



ООО «БФТ»

129085, г. Москва, ул. Годовикова, д. 9, стр. 17  
+7 (495) 784-70-00

ined@bftcom.com  
bftcom.com

**Утвержден**

БАРМ.00003-45 32 01-1-ЛУ

# **Система автоматизации процесса управления государственными и муниципальными закупками – Автоматизированный Центр Контроля – Государственный и муниципальный заказ «АЦК-Госзаказ»/«АЦК-Муниципальный заказ»**

## **Блок администрирования Подсистема администрирования системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» Администрирование системы**

Руководство администратора

БАРМ.00003-45 32 01-1

Листов 159

© 2022 ООО «БФТ»

# АННОТАЦИЯ

В документе приводится описание работы в подсистеме «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» содержащее общие принципы работы, основные понятия и термины, содержание операций при администрировании системы.

Содержание документа соответствует ГОСТ 19.503-79 «Единая система программной документации. РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА. Требования к содержанию и оформлению».

«Система автоматизации процесса управления государственными закупками - Автоматизированный Центр Контроля – Государственный заказ» («АЦК-Госзаказ») зарегистрирована в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам, Свидетельство № 2008610925 от 21 февраля 2008 г. «Система автоматизации процесса управления муниципальными закупками - Автоматизированный Центр Контроля - Муниципальный заказ» («АЦК-Муниципальный заказ») зарегистрирована в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам, Свидетельство № 2009615485 от 02 октября 2009 г.

ООО «БФТ» оставляет за собой право вносить изменения в программное обеспечение без внесения изменений в эксплуатационную документацию.

Оперативное внесение изменений в программное обеспечение отражается в сопроводительной документации к выпускаемой версии.

Документ соответствует версии системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» – 1.45.0. Последние изменения внесены 28.06.2022 г.

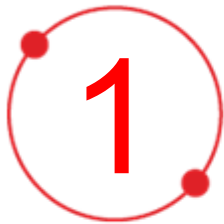
# СОДЕРЖАНИЕ

1	Сведения о системе.....	7
1.1	Назначение программы.....	8
1.2	Функциональное назначение.....	13
2	Структура программы.....	16
2.1	Архитектура системы.....	17
2.2	Компоненты системы.....	17
2.2.1	Серверные компоненты.....	18
2.2.1.1	СУБД.....	18
2.2.1.2	Сервер приложений.....	18
2.2.1.3	Web-сервер.....	20
2.2.2	Клиентские компоненты.....	22
2.2.2.1	Online-клиент.....	22
2.2.2.2	Offline-клиент.....	22
2.2.2.3	Web-клиент.....	22
2.3	Взаимодействие компонентов.....	23
2.3.1	Взаимодействие по протоколу HTTP.....	23
2.3.2	Взаимодействие по протоколу CORBA.....	24
2.4	Стандартная сборка системы.....	24
2.5	Проверка целостности сборки.....	26
3	Системные требования.....	30
3.1	Требования к серверам.....	31
3.1.1	Общие требования.....	31
3.1.2	Сервер приложений АЦК.....	36
3.1.3	Сервер для интеграции с ЕИС и ЭТП.....	37
3.1.4	Сервер базы данных АЦК.....	37
3.1.5	Совмещенный сервер приложений и сервер базы данных.....	38
3.1.6	Сервер лицензий.....	39
3.1.7	Сервер ЭП.....	39
3.1.8	Сервер обновлений.....	40
3.1.9	Сервер web-сайта.....	41

3.1.10	Сервер web-клиента.....	41
3.1.11	Прoxy-сервер.....	42
3.2	Требования к рабочим местам.....	43
3.2.1	Общие требования.....	43
3.2.2	АРМ Уполномоченного органа (Corba и Http).....	44
3.2.3	АРМ Уполномоченного органа (WEB-клиент).....	44
3.2.4	АРМ Заказчика (Corba и Http).....	45
3.2.5	АРМ Заказчика (WEB-клиент).....	46
3.2.6	Требования к программному окружению при использовании криптографических функций.....	47
3.2.7	Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ.....	48
3.3	Требования к линиям связи.....	48
3.3.1	Общие требования к линиям связи.....	48
3.3.2	Линии связи между серверами системы.....	48
3.3.3	Линии связи между сервером приложения и АРМ Уполномоченного органа.....	49
3.3.4	Линии связи между сервером приложения и АРМ Заказчика.....	49
3.4	Требования к безопасности.....	49
3.4.1	Демилитаризованная зона (DMZ).....	49
4	Установка и настройка программы.....	52
4.1	Настройка операционной системы.....	53
4.2	Установка и настройка СУБД.....	54
4.2.1	Настройка СУБД «Firebird».....	54
4.2.2	Настройка СУБД «Oracle».....	54
4.2.3	Настройка PostgreSQL.....	55
4.2.4	Резервное копирование БД.....	55
4.2.4.1	Резервное копирование СУБД «Firebird».....	55
4.2.4.2	Резервирование СУБД «Oracle».....	67
4.2.4.3	Резервное копирование СУБД PostgreSQL.....	75
4.2.4.4	Проверка качества созданной резервной копии БД.....	78
4.3	Установка «Sentinel System Driver».....	84
4.4	Установка «CryptoPro-CSP».....	84
4.5	Установка системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ».....	85

4.5.1	Установка в операционной системе «Linux».....	85
4.5.1.1	Установка системы в ручном режиме.....	85
4.5.1.2	Автоматическая установка системы.....	89
4.5.2	Установка в операционной системе «Windows».....	89
4.5.2.1	Настройка ролей серверов приложений.....	89
4.5.2.2	Настройка файлового хранилища.....	91
4.5.3	Настройка запуска комплекса серверов приложений.....	94
4.5.4	Настройка базы данных.....	94
4.5.4.1	Создание базы данных.....	95
4.5.4.2	Загрузка и выгрузка произвольных данных.....	98
4.5.4.3	Создание резервной копии БД.....	99
4.5.4.4	Восстановление БД из резервных копий.....	103
4.5.5	Установка и настройка сервера приложений.....	105
4.5.5.1	Файл настроек сервера приложений.....	107
4.5.6	Установка и настройка online-клиентов.....	119
4.5.7	Установка OLAP сервера.....	123
4.5.8	Настройка сжатия данных, передаваемых/отсылаемых между сервером и клиентом .....	123
4.6	Настройка системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ».....	124
4.6.1	Настройка ролей организаций.....	124
4.6.2	Настройка ролей пользователей.....	124
4.6.3	Настройка пользовательского меню для Win-клиента.....	124
4.6.4	Заполнение справочников системы.....	125
4.6.5	Выполнение XML-заданий.....	125
4.6.6	Настройка планировщика заданий.....	125
4.6.7	Настройка подсистемы ЭП.....	126
4.6.8	Подключение шаблонов печатных форм.....	126
4.7	Настройка обновления системы.....	128
4.7.1	Обновление структуры БД.....	128
4.7.2	Обновление сервера приложений.....	130
4.7.3	Обновление данных в БД.....	130
4.7.4	Обновление online-клиентов.....	131
4.7.5	Обновление web-сайта.....	131
4.8	Настройка локальной сети.....	132
4.9	Настройка журнализации событий и отладочных режимов.....	132

4.10	Настройка оптимизации работы системы.....	133
4.11	Настройка механизма журналирования изменений системных параметров .....	134
5	Сопровождение системы.....	136
5.1	Задачи сопровождения системы.....	137
5.2	Резервное копирование и архивирование БД и журнального файла сервера приложений.....	137
5.3	Идентификация ошибок и их устранение.....	138
5.4	Поддержка пользователей системы.....	139
5.5	Другие функции администратора системы.....	139
6	Порядок технического обслуживания.....	140
6.1	Регламентно-профилактические мероприятия.....	141
6.1.1	Регламентно-профилактические мероприятия по ППО.....	141
6.1.1.1	Установка эталонной сборки.....	143
6.1.1.2	Параметры распределения JAVA памяти.....	144
6.1.1.3	Параметры настроек в конфигурационных файлах АЦК.....	145
6.1.1.4	Использование утилиты «jvisualvm».....	148
6.1.2	Регламентно-профилактические мероприятия по СУБД.....	149
6.1.2.1	Рекомендации по параметрам СУБД.....	150
6.1.3	Регламентно-профилактические мероприятия по СПО.....	152
6.1.4	Мониторинг виртуальной инфраструктуры.....	153
6.1.5	Наличие тестового СП.....	153
6.1.6	Мониторинг структуры системы хранения данных.....	153
6.1.6.1	Оценка производительности СХД.....	154
6.1.6.2	Повышенный Service Time СХД.....	154
6.1.6.3	Утилизация CPU СХД.....	155
6.1.7	Проверка работоспособности.....	155
6.1.8	Проверочное восстановление резервной копии.....	157
7	Аванпост FAM.....	158



# Сведения о системе



## 1.1 Назначение программы

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» предназначена для организации контроля и анализа всего процесса подготовки, и проведения закупки: от планирования закупок и определения поставщика до контроля результатов полученных по итогам исполнения контрактов.

Работы системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» осуществляется на основании законодательства РФ:



Рисунок 1 – Законодательство РФ, служащее основанием для работы системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ»

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» позволяет всем участникам процедур закупок оперативно, в режиме реального времени, просматривать и анализировать информацию о планировании, формировании, определении поставщика и исполнении закупки, а также принимать эффективные управленческие решения, обеспечивая гласность и прозрачность осуществляемых закупок.

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» предназначена для использования следующими участниками процедур закупок:



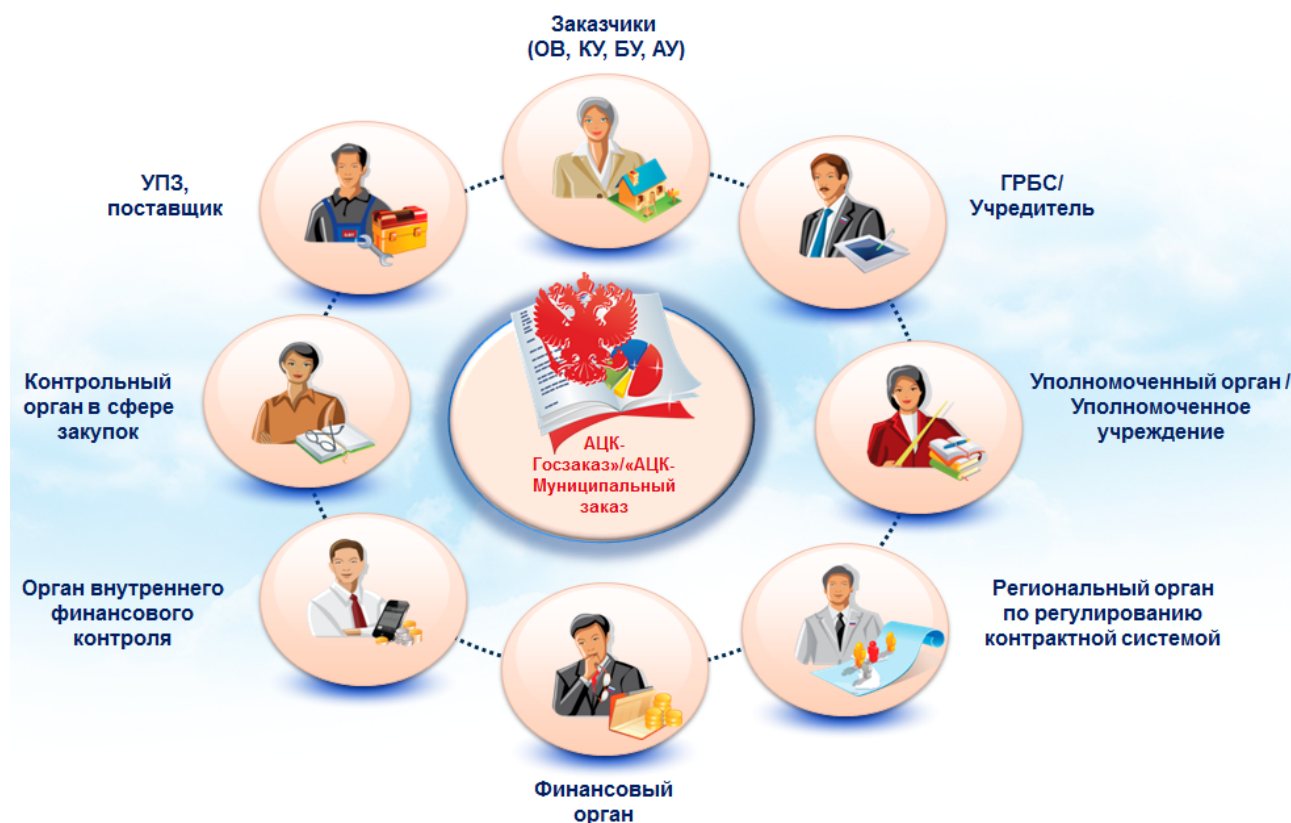


Рисунок 2 – Участники процедур закупок

Этапы закупок:

## 1. Планирование заказа

Планирование заказа – итерационный, многоступенчатый процесс. В системе автоматизация данного процесса представлена в подсистеме «Планирования процедур государственного (муниципального) заказа».

Этап планирования закупок делится на подэтапы:

- **Формирование планов-графиков**

ЭД «План-график» формируется заказчиками ежегодно на очередной финансовый год в сроки, установленные высшими исполнительными органами государственной власти субъектов Российской Федерации (местными администрациями).

ЭД «План-график» формируют после внесения проекта закона (решения) о бюджете на рассмотрение законодательного (представительного) органа субъекта Российской Федерации (представительного органа муниципального

образования).

Сформированный ЭД «План-график» при необходимости уточняют, после уточнения и доведения до государственного заказчика или муниципального заказчика объема прав в денежном выражении на принятие и (или) исполнение обязательств, в соответствии с бюджетным законодательством Российской Федерации, сформированный ЭД «План-график» утверждают.

В ЭД «План-график» включается информация в отношении каждой закупки:

- информация об источниках финансирования закупки (классификация);
- информация о кодификаторах номенклатуры закупки (ОКПД, ОКДП, Справочник товаров, работ и услуг);
- информация о виде экономической деятельности (ОКВЭД);
- наименование и описание объекта закупки с указанием характеристик такого объекта, количество поставляемого товара, объем выполняемой работы, оказываемой услуги, планируемые сроки, периодичность поставки товара, выполнения работы или оказания услуги, начальная (максимальная) цена контракта, цена контракта, заключаемого с единственным поставщиком (подрядчиком, исполнителем), обоснование закупки (в виде файлов), размер аванса (если предусмотрена выплата аванса), этапы оплаты (если исполнение контракта и его оплата предусмотрены поэтапно);
- дополнительные требования к участникам закупки (при наличии таких требований) и обоснование таких требований;
- способ определения поставщика (подрядчика, исполнителя) и обоснование выбора этого способа (в виде файлов);
- дата начала закупки;
- информация о размере предоставляемых обеспечения соответствующей заявки участника закупки и обеспечения исполнения контракта;
- информация о банковском сопровождении контракта.

## 2. Формирование заявок на закупку

На промежуточном этапе формирования ЭД «Заявка на закупку» выполняются действия:

- подача заявок на закупку заказчиками;
- проверка и контроль выполнения графика закупок;
- заверение подлинности документов электронной подписью;
- согласование с вышестоящими организациями (РБС, УО) способов закупки, сроков проведения, объемов закупки;
- консолидация заявок подведомственных организаций.

Электронные документы, которые используются на этапе формирования заявок на закупку, «Заявка на закупку» и «Консолидированная заявка на закупку».

### 3. Размещение заказа и проведение процедур закупки

На этапе размещения заказа выполняются действия:

- проведение процедур закупки (предварительного отбора, закупки у единственного источника, конкурсов, запросов котировок, аукционов), подготовка конкурсной документации, согласование с вышестоящими организациями;
- информирование о процедурах закупки;
- прием и обработка заявок поставщиков;
- сравнение предложений претендентов;
- определение поставщиков из претендентов;
- подготовка проекта контрактов и договоров.

