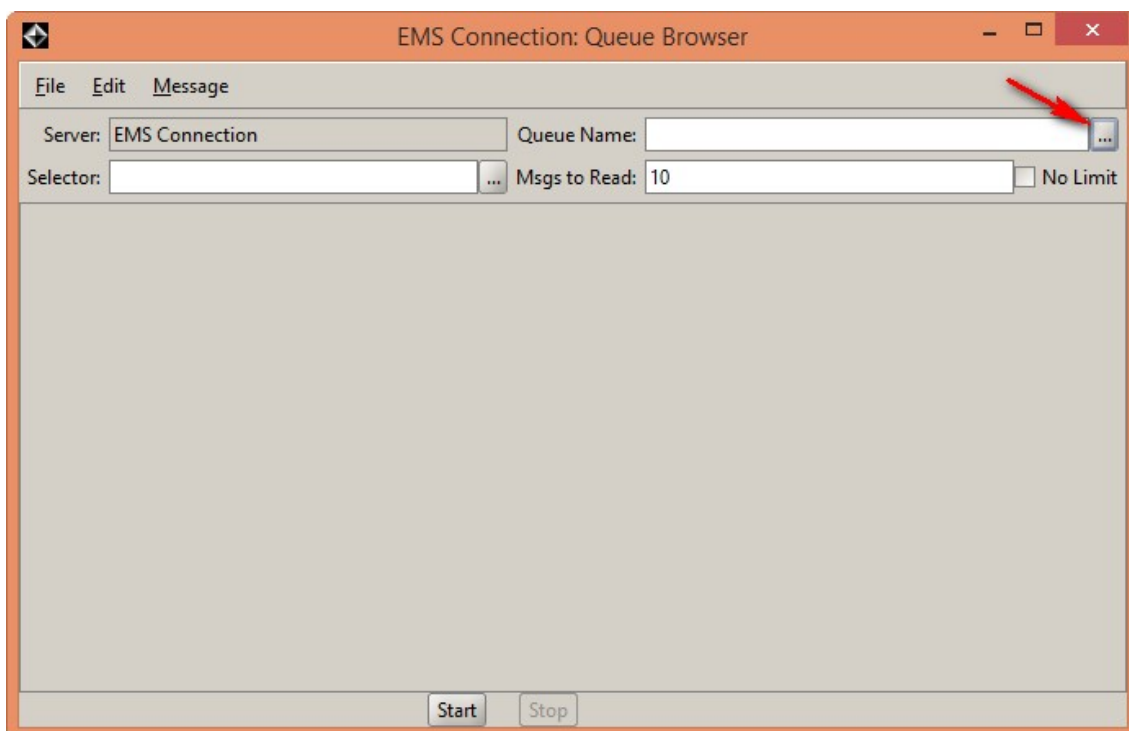
- Alias — имя, которое будет отображаться слева в меню
- URL — URL подключения к Tibco
- Username\password — данные для подключения



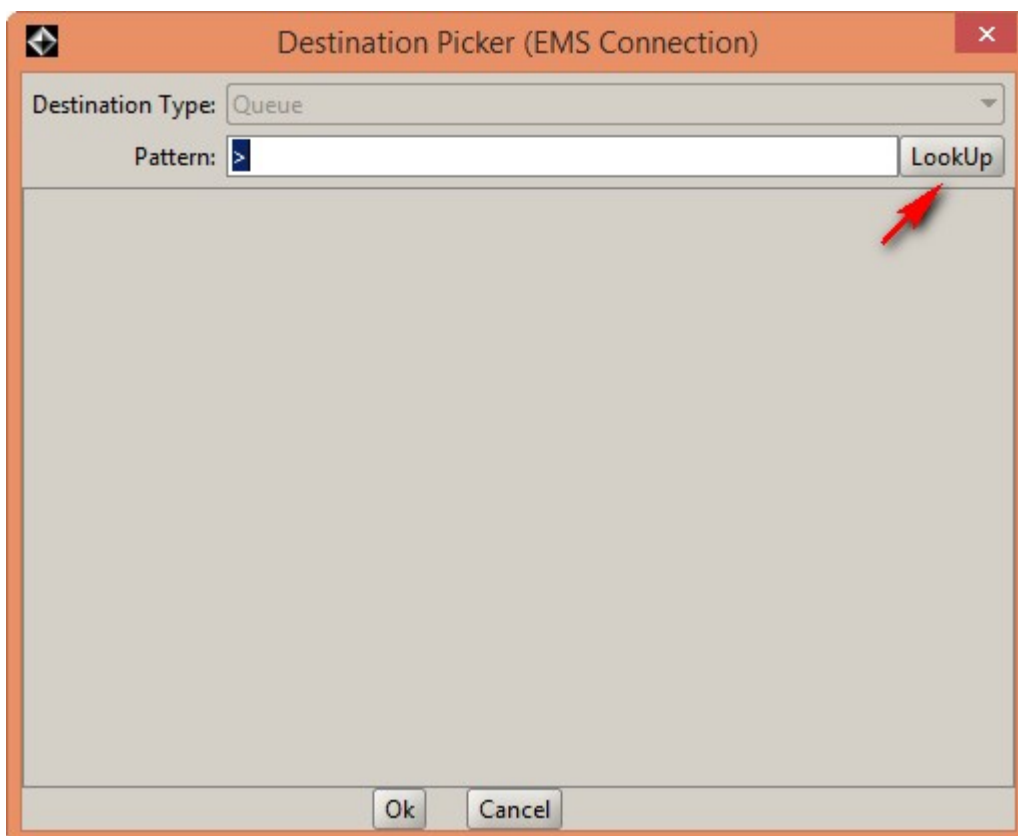
4. Выделить соединение — Queues – Queue Browser



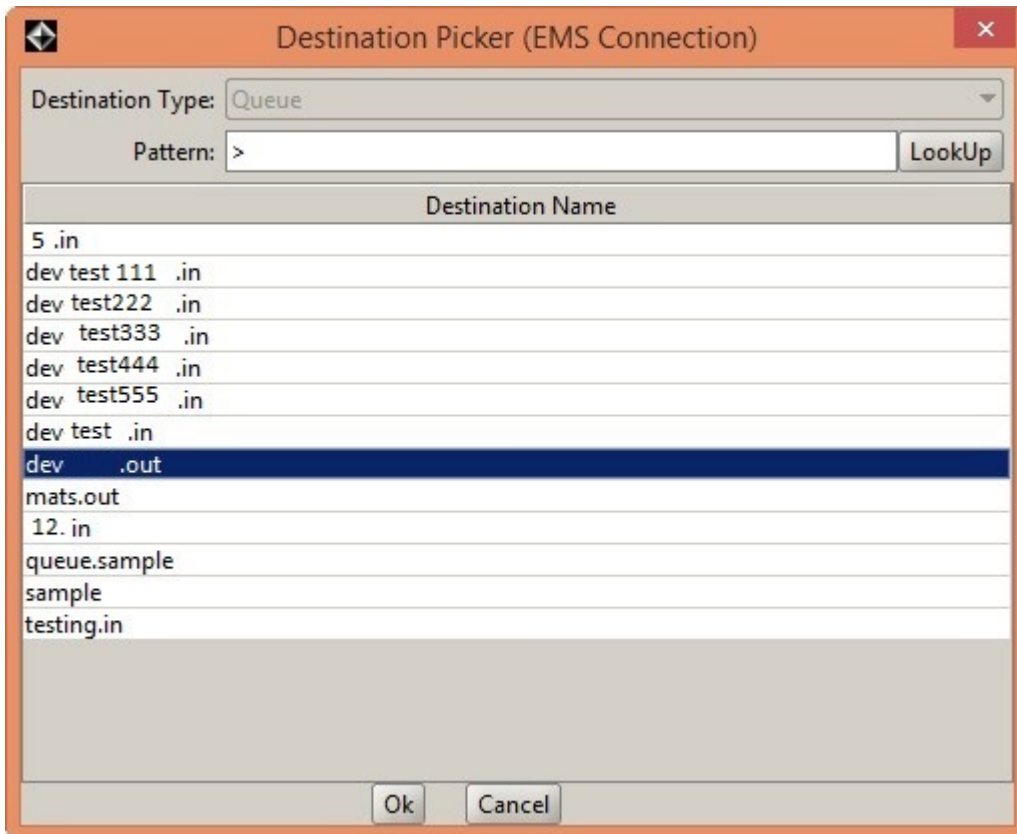
5. Нажать «...» около Queue Name



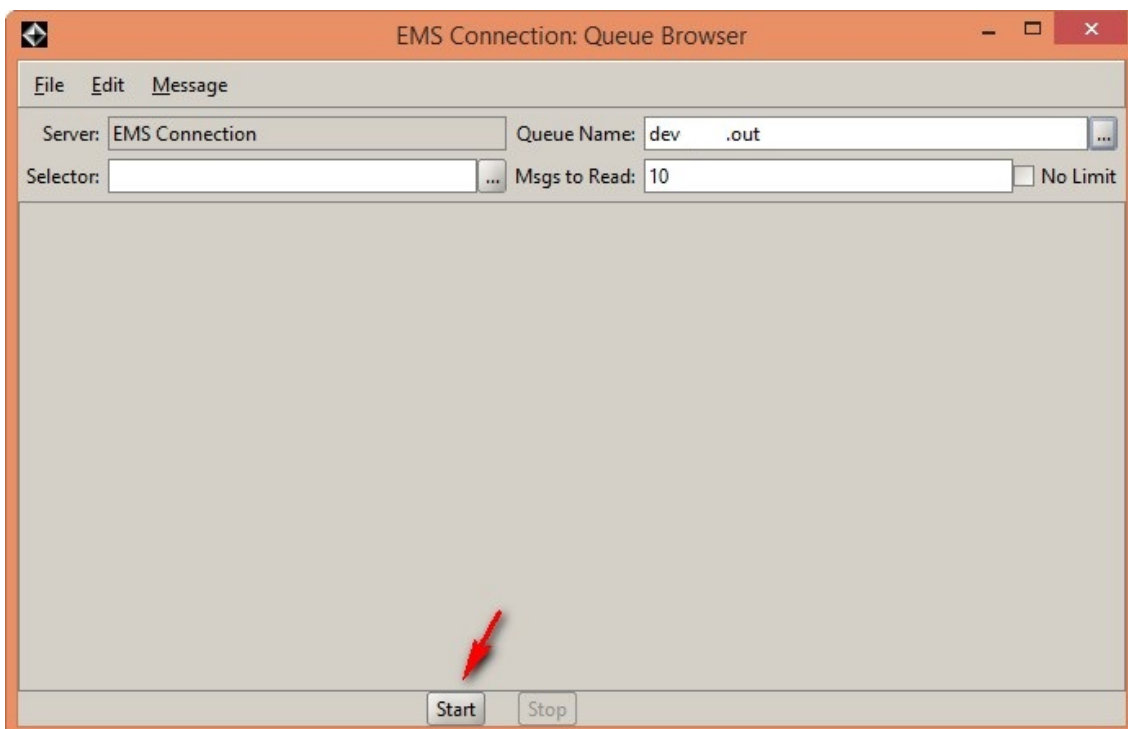
6. LookUp — система покажет все доступные очереди