Электронные документы этапа размещения заказа и проведения процедур закупок:

- решения о размещении заказа:
  - «Решение о проведении конкурса»;
  - «Решение о проведении запроса котировок»;
  - «Решение о закупке у единственного источника»;
  - «Решение о проведении предварительного отбора»;
  - «Решение о проведении торгов на ЭТП»;

- «Решение о проведении запроса предложений».
- «Заявки и предложения поставщиков»:
  - «Конкурсная заявка поставщика»;
  - «Котировочная заявка поставщика»;
  - «Заявка на участие в предварительном отборе»;
  - «Заявка на участие в запросе предложений»;
  - «Заявка на участие в открытом аукционе в электронной форме».

#### 4. Исполнение заказа

На этапе исполнения заказа выполняются действия:

- контроль и согласование заключенных контрактов и договоров;
- принятие бюджетных обязательств;
- финансовое исполнение договоров;
- контроль за исполнением договоров поставщиками;
- контроль бездоговорных закупок.

Электронные документы этапа исполнения заказа:

- «Контракт»;
- «Сведения об исполнении контракта»;
- «Договор»;
- «Счет»;
- «Факт поставки»;
- «Иски»;
- «Претензии».

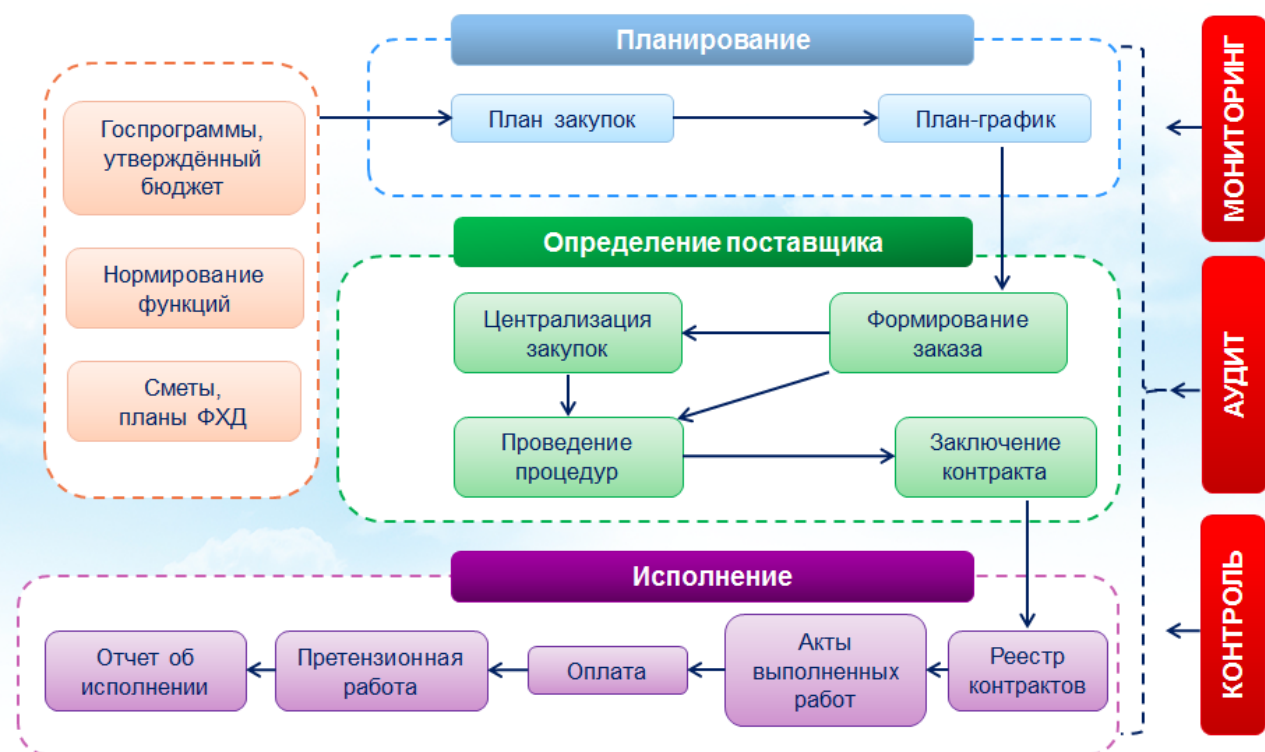


Рисунок 3 – Основные этапы закупочного цикла

## 1.2 Функциональное назначение

Принципы, которые лежат в основе электронного документооборота:

- единая система прав доступа к документам;
- однократный ввод данных;
- необратимость ввода данных;
- авторизация ввода данных.

После внесения документа в систему данные в документе, как правило, не подлежат изменению, изменяется только статус документа.

Исправление ошибок и внесение изменений в документы производится введением корректирующих документов, либо созданием нового документа.

Документооборот в системе основан на следующем:

- По отношению к зарегистрированному в системе документу может быть

предпринят ряд действий, которые составляют этапы документооборота. Результатом выполнения одного такого этапа может быть модификация данных самого документа, дополнение его информацией, регистрация других документов и объектов учета системы.

- Однажды выполненный этап документооборота можно отменить, то есть удалить из базы данных информацию о его выполнении, в частности, снять с самого документа отметку о выполнении этапа.
- Возможность выполнения или отмены каждого этапа можно поставить в зависимость от состояния других этапов (выполнен или нет) для этого же документа. Право выполнять этап может предоставляться ограниченному кругу пользователей.
- Для каждого документа можно оперативно просмотреть список связанных (родительских и дочерних) документов.

При передаче электронных документов применяется эквивалент традиционной подписи – электронная подпись (далее – ЭП). ЭП позволяет удостовериться в подлинности сообщения. С ее помощью можно доказать не только то, что сообщение было послано тем или иным пользователем, но и то, что информация не была искажена во время передачи.

ЭП не обеспечивает защиты от несанкционированного просмотра, а позволяет:

- контролировать неизменность документов;
- подтверждает авторство (ответственность).

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» представляет собой многозвенное приложение электронного документооборота, позволяющее осуществить:

- единое хранение данных;
- единую систему прав доступа к информации;
- однократный ввод данных в систему;
- хранение истории обработки документов;
- хранение истории действий пользователей;
- одновременную работу с данными неограниченного количества пользователей (ограничение связано только с аппаратными возможностями серверов);

- использование ЭП для контроля неизменности передаваемого в систему электронного документа и для подтверждения авторства документа.

Документооборот в системе реализован следующим образом:

- создается документ с внесением всех необходимых данных;
- документ направляется в обработку, в процессе которой у документа меняется статус посредством выполнения над ним определенного действия или метода;
- действия и методы документов на определенных статусах доступны в соответствии с настройкой ролей пользователей;
- в некоторых случаях при выполнении действия или метода над документом помимо смены статуса автоматически выполняются другие действия, например, автоматическое создание нового документа, связанного с обрабатываемым.

Действия и методы, доступные для документа, отражаются в дереве сценариев. Существует Редактор сценариев и Пользовательский редактор сценариев. В Редактор сценариев отражается порядок обработки электронных документов, определенные разработчиком. Вносить изменения в Редактор сценариев нельзя. Администратор системы может вносить дополнительные команды и условия в действия, внося изменения в Пользовательский редактор сценариев. Также системный администратор создает учетные записи пользователей и настраивает для них функциональные и организационные роли.

Для заполнения полей электронных документов (далее – ЭД) используются справочники.

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» может интегрироваться с системой казначейского исполнения бюджета (далее – СКИБ) «АЦК-Финансы».



# 2

## Структура программы



## 2.1 Архитектура системы

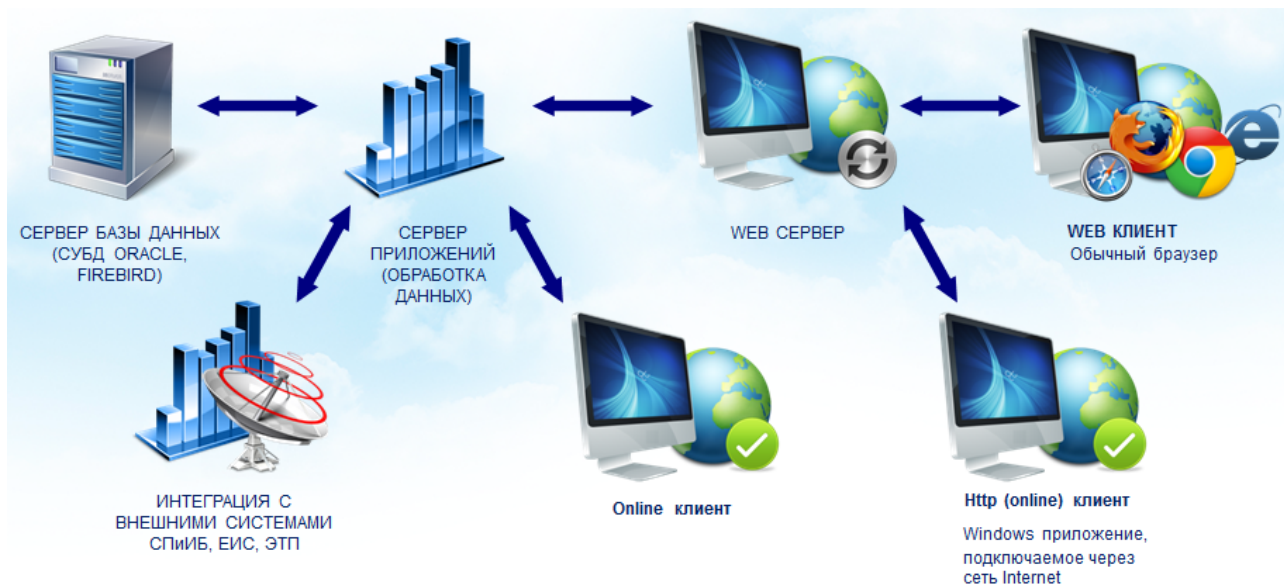


Рисунок 4 – Схема архитектуры системы

Данная архитектура системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» обеспечивает следующие преимущества:

- высокую степень информационной безопасности;
- целостность базы данных;
- масштабируемость;
- высокую производительность системы.

## 2.2 Компоненты системы

Система состоит из следующих взаимосвязанных компонентов:

- системы управления базами данных – СУБД;
- сервера приложений (серверов приложений может быть несколько);
- online-клиента;
- web-сервера;
- web-клиента;

- offline-клиента.

Использование web-сервера, web-клиента и offline-клиента является опциональным.

## 2.2.1 Серверные компоненты

### 2.2.1.1 СУБД

Система поддерживает работу с СУБД **Firebird-2.5.2**, **Oracle 11g** (рекомендуемая 11.2.0.4), **PostgreSQL** (рекомендуемая 9.5.3).

Данные системы хранятся в БД, непосредственный доступ к которой имеет только сервер (серверы) приложений. СУБД может работать как под управлением операционной системы **Windows 2008** (или выше), так и под управлением операционной системы **Linux**.

Сервер управления базами данных желательно размещать на выделенном сервере, соединенным с сервером (серверами) приложений высокоскоростным каналом. Рекомендуется организовать дисковый RAID-массив для обеспечения отказоустойчивости и более высокого быстродействия БД.

### 2.2.1.2 Сервер приложений

Сервер приложений является основным компонентом системы. Назначение сервера приложений заключается в обработке запросов пользователей и обработке данных. Функции сервера приложений:

- обработка прикладной логики;
- контроль целостности прикладных данных;
- передача данных в БД для хранения;
- передача информации;
- проверка ЭП документов;
- прием и передача данных во внешние системы.

Сервер приложений предназначен для выполнения в **JRE** и является платформо-независимым приложением. Сервер приложений может работать под управлением ОС **Windows** или **Linux**. С другими ОС возможна работа при наличии **JAVA**.

*Примечание. ОС на базе **Linux** используется в качестве платформы для обеспечения надежной и стабильной работы, а также для повышения быстродействия. ОС на базе **Windows** используется в качестве платформы при наличии достаточного запаса мощности аппаратной платформы и при отсутствии возможности сопровождения **Linux-систем**.*

Для нормальной работы сервера приложений необходимо установить **JAVA 8 Update 40** и определить переменную окружения **JAVA\_HOME**, указывающую на корневой каталог установленной **JDK**. Также необходимо переопределить переменную окружения **PATH** так, чтобы при поиске **Java-машины** (для ОС **Win32** – это **java.exe**) вызывался файл именно из каталога **/bin**, установленной **JDK**.

Для разделения вычислительной нагрузки и повышения производительности устанавливается несколько серверов приложений (кластер серверов приложений), работающих с одной БД.

Совокупность БД и сервера приложений (кластера серверов приложений), работающего с БД, называется сайтом. Каждый сайт должен иметь уникальный **ID**. Сайт, на котором выполняются все движения по статусам документов, называется главным (**Master**). Остальные сайты являются подчиненными (**Slave**) сайтами. На подчиненных сайтах все движения по статусам документов сопровождаются отправкой на **Master-сайт** запроса о переводе документа на другой статус и после получения подтверждения с главного сайта документ переходит на другой статус и на подчиненном сайте. Тем самым обеспечивается централизованность обработки всех документов, вне зависимости от их происхождения. Подчиненные сайты используются для реализации **offline-клиентов**. Актуальность данных в подчиненных сайтах поддерживается при помощи репликации и обмена документами. Репликация осуществляется со стороны главного сайта в сторону подчиненных сайтов. Обмен документами осуществляется в обе стороны (с вышеописанным условием перехода по статусам).

Между собой сайты взаимодействуют посредством сообщений. Сообщения представляют собой **xml-последовательности** определенной структуры.

В зависимости от выполняемых функций можно выделить следующие роли сервера приложений:

Таблица 1 – Роли сервера приложений

Название сервера приложений	Роль сервера приложений
Сервер обработки запросов online-клиентов	Взаимодействует с <b>online-клиентами</b> .
Сервер проверки подлинности ЭП	Осуществляет проверку подлинности ЭП. Необходима установка <b>CryptoPro CSP</b> (на данный момент <b>CryptoPro CSP</b> доступно только под <b>Win32</b> ).
Сервер лицензий	Осуществляет централизованное использование аппаратного ключа защиты <b>Sentinel</b> и файла лицензии всем кластером серверов приложений (на данный момент разработчиком <b>Sentinel</b> реализовано использование системы защиты только под <b>Win32</b> ).
Сервер обработки web-сервера	Взаимодействует с <b>web-сервером</b> , обрабатывая запросы <b>web-клиентов</b> .
Сервер взаимодействия со СКИБ	Взаимодействует с системой казначейского исполнения бюджета «АЦК-Финансы». (рекомендовано реализовывать под <b>Win32</b> ).
Сервер обработки offline-клиентов	Взаимодействует с <b>offline-клиентами</b> (подчиненными сайтами).

***Примечание.** Для выполнения ролей сервера лицензий и сервера проверки подлинности ЭП особых вычислительных ресурсов не требуется и роли могут быть объединены в одном сервере приложений вместе с сервером взаимодействия со СКИБ. Объединение других ролей, либо их еще большее разделение рекомендуется производить проанализировав текущую нагрузку. Минимально возможный вариант – это реализация всех ролей в рамках одного сервера приложений (следуя ограничениям – сервер должен работать под управлением **Win32**).*

Для сервера приложений в роли сервера лицензий необходимо наличие аппаратного ключа **Sentinel** и его драйвера.

**Внимание!** Ключ **Sentinel** доступен только в версии для **MS Windows**.

### 2.2.1.3 Web-сервер

Web-сервер представляет собой http-сервер **Tomcat 8.5** (работает в **JDK**) и обслуживаемый им сайт (реализован на **Java Server Pages**). Сервер приложений является платформо-независимым.

Web-сервер может выполнять две функции:

- Web-сервер является промежуточным звеном между сервером приложений и **win-клиентами**, позволяя последним взаимодействовать с сервером приложений. При таком режиме работы нагрузка на **web-сервер** не значительная. **Web-сервер** представляет собой клиента по отношению к серверу приложений. Однако он обслуживает запросы **win-клиентов** (браузеров удаленных пользователей).



Рисунок 5 – Работа web-сервера с win-клиентом

- Web-сервер используется для формирования web-интерфейса.

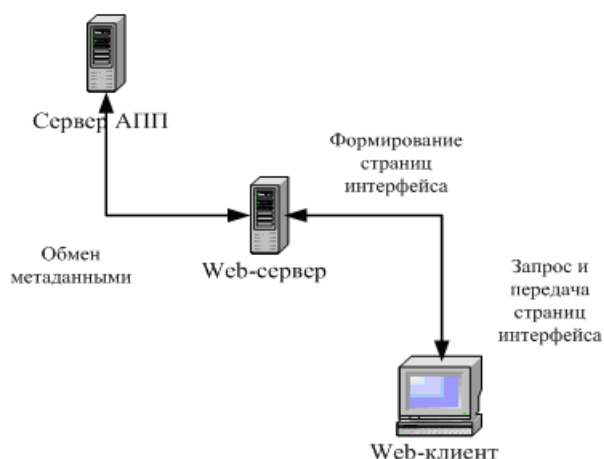


Рисунок 6 – Работа web-сервера с web-клиентом

## 2.2.2 Клиентские компоненты

Клиентское приложение предназначено для подготовки и отправки заданий на сервер приложений, а также представления данных, полученных от сервера приложений, в доступной для восприятия пользователем форме.

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» поддерживает два вида клиентских компонентов: **win-клиент** и **web-клиент**.

### 2.2.2.1 Online-клиент

Online-клиент представляет собой Windows-приложение.

Главное назначение online-клиента – это получение данных от пользователя, передача их на обработку серверу приложений от имени данного пользователя, а также получение и вывод пользователю результата этой обработки, т.е. вывод ответа сервера приложений. Online-клиент связан только с сервером приложений и не имеет прямого доступа к БД.

### 2.2.2.2 Offline-клиент

Offline-клиент представляет собой следующую связку, настроенную на автономную работу: СУБД + Сервер приложений + Online-клиент.

Offline-клиент позволяет не иметь постоянное соединение (в отличие от online-клиента и web-клиента) с главным сайтом. Пользователь может создавать документы автономно, связываясь с главным сайтом только в момент отправки документа в обработку и для получения результата обработки с главного сайта.

Набор необходимых справочников реплицируется однонаправлено с главного сайта.

### 2.2.2.3 Web-клиент

Работа с web-клиентом системы осуществляется непосредственно в браузере. Браузер должен поддерживать обработку **JavaScript** и элементов **ActiveX**, а также должна быть включена поддержка **cookies**.

Рекомендуемый браузер **Mozilla Firefox 44.0**. Поддерживаемые браузеры: **Google Chrome 48.0, Microsoft Internet Explorer** (версии 10.0/11.0).

## 2.3 Взаимодействие компонентов

### 2.3.1 Взаимодействие по протоколу HTTP

По стандартному протоколу **HTTP** осуществляется взаимодействие web-сервера и web-клиента – обмен гипертекстовыми документами (web-страницами).

Коннектор **HTTP** может использоваться для взаимодействия win-клиента и сервера приложений, и взаимодействия между серверами приложений.

Применяется при необходимости обеспечения взаимодействия через шлюзовые системы, туннели, фильтры, низкоскоростные каналы связи. Позволяет работать через проху-системы.

По сравнению с **CORBA** обладает более медленным откликом и повышенной устойчивостью на низко скоростных линиях связи.

Для работы по протоколу **HTTP** необходимо выполнить следующие действия:

- в файле запуска **StartServer.bat** указать порт, по которому будет осуществляться связь с клиентом: **set SERVER\_HTTP\_PORT=8083**;
- в файле **Azk2CInt.ini** выполнить следующие настройки:
  - указать режим подключения к серверу:

**[SERVER]**

**ServerMode=HTTP**

где:

- **ServerMode** – режим подключения клиента к серверу приложений. Доступны следующие значения:
  - **CORBA** – значение для подключения по протоколу **CORBA**;
  - **HTTP** – значение для подключения по протоколу **HTTP**.
- указать путь к серверу.
- если используется стандартный режим:

*[CORBA]*

*Server=localhost:2001*

- если режим **HTTP**:

*URL=http://localhost:8083/exec*

где:

- 8083 – параметр **SERVER\_HTTP\_PORT**, указанный в файле **StartServer.bat**.

### 2.3.2 Взаимодействие по протоколу CORBA

Коннектор **CORBA** может использоваться для взаимодействия win-клиента и сервера приложений, и взаимодействия между серверами приложений.

Применяется при необходимости обеспечения взаимодействия в пределах локальной сети, при наличии скоростных линий связи. При наличии в канале связи *проху*-систем применение затруднительно.

## 2.4 Стандартная сборка системы

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» поставляется в виде сборки, предназначенной как для установки системы, так и для ее обновления на более новую версию.

**Внимание!** Обновление на более раннюю версию в системе не предусмотрено.

Сборка представляет собой набор каталогов и файлов. В большинстве скриптов для указания пути доступа к файлам сборки используется корневой каталог сборки, т.е. каталог в котором размещена сборка.

**Внимание!** Ограничений на именование корневого каталога нет. Однако рекомендуется учитывать особенности функционирования **JVM** и избегать имен каталогов с пробелами. Не допускается переименовывать каталоги в самой сборке.

Корневой каталог сборки содержит следующие подкаталоги и файлы:



Таблица 2 – Подкаталоги и файлы корневого каталога сборки

Название каталога или файла	Описание
Bin	Содержит скрипты и утилиту для создания БД, оптимизации записей в ней для работы с xml-данными и др. Скрипты содержат комментарии.
Client	Содержит файлы online-клиента. В каталоге отсутствуют необходимые для работы online-клиента дополнительные программные библиотеки.
ClientX	Содержит файлы для инсталляции и обновления online-клиента. Файлы инсталляции и обновления содержат все необходимые компоненты, в том числе и дополнительные программные библиотеки, отсутствующие в каталоге <b>Client</b> .
Doc	Содержит краткие руководства для администратора системы, а также примеры заполнения некоторых файлов.
Embedded	Содержит файлы, которые используются для работы сервера приложений в embedded-режиме.
module	Содержит файлы модулей системы.
Server	Содержит скомпилированные файлы сервера приложений.
SQL	Содержит скрипты для первоначального формирования структуры БД, а также для ее обновления (подкаталог <b>/log</b> ). Подкаталог <b>/ServiceScript</b> содержит сервисные скрипты. Их применение возможно только после предварительной проверки на тестовом стенде.
UpdateOffline	Содержит скрипты для реализации автоматического обновления offline-клиентов с Master-сайта.
web	Содержит web-клиент системы (в виде специального архива).
website	Содержит web-сайт системы (в виде специального архива), а также краткое руководство для администратора.
XML	Содержит xml-скрипты для первоначального формирования данных в БД, а также для их обновления.
azk.chk	Файл в котором перечислены файлы поставки дистрибутива.
Startsv	Пример скрипта для запуска в ОС <b>Linux</b> .
StartServer.bat	Пример скрипта для запуска в <b>Win32</b> .
Server.properties	Файл параметров сервера приложений. В файле указываются основные параметры, необходимые системе при старте.
changelog.1.XX.txt	Файлы с описанием сделанных в каждой версии изменений, где 1.XX – номер версии.
Order_ib.log	Журнальный файл (log-файл). По умолчанию название log-файла – <b>order_ib.log</b> .
storages.xml	Файл параметров для файловых УХЗ, используемых системой, по умолчанию <b>storages.xml</b> . Расположение данного файла определяется в файле <b>Server.properties</b> .
integrity_check.cmd	Командный файл для проверки целостности установки системы.
IntegrityCert.cer	Сертификат, необходимый для удостоверения подлинности компонентов дистрибутива.

Сервер приложений состоит из каталога **Server** и файлов **startsrv** (для **Linux**), **StartServer.bat** (для **Win32**) и **Server.properties**.

**Online-клиент** рекомендуется устанавливать инсталлятором из папки **ClientX**.

Для функционирования web-сайта необходим http-сервер **Tomcat 6.0** (он расположен в каталоге **web** в виде zip-архива).

*Примечание.* При необходимости администратору могут быть высланы недостающие в папке **IClient** дополнительные программные библиотеки, если он решит запускать online-клиент без предварительной установки.

## 2.5 Проверка целостности сборки

Для проверки целостности сборки запускается файл **integrity\_check.cmd**. При запуске файла осуществляется проверка:

- целостности сборки по списку файлов;
- подлинности каждого файла с помощью ЭП, формируемой для каждого файла сборки.

*Примечание.* Проверке подлежат только файлы формата: «.cmd», «.bat», «.exe», «.osx», «.bpl», «.dll», «.jar», «.class».

В форме редактирования файла **azk.chk** каждая строка содержит информацию об одном файле сборки: <путь к файлу от корневого каталога сборки (включая имя файла) ><пробел><хеш электронной подписи>. Последняя строка содержит хеш электронной подписи для chk-файла.

### **Пример.**

`SQL\DBUpdate.cmd`

`[4TTk p0v68ugzqMtZsMmkP7Bfx1MhZxjk YbMC3jd3ZjXEp75XmOg6dWFNFKfk Q0Y/x+yk+k G8+l2PA CQveaOLYQ==]`

`SQL\back up.cmd`

`[jDdhnuv2uYH80EYIiT5JC2FSSIDotDHwFdMk 1ByXDuN6XzrzN3grx+MojdLr9MBc/IMAm7/17UDIZouJ 0XrIA==]`

```
server\lib\xml-apis.jar  
[F/PDWeBgynOyzhxBKbr/9Fm+rH9SulJGjunoNAhknUWb9KnwAQ8ENJkV246HXWBGs0Qw2rBC7T  
0Qgaz68bgSUQ==]  
server\lib\xmlsec-1.2.1.jar  
[umcTsJgaOaSb0IKltyvkUctIpgWARvu7HA4h39HCHoJKFLAxuzpYGmvPDBhiDcaZzoTusKFwCUbt  
zCrc+LFBw==]  
server\loader.jar  
[gWLLLqjEEwFzTS0+VfcmFt39YqDhLZSfp5P+U/ell3fahHUcWhck2BI2mbUYIT/AF+kRGoZzi01hbp  
MNfih2kA==]  
[V6sUq4AgZcFqSIxH3eqNAcKgb3K5ISdk1Hdx/1GdP4tjE2Cx0OQb8Tj5H0IxSEjFsMk6BIcRVNaU+Vi  
18UM1XQ==]
```

В случае наличия в дистрибутиве файла **integrity\_check.cmd**, для настройки механизма проверки целостности дистрибутива необходимо:

- в переменной **builddir** указать путь к корневому каталогу дистрибутива;
- в переменной **certfile** указать путь к файлу **integrityCert.cer**;
- в переменной **cfgfile** указать путь к файлу конфигурации;
- в переменной **resultfile** указать путь к файлу **azk.chk**;
- в переменной **config** указать путь к файлу конфигурации сервера приложений **server.properties**.

Для указания путей к файлам рекомендуется использовать пути локальных файловых систем (не сетевые имена). В случае возникновения необходимости осуществить проверку целостности дистрибутива, находящегося в сети, рекомендуется использовать **mapping** дисков (монтирование в сетевого диска в локальную файловую систему).

**Пример.** Пример файла **integrity\_check.cmd**:

```
set builddir=%~dp0%  
set certfile=%builddir%\integrity.cer  
set resultfile=%builddir%\azk.chk  
set hashalg=  
set signalg=  
IF not "%hashalg%"==" " ( set HASHALGPARAM=-hashalg %hashalg% )  
IF not "%signalg%"==" " ( set SIGNALGPARAM=-signalg %signalg% )  
set ALGPARAM=%HASHALGPARAM% %SIGNALGPARAM%
```

```
:start
```

```
call %builddir%\server\bin\start com.bssys.tools.integrity.SuiteIntegrity -builddir %builddir% -certfile  
%certfile% -resultfile %resultfile% %ALGPARAM% check
```

**Примечание.** Проверить целостность файлов сборки с именами можно из файла конфигурации, в котором прописаны пути к файлам для обработки (каждый в новой строке). Если параметр **cfgfile** не задан, то файлы проверяются из каталога текущей сборки (**builddir**). Проверка целостности и достоверности осуществляется через результирующий файл **resultfil**. Проверка выполняется с помощью команды **check**.

Для запуска процесса проверки целостности дистрибутива запустите на исполнение файл **integrity\_check.cmd**.

В процессе проверки целостности дистрибутива происходят следующие операции:

- файл **azk.chk** считывается в память и производится проверка его подлинности:
  - в случае отсутствия файла на экран выводится сообщение об ошибке и процесс проверки завершается;
  - в случае повреждения файла на экран выводится сообщение об ошибке и процесс проверки завершается.
- составляется список файлов, подлежащих проверке и находящихся в каталоге и всех подкаталогах для указанного в переменной **builddir** каталога;
- для каждого файла проверяется соответствие ЭП, указанной в файле **azk.chk**.

Варианты окончания проверки целостности дистрибутива:

- в случае успешной проверки ЭП для каждого файла на экран выводится сообщение вида: *file SQL\sql.cmd is Ok*;
- в случае отсутствия файла **azk.chk** на экран выводится сообщение вида: *<системные время и дата>, main, ERROR, SuiteIntegrity, java.io.FileNotFoundException: .\azk.chk (Не удается найти указанный файл)*;
- в случае повреждения файла **azk.chk** на экран выводится сообщение вида: *file .\azk.chk is wrong*;
- в случае повреждения **jar-файла** на экран выводится сообщение вида: *<системные время и дата>, main, ERROR, SuiteIntegrity, com.bssys.tools.scanner.BuildException: problem reading .\server\Azk Server.jar*;

- в случае повреждения файла (не **azk.chk** или **JAR**) на экран выводится сообщение вида: *file server\AzkServer.jar is wrong*;
- в случае отсутствия проверяемого файла на экран выводится сообщение вида: *file filloutdiff.cmd missed*;
- в случае появления в каталоге с дистрибутивом не подписанного файла на экран выводится сообщение вида: *file Uninstall.exe is unknown*.



# Системные требования

## 3.1 Требования к серверам

### 3.1.1 Общие требования

Все серверы системы должны соответствовать указанным в [таблице](#)<sup>[31]</sup> требованиям, быть изготовленными известными производителями серверного оборудования (IBM, HP и т.п.), и оснащены системами бесперебойного питания, способными завершать работу сервера в штатном режиме.

Серверы системы не должны выполнять посторонних функций, быть контроллером домена, прокси-сервером и т.д. Запрещается установка на них постороннего программного обеспечения, которое не имеет отношение к функционированию системы. Обслуживание серверов должно осуществляться только ответственным сотрудником.

Установка средств защиты допускается только в тех случаях, если они не мешают функционированию системы. Рекомендуется установка антивирусных программ Dr.Web, McAfee. Не рекомендуется установка антивируса лаборатории Касперского.

Доступ к серверам и его ресурсам не должен быть свободным. На серверы могут устанавливаться операционные системы и программное обеспечение в соответствии с [таблицей](#)<sup>[31]</sup>.

Для серверов рекомендуется устанавливать 64-х разрядную операционную систему, для возможности использования большего объема оперативной памяти.

Для всех конфигураций серверного оборудования: источник бесперебойного питания, опционально FDD, CD-ROM.

При использовании процессоров, отличных от указанных Intel Xeon, необходимо подбирать производительность сервера, ориентируясь относительно производительности указанных в таблицах процессоров.