7. Выбрать очередь, щелкнуть дважды. Выбираем исходящую очередь:

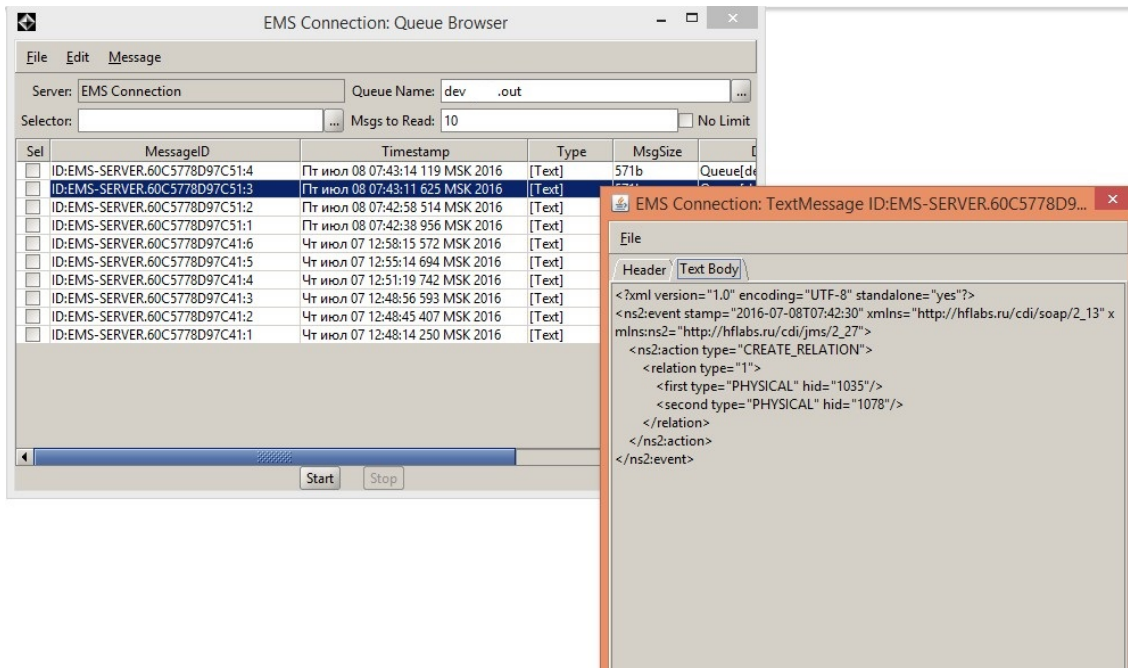


8. Нажать Start



9. Выбрать любое сообщение из списка. Если список пуст, сгенерить сообщение вручную, например, отправив запрос [save](#)

На вкладке Text Body будет текст JMS-события



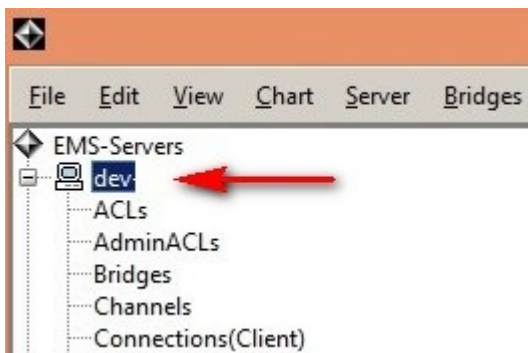
8.5.8 Как послать в Tibco входящее JMS-сообщение

8.5.8.1 Подготовка

Скачать утилиту Gems : распаковать из [архива](#). Утилита в архиве уже настроена, содержит библиотеки нужной версии под нашу внутреннюю tibco, а также уже содержит параметры подключения до тибко на dev-орен.

8.5.8.2 Отправка входящего JMS-сообщения со стандартными заголовками

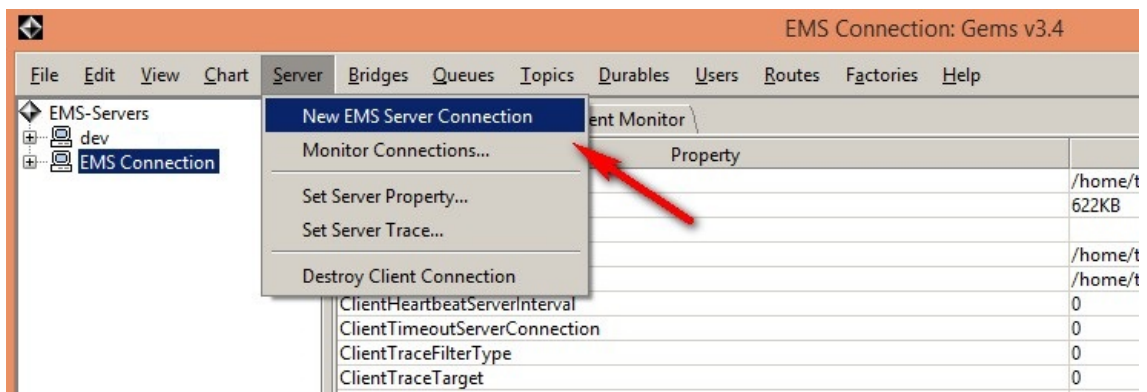
1. Запустить Gems — `rungems.cmd`
2. Если уже есть EMS Server Connection — выделить его



Если соединения еще нет, создаем:

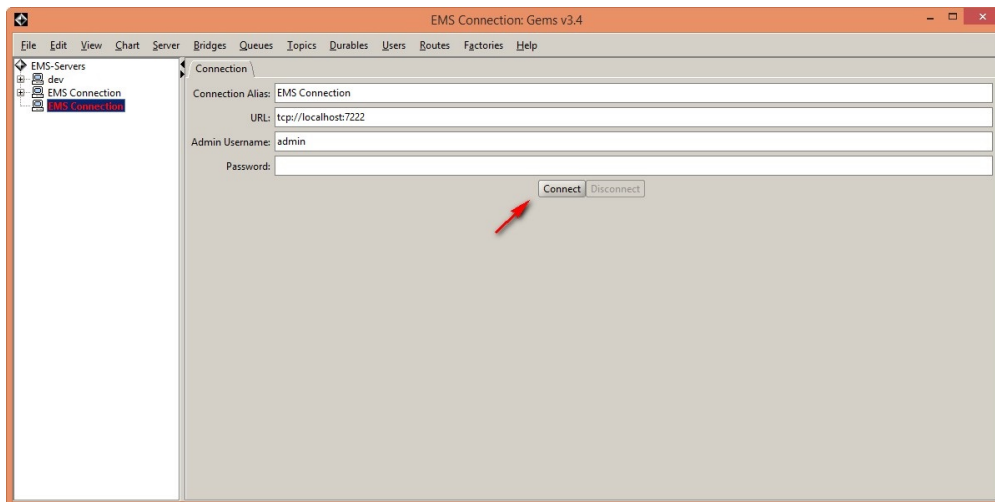
Как создать

Server — New EMS Server Connection

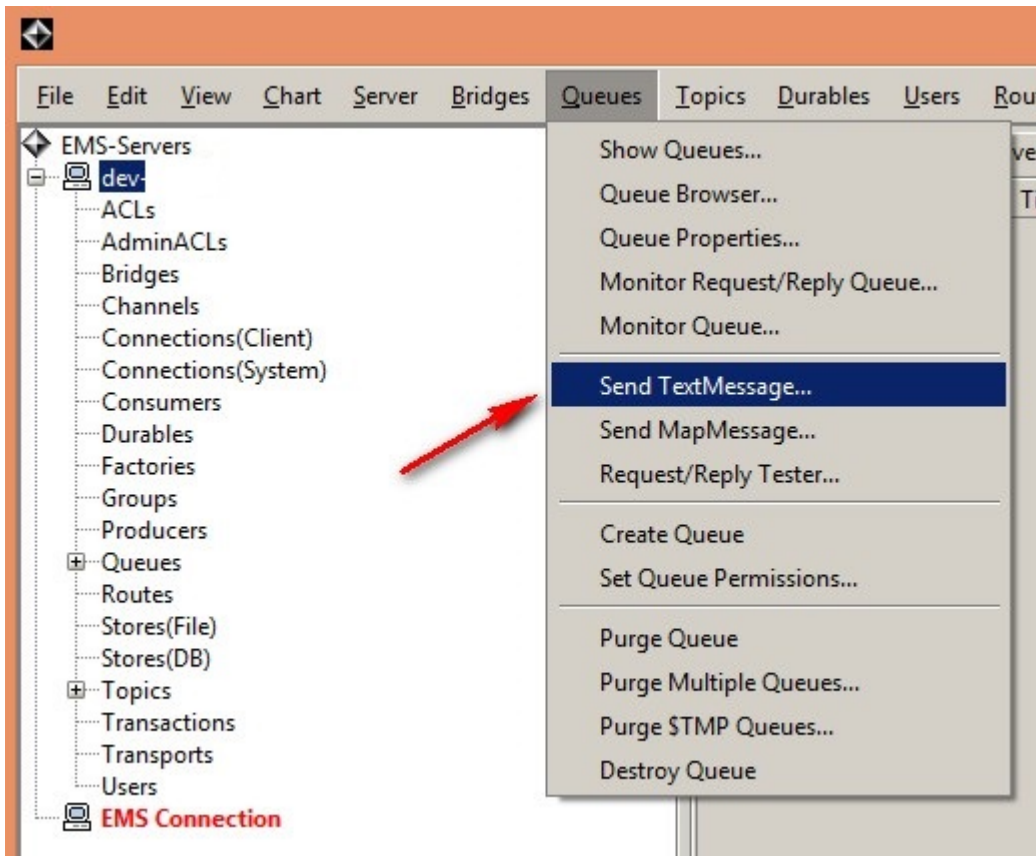


Указать настройки подключения, нажать Connect :

- Alias — имя, которое будет отображаться слева в меню
- URL — URL подключения к Tibco
- Username\password — данные для подключения



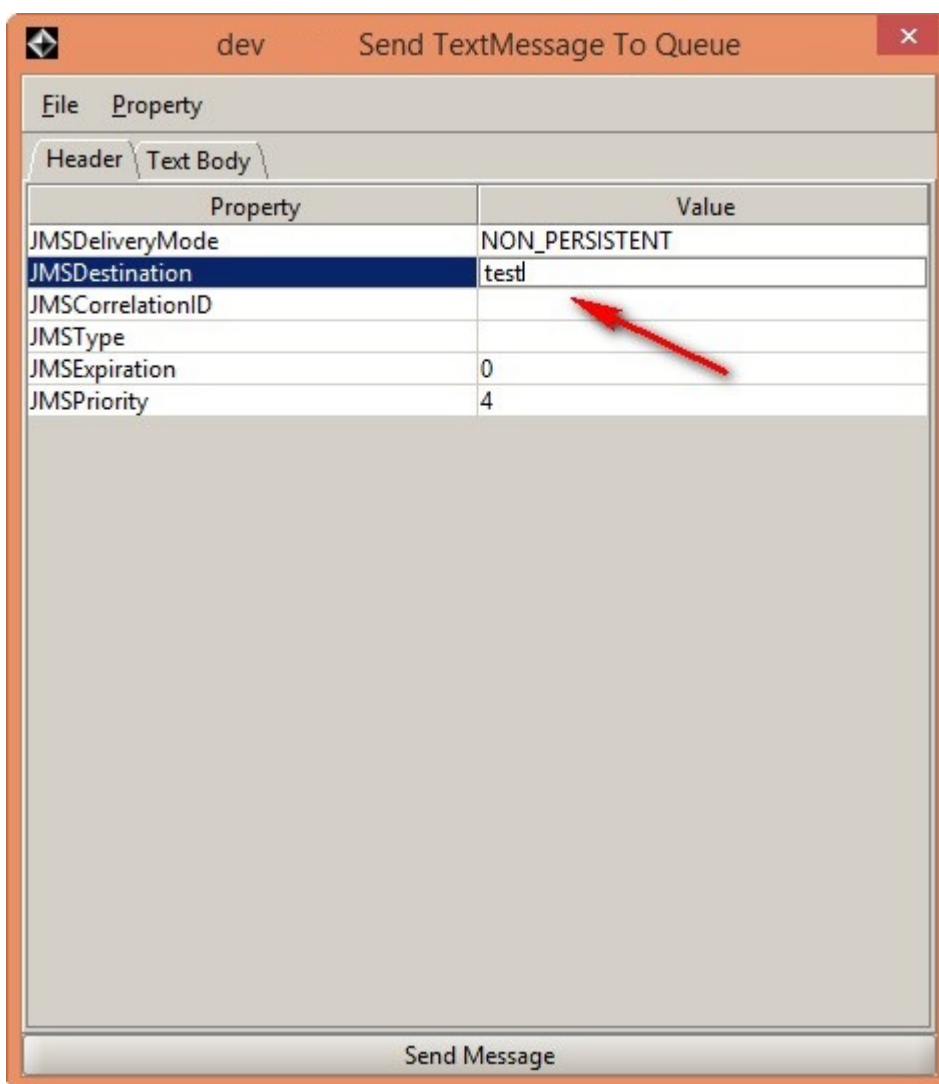
3. Queues – Send TextMessage



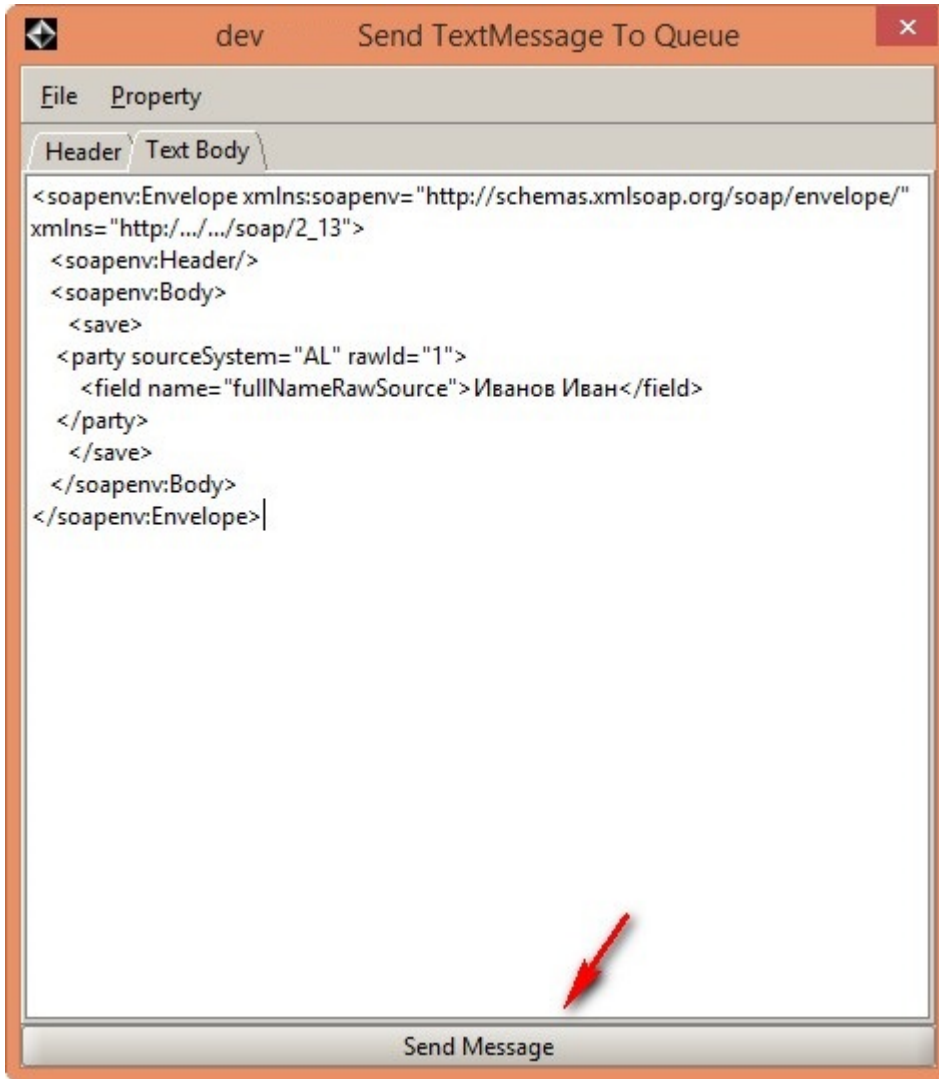
4. Указать параметр `JMSDestination` – место назначения.

Для системы «Гражданский фактор» 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)

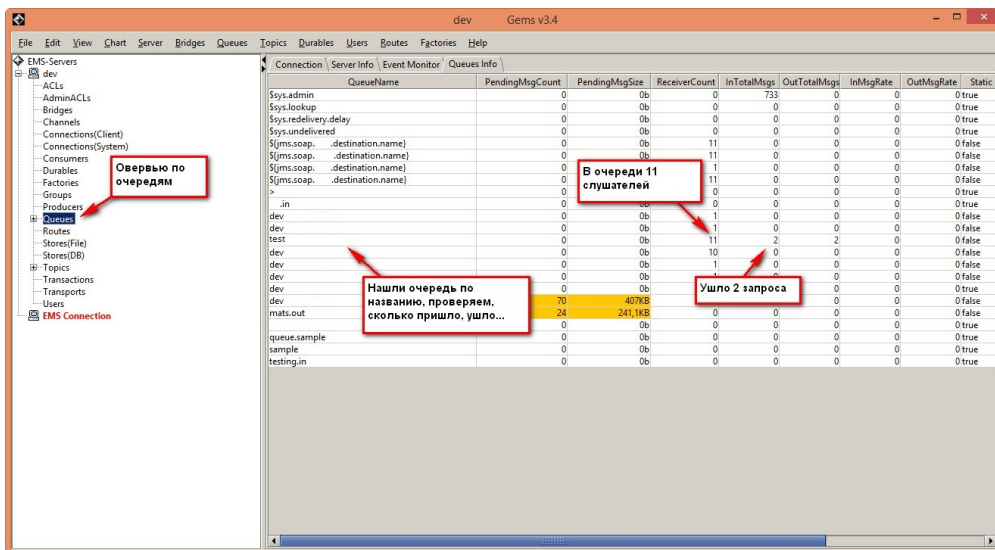


5. Все, стандартные заголовки есть, переключаемся на вкладку `Text Body` и отправляем запрос. Запрос обычный, без особых тегов, можно полностью скопировать из `Soap Ui`.