Таблица 3 – Системные требования к конфигурации для различных типов серверов

№	Тип сервера	Примечание	Программное окружение
1	Сервер СУБД АЦК	ОС в соответствии с требованиями и к СУБД	Рекомендуемые: Oracle Enterprise Linux 4/5/6/7 RH Linux AS 4/5/6/7 SUSE Linux 10/11 MS Windows 2008/7 (только для малых объектов с количеством заказчиков до 25)

			<p>Windows Server 2019 RedOS 7.1 (для СУБД Oracle и Postgres) RedOS 7.2 (для СУБД Postgres) ALT Linux 8.2 (для СУБД Postgres) ALT Linux 9.1 (для СУБД Postgres) Astra Linux 2.12 (для СУБД Postgres) РОСА «Кобальт» 7.3 (для СУБД Postgres)</p>
		СУБД Oracle	<p>Oracle 11g (рекомендуемая 11.2.0.4) Oracle 12c (рекомендуемая 12.1.0.2, 12.2 не поддерживается)</p>
		СУБД Firebird	Firebird-2.5.2.
		СУБД Postgres	<p>PostgreSQL 9.6.11 и выше PostgreSQL 11.5-11.8 (рекомендуемая)</p>
		СУБД RedDatabase	<p>RedDatabase 2.5.0.10604 (Для Linux минимальная рекомендованная версия ядра - 2.6.32, и GLIBC - 2.11.1. Также в Red Hat Enterprise Linux 5.5 и более поздних версиях имеются необходимые исправления. На более ранних версиях возможны проблемы со стабильностью работы.)</p>
2	Сервер приложения подсистем	ОС	<p>Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10 SP2/11 MS Windows 2008/7 (только для малых объектов с количеством заказчиков до 25)/2012 R2(642) Windows Server 2019 RedOS 7.1 RedOS 7.2 ALT Linux 8.2 ALT Linux 9.1 Astra Linux 2.12 РОСА «Кобальт» 7.3</p>
		JDK	Sun Java SE 8 update от 40 до 160
3	Сервер лицензий	ОС	MS Windows 2008 Server / Windows 7
		JDK	Java SE 8 (x86) update 40 и выше
		Sentinel	Не ниже 7.4.2
4	Сервер ЭП	ОС	<p>MS Windows 2008/7/2012 R2(642) Windows Server 2019 для КриптоПро CSP 5.0 (ограничение: не работает ЭП для типа подписи «усиленный с доказательством подлинности», другие типы подписи работают) Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10 SP2/11 AltLinux centaurus 7.0.5 RedOS 7.1</p>



			<p>RedOS 7.2 (ограничение: не работает ЭП для типа подписи «усиленный с доказательством подлинности», другие типы подписи работают)</p> <p>ALT Linux 8.2 ALT Linux 9.1 Astra Linux 2.12</p>
		JDK	Sun Java SE 8 update от 40 до 160
		ЭП	<p>При использовании ОС MS Windows одно из поддерживаемых СКЗИ:</p> <ul style="list-style-type: none"> <li>• КриптоПро CSP 3.6-4.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»). При этом необходима установка «Cades runtime».</li> </ul> <p>Опционально – КриптоПро OSCP Client и КриптоПро TSP Client;</p> <ul style="list-style-type: none"> <li>• VPNet CSP 3.2.</li> <li>• КриптоПро ФКН CSP 3.9</li> </ul> <p>При использовании ОС Linux, ОС MS Windows: КриптоПро JCP 2.01*.</p>
5	Сервер СУБД	ОС в соответствии с требованиями и к СУБД	<p>Рекомендуемые:</p> <ul style="list-style-type: none"> <li>• Oracle Enterprise Linux 4/5/6/7</li> <li>• RH Linux AS 4/5/6/7</li> <li>• SUSE Linux 10/11</li> <li>• MS Windows 2008/7 (только для малых объектов с количеством заказчиков до 25)</li> </ul>
		СУБД Oracle	<p>Oracle 11g (рекомендуемая 11.2.0.4) Oracle 12c (рекомендуемая 12.1.0.2)</p>
		СУБД Firebird	Firebird-2.5.2
		СУБД Postgres	<p>PostgreSQL 9.6.11 и выше PostgreSQL 11.5-11.8 (рекомендуемая)</p>
		СУБД RedDatabase	<p>RedDatabase 2.5.0.10604</p> <p>Для Linux минимальная рекомендованная версия ядра – 2.6.32, и GLIBC – 2.11.1. Также в Red Hat Enterprise Linux 5.5 и более поздних версиях имеются необходимые исправления. На более ранних версиях возможны проблемы со стабильностью работы.</p>
6	Сервер для интеграции с ЕИС и ЭТП	ОС	<p>Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10/11 RedOS 7.1 RedOS 7.2 ПОСА «Кобальт» 7.3 Astra Linux 2.12 ALT Linux 9.1 MS Windows 2008 Server</p>

			Windows 7 Windows Server 2019 Suse Linux Enterprise Desktop Red Hat Enterprise Linux WS Red Hat Desktop.
		JDK (JRE)	Sun Java SE 8 update 40 и выше
		КриптоПро JCP/JTLS	Не ниже 2.0 При интеграции с ЕИС для 44-ФЗ необходим вариант лицензии, в котором доступны функции проверки подписи (дешифровки), необходимые для организации одностороннего защищенного канала. При интеграции с ЕИС для 223-ФЗ необходим вариант лицензии, в котором доступны функции наложения и проверки подписи (шифровки и дешифровки), необходимые для организации двухстороннего защищенного канала.
7	Сервер обновлений	ОС	Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10/11 MS Windows 2008/7 (32)/2012 R2(64)
		JDK	Sun Java SE 8 update 40 - 51
8	Сервер Http-клиента и web-клиента	ОС	Рекомендуемые: <ul style="list-style-type: none"> <li>• Oracle Enterprise Linux 4/5/6/7</li> <li>• RH Linux AS 4/5/6/7</li> <li>• SUSE Linux 10/11</li> <li>• MS Windows 2008/7</li> </ul>
		JDK	Firebird-2.5.2
		Apache Tomcat	8.5 и выше
9	Сервер WEBSITE	ОС	Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10 SP2/11 MS Windows 2008/7 (32)/2012 R2(642) Windows Server 2019 RedOS 7.1 RedOS 7.2 ALT Linux 8.2 ALT Linux 9.1 Astra Linux 2.12
		JDK	Sun Java SE 8 update от 40 до 77
		Apache Tomcat	6.0.29

1 0	Сервер СУБД для WEB-сайта	ОС соответствии с требованиям и к СУБД	в	Рекомендуемые: Oracle Enterprise Linux 4/5/6/7 RH Linux AS 4/5/6/7 SUSE Linux 10/11 MS Windows 2008/7 Windows Server 2019 RedOS 7.1 RedOS 7.2 ALT Linux 8.2 для СУБД Firebird-2.5.2 ALT Linux 9.1 только для СУБД PostgreSQL 11.5-11.8 Astra Linux 2.12	
				СУБД Firebird	Firebird-2.5.2
				СУБД Postgres	PostgreSQL 9.6.11 и выше PostgreSQL 11.5-11.8 (рекомендуемая)
1 1	Сервер СУБД для журнала действий пользователя	ОС соответствии с требованиям и к СУБД	в	Рекомендуемые: Oracle Enterprise Linux 4/5/6/7 RH Linux AS 4/5/6/7 SUSE Linux 10/11 MS Windows 2008/7 Windows Server 2019 RedOS 7.1 RedOS 7.2 ALT Linux 8.2 для СУБД Firebird-2.5.2 ALT Linux 9.1 только для СУБД PostgreSQL 11.5-11.8 Astra Linux 2.12	
				СУБД Firebird	Firebird-2.5.2
				СУБД Postgres	PostgreSQL 9.6.11 и выше PostgreSQL 11.5-11.8 (рекомендуемая)
1 2	Сервер для интеграции с АЦК-Финансы, АЦК- Планирование, АЦК- Бюджетный учет	ОС	в	Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10 SP2/11 RedOS 7.1 RedOS 7.2 ПОСА «Кобальт» 7.3 Astra Linux 2.12 ALT Linux 9.1 MS Windows 2008/7 (только для малых объектов с количеством заказчиков до 25)/2012 R2(642) Windows Server 2019	
				JDK (JRE)	Sun Java SE 8 update от 40 до 160

**Примечание.** Для 64-разрядной ОС рекомендуется не менее 4Gb оперативной памяти.

### 3.1.2 Сервер приложений АЦК

Сервер приложения – это сервер, на который устанавливается основная серверная часть системы. Данный сервер осуществляет обработку запросов клиентских частей системы и передачу этих запросов базе данных системы, а также получение данных из базы и передача их клиентским частям. Сервер приложения выступает промежуточным звеном между базой данных и клиентскими частями, обеспечивая тем самым защиту данных и распределение нагрузки. Ниже приведены требования конфигурации серверов для установки сервера приложений каждому виду объекта:

Таблица 4 – Требования к конфигурации серверов для установки сервера приложений

Количество on-line пользователей		до 50	до 250	250 -500	500 – 1000	Свыше 1000
Минимальная конфигурация	CPU	1x4Core 3GHz (2x2Core 3GHz) Intel Xeon 64 bit	2x4Core 3GHz Intel Xeon 64 bit	2 сервера 2x4Core 3GHz Intel Xeon 64 bit	От четырёх серверов 2x4Core 3GHz Intel Xeon 64 bit	От пяти серверов 4x4Core 3GHz Intel Xeon 64 bit
	RAM	16GB	32GB	64GB на сервер	64GB на сервер	64GB на сервер
	HDD	4x36GB SCSI, SAS	4x36GB SCSI, SAS	4x36GB SCSI, SAS на сервер	4x36GB SCSI, SAS на сервер	4x36GB SCSI, SAS на сервер
	Ethernet	1Gbit	2Gbit	4Gbit	6Gbit	8Gbit
Рекомендуемая конфигурация	CPU	2x4Core 3GHz Intel Xeon 64 bit	4x4Core 3GHz Intel Xeon 64 bit	3 сервера 4x4Core 3GHz Intel Xeon 64 bit	От пяти серверов 4x4Core 3GHz Intel Xeon 64 bit	От пяти серверов 4x4Core 3GHz Intel Xeon 64 bit
	RAM	32GB	64GB	64GB на сервер	64GB на сервер	64GB на сервер
	HDD	4x73GB SCSI, SAS	4x73GB SCSI, SAS	4x73GB SCSI, SAS на сервер	4x73GB SCSI, SAS на сервер	4x73GB SCSI, SAS на сервер
	Ethernet	2Gbit	2-4Gbit	4Gbit	6Gbit	8Gbit

Возможен запуск сервера web-клиента на одном компьютере с сервером приложений для клиентских запросов, к которому он подключен, при условии, что выделенные технические ресурсы соответствуют сумме требуемых технических ресурсов, необходимых для запуска сервера приложений и web-клиента на отдельных компьютерах.

### 3.1.3 Сервер для интеграции с ЕИС и ЭТП

Сервер для интеграции с ЕИС и ЭТП – это сервер, на который устанавливается серверная часть системы. Данный сервер осуществляет прием и отправку сообщений при интеграции с ЕИС и ЭТП.

Ниже приведены требования к конфигурации серверов:

Таблица 5 – Требования к конфигурации серверов

Вид сервера	Рекомендуемые технические требования
Сервер для интеграции с ЕИС и ЭТП	CPU-3GHz Intel Xeon, 8Gb RAM, 18GB HDD SCSI, сетевая карта Ethernet 1Gb

### 3.1.4 Сервер базы данных АЦК

Сервер базы данных АЦК – это сервер, на который устанавливается система управления базами данных. Данный сервер осуществляет хранение и обработку данных системы. К этому серверу предъявляются особые требования по производительности и надежности.

Ниже приведены требования к конфигурации серверов для каждого из видов объектов:

Таблица 6 – Требования к конфигурации серверов

Количество on-line пользователей		до 50	до 250	250 -500	500 – 1000	Свыше 1000
Минимальная конфигурация	CPU	1x2Core 2GHz Intel Xeon	2x2Core 3GHz Intel Xeon 64 bit	2x4Core 3GHz Intel Xeon 64 bit	-	-
	RAM	8Gb	16Gb	32Gb	-	-
	HDD	6x36Gb SCSI/SAS 15k Rpm	8x73Gb SCSI/SAS 15k Rpm	8x73Gb SCSI/SAS 15k Rpm	-	-

	RAID Controller	Ultra320 SCSI или SAS с поддержкой не менее чем 6 дисков, RAID-кеш 128MB	Ultra320 SCSI или SAS с поддержкой не менее чем 8 дисков, RAID-кеш 256MB	Ultra320 SCSI или SAS с поддержкой не менее чем 8 дисков, RAID-кеш 256MB	-	-
	RAID Level	10	10	10	-	-
	Ethernet	1Gbit	1Gbit	1Gbit	-	-
Рекомендуемая конфигурация	CPU	1x4Core 2GHz (2x2Core 2GHz) Intel Xeon 64 bit	2x4Core 3GHz Intel Xeon 64 bit	4x4Core 3GHz Intel Xeon 64 bit	4x8Core 3GHz Intel Xeon 64 bit	От 4x8Core 3GHz Intel Xeon 64 bit
	RAM	8Gb	32Gb	64Gb	128Gb	От 128Gb
	HDD	6x36Gb SCSI/SAS 15k Rpm	Использование СХД	Использование СХД	Использование СХД	Использование СХД
	RAID Controller	Ultra320 SCSI или SAS с поддержкой не менее чем 6 дисков, RAID-кеш 128Mb	-	-	-	-
	RAID Level	10	-	-	-	-
	Ethernet	1Gbit	2Gbit	4Gbit	8Gbit	16Gbit

### 3.1.5 Совмещенный сервер приложений и сервер базы данных

Совмещенный вариант сервера приложений и сервера БД возможен только при низкой нагрузке и малом количестве пользователей (до пятидесяти человек).

В таком варианте использования ОС MS Windows 2003/2008/7 допустимо только для малых объектов с количеством on-line пользователей до двадцати пяти человек.

Ниже приведены требования к конфигурации совмещенного сервера приложений и сервера БД:

Таблица 7 – Требования к конфигурации совмещенного сервера приложений и сервера баз данных

Количество on-line пользователей	до 50
----------------------------------	-------

<b>Рекомендуемая конфигурация</b>	CPU	1x4Core 3GHz (2x2Core 3GHz) Intel Xeon 64 bit
	RAM	16Gb
	HDD	6x36Gb SCSI/SAS 15k Rpm
	RAID Controller	Ultra320 SCSI/SAS с поддержкой не менее чем 6 дисков, RAID-кеш 128MB
	RAID Level	10
	Ethernet	1Gbit

### 3.1.6 Сервер лицензий

Сервер лицензий – это сервер, который осуществляет проверку доступа к функциям системы на основании лицензии.

Ниже приведены требования к конфигурации сервера лицензий:

Таблица 8 – Требования к конфигурации сервера лицензий

Вид сервера	Рекомендуемые технические требования
Сервер лицензий	Intel Xeon -3Гц, 2Gb RAM, 18Gb HDD SCSI, сетевая карта Ethernet 1Gb

### 3.1.7 Сервер ЭП

Сервер электронной подписи (ЭП) – это сервер приложений АЦК, на который вынесены функции валидации электронных подписей, сформированных в системе. Вынесение данной функциональности на отдельный сервер способствует масштабированию и балансировке нагрузки на серверную часть системы, а также обеспечивает возможность использования на основном сервере приложений ОС семейства Unix при одновременном использовании ОС класса Windows на сервере ЭП. Это может быть особенно актуально при использовании Windows-версии средства криптографической защиты информации (СКЗИ).

Ниже приведены требования к конфигурации сервера ЭП:

Таблица 9 – Системные требования к конфигурации сервера электронной подписи

Максимальное количество пользователей, одновременно выполняющих проверку ЭП для группы ЭД	Вид используемой ЭП	Рекомендуемая конфигурация		
		CPU	RAM	Ethernet
до 5	Усиленная (64Б), Усиленная (XML), Усиленная (со штампом времени)	2 Cores 3GHz Intel Xeon 64 bit	4GB	1Gb it
	Усиленная (с доказательствами подлинности)	4 Cores 3GHz Intel Xeon 64 bit		
до 10	Усиленная (64Б), Усиленная (XML), Усиленная (со штампом времени)	4 Cores 3GHz Intel Xeon 64 bit		
	Усиленная (с доказательствами подлинности)	8 Cores 3GHz Intel Xeon 64 bit		
до 20	Усиленная (64Б), Усиленная (XML), Усиленная (со штампом времени)	8 Cores 3GHz Intel Xeon 64 bit		
	Усиленная (с доказательствами подлинности)	16 Cores 3GHz Intel Xeon 64 bit		

*Примечание.* Для поддержки большего количества пользователей (свыше двадцати), одновременно выполняющих множественную проверку ЭП, рекомендуется развертывание дополнительного сервера ЭП.

### 3.1.8 Сервер обновлений

Сервер обновлений – это сервер, на который устанавливается серверная часть системы. Данный сервер осуществляет обновление клиентской АРМ, установленной у пользователя.

Ниже приведены требования к конфигурации сервера:



Таблица 10 – Системные требования к конфигурации сервера

Вид сервера	Минимальные технические требования
Сервер обновлений	Intel Xeon 3Гц, 4Gb RAM, 18Gb HDD SCSI, сетевая карта Ethernet 1Gb

### 3.1.9 Сервер web-сайта

Данный сервер обеспечивает функционирование web-сайта системы, осуществляет обработку запросов сайта и передачу запросов на сервер приложения, а также получение данных от сервера приложений и передачу их сайту.

Ниже приведены требования к конфигурации сервера:

Таблица 11 – Системные требования к конфигурации сервера

Минимальная конфигурация	CPU	1x4Core 2GHz Intel Xeon 64 bit	Рекомендуемая конфигурация	CPU	1x4Core 3Ghz (2x2Core 3GHz) Intel Xeon 64 bit
	RAM	8GB		RAM	16GB
	HDD	1x36GB SCSI, SAS		HDD	2x36GB SCSI, SAS
	Ethernet	1Gbit		Ethernet	1Gbit

### 3.1.10 Сервер web-клиента

В качестве клиентского приложения используется интернет браузер. При количестве пользователей более пятисот рекомендуется распределить веб-сервер по разным физическим узлам в рамках рекомендуемой конфигурации. Для распределения конечных пользователей по узлам кластера веб-серверов могут быть применены стандартные аппаратные или программные балансировщики нагрузки, либо распределение пользователей по серверам может быть осуществлено организационными мерами. Рекомендуется подбирать конфигурации с одинаковым количеством серверов приложения и веб-серверов, в этом случае возможно выделение для каждого веб-сервера собственного сервера приложения без ресурсозатрат на мультиплексирование.

Ниже приведены требования к конфигурации серверов в зависимости от количества клиентских рабочих мест:

Таблица 12 – Требования к конфигурации серверов в зависимости от количества клиентских рабочих мест

Количество on-line пользователей		до 250	От 250 до 500	от 500 до 1000	Свыше 1000
Минимальная конфигурация	CPU	1x4Core 2GHz Intel Xeon 64 bit	2x4Core 3GHz Intel Xeon 64 bit	2 сервера 2x4Core 3GHz Intel Xeon 64 bit	От 4-х серверов 2x4Core 3GHz Intel Xeon 64 bit
	RAM	8GB	16GB	По 16GB	По 16GB
	HDD	1x36GB SCSI, SAS	1x36GB SCSI, SAS	По 1x36GB SCSI, SAS	По 1x36GB SCSI, SAS
	Ethernet	1Gbit	1Gbit	2Gbit	2Gbit
Рекомендуемая конфигурация	CPU	1x4Core 3Ghz (2x2Core 3GHz) Intel Xeon 64 bit	4x8 Core 3GHz Intel Xeon 64 bit	2 сервера 4x8Core 3GHz Intel Xeon 64 bit	От 4-х серверов 4x8Core 3GHz Intel Xeon 64 bit
	RAM	16GB	32GB	По 32GB	По 32GB
	HDD	2x36GB SCSI, SAS	2x36GB SCSI, SAS	2x36GB SCSI, SAS	2x36GB SCSI, SAS
	Ethernet	1Gbit	2Gbit	4Gbit	8Gbit

В случае необходимости организации подключений к web-серверу из сети Интернет, web-сервер должен располагаться в безопасной части сети – [«демитилизированной зоне»](#)<sup>49</sup>.

### 3.1.11 Proxy-сервер

Данный сервер является промежуточным звеном между клиентами и сервером приложений, осуществляет балансировку нагрузки, а так же перенаправление соединений толстых клиентов по протоколу **http** на сервер приложений. В качестве прокси-сервера можно использовать **nginx** – продукт, использующийся не только в качестве веб-сервера, но и как прокси-сервер.

*Примечание. При настройке приложения для балансировки нагрузки необходимо отключить кэширование файлов.*

Ниже приведены требования к конфигурации сервера:

Таблица 13 – Системные требования к конфигурации сервера

Минимальные требования	Рекомендуемые требования
1x4Core 2GHz Intel Xeon 64 bit, 8 ГБ RAM, Ethernet 1Gbit	1x4Core 2GHz Intel Xeon 64 bit, 16 ГБ RAM, Ethernet 1Gbit

## 3.2 Требования к рабочим местам

### 3.2.1 Общие требования

Все рабочие места должны соответствовать указанным требованиям, иметь надежную связь и программное окружение. Не допускается установка на эти АРМ программного обеспечения, изменяющего стандартные функции операционной системы и другое ПО, способное нарушить функционирование систем. Все требования разделены на две категории: минимальные и рекомендуемые.

Минимальные требования – это требования к конфигурации компьютера, который позволит запустить систему и работать с ней. Работа на таком компьютере будет крайне затруднительна. При работе с большим объемом данных устойчивая и безошибочная работа системы не гарантируется.

Рекомендуемые требования – это требования к конфигурации компьютера, который позволит достичь комфортного режима работы с системой. Исключит вероятность возникновения сбоев в работе из-за нехватки ресурсов системы. Компьютер признается соответствующим требованиям, если:

- компьютер имеет конфигурацию не ниже указанной в требованиях, как рекомендуемая;
- компьютер подключен к линиям связи;
- к компьютеру подключены необходимые устройства и периферия;
- все устройства настроены (установлены драйвера, отсутствуют конфликты).

### 3.2.2 АРМ Уполномоченного органа (Corba и Http)

АРМ Уполномоченного органа – это рабочее место сотрудника уполномоченного органа, которое является клиентским приложением системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» и предоставляет пользователю доступ к данным и функциям системы посредством графического интерфейса ОС семейства Windows.

Ниже приведены системные требования к конфигурации компьютера и его оснащению:

Таблица 14 – Системные требования к конфигурации рабочего места

Минимальные требования	Рекомендуемые требования
Процессор: P4-2,8 Ghz Память: 2 Gb Диск: 20 Gb (свободно) Монитор и карта: SVGA 1280x1024 Подключение: Ethernet 100 Mbit ОС: Windows 7 ПО: <ul style="list-style-type: none"><li>офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx<sup>1</sup></li><li>MS Excel/Word 2007/2010/2013<sup>2</sup></li></ul>	Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 20 Gb (свободно) Разрешение экрана: 1280x1024 или 1440x900 Подключение: Ethernet 100 Mbit ОС: Windows 7/8 ПО: <ul style="list-style-type: none"><li>офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx<sup>1</sup></li><li>MS Excel/Word 2007/2010/2013<sup>2</sup></li></ul>

**Примечания:**

<sup>1</sup> При использовании механизма формирования отчетов на сервере

<sup>2</sup> При использовании механизма построителя пользовательских отчетов и/или механизма формирования отчетов на клиенте

**Примечание.** Для 64-разрядной ОС рекомендуется не менее 4Gb оперативной памяти.

### 3.2.3 АРМ Уполномоченного органа (WEB-клиент)

АРМ Уполномоченного органа – это рабочее место сотрудника уполномоченного органа, доступ к системе «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» с помощью веб-браузера, без установки клиентского программного обеспечения на рабочую станцию.

Ниже приведены системные требования к конфигурации компьютера и его оснащению:

Таблица 15 – Системные требования к конфигурации рабочего места

Минимальные требования	Рекомендуемые требования
Процессор: P-4 2.8 GHz Память: 1 Gb Диск: 1 Gb (свободно) Разрешение экрана: 1024x768 Сеть: 2Mbit ОС: Windows 7 Браузер: Mozilla Firefox 60.0, MS Internet Explorer 10.0, 11.0, Google Chrome 67.0 ПО офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx СКЗИ: КриптоПро CSP 3.6 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»), КриптоПро ФКН CSP 3.9, ViPNet CSP 3.2	Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 1 Gb (свободно) Разрешение экрана: 1280x1024 и выше Сеть: 10Mbit ОС: Windows 7/Windows 8 Браузер: Mozilla Firefox 60.0, MS Internet Explorer 10.0, 11.0, Google Chrome 67.0 ПО: офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx СКЗИ: КриптоПро CSP 3.6 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»), КриптоПро ФКН CSP 3.9, ViPNet CSP 3.2

*Примечание. Для 64-разрядной ОС рекомендуется не менее 4Gb оперативной памяти.*

### 3.2.4 АРМ Заказчика (Corba и Http)

АРМ Заказчика – это рабочее место сотрудника заказчика, которое является клиентским приложением системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» и предоставляет пользователю доступ к данным и функциям системы посредством графического интерфейса ОС семейства Windows.

Ниже приведены системные требования к конфигурации компьютера и его оснащению:

Таблица 16 – Системные требования к конфигурации рабочего места

Минимальные требования	Рекомендуемые требования
Процессор: P4-2,8 GHz Память: 2 Gb Диск: 20 Gb (свободно) Монитор и карта: SVGA 1280x1024 Подключение: Ethernet 100Mbit ОС: Windows 7 ПО:	Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 20 Gb (свободно) Разрешение экрана: 1280x1024 или 1440x900 Подключение: Ethernet 100 Mbit ОС: Windows 7/8 ПО:

Минимальные требования	Рекомендуемые требования
<ul style="list-style-type: none"> <li>офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx<sup>1</sup></li> <li>MS Excel/Word 2007/2010/2013<sup>2</sup></li> </ul>	<ul style="list-style-type: none"> <li>офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx<sup>1</sup></li> <li>MS Excel/Word 2007/2010/2013<sup>2</sup></li> </ul>

**Примечания:**

<sup>1</sup> При использовании механизма формирования отчетов на сервере

<sup>2</sup> При использовании механизма построителя пользовательских отчетов и/или механизма формирования отчетов на клиенте

**Примечание.** Для 64-разрядной ОС рекомендуется не менее 4Gb оперативной памяти.

### 3.2.5 АРМ Заказчика (WEB-клиент)

АРМ Заказчика – это рабочее место сотрудника заказчика, доступ к системе «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» с помощью веб-браузера, без установки клиентского программного обеспечения на рабочую станцию.

Ниже приведены системные требования к конфигурации компьютера и его оснащению:

Таблица 17 – Системные требования к конфигурации рабочего места

Минимальные требования				Рекомендуемые требования			
Процессор: P-4 2.8 GHz Память: 1 Gb Диск: 1 Gb (свободно) Разрешение экрана: 1024x768 Сеть: 2Mbit ПО: офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx				Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 1 Gb (свободно) Разрешение экрана: 1280x1024 и выше Сеть: 10Mbit ПО: офисный пакет с поддержкой чтения файлов doc, docx, xls, xlsx			
ОС	Windows 7	Scientific Linux 7.2	Linux	ОС	Windows 7/8/10	Scientific Linux 7.2	Linux
Браузер	Mozilla Firefox 60.0, MS Internet Explorer 10.0, 11.0, Google Chrome 67.0	Google Chrome 67.0	Chrome	Браузер	MS Internet Explorer 10.0, 11.0, Mozilla Firefox 60.0, Google Chrome 67.0	Google Chrome 67.0	Chrome

*Примечание. Для 64-разрядной ОС рекомендуется не менее 4Gb оперативной памяти.*

### 3.2.6 Требования к программному окружению при использовании криптографических функций

При использовании функций наложения ЭП и аутентификации по сертификату на компьютере должно быть установлено следующее программное обеспечение:

Таблица 18 – Требования к программному окружению при использовании криптографических функций

При использовании СКЗИ КриптоПро			При использовании СКЗИ ViPNet		
СКЗИ: КриптоПро CSP 3.6-4.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»), КриптоПро ФКН CSP 3.9. При этом необходима установка «Cades runtime». Опционально: КриптоПро OSCP Client и КриптоПро TSP Client.			СКЗИ: ViPNet CSP 3.2-4.2		
Windows-клиент					
ОС	Windows 7/8/10		ОС	Windows 7/10	
WEB-клиент					
ОС	Windows 7/8/10	Scientific Linux 7.2	ОС	Windows 7/10	
Доп.ПО	Сервис ЭП АЦК 1.0.3.9	Плагин nmsigner-1.0.9-1	Доп. ПО	Сервис ЭП АЦК 1.0.3.9	

Сервис ЭП АЦК – программное окружение, обеспечивающее использование криптографических функций при работе в браузерах Google Chrome, Mozilla Firefox, Internet Explorer под ОС Windows.

Плагин – программное окружение, обеспечивающее использование криптографических функций при работе только в браузере Google Chrome под ОС Linux.

*Примечание. При использовании функций наложения ЭП на прикрепленные к ЭД файлы (вложения), рекомендуемый размер подписываемого файла не должен превышать 20МБ.*

### 3.2.7 Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ

При использовании криптографического протокола HTTPS в соответствии с требованиями ГОСТ к алгоритмам шифрования данных на компьютере должно быть установлено следующее программное обеспечение:

Таблица 19 – Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ

При использовании СКЗИ КриптоПро	При использовании СКЗИ ViPNet
ОС: Windows 7/8 СКЗИ: КриптоПро CSP 3.6-4.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»), КриптоПро ФКН CSP 3.9 Опционально: КриптоПро TSP Client, КриптоПро OSCP Client Браузер: КриптоПро Fox 45.0, MS Internet Explorer 10.0, 11.0	ОС: Windows 7 СКЗИ: ViPNet CSP 3.2-4.2 Браузер: MS Internet Explorer 10.0, 11.0

*Примечание. Только при использовании web-клиента!*

## 3.3 Требования к линиям связи

### 3.3.1 Общие требования к линиям связи

Все линии связи должны обеспечивать надежную и устойчивую связь. Все серверы и службы, обеспечивающие функционирование сети, должны быть тщательно настроены и иметь механизмы дублирования их функций.

### 3.3.2 Линии связи между серверами системы

Все используемые линии связи должны поддерживать сетевой протокол TCP/IP и обеспечивать работу со скоростью не ниже 1 Гбит. Реальная скорость передачи данных между серверами должна быть не ниже 20 Мб/сек. Подключение сервера приложения АЦК с сервером базы данных, а также web-сервера должны осуществляться со скоростью 1 Гбит.



### 3.3.3 Линии связи между сервером приложения и АРМ Уполномоченного органа

Все используемые линии связи должны поддерживать сетевой протокол TCP/IP и обеспечивать работу со скоростью не ниже 100 Мбит. Реальная скорость передачи данных от любого рабочего места до сервера, в момент пиковых нагрузок, должна быть не ниже 2 Мб/сек.

### 3.3.4 Линии связи между сервером приложения и АРМ Заказчика

Все используемые линии связи должны поддерживать сетевой протокол TCP/IP и обеспечивать работу со скоростью не ниже 2 Мб/сек.

## 3.4 Требования к безопасности

### 3.4.1 Демилитаризованная зона (DMZ)

Когда необходим доступ пользователей к web-сервисам системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» из общественной сети Интернет, необходимо обеспечить безопасность и защиту данных внутренней локальной сети предприятия. Для этой цели используется технология «демилитаризованных зон» или DMZ.

Суть DMZ заключается в том, что она не входит непосредственно ни во внутреннюю, ни во внешнюю сеть, и доступ к ней может осуществляться только по заранее заданным правилам межсетевого экрана. В DMZ нет пользователей – там располагаются только серверы. Демилитаризованная зона, как правило, служит для предотвращения доступа из внешней сети к хостам внутренней сети за счет выноса из локальной сети в особую зону всех сервисов, требующих доступа извне. Фактически получается, что эта зона будет являться отдельной подсетью с публичными адресами, защищенной (или – отделенной) от публичных и корпоративных сетей межсетевыми экранами.

Рекомендуется использовать два межсетевых экрана, отделяющих DMZ от внешней и внутренней сети. Внешняя сеть оказывается между маршрутизатором провайдера и первым межсетевым экраном, в то время как демилитаризованная зона размещается между межсетевыми экранами №1 и №2. Архитектура с двумя межсетевыми экранами требует,

чтобы межсетевой экран №1 мог обрабатывать достаточный объем трафика, если системы в DMZ будут работать с большим объемом трафика. Межсетевой экран №2 может быть менее производительной системой, так как он обрабатывает только внутренний трафик. При этом сервер интеграции с ЕИС должен иметь выход в Интернет по протоколам FTP без прокси.