Потом проверяем внутри приложения — пришел запрос? Изменились данные?

В Gem можем тоже проверить — выделяем в левом меню Queues и видим overview по всем очередям

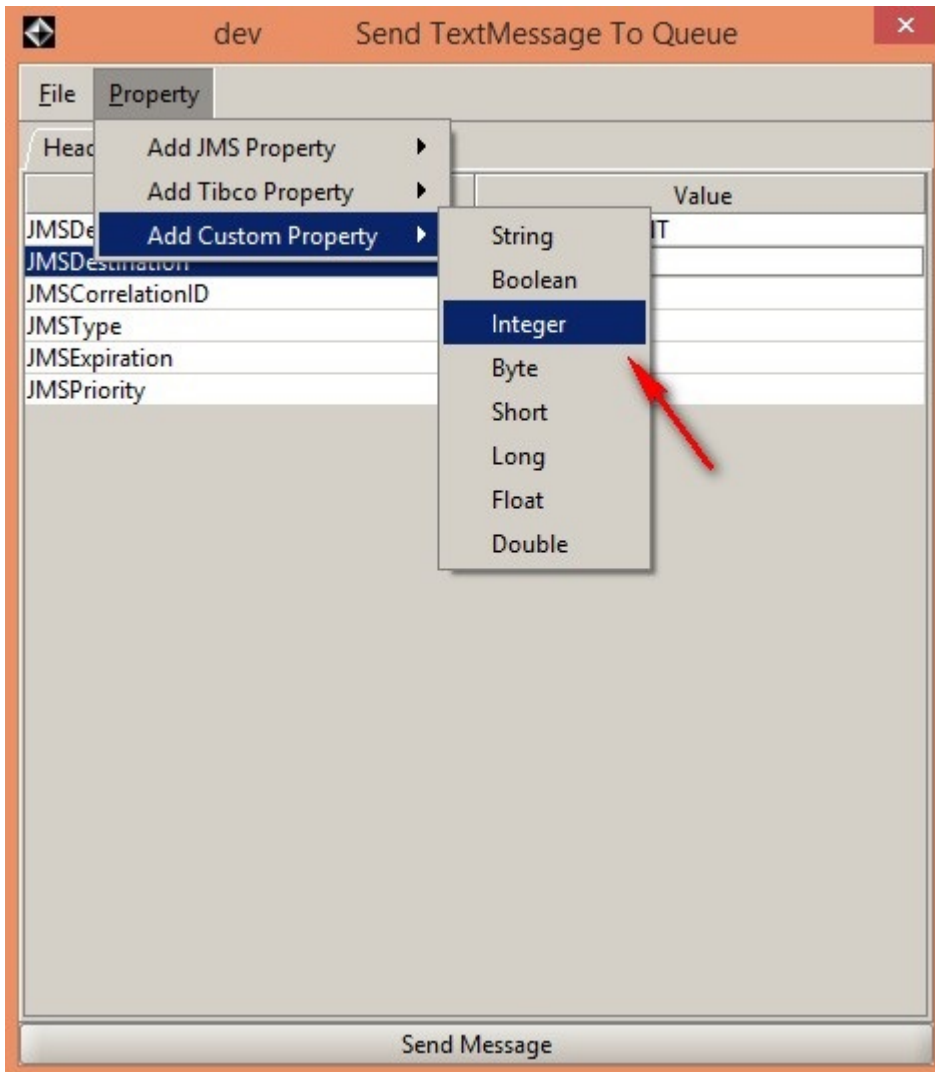


8.5.8.3 Нестандартные заголовки

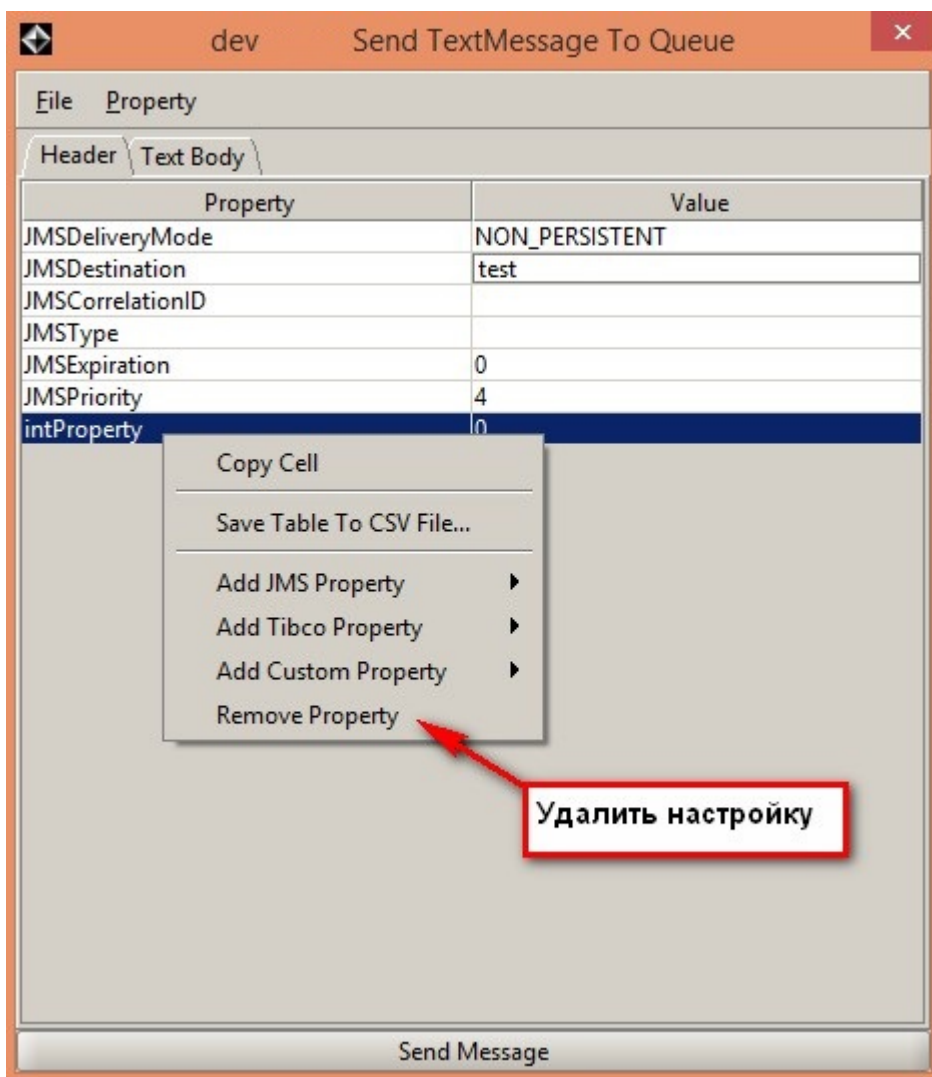
Если у нас не один читатель очереди, а 10, их надо как-то разделять. Для этого во входящее сообщение добавляется новый параметр в заголовке. Допустим, это будет ID с типом Integer.

Как проверить, что можно послать в приложение сообщение с ID в заголовке?

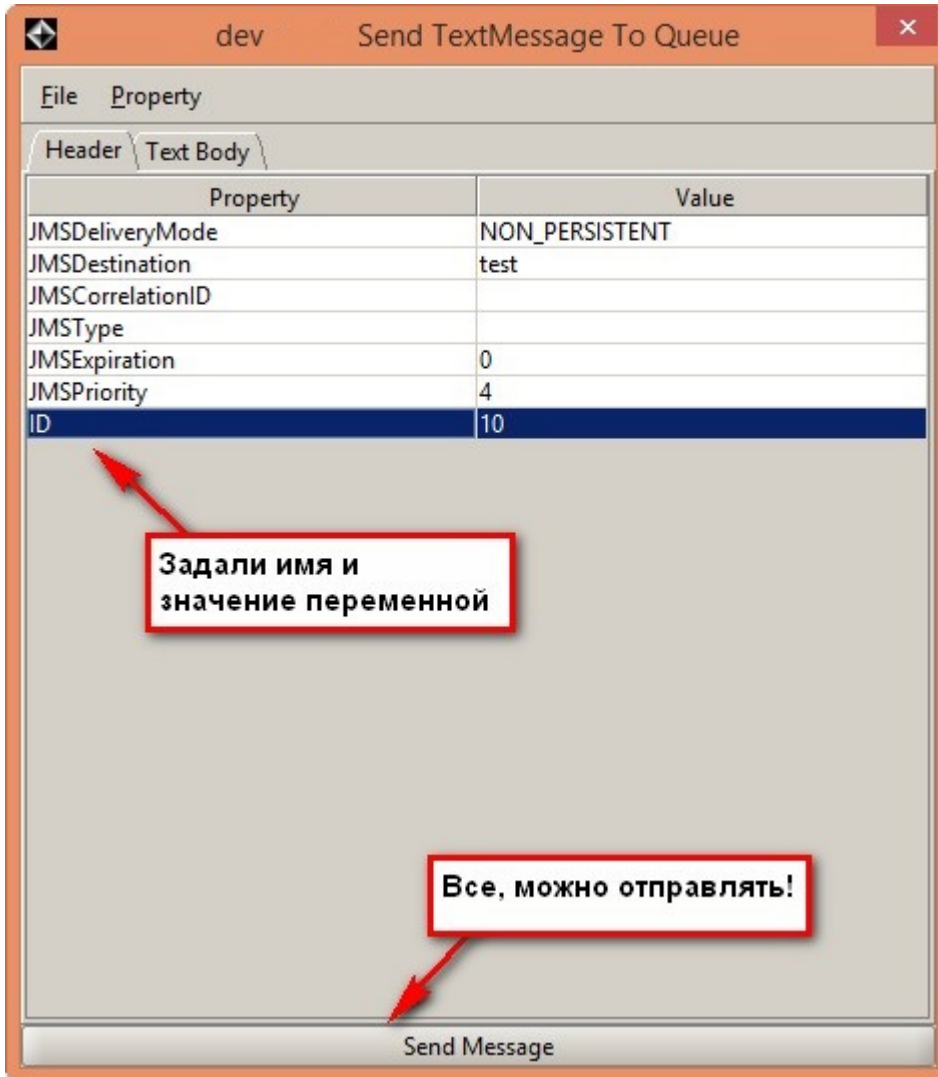
1. Queues – Send TextMessage
2. Property – Add Custom Property – <выбираем тип>