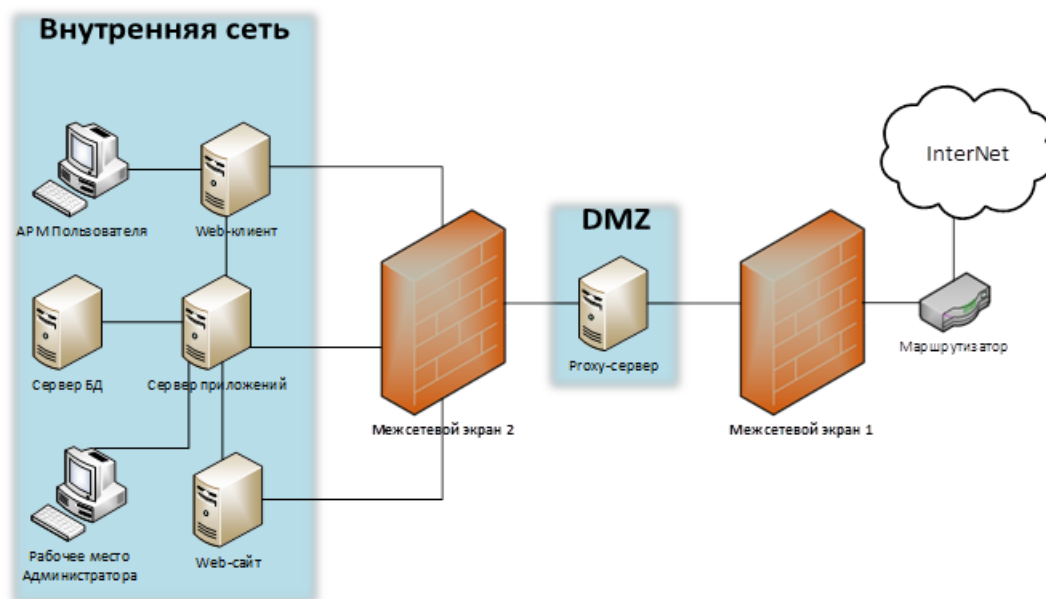


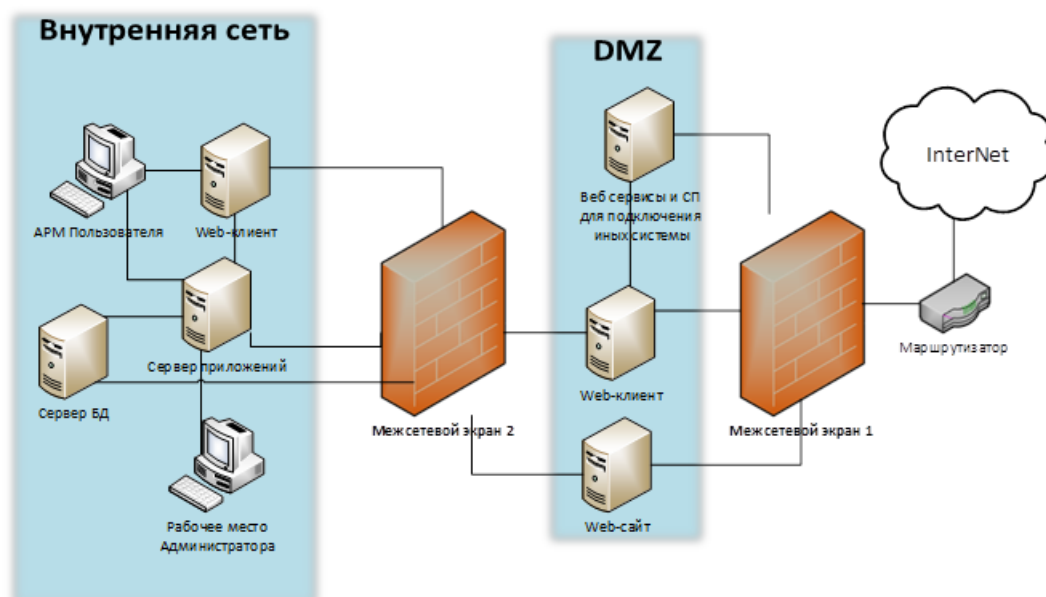
Рисунок 7 – Схема работы комплекса с использованием прокси-сервера для обработки внешних HTTP запросов от win-клиентов

Из внешней сети через межсетевой экран №1, приходит запрос, адресованный серверу демилитаризованной зоны, в которой находится прокси-сервер. Основная функция прокси-сервера – трансляция сетевых пакетов с прослушиваемого публичного порта (чаще всего 80 порт) во внутреннюю сеть на определенный, заранее настроенный порт сервера приложений. Далее, после осуществления проксирования HTTP пакетов от внешнего клиента, они через межсетевой экран №2 поступают во внутреннюю сеть на сервера web-сервисов (web-клиент, web-сайт, web-сервисы интеграции с РТС-Тендер и АЦК-Бюджетный учет, сервера приложений для интеграции с АЦК-Финансы и АЦК-Планирование), который, взаимодействуя с серверами приложений или базы данных, осуществляет основную обработку бизнес-логики системы. Кроме этого, сервера во внутренней сети также обслуживают обращения от клиентов и серверов внутренней сети.

Работа пользователей web-клиентов внешней сети организована за счет web-сервера в демилитаризованной зоне(DMZ), который через межсетевой экран обменивается пакетами с сервером приложений внутренней сети.

Доступ к публичной части (web-сайт) реализуется сервером web-сайта демилитаризованной зоны, который так же через межсетевой экран обменивается пакетами с сервером приложений DMZ.

Далее рассмотрена схема без использования проху-сервера. Данная схема требует вывода в DMZ всех серверов (веб-клиент, web-сайт, веб-сервисы и серверы приложений для интеграции с иными системами). При этом для каждого экземпляра веб-сервера необходимо открывать отдельный порт на межсетевом экране №1. Обмен web-серверов (web-клиент, web-сайт, веб-сервисы для интеграции с иными системами) с сервером приложений и серверов приложений для интеграции с иными системами с сервером БД внутренней сети через межсетевой экран №2.



**Рисунок 8 – Схема работы комплекса с использованием сервера приложений в DMZ для обработки внешних HTTP запросов от win-клиентов**

Возможно использовать совмещенную схему, когда часть запросов обрабатываются через проху-сервер, а часть web-серверов вынесено для работы в DMZ.



# 4

## Установка и настройка программы

## 4.1 Настройка операционной системы

При работе сервера приложений под управлением ОС **Windows** для корректного ввода дат и чисел настраиваются форматы. Для настройки форматов необходимо выполнить следующие действия:

1. Через меню **Start**→**Settings**→**Control Panel**→**Regional Settings** открыть форму *Regional Options*.
2. Перейти на закладку **Date** и в поле **Short date format** выбрать формат даты *dd.MM.yyyy*.

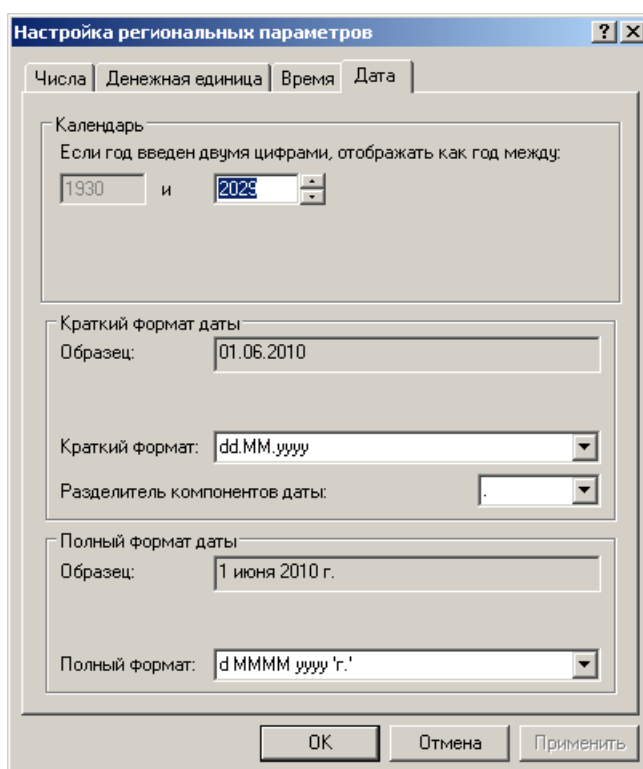


Рисунок 9 – Выбор формата даты

**Примечание.** Формат даты *dd.MM.yyyy* предусмотрен ГОСТ 6.30-2003 «Унифицированная система организационно-распорядительной документации. Требования к оформлению документов».

## 4.2 Установка и настройка СУБД

Поддерживается совместимость и оптимальная работа с СУБД **Oracle**, **Firebird**, **PostgreSQL**.

*Примечание.* К наиболее наглядным отличиям СУБД **Oracle** и СУБД **PostgreSQL** от СУБД **Firebird** следует отнести то, что в **Firebird** база данных представлена в виде отдельного файла, а в **Oracle** и **PostgreSQL** база данных состоит из нескольких файлов. СУБД **Oracle** и СУБД **PostgreSQL**, в отличие от **Firebird**, поддерживают схемы, которые, по сути, являются отдельными базами данных внутри данной базы.

СУБД **PostgreSQL** использует процедурный язык **PL/pgSQL**, но также предоставляет возможность использовать **PL/Perl**, **PL/Python**, **PL/Tcl**, как СУБД **Oracle** использует только **PL/SQL**.

Установка сервера базы данных выполняется в соответствии с документацией на систему управления БД.

### 4.2.1 Настройка СУБД «Firebird»

После установки необходимо настроить автоматический запуск службы БД.

Для корректного взаимодействия сервера приложений и БД необходимо установить дополнительные подсистемы к БД.

Дополнительные подсистемы находятся в дистрибутиве системы. Чтобы установить дополнительные подсистемы, в каталог <каталог\_FireBird>/UDF/ необходимо скопировать из дистрибутива следующие файлы:

- **rpl\_version.so** (**rpl\_version.dll** для ОС **Windows**);
- **rpl\_version2.dll**;
- **rpl\_version2.so**.

### 4.2.2 Настройка СУБД «Oracle»

СУБД **Oracle 11g** (рекомендуемая **11.2.0.4**) может работать под управлением ОС

**Windows** и **Linux**. Установка и первоначальная настройка СУБД рассмотрены в справочном руководстве **Oracle**.

## 4.2.3 Настройка PostgreSQL

СУБД **PostgreSQL** (рекомендуемая 9.5.3) может работать под управлением ОС **Windows** и **Linux**. В процессе эксплуатации СУБД под разными платформами есть отличия только в процессе ее инсталляции.

## 4.2.4 Резервное копирование БД

### 4.2.4.1 Резервное копирование СУБД «Firebird»

#### 4.2.4.1.1 Резервное копирование с помощью командной строки

Наиболее универсальным инструментом, позволяющим осуществить резервное копирование базы данных на любой платформе (OS Windows, OS Linux), является **gbak** – утилита командной строки, входящая в поставку Firebird.

Для того чтобы создать резервную копию базы данных, необходимо воспользоваться следующим образцом запуска **gbak**:

```
gbak [-b] [options] <база_данных-источник> <файл резервной копии>
```

Где:

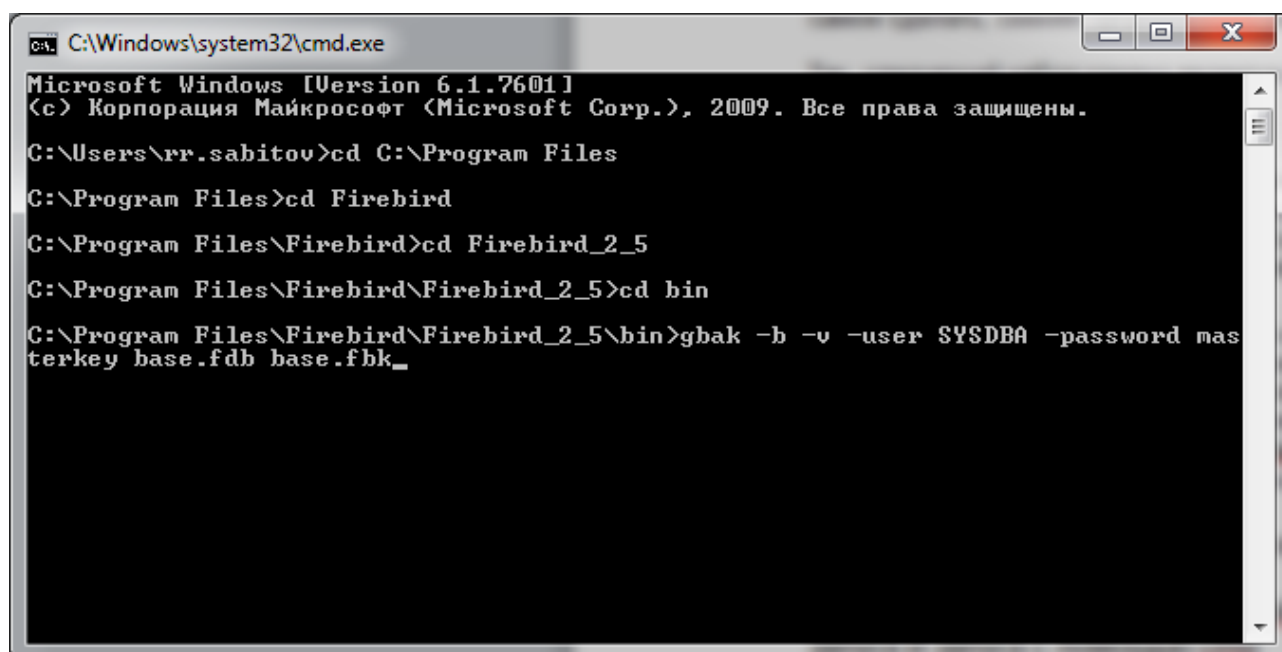
- **-b** – означает, что необходимо выполнить резервное копирование базы данных, путь к которой указан как <база\_данных-источник>, а результаты резервного копирования упаковать в файл, указанный как <файл резервной копии>:
- <база\_данных-источник> – путь к базе данных, для которой будет создана резервная копия (например, *base.fdb*);
- <файл резервной копии> – результат резервного копирования в виде файла (например, *base.fbk*);
- **options** – дополнительные опции из следующего списка:
  - **-e**(*xpand*) – не производить сжатие резервной копии;
  - **-g**(*arbage\_collect*) – не собирать «мусор» во время резервного копирования;
  - **-ig**(*ignore*) – игнорировать контрольные суммы;

- **-m(etadata)** – произвести резервное копирование только метаданных;
- **-u(ser) name** – имя пользователя, который подключается к базе данных для резервного копирования;
- **-pas(sword) text** – пароль пользователя, подключающегося к базе данных для резервного копирования;
- **-v(erbosc)** – включить показ подробного протокола действий *gbak* во время *backup*;
- **-y <path>** – направлять сообщения в файл (файла с таким именем не должно существовать) или подавить вывод сообщений;
- **-z** – показать версию *gbak* и версию ядра Firebird-сервера.

Далее указаны примеры запуска **gbak** на платформах OS Linux и OS Windows.