Настройка добавлена! При желании ее можно удалить по правой кнопке мыши. Стандартные заголовки удалять нельзя



3. Указываем имя и значение переменной — Send Message



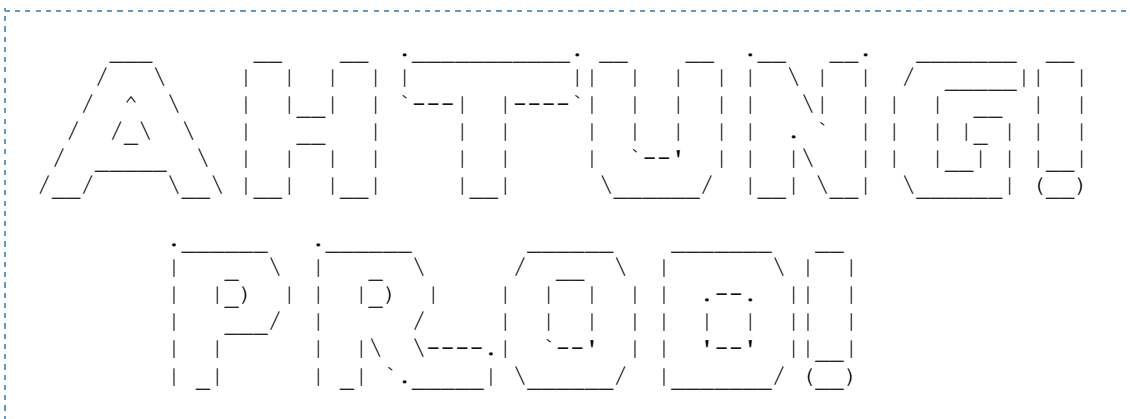
Все, наше сообщение отправлено с новым заголовком!

8.5.9 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.

8.5.9.1 Настройка

Указать приветствие в файле `/etc/motd`



Пример файла — [motd](#)