#### 4.2.4.1.1.1 Рекомендации для OS Windows

В OS Windows утилита **gbak** обычно расположена по пути **C:\Program Files\Firebird\Firebird\_2\_5\bin\gbak.exe**. Пример ввода команд для запуска утилиты **gbak** с параметрами для резервного копирования (командой **cd** осуществляется переход в другой каталог):



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\rr.sabitov>cd C:\Program Files
C:\Program Files>cd Firebird
C:\Program Files\Firebird>cd Firebird_2_5
C:\Program Files\Firebird\Firebird_2_5>cd bin
C:\Program Files\Firebird\Firebird_2_5\bin>gbak -b -v -user SYSDBA -password masterkey base.fdb base.fbk_
```

Рисунок 10 – Пример ввода команд

Если добавить **-y log.txt**, то процесс резервного копирования будет сохраняться в журнал изменений в указанный файл **log.txt**.

**Пример.** `gbak -b -v -y log.txt -user SYSDBA -password masterkey base.fdb base.fbk`



Где:

- **-b** – ключ, отвечающий за необходимость делать резервное копирование;
- **-user SYSDBA** – имя пользователя, который подключается к базе данных для резервного копирования;
- **-pass masterkey** – пароль пользователя, подключающегося к базе данных для резервного копирования;
- **-v** – включить показ подробного протокола действий **gbak** во время **backup**;
- **-y log.txt** – направлять сообщения в файл **log.txt**;
- **base.fdb** – база данных-источник;
- **backup.fbk** – файл резервной копии.

Далее запустится процесс формирования резервной копии базы.

#### 4.2.4.1.1.2 Рекомендации для OS Linux

Для запуска **gbak** на OS Linux, необходимо:

1. Открыть терминал: **Приложения**→**Терминал** или **Alt+F2**→**gnome-terminal**.
2. Сменить пользователя на администратора (для упрощения используется учетная запись *root*, но в реальных условиях так делать не следует). Для смены учетной записи ввести команду **su** и указать пароль (при вводе пароль не отображается звездочками, точками и т.д.).

```
$ su root
```

```
Пароль:
```

```
#
```

3. Утилиты Firebird располагаются в **/opt/firebird/bin**.

```
# cd /opt/firebird/bin
```

4. Перед первым запуском необходимо поменять пароль на пользователя **SYSDBA**. Для смены пароля пользователя необходимо запустить **changeDBAPassword.sh**.

Пароль пользователя **SYSDBA**, который был сгенерирован при установке, находится в файле **SYSDBA.password**.

Ниже показан процесс смены пароля после запуска файла

### **changeDBAPassword.sh:**

```
# cat SYSDBA.password
Firebird generated password for user SYSDBA is:
ISC_USER=sysdba
ISC_PASSWD=ASddsaQ21
.....
# ./changeDBAPassword.sh
Please enter current password for SYSDBA user: ASddsaQ21
Please enter new password for SYSDBA user: masterkey
# cat SYSDBA.password
Firebird generated password for user SYSDBA is:
ISC_USER=sysdba
ISC_PASSWD=masterkey
```

Ключи команд и параметры запуска утилиты **gbak** в OS Linux аналогичны ключам **gbak** в OS Windows.

*Пример.* Следующая команда выполнит резервное копирование базы **base.fdb** в файл **base.fbk** с подробным выводом информации:

```
gbak -b -v -y log.txt -user SYSDBA -password masterkey base.fdb base.fbk
```

Где:

- **-b** – ключ, отвечающий за необходимость делать резервное копирование;
- **-user SYSDBA** – имя пользователя, который подключается к базе данных для резервного копирования;
- **-pass masterkey** – пароль пользователя, подключающегося к базе данных для резервного копирования;
- **-v** – включить показ подробного протокола действий **gbak** во время **backup**;
- **-y log.txt** – направлять сообщения в файл **-y log.txt**;
- **base.fdb** – база данных-источник;
- **backup.fbk** – файл резервной копии.

**Внимание!** Показателем корректного резервного копирования будет отсутствие ошибок в логе. В противном случае, необходимо обратиться к официальной документации (<https://www.ibase.ru/gbak/>) и исходить из информативности ошибки.

#### 4.2.4.1.2 Резервирование с помощью пакетного файла (скрипта)

Для упрощения процедуры резервирования можно использовать запускаемый пакетный файл (скрипт).

##### 4.2.4.1.2.1 Рекомендации для OS Windows

Чтобы создать исполняемый файл *.bat*, необходимо выполнить следующие действия:

1. Запустить Блокнот или любой другой редактор файлов.

Блокнот позволяет написать код в виде текста, а затем сохранить его в виде *bat*-файла. Блокнот можно запустить, нажав **Пуск**→**Программы**→**Стандартные**→**Блокнот**. Также можно ввести *notepad* в поле **Выполнить**.

Наиболее простой набор команд включает в себя следующий код:

```
gbak -b -v -user SYSDBA -password masterkey %1 %2
```

Где:

- *%1* – это путь до базы с расширением *db*
- *%2* – это путь, куда нужно сохранить резервную копию базы с расширением *fbk*.

2. Указанный код необходимо вписать в блокнот.

Для сохранения файла необходимо выполнить следующие действия:

1. Нажать **Файл**→**Сохранить как**.
2. Нажать выпадающее меню **Тип файла**.
3. Выбрать *Все файлы*.
4. Ввести имя программы, а затем – расширение *.bat* или *.cmd*.

Для запуска *bat*-файла необходимо выполнить следующие действия:

1. Указать путь до места расположения файла.
2. Написать следующую команду: ***name.bat*** ***D:\BASE\base.fdb*** ***D:***

**`\BACKUP\base.fbk`**. Это запустит процесс выполнения резервной копии.

#### 4.2.4.1.2.2 Рекомендации для OS Linux

Необходимо создать запускаемый файл `gbak.sh` и дать ему права на запуск:

```
# touch gbak.sh
# chmod +x gbak.sh
```

Далее потребуется открыть созданный файл в редакторе. В примере используется командный редактор `nano`:

```
# nano gbak.sh
```

Откроется пустой созданный файл. В него необходимо вписать содержание будущего скрипта:

```
#!/bin/bash
# Скрипт выполняет резервное копирование базы
user=SYSDBA
password=masterkey
# Полный путь до базы данных
data_base_file=/home/oracle/Downloads/BASE_NAME.FDB
# Полный путь до бэкапа (перезапретится если будет указан существующий)
backup_base_file=/home/oracle/Downloads/BACKUP_NAME.bak
# В лог выведется вся информация о процессе выполнения скрипта
log=/home/oracle/Downloads/backup.log
# Утилита gbak выполняет резервное копирование и направляет вывод в лог
/opt/firebird/bin/gbak -b -g -v -user $user -password $password $data_base_file $backup_base_file
>> $log
echo "Резервное копирование выполнено"
```

Для сохранения файла в редакторе `nano` используется сочетание клавиш **Ctrl+X**, после чего необходимо нажать **Y** и **Enter** для подтверждения сохранения.

Скрипт запускается под пользователем `root` командой `./gbak.sh`.

Журнал изменений можно увидеть в файле `$log`, который указан в переменной `$log`.

Для того чтобы автоматически запускать скрипт `gbak` в ОС Linux, необходимо использовать планировщик задач – CRON.

**CRON** – классический планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определенное время. Регулярные действия описываются инструкциями, помещенными в файлы *crontab* и в специальные директории. Для этого необходимо ввести команду:

```
# crontab -e
```

Это откроет на редактирование таблицу задач планировщика. Чтобы добавить строку, необходимо нажать клавишу **a** и ввести примерно следующий код:

```
### backup databases #####  
00 3 * * * root /opt/admin/gbak.sh
```

Для сохранения нажать **Esc**, затем ввести **wq** и нажать **Enter**.

После сохранения, планировщик будет выполнять ежедневное резервирование в 3 часа 00 минут.

#### 4.2.4.1.3 Резервное копирование с помощью специальных программ

**IBExpert** – графическая программа, предназначенная для разработки и администрирования баз данных InterBase и Firebird, а также для выбора и изменения данных, хранящихся в базах.

Для выполнения резервирования необходимо открыть меню **Службы→Резервирование базы данных:**

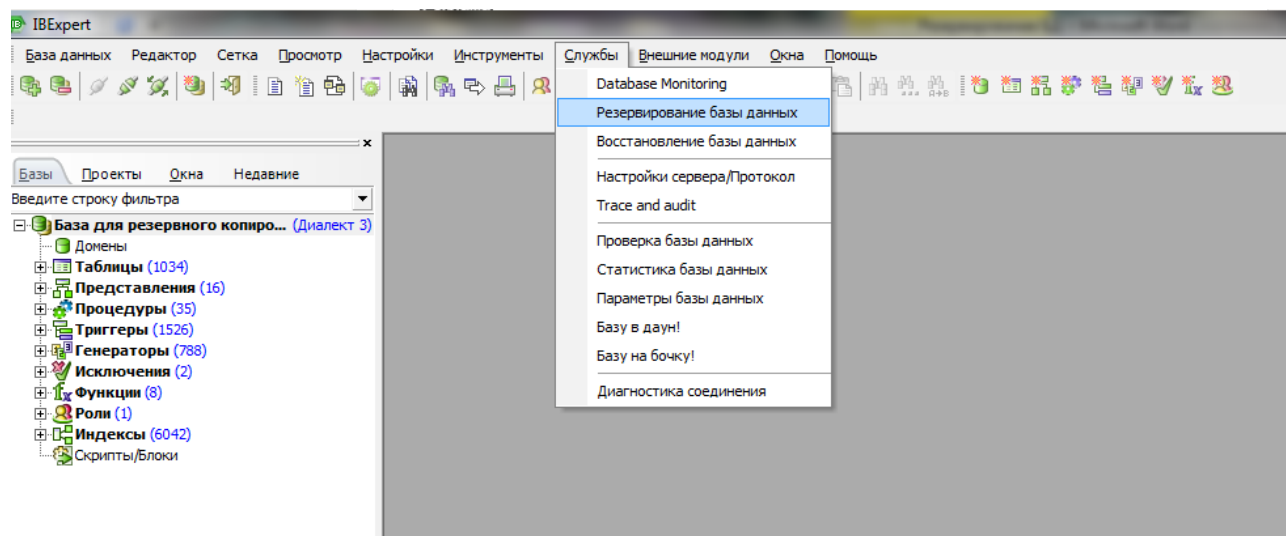


Рисунок 11 – Резервирование базы данных

Откроется окно со следующими настройками:

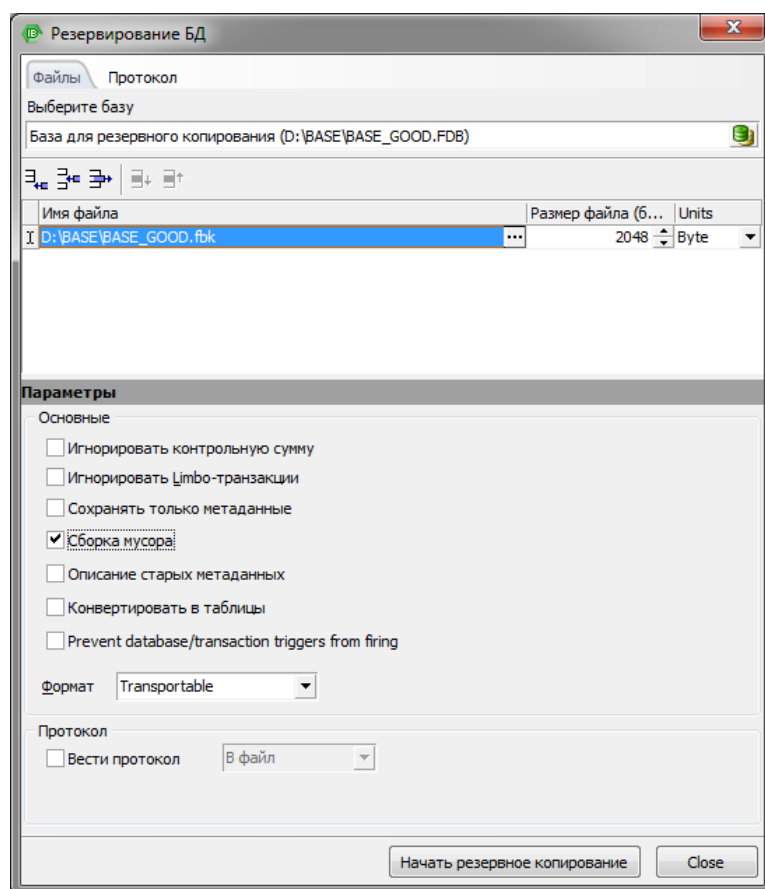


Рисунок 12 – Параметры резервирования базы данных

Для того что бы запустить процесс резервного копирования, необходимо выбрать

базу и указать параметры:

- **Игнорировать контрольную сумму** – позволяет игнорировать ошибки контрольных сумм страниц и не останавливать резервное копирование из-за этих ошибок.
- **Игнорировать limbo-транзакции** – не сохраняет в резервной копии БД версии записей, которые созданы транзакциями, находящимися в состоянии *in limbo*. Такое состояние может быть только у не завершившихся транзакций двухфазного коммита (2PC).
- **Сохранять только метаданные** – данные в резервную копию не сохраняются. Используется, когда необходимо сделать копию пустой БД.
- **Сборка мусора (по умолчанию)** – самый важный параметр. Параметр запрещает серверу проверять читаемые записи на наличие мусорных, что ускоряет процесс бэкапа. Более подробно это [описано в документе](#).
- **Описание старых метаданных** – если параметр включен, то старые описания метаданных включены в базу данных резервного копирования. Это предусмотрено в целях обеспечения совместимости для старых InterBase версий.
- **Конвертировать в таблицы** – параметр преобразует данные базы данных для таблиц в резервной копии. Это касается внешних файлов.
- **Prevent database/transaction triggers from firing** – это равно *isc\_dpb\_no\_db\_triggers* опции в **дополнительных параметрах подключения**.
- **Формат** – выбор формата данных для файла резервной копии базы данных. Рекомендуемое значение по умолчанию – *Переносные*, так как он позволяет восстановить в различные версии Firebird/InterBase, если они **пожелают**. Он сохраняет данные и метаданные в универсальном формате, в отличие от значения *Non-транспортабельные*.
- **Протокол** – включить признак **Вести Протокол**. Протокол необходим для сохранения операции (в файле), которые производились во время резервного копирования базы.

#### 4.2.4.1.4 Планировщик АЦК

В некоторых системах линейки продуктов АЦК (АЦК-Финансы, АЦК-Планирование, АЦК-Госзаказ) для резервного копирования баз FireBird существует задание планировщика задач. Для настройки резервного копирования БД FireBird необходимо в файле конфигурации сервера приложения (далее СП) **azk2server.properties** указать путь до файла **gbak.exe**. Путь к GBAK (только для FireBird):