8.5.10 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);
```

8.6 Запуск задач / триггеров

В: Уточните, пожалуйста. Задачу passportInvalidFlagActualization безопасно в рабочее время запускать?

О: Запускать в рабочее время безопасно, идти по ветке «Обработка всей базы».

Перед проверкой рекомендую [обновить](#) перечень недействительных документов на серверах, если этот процесс ещё не автоматизирован

8.6.1 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](#)

8.6.1.1 Зачем нужна миграция

Переезд на исторические таблицы нужен для переноса части БД на медленный диск. Все таблицы, в которых хранились версии будут разделены на две:

PHYSICAL_PARTY → PHYSICAL_PARTY + PHYSICAL_PARTY_HIST

В `PHYSICAL_PARTY` останутся только актуальные записи (`version = 0`), а все старые версии (изменения имени, даты рождения, даты актуальности и другие) будут храниться отдельно.

Это поможет:

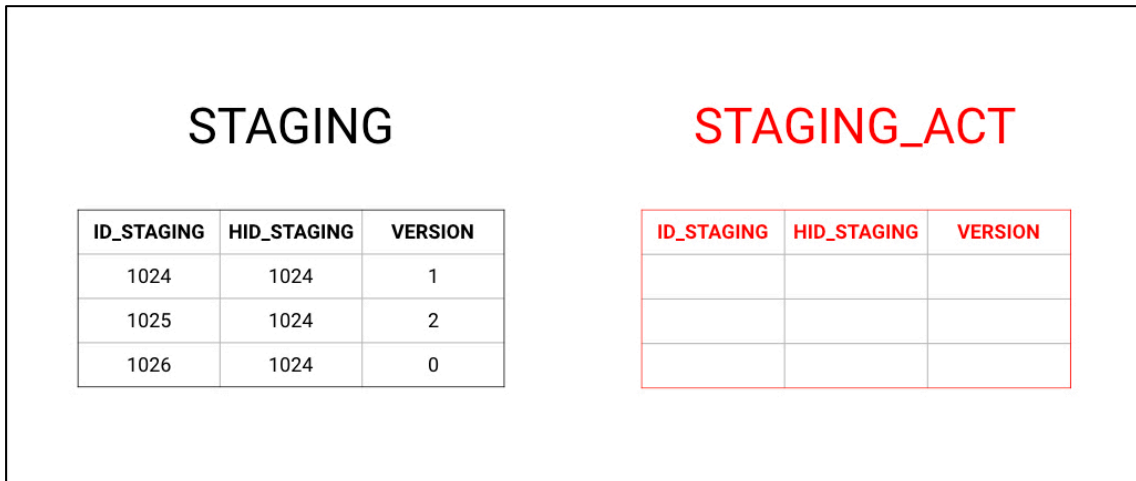
1. Ускорить задачи:
 - a. Перестроение индекса — не нужно будет делать отсев прошлых версий в селекте
 - b. Перестроение мат. view — чем меньше данных в таблице, тем быстрее работает select
2. Купить под актуальные данные в БД SSD-диск, а старые хранить на обычном. SSD стоит дорого, покупать его под большую базу невыгодно.

Ускорение будет заметно только на больших базах, где уже накопилась история: записей с `version ≠ 0` сильно больше, чем актуальных (`version = 0`).

8.6.1.2 Как проходит миграция

Миграция проходит в 4 фазы, чтобы избежать длительного даунтайма сервиса.

8.6.1.2.1 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:

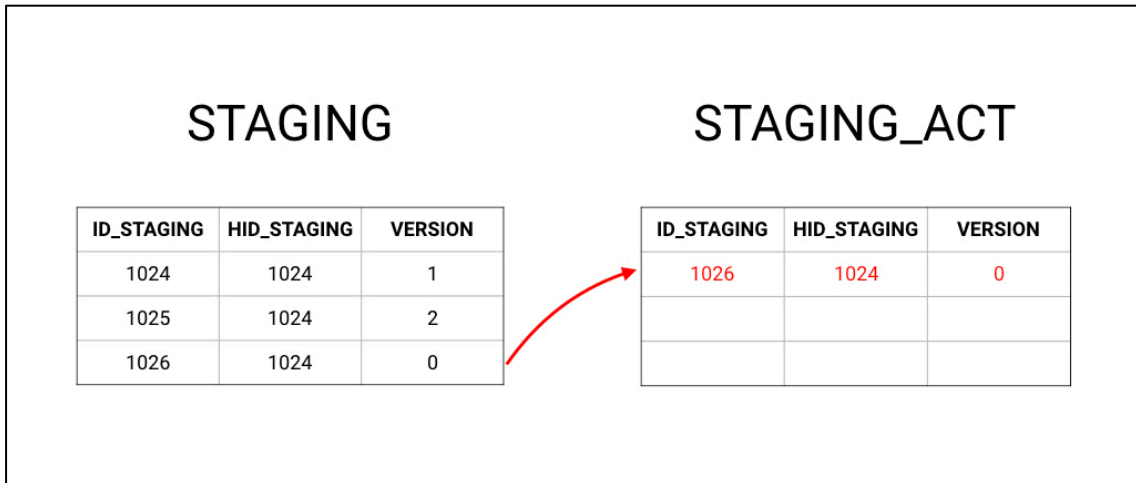
STAGING_ACT

| ID_STAGING | HID_STAGING | VERSION |
|------------|-------------|---------|
| | | |

Операция повторяется для каждой версионной таблицы.

Фаза проходит быстро и автоматически. Сотрудники службы сопровождения вендора решения добавляют скрипты автоматизации в поставку, вручную ничего делать не нужно.

8.6.1.2.2 Фаза 2. Заполнение таблиц _ACT



В ней сотрудники службы сопровождения заполняют _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 |

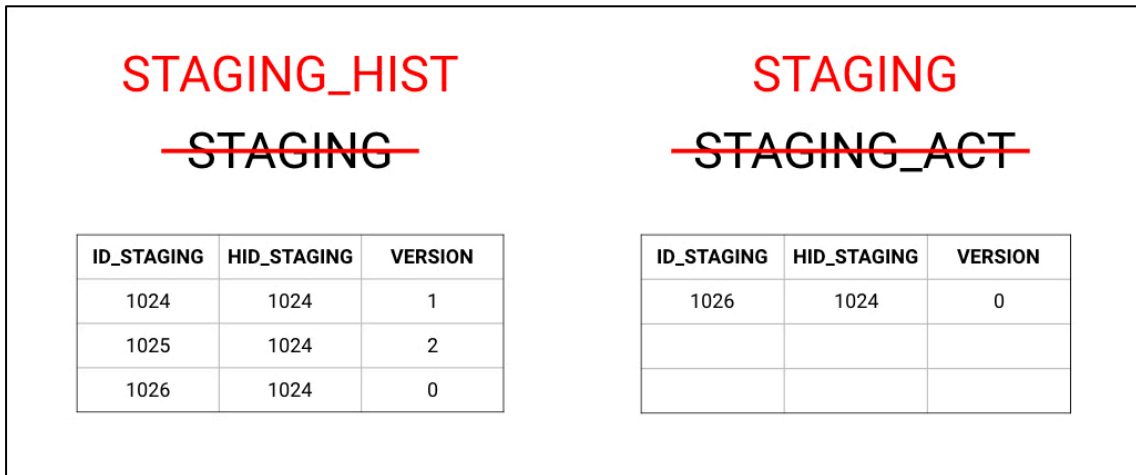
Пока это подготовительный этап, система все еще живет на старых таблицах.

Наполнение _ACT таблиц происходит двумя путями:

- самостоятельно — любое изменение / создание записи инициирует ее 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.

8.6.1.2.3 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:

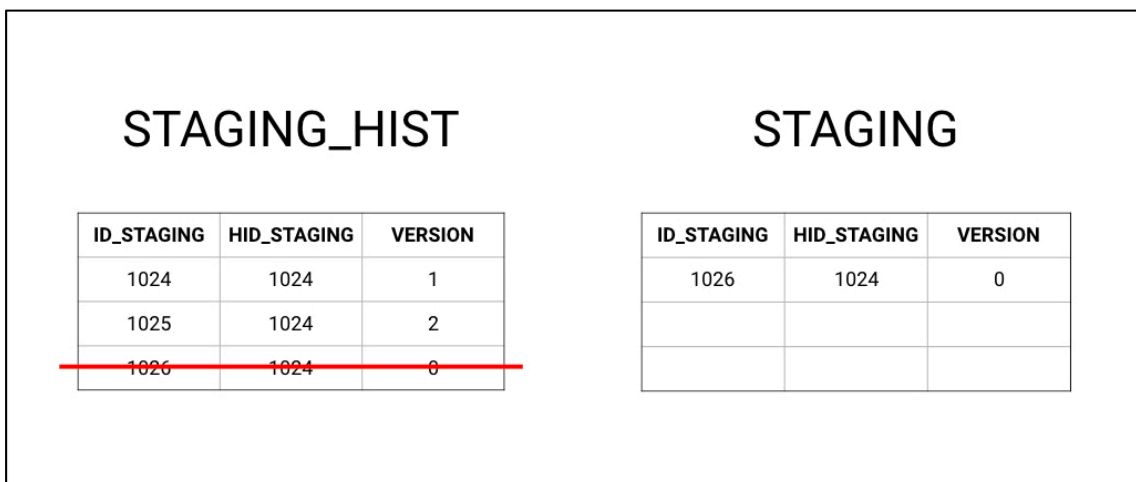
- все текущие таблицы переименовываются в `_HIST . STAGING` → `STAGING_HIST`;
- все актуальные таблицы переименовываются, оттуда удаляется префикс `_ACT . STAGING_ACT` → `STAGING`;
- все view пересоздаются, чтобы они умели работать с новой структурой БД.

Команда службы сопровождения вендора решения выдает скрипт ручной миграции. Нужно остановить сервис и запустить его.

8.6.1.2.4 Фаза 4. Удаление дублированных данных

Эта фаза необратимо удаляет данные. Поэтому сначала надо убедиться, что все прошло хорошо:

- неделю пожить в фазе 3
- перед запуском задачи обязательно снять бекап с базы



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.

8.6.2 Что делать, если инкремент упал с ошибкой

8.6.2.1 Вариант 1 — исправление проблемы

1. Посмотреть в админке или в логах текст ошибки.
2. Найти «виноватых» и устранить ошибку.
3. Сделать перевыгрузку данных

И все хорошо, инкремент прогружается, `oldMaxId` меняется. Собственно, он потому и не меняется при ошибках — чтобы была возможность исправить инкремент и снова запустить задачу.

8.6.2.2 Вариант 2 — пока поправить не можем

Бывает такое, что устранить ошибку быстро нельзя. Тогда нужно [деактивировать исходные записи](#) → инкремент прогрузится успешно, а записи всегда можно будет найти.

1. Отредактировать задачу [disableStagingRecords](#), указав ей список `stagingHids`, которые надо деактивировать. `stagingHid` можно найти в логах, там перечислены все проблемные случаи.
2. Запустить задачу.
3. Запустить задачу загрузки инкремента → она пройдет успешно, не пытаясь трогать «плохие записи», счетчик `oldMaxId` обновится.
4. После исправления проблемы [активировать записи](#) и запустить задачу `transform`.

Но резюме такое, что ошибки в системе не надо оставлять, с ними надо разбираться

8.7 FAQ для сопровождающих

1. Нужно добавить задачу / триггер прямо в war-файл хотфиксом, как?

WAR-ник → WEB-INF → classes

8.8 Фактор (FAQ)

- [Есть ли doBatchClean для телефонов?](#)

8.8.1 Есть ли doBatchClean для телефонов?

Q: Можно ли вызвать [CleanService.phone](#) через `doBatchClean`? Как?

A: Можно, как и любой другой меппинг. Пример:

Запрос

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cle="http://clean.soap.cleaner.cleandata.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.cleanner.cleandata.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>
      <data>
        <dataFields>id 1</dataFields>
        <dataFields>мобильный</dataFields>
        <dataFields>OTHER</dataFields>
        <dataFields>79262235443</dataFields>
        <dataFields>7</dataFields>
        <dataFields>926</dataFields>
        <dataFields>2235443</dataFields>
        <dataFields/>
        <dataFields/>
        <dataFields>GOOD</dataFields>
        <dataFields>MOBILE</dataFields>
        <dataFields>ПАО "МегаФон"</dataFields>
        <dataFields>Москва и Московская область</dataFields>
        <dataFields/>
        <dataFields>РОССИЯ</dataFields>
        <dataFields>Москва и Московская область</dataFields>
        <dataFields/>
        <dataFields/>
        <dataFields/>
        <dataFields>UTC+3</dataFields>
        <dataFields>MSK+0</dataFields>
      </data>
    </ns2:doBatchCleanResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

8.9 Интеграция (faq)

- [Как передать две записи с одной связью](#)

8.9.1 Как передать две записи с одной связью

Вопрос: как передать в онлайн две новые записи, 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 обе записи сразу через запрос [saveAndMerge](#), тогда ошибки не будет. Просто при выгрузке из своей системы, если видим, что есть связь, передаем сразу обе записи

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="http://cleandata.ru/cdi/soap/2_13">
  <soapenv:Header/>
  <soapenv:Body>
    <saveAndMerge>
      <party type="PHYSICAL" sourceSystem="SOZ" rawId="1">
        <field name="fullNameRawSource">Первый Олег
Михайлович</field>
        <relation type="4">
          <second type="PHYSICAL" sourceSystem="SOZ"
rawId="2"/>
        </relation>
      </party>
      <party type="PHYSICAL" sourceSystem="SOZ" rawId="2">
        <field name="fullNameRawSource">Второй Олег
Михайлович</field>
        <relation type="4">
          <first type="PHYSICAL" sourceSystem="SOZ" rawId="1"/>
        </relation>
      </party>
    </saveAndMerge>
  </soapenv:Body>
</soapenv:Envelope>
```

9 Обновление справочников

9.1 Загрузка и обновление ЕГРЮЛ

9.1.1 Первичная загрузка справочников ЕГРЮЛ и ЕГРИП в систему «Гражданский фактор»

Для хранения справочника ЕГРЮЛ и проверки базы записей 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

| | |
|--|--------------|
| Группа import | |
| Название EgrDictionaryImport | |
| Параметры задачи | |
| Название параметра | Значение |
| massiveChanging | true |
| sourceFileName | /home/egrul2 |
| sourceDataReadBlockSize | 1000 |
| sourceDataEncoding | UTF-8 |
| poolSize | 4 |
| essenceType | EGR |
| | |
| | |
| <input type="button" value="Сохранить"/> <input type="button" value="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

9.1.2 Обновление справочников ЕГРЮЛ

При загрузке инкрементального ЕГРЮЛ необходимо ежедневное обновление справочника

- 9.1.2.1 Загрузка справочников ЕГРЮЛ и ЕГРИП в систему «Гражданский фактор»
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`.

| Название параметра | Значение |
|-------------------------|--------------|
| massiveChanging | false |
| sourceFileName | /home/egrul2 |
| sourceDataReadBlockSize | 1000 |
| sourceDataEncoding | UTF-8 |
| poolSize | 4 |
| essenceType | EGR |
| | |
| | |

Сохранить Отмена

3. Запустить задачу `EgrDictionaryImport`.

9.2 Обновление справочников Подсказок

9.2.1 Справочник email

1. Скачайте справочник из [репозитория Cleandata](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида `email-rrrrmmdd.zip`
2. Распакуйте архив на сервере в `/opt/suggestions/dictionaries/email`
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.

9.2.2 Справочник ip-адресов

1. Скачайте справочник из [репозитория Cleandata](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида `ip-rrrrmmdd.zip`
2. Распакуйте архив на сервере в `/opt/suggestions/dictionaries/ip`
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.

9.2.3 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 Cleandata](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида `fias-rrrrmmdd.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.

9.2.4 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 Cleandata](#). Выберите 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.

9.2.5 Справочник адресов (20.2–20.8)

1. Скачайте справочник из [репозитория Cleandata](#). 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 ADDR0B*.DBF HOUSE*.DBF SOCRBASE.DBF -  
d /opt/suggestions/dictionaries/address
```

3. Если нужны иностранные адреса: загрузите справочник `geonames-ггггммдд.zip` из [репозитория Cleandata](#), 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 Cleandata](#), 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 Cleandata](#), 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 Cleandata](#), 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/resourc  
e' -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.

9.2.6 Справочник адресов (20.9+)

- [Основной справочник адресов \(Россия\)](#)
- [Справочник квартир \(flat\)](#)
- [Справочник земельных участков \(stead\)](#)
- [Справочник GeoNames \(иностранные адреса кроме Беларуси\)](#)
- [Справочник OSM \(Беларусь\)](#)

9.2.6.1 Основной справочник адресов (Россия)

1. Скачайте справочник из [репозитория Cleandata](#). 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 переходите с версии 20.8 или раньше] Удалите *.DBF файлы старого распакованного справочника:

```
rm -f /opt/suggestions/dictionaries/address/*.DBF
```

4. Если нужны ISO-коды для стран и регионов: загрузите справочник iso3166-ггггммдд.zip из [репозитория Cleandata](#), 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 Cleandata](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address/house
```

6. Если используете поиск по географическим координатам: загрузите справочник geo-ггггммдд.zip из [репозитория Cleandata](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/address/geo
```

7. 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"] }'
```

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.

9.2.6.2 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.

9.2.6.3 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.

9.2.6.4 Справочник GeoNames (иностранные адреса кроме Беларуси)

1. Загрузите справочник geonames-ггггммдд.zip из [репозитория Cleandata](#), и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/geonames
```

2. Запустите обновление справочника:

```
curl -X POST
'http://localhost:8080/suggestions/api/4_1/rs/manage/resourc
e' -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.

9.2.6.5 Справочник OSM (Беларусь)

1. Загрузите справочник osm-ггггммдд.zip из репозитория и распакуйте архив на сервере в каталог

```
/opt/suggestions/dictionaries/osm
```

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.

9.2.7 Справочник банков

1. Скачайте справочник из [репозитория Cleandata](#). Выбирайте каталог с наиболее свежей датой, в нём архив вида ed807-ггггммдд.zip
2. Распакуйте архив на сервере в /opt/suggestions/dictionaries/bank
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.

9.2.8 Обновление внешних справочников

Требует лицензию 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+ |

9.2.8.1 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. 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 $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)

9.2.9 Справочник имен

1. Скачайте справочник из репозитория. Выбирайте каталог с наиболее свежей датой, в нём архив вида `fio-ггггммдд.zip`
2. Распакуйте архив на сервере в `/opt/suggestions/dictionaries/fio`
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.

9.2.10 Справочник компаний

1. Скачайте [основной справочник](#) из репозитория. Выбирайте каталог с наиболее свежей датой, в нём файл вида `egr-sgt-ггггммдд-bson.dic`.

2. Положите основной справочник в каталог `/opt/suggestions/dictionaries/party`
3. Скачайте [вспомогательный справочник](#) из репозитория. Выбирайте каталог с наиболее свежей датой, в нём файл вида `party_ext-ггггммдд.zip`.
4. Распакуйте вспомогательный справочник в каталог `/opt/suggestions/dictionaries/party/party_ext`. Каталог `party_ext` должен находиться внутри каталога `party`.
5. 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"] }'
```

6. 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)
```

7. 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.

9.2.11 Справочник координат

Начиная с версии подсказок 20.4 обновляется вместе совместно со [справочниками адресов](#)

Для «Подсказок» до версии 20.4

1. Скачайте справочник из [репозитория](#). 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](#)

9.3 Обновление справочников Фактора

9.3.1 База данных перенесенных номеров

Горячее обновление — обновление без остановки «Фактора». Доступно для «Фактора» 8.7 и выше.

9.3.1.1 Как запустить обновление

1. [Скачать из хранилища](#) новый справочник вида *Port_All_New-ГГГММДДЧММ_NNNN.zip* и его контрольную сумму (один из файлов с его названием и расширением sha1/sha256) на сервер с «Фактором». Размер справочника около 24Мб.

При необходимости справочник и его контрольную сумму можно расположить в пользовательской директории. Для этого нужно задать переменную `FACTOR_PHONE_CHANGE_MOBILE_PROVIDER_PATH` и указать путь. `FACTOR_PHONE_CHANGE_MOBILE_PROVIDER_PATH` можно задать как переменную окружению или как параметр в *standalone.conf*:

```
JAVA_OPTS="$JAVA_OPTS -
DFACTOR_PHONE_CHANGE_MOBILE_PROVIDER_PATH=/data/dict/phone/"
```

Code Block 15 Пример указания пути в standalone.conf

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}/phone/resources/change_mobile_provider"`.

2. **После завершения копирования** справочника и его контрольной суммы в той же директории создать файл *<название справочника>.ready*.

Например, если справочник называется *Port_All_New-201709260000_1406.zip*, то необходимо создать файл *Port_All_New-201709260000_1406.zip.ready*.

После этого начнётся загрузка нового справочника.

«Фактор» проверяет новый справочник на корректность.

- Если справочник корректный, то «Фактор»:
 - i. распаковывает справочник в локальную директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
 - ii. 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.

9.3.1.2 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.

9.3.1.2.1 Признаки успешного обновления

1. Файл *<название справочника>.ready* «Фактор» переименует в файл со статусом SUCCESS, названием справочника, датой и временем обновления в названии. Дата и время выводятся в формате уууymmdd_hhmmss. Например,
Port_All_New-201709220000_1402.zip_20170926_175003.success
2. В логе Фактора *factor.log* появятся строки:

```
(ResourceReload-changeMobileProvider-1) [INFO]
(LuceneIndexBuilder) Indexing 9807581 entities of
changeMobileProvider took 42,33 s
(ResourceReload-changeMobileProvider-1) [INFO]
(LuceneIndexManager) changeMobileProvider resource from
'file:///Z:/factor_indexes/index_changeMobileProvider_47639e
61-d60d-4a4b-beea-617ecfd3aa1c/' (19.04.2019 00:00:00)
loaded
(ResourceReload-changeMobileProvider-1) [INFO]
(ResourceHolder) Obsolete changeMobileProvider resource from
'Z:\factor_indexes\index_changeMobileProvider_b0afaae2-dc1f-
4d09-8c1a-f612fa4dled9' removed in 16,65 ms
(ResourceReload-changeMobileProvider-1) [INFO]
(ResourceChangeListener) Process
'file:///Z:/Dictionaries/mobile/Port_All_New-
201904190000_1976.zip' resource took '43,00 s'
```

9.3.1.2.2 Если что-то пошло не так

В директории с новым справочником «Фактор» переименует файл *<название справочника>.ready* в файл со статусом FAILED, названием справочника, датой и временем обновления.

1. Например,
Port_All_New-201709250000_1405.zip_20170926_175533.failed
2. В логе *factor.log* Фактор сообщит об ошибке:

```
(ResourceReload-changeMobileProvider-1) [ERROR] Can't process
'file:///data/Data_doc/wildfly-
9.0.2.Final/user_lib/phone/resources/change_mobile_provider/
Port_All_New-201709250000_1405.zip'
java.io.IOException: Zip file </data/Data_doc/wildfly-
9.0.2.Final/user_lib/phone/resources/change_mobile_provider/
Port_All_New-201709250000_1405.zip> has not entries
```

9.3.1.3 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.

9.3.1.4 Проверка версии справочников

Проверить версию справочников можно по [инструкции](#).

9.3.2 Обновление ЕГРЮЛ и ЕГРИП

Средства обновления:

| Доступно из коробки | Необходима лицензия |
|---------------------|---------------------|
| через ready-файлы | через апдейтер |
| через API | |

9.3.2.1 Обновление через ready-файлы

9.3.2.1.1 Предусловие:

«Фактор» запущен.

9.3.2.1.2 Шаги для запуска обновления:

1. Скачайте из хранилища 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»:
 - a. копирует справочник в директорию, заданную в переменной `FACTOR_EGR_CURRENT_PATH`;
 - b. 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.

9.3.2.1.3 Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

9.3.2.1.4 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) egdul2 resource from
      '/data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/current/egdul2-
      20170723.v2.mv.db' (23.07.2017 04:34:06) loaded
INFO] (ResourceHolder) Obsolete egdul2 resource from
      '/data/Data_doc/wildfly-9.0.2.Final/user_lib/egrul2/current/egdul2-
      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/egdul2-20170723.v2.mv.db'
      resource took '1,286 min'
```

9.3.2.1.5 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.cleandata.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.cleandata.cleaner.egrul2.storage.H2EgrDirectory.detect (H2EgrDirectory.java:114)
```

9.3.2.2 Обновление через API

Доступно с версии «Фактора» 9.3+.

Описание в разделе [обновления через API](#).

9.3.2.3 Обновление через Апдейтер

Описание в разделе [обновления через Апдейтер](#).

9.3.3 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.

```
fias-19.10-20191014.zip
fias-19.10-20191014.zip.md5
```

Code Block 16 Пример ФИАС и его контрольной суммы

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**:

```
fias-19.4-20190328.zip.ready
```

Code Block 17 Пример имени файла .ready для файла с ФИАС «fias-19.4-20190328.zip»

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.

9.3.3.1 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.

9.3.3.1.1 Как понять, завершено ли обновление ФИАС

В логе «Фактора» `factor.log` должны появиться записи вида:

«Фактор» 19.4+

```
[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
```

Code Block 18 Шаблон сообщения в factor.log

«Фактор» до 19.3 включительно

```
[INFO] (DefaultAddressClassifierProvider) Загружен ФИАС [date]
```

Code Block 19 Шаблон сообщения в factor.log

9.3.3.1.2 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`.


```
[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
```

Code Block 20 Шаблон сообщения в factor.log

- 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

```
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.cleandata.cleaner.exception.FactorConfigurationException: FIAS file
corrupted/has wrong version
```

Code Block 21 Шаблон сообщения в factor.log

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.

9.3.4 Обновление доменов первого уровня

Горячее обновление — обновление без остановки «Фактора». Доступно для «Фактора» 19.8 и выше.

- [Как запустить обновление](#)
- [Что происходит после завершения обновления](#)
 - [Признаки успешного обновления](#)
- [Время обновления](#)
- [Проверка версии справочников](#)

9.3.4.1 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 *standalone.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»:
 - i. распаковывает справочник в локальную директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
 - ii. 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.

9.3.4.2 Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

9.3.4.2.1 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_47
f22729-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_48
b50545-22db-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'
```

9.3.4.3 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.

9.3.4.4 Проверка версии справочников

Проверить версию справочников можно по [инструкции](#).

9.3.5 Перечень недействительных паспортов

Если серия или номер в справочнике пустая, то при построении индексов такие строки игнорируем, т.к. считаем серию и номер обязательными атрибутами.

9.3.5.1 Для одного «Фактора»

1. [Скачать из хранилища](#) новый справочник `passports-expired-ГТТГММДД.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»:

9.3.5.1.1 С остановкой «Фактора»

Остановить и запустить «Фактор».

Из нескольких ПНП «Фактор» выберет наиболее новый по дате модификации.

9.3.5.1.2 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 уууymmdd_hhmmss.

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.

9.3.5.2 Для нескольких «Факторов» без остановки

Обратитесь к сотруднику 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 ууу-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.

9.3.5.3 Время подключения (обновления)

Для работы со справочником «Фактор» формирует lscene-индекс. Формирует 10-30 минут, в зависимости от мощности сервера. В это время «Фактор» работает со старым индексом.

9.3.5.4 Сколько занимают рабочие файлы

Файлы с индексом занимают около 1 Гб.

9.3.5.5 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.

9.3.5.6 Как понять, что подключение завершено

В логе «Фактора» *factor.log* должны появиться записи вида:

```
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'
```

Code Block 22 Шаблон сообщения в factor.log

9.3.5.7 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.

9.3.6 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>"
```

9.3.7 Справочник Россвязи

Средства обновления:

- через ready-файлы
- через API

9.3.7.1 Обновление через ready-файлы

9.3.7.1.1 Предусловие: «Фактор» запущен.

9.3.7.1.2 Шаги для запуска обновления:

1. [Скачайте из хранилища](#) 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»:
 - a. строит индексы в `FACTOR_INDEXES_PATH`;
 - b. 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.

9.3.7.1.3 Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

9.3.7.1.4 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'
```

9.3.7.1.5 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

9.3.7.2 Обновление через API

Доступно с версии «Фактора» 19.7+.

Описание в разделе [обновления через API](#).

9.3.8 Справочник геокоординат

- [Как запустить обновление](#)
- [Что происходит после завершения обновления](#)
 - [Признаки успешного обновления](#)
 - [Если что-то пошло не так](#)
- [Время обновления](#)
- [Проверка версии справочников](#)

Запустить горячее обновление справочника геокоординат без остановки «Фактора».

9.3.8.1 Как запустить обновление

1. Скачайте справочник и его контрольную сумму из репозитория . Выберите каталог с наиболее свежей датой, а в нём файлы вида:

- `index_fiasIdGeoCodes-ГТТГММДД.idx` — справочник;

- o `index_fiasIdGeoCodes-ГГГГММДД.idx.md5` — его контрольная сумма.

2. Поместите справочник и его контрольную сумму в директорию, заданную в переменной `FACTOR_OFFLINE_FIAS_ID_GEO`.
`FACTOR_OFFLINE_FIAS_ID_GEO` может быть задана как переменная окружения или как параметр в `standalone.conf.bat` для windows или `standalone.conf` для linux:

```
set "JAVA_OPTS=%JAVA_OPTS% -DFACTOR_OFFLINE_FIAS_ID_GEO  
=C:\offlinegeo"
```

Code Block 23 Пример указания пути в `standalone.conf.bat`

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/geocoder"`.

3. **После завершения копирования** нового справочника и его контрольной суммы в той же директории создайте файл *<название справочника>.ready*.
Например, если справочник называется `index_fiasIdGeoCodes-20161226.idx`, то необходимо создать файл `index_fiasIdGeoCodes-20161226.idx.ready`.
После этого начнется 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. проверяет на корректность распакованный справочник;
 - c. 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.

9.3.8.2 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.

9.3.8.2.1 Признаки успешного обновления

1. Файл *<название справочника>.ready* «Фактор» переименует в файл со статусом `SUCCESS`, названием справочника, датой и временем обновления в названии. Дата и время выводятся в формате `ууууммдд_ххммсс`.
Например,
`index_fiasIdGeoCodes-20161226.idx_20170619_173610.success`
2. В логе «Фактора» `factor.log` появятся строки:


```
08.11.2019/10:58:35 127.0.0.1#anonymous (ResourceReload-fiasIdGeoCodes-1)
[INFO] (ResourceChangeListener) Processing 'c:\wildfly-16.0.0.Final-18080-
factor\user_lib\address2\resources\geocoder\index_fiasIdGeoCodes-
20190712.idx' resource
08.11.2019/10:58:35 127.0.0.1#anonymous (ResourceReload-fiasIdGeoCodes-1)
[INFO] (LuceneIndexAccessor) Unpack fiasIdGeoCodes index from 'c:\wildfly-
16.0.0.Final-18080-
factor\user_lib\address2\resources\geocoder\index_fiasIdGeoCodes-
20190712.idx'
08.11.2019/11:02:20 127.0.0.1#anonymous (ResourceReload-fiasIdGeoCodes-1)
[INFO] (LuceneIndexAccessor) Unpack fiasIdGeoCodes index took 3,751 min
08.11.2019/11:02:20 127.0.0.1#anonymous (ResourceReload-fiasIdGeoCodes-1)
[INFO] (LuceneIndexManager) fiasIdGeoCodes resource from
'file:///c:/wildfly-16.0.0.Final-18080-
factor/standalone/data/index_fiasIdGeoCodes_f61a8d8e-ba8c-42f8-bb71-
b65c0082c621/' (12.07.2019 00:00:00) loaded
08.11.2019/11:02:20 127.0.0.1#anonymous (ResourceReload-fiasIdGeoCodes-1)
[INFO] (ResourceChangeListener) Process 'c:\wildfly-16.0.0.Final-18080-
factor\user_lib\address2\resources\geocoder\index_fiasIdGeoCodes-
20190712.idx' resource took '3,751 min'
```

9.3.8.2.2 Если что-то пошло не так

1. В директории с новым справочником «Фактор» переименует файл *<название справочника>.ready* в файл со статусом FAILED, названием справочника, датой и временем обновления.
Например,
index_fiasIdGeoCodes-20191013.idx_192.168.0.181_20191108_111537.failed
2. В логе *factor.log* «Фактор» сообщит об ошибке:

```
08.11.2019/11:15:38 127.0.0.1#anonymous (ResourceReloadLockWatcher-1)
[ERROR] (ResourceReloadLockWatcher) fiasIdGeoCodes by path
'file:///c:/wildfly-16.0.0.Final-18080-
factor/user_lib/address2/resources/geocoder/index_fiasIdGeoCodes-
20191013.idx' corrupted. Cause by: Stream is not in the BZip2
format
```

9.3.8.3 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.

9.3.8.4 Проверка версии справочников

Проверить версию справочников можно по [инструкции](#).

9.3.9 Справочник локальных доменов

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](#)

9.3.9.1 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`:

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_EMAIL_LOCAL_PATH=/data/dict/email/"
```

Code Block 24 Пример указания пути в `standalone.conf`

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.

9.3.9.2 Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

9.3.9.2.1 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 `ууууmmdd_hhmmss`.
Например,
`local-domains.txt_10.0.63.16_20191227_203927.success`

2. В лог 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.Final\user_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.Final\user_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_648fbca
9-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_866dd2c
b-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.Final\user_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](#).

9.3.10 Справочник одноразовых e-mail

Горячее обновление — обновление без остановки «Фактора». Доступно для «Фактора» 19.4 и выше.

9.3.10.1 Как запустить обновление

[Скачать из хранилища](#) 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*:

```
JAVA_OPTS="$JAVA_OPTS -DFACTOR_EMAIL_DISPOSABLE_PATH=/data/dict/email/"
```

Code Block 25 Пример указания пути в standalone.conf

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»:
 - i. распаковывает справочник в локальную директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
 - ii. переключается на работу с новым справочником.
 - Если справочник некорректный, то «Фактор» продолжит работать со старым справочником.

9.3.10.2 Что происходит после завершения обновления

Данные, которые начали обрабатываться до обновления, обрабатываются при помощи старой версии справочника. Новые данные обрабатываются с использованием нового справочника.

9.3.10.2.1 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'
```

9.3.10.3 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.

9.3.10.4 Проверка версии справочников

Проверить версию справочников можно по [инструкции](#).

9.3.11 Справочник площадей квартир

9.3.11.1 Как запустить обновление

1. Скачайте справочник и его контрольную сумму из хранилища. Выбирайте каталог с наиболее свежей датой, а в нём файлы вида:
 - o index_flatArea-ГГГГММДД.idx — справочник;
 - o 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»:
 - a. распаковывает справочник в директорию, заданную в переменной `FACTOR_INDEXES_PATH`;
 - b. проверяет на корректность распакованный справочник;
 - c. 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.

9.3.11.2 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.

9.3.11.2.1 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 логге Фактора *factor.log* появятся строки:

```
(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'
```

Code Block 26 Успешное обновление

9.3.11.2.2 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:

```
(ResourceReload-flatArea-2) [ERROR] Can't process  
'file://overmind/public/elizavetam/flat_area/index_flatArea-20160909.idx'  
ru.cleandata.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)
```

Code Block 27 Попытка обновления на более старый 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
```

Code Block 28 Попытка обновления на битый справочник

9.3.12 Справочник стоимости квадратного метра

- [Описание](#)
- [Горячее обновление справочника стоимостей](#)

9.3.12.1 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.

9.3.12.2 Горячее обновление справочника стоимостей

1. [Скачать из хранилища](#) 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)