```
azk.update.backup.gbak=C:/fb/bin/gbak.exe
```

Для автоматизации запуска сервисных задач, таких как внутренние службы, в составе сервера приложений постоянно функционирует фоновый поток планировщика выполнения задач. Его основной задачей является отслеживание расписания запуска заданий и запуск последних в случае, если прошел достаточный интервал времени. Настройка планировщика задач осуществляется из клиентского приложения посредством пункта меню **Справочники**→**Планировщик**.

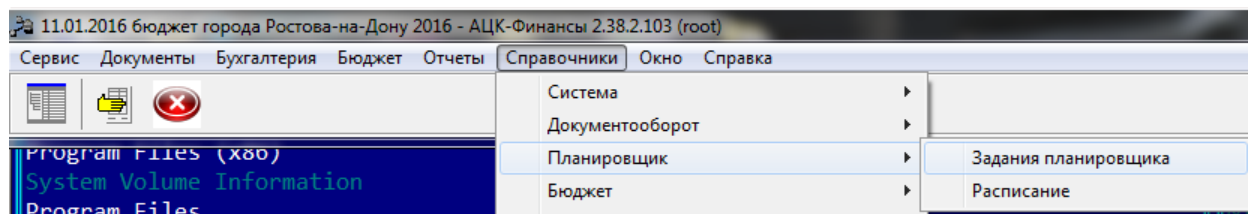


Рисунок 13 – Доступ к заданиям планировщика

В справочнике *Задания планировщика* перечислены процедуры, которые могут быть добавлены в справочник *Расписание планировщика* для выполнения. Существующие способы определения момента запуска процедуры на исполнение позволяют гибко настраивать работу планировщика.

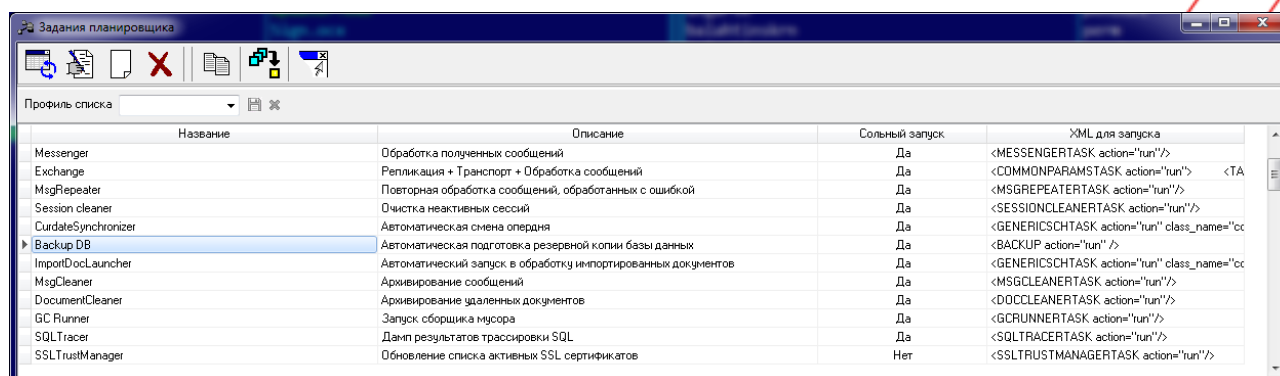


Рисунок 14 – Справочник «Задания планировщика»



При определении задания для планировщика можно ограничить уровень изоляции задания, выключив признак **Запуск на всех серверах приложений**. В этом случае задача будет выполняться только на одном сервере приложений, который входит в кластер.

Наименование	Задание	Запуск	Активно	Время последнего запуска	Время последнего завершения	Параметры	Результат последнего выполнения
Backup DB	Backup DB	каждые 0 дней, 06:00:00	Нет	11.06.2015 16:07:52	11.06.2015 16:27:11	cmd="/opt/firebird/bin/gbak; cmdZip="/usr/bin/gzip -9 4; p <BACKUP completed="true"/>	
ConsReplauncher	ConsReplauncher	ежедневно в 17:29:44	Нет			ProcessClasses=126	
Consistency check	Consistency check	каждые 0 дней, 03:00:00	Нет				
CurdateSynchronizer	CurdateSynchronizer	ежедневно в 00:00:00	Да	11.01.2016 00:00:10	11.01.2016 00:00:15		<GENERICISCHTASK completed="true"/>
DocumentCleaner	DocumentCleaner	ежедневно в 23:30:00	Да	11.01.2016 13:42:10	11.01.2016 13:43:47	path="/deletedDocuments; delete=true; nosave=false	<DOCLEANERTASK completed="true"/>
Exchange	Exchange	каждые 0 дней, 01:05:00	Да	10.05.2016 17:33:32	10.05.2016 17:34:37		<COMMONPARAMTASK completed="true"/>
GC_Runner	GC_Runner	каждые 0 дней, 00:05:00	Нет	21.10.2014 12:54:33	21.10.2014 12:54:38		<GCRUNNERTASK completed="true"/>
GZ_Exchange	Exchange	каждые 0 дней, 00:02:00	Нет			SITE_ID=1	
GZ_Messenger	Messenger	каждые 0 дней, 00:02:00	Да	11.01.2016 16:39:23	11.01.2016 16:39:28	SITE_ID=1	<MESSENGERTASK completed="true"/>
ImportDocLauncher	ImportDocLauncher	каждые 0 дней, 00:05:00	Да	11.01.2016 16:38:10	11.01.2016 16:36:15	OnlyWhenControlsDn=1; ProcessClasses=3.5.31.11.24.4;	<GENERICISCHTASK completed="true"/>
MailCleaner	MailCleaner	ежедневно в 02:00:00	Да	11.01.2016 16:36:10	11.01.2016 12:01:47		<MAILCLEANERTASK completed="true"/>
Messenger	Messenger	каждые 0 дней, 00:05:00	Да	11.01.2016 16:36:54	11.01.2016 16:36:59		<MESSENGERTASK completed="true"/>
MsgCleaner	MsgCleaner	ежедневно в 00:00:00	Да	11.01.2016 00:00:10	11.01.2016 00:00:15	path="/completedMessages; format=true; dispstatus=3; di	<MSGCLEANERTASK completed="true"/>
MsgRepeater	MsgRepeater	каждые 0 дней, 00:30:00	Да	11.01.2016 16:20:48	11.01.2016 16:20:53	CHECK_FAILED	<MSGREPEATERTASK completed="true"/>

Рисунок 15 – Расписание планировщика

Задания планировщика хранятся в таблице SCHEDULE, которая обычно заполняется из файла `./xml/schedule.xml`. В расписании перечислены задания, параметры заданий, время последнего запуска, результаты последнего запуска заданий. Расписание хранится в таблице SCHEDULE.

Наименование	Задание	Запуск	Активно	Время последнего запуска	Время последнего завершения	Параметры	Результат последнего выполнения
Backup DB	Backup DB	каждые 0 дней, 06:00:00	Нет	11.06.2015 16:07:52	11.06.2015 16:27:11	cmd="/opt/firebird/bin/gbak; cmdZip="/usr/bin/gzip -9 4; p <BACKUP completed="true"/>	
ConsReplauncher	ConsReplauncher	ежедневно в 17:29:44	Нет			ProcessClasses=126	
Consistency check	Consistency check	каждые 0 дней, 03:00:00	Нет				
CurdateSynchronizer	CurdateSynchronizer	ежедневно в 00:00:00	Да	11.01.2016 00:00:10	11.01.2016 00:00:15		<GENERICISCHTASK completed="true"/>
DocumentCleaner	DocumentCleaner	ежедневно в 23:30:00	Да	11.01.2016 13:42:10	11.01.2016 13:43:47	path="/deletedDocuments; delete=true; nosave=false	<DOCLEANERTASK completed="true"/>
Exchange	Exchange	каждые 0 дней, 01:05:00	Да	10.05.2016 17:33:32	10.05.2016 17:34:37		<COMMONPARAMTASK completed="true"/>
GC_Runner	GC_Runner	каждые 0 дней, 00:05:00	Нет	21.10.2014 12:54:33	21.10.2014 12:54:38		<GCRUNNERTASK completed="true"/>
GZ_Exchange	Exchange	каждые 0 дней, 00:02:00	Нет			SITE_ID=1	
GZ_Messenger	Messenger	каждые 0 дней, 00:02:00	Да	11.01.2016 16:39:23	11.01.2016 16:39:28	SITE_ID=1	<MESSENGERTASK completed="true"/>
ImportDocLauncher	ImportDocLauncher	каждые 0 дней, 00:05:00	Да	11.01.2016 16:38:10	11.01.2016 16:36:15	OnlyWhenControlsDn=1; ProcessClasses=3.5.31.11.24.4;	<GENERICISCHTASK completed="true"/>
MailCleaner	MailCleaner	ежедневно в 02:00:00	Да	11.01.2016 16:36:10	11.01.2016 12:01:47		<MAILCLEANERTASK completed="true"/>
Messenger	Messenger	каждые 0 дней, 00:05:00	Да	11.01.2016 16:36:54	11.01.2016 16:36:59		<MESSENGERTASK completed="true"/>
MsgCleaner	MsgCleaner	ежедневно в 00:00:00	Да	11.01.2016 00:00:10	11.01.2016 00:00:15	path="/completedMessages; format=true; dispstatus=3; di	<MSGCLEANERTASK completed="true"/>
MsgRepeater	MsgRepeater	каждые 0 дней, 00:30:00	Да	11.01.2016 16:20:48	11.01.2016 16:20:53	CHECK_FAILED	<MSGREPEATERTASK completed="true"/>

Рисунок 16 – Расписание планировщика

Существует несколько моделей задания времени запуска:

- **Один раз** – если требуется установить дату и время запуска, а также при необходимости повтора задания.
- **Ежегодно** – если необходимо включать контроль каждый год.
- **Периодически** – если требуется указать количество дней, то есть промежуток, через которое задание повториться в назначенное время.
- **Ежедневно** – если задание запускается каждый день в назначенное время.

- **Еженедельно** – если задание запускается в назначенное время в определенные дни недели.

Использование фоновых процессов влечет повышение нагрузки на сервер приложений, поэтому все неиспользуемые службы на сервере приложений рекомендуется удалить из расписания планировщика. Процесс работы планировщика отображается в файле журнализации СП (*azk2.log*).

Теперь рассмотрим ближе содержание задание планировщика BACKUP.

Описание задания в расписании XML для запуска:

```
<BACKUP action="run" />
```

**Параметры:** *cmd, cmdZip, path, name.*

**Описание:** *Автоматическая подготовка резервной копии базы данных.*

**Рекомендуемый период запуска:** *3 часа.*

*Процедура предназначена для автоматизированной подготовки резервных копий БД на Firebird для сервера приложений.*

*Принимает следующие параметры:*

- **cmd=<command>** – команда для вызова утилиты командной строки для подготовки резервной копии БД.
- **cmdZip=<command>** – команда для вызова утилиты сжатия программ; может включать в себя дополнительные параметры командной строки.
- **path=<path>** – путь, по которому будут размещаться файлы резервных копий.
- **fbk=<suffix>** – дополнительный суффикс к имени файла резервной копии.

*Пример использования параметров:*

```
cmd=gbak; cmdZip=gzip -9 -f; path=/backup s; fbk=Tomsk
```

*Значения параметров:*

*cmd=gbak, можно указать полное имя файла gbak (по умолчанию gbak).*

*cmdZip=gzip -9 -f, можно указать другой архиватор с валидными ключами (по умолчанию не указан).*

*path=c:\dir, или host:c:\dir .. и т.п. (по умолчанию .\backups).*

*fbk=Tomsk – желательно указывать название объекта реализации, обязательно латинскими буквами.*

*Реализован формат наименования резервной копии БД:*

```
base_X.X.X.X_ГГГГММД Д_ЧЧММСС.fbk,
```

*где:*

- **base** – имя файла, определяется по параметру *name*;

- *X.X.X.X* – версия билда;
- *ГГГГММДД* – текущая дата;
- *ЧЧММСС* – текущее время

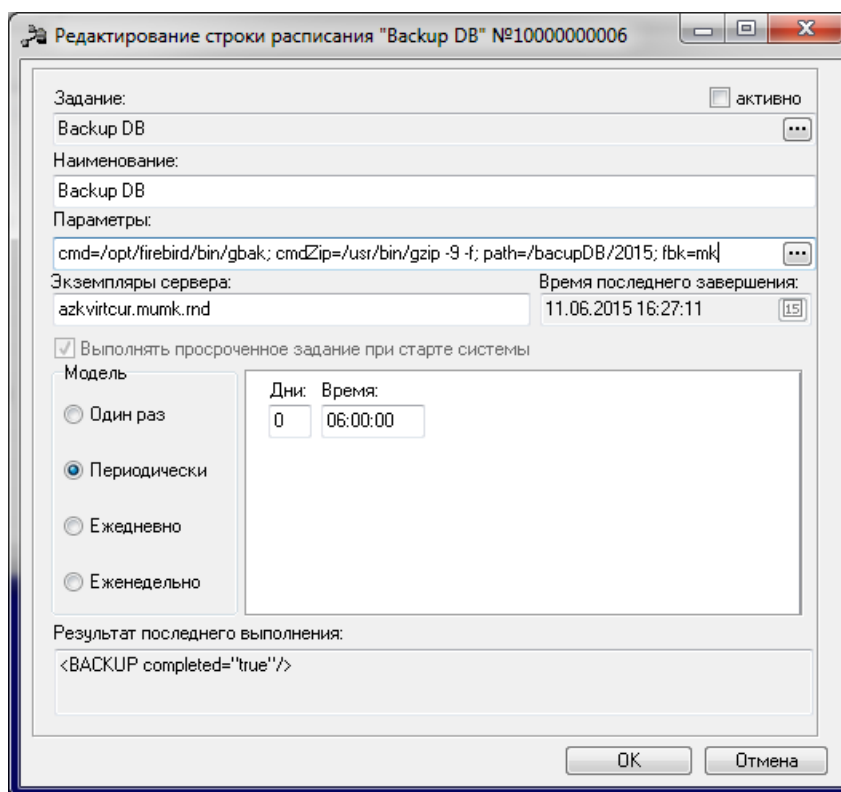


Рисунок 17 – Форма редактирования строки расписания

Можно не задавать ни одного параметра, но архивные копии будут создаваться на том же хосте, что и БД, что крайне нежелательно.

Данный способ рекомендуется для выполнения резервного копирования баз Firebird, т.к. при этом используется утилита **gbak**, входящая в стандартный состав утилит Firebird и планировщик АЦК упрощает с ней работу.

#### 4.2.4.2 Резервирование СУБД «Oracle»

##### 4.2.4.2.1 Резервное копирование с помощью командной строки

Резервное копирование БД Oracle осуществляется утилитой **expdp**. Она входит в пакет стандартных утилит.

Для запуска утилиты **expdp** (в OS Windows утилита располагается в папке C:\Program Files\Oracle\expdp\bin) необходимо выполнить следующие команды:

\Oracle\ora92\bin\expdp, в OS Linux утилита располагается /usr/oracle/bin/expdp), подготовлен скрипт **exportdp.cmd** (для Linux **exportdp.sh**), который предварительно необходимо создать и настроить (данные строки могут отличаться):

- SYSTEM – имя администратора СУБД.
- PASSWD – пароль администратора СУБД.
- URL – путь для развертывания БД.
- ORACLE\_HOME – путь к каталогу Oracle.

Формат вызова утилиты **exportdp**:

```
exportdp <имя схемы> <название файла>
```

Где:

- <имя схемы> – имя экспортируемой схемы;
- <название файла> – имя получаемого файла дампа (необязательный параметр, по умолчанию создается файл с именем, включающим в себя имя схемы, дату и время создания дампа).

#### 4.2.4.2.1.1 Рекомендации для OS Windows

Содержание скрипта **exportdp.cmd**:

```
@echo off
rem -----
set SYSTEM=system
set PASSWD=oracle
set URL=@localhost/xe
set ORACLE_HOME=C:\Oracle\ora92
rem -----
if "%1" == "" goto hlp
set USERNAME=%1
set FILENAME=%2
set LOGFILE=back up.log
for /f "tokens=3 delims=" %%D in ("%DATE%") do set YEAR=%%D
for /f "tokens=2 delims=" %%D in ("%DATE%") do set MONTH=%%D
for /f "tokens=1 delims=" %%D in ("%DATE%") do set DAY=%%D
```

```
for /f "tokens=1 delims=:" %%T in ("%TIME%") do set HOUR=%%T
for /f "tokens=2 delims=:" %%T in ("%TIME%") do set MINUTE=%%T
for /f "tokens=3 delims=:", %%T in ("%TIME%") do set SECOND=%%T
if "%FILENAME%" == "" (
    set FILENAME=%1_%YEAR%-%MONTH%-%DAY%_%HOUR%%MINUTE%%SECOND%.dmp
    set LOGFILE=%LOGFILE
)
if not "%ORACLE_HOME%" == "" (
    set CMD="%ORACLE_HOME%/bin/expdp"
) else (
    set CMD=expdp
)
echo %CMD%
if exist %FILENAME% (
    echo ERROR!
    echo File %FILENAME% is already exist.
    goto end
)
echo -----
echo Backing-up %USERNAME% into %FILENAME% ...
echo -----
set NLS_TIMESTAMP_FORMAT=YYYY-MM-DD HH24:MI:SS
%CMD%          %SYSTEM%/ %PASSWD% %URL%          SCHEMAS=%USERNAME%
DIRECTORY=backup_dir EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS DUMPFILE=%
FILENAME% LOGFILE=%LOGFILE% FLASHBACK_TIME="TO_TIMESTAMP(sysdate)"
goto end
```

После сохранения скрипт запускается в командной строке. Например:

```
exportdp.cmd LNXSTEND23 Inx.dmp
```

Где:

- *LNXSTEND23* – имя схемы.
- *Inx.dmp* – имя файла, в который сохранится резервная копия.

Путь выгрузки файла бэкапа базы данных можно определить, задав переменную, в нашем случае это переменная **backup\_dir**. Сделать это можно следующим образом:

1. заходим на сервер СУБД в **sqlplus**;
2. определяем данную переменную **backup\_dir**:

```
sqlplus /nolog
conn sys/sys@azk11 as sysdba
SQL> create or replace directory backup_dir as 'd:\backup';
Directory created.
```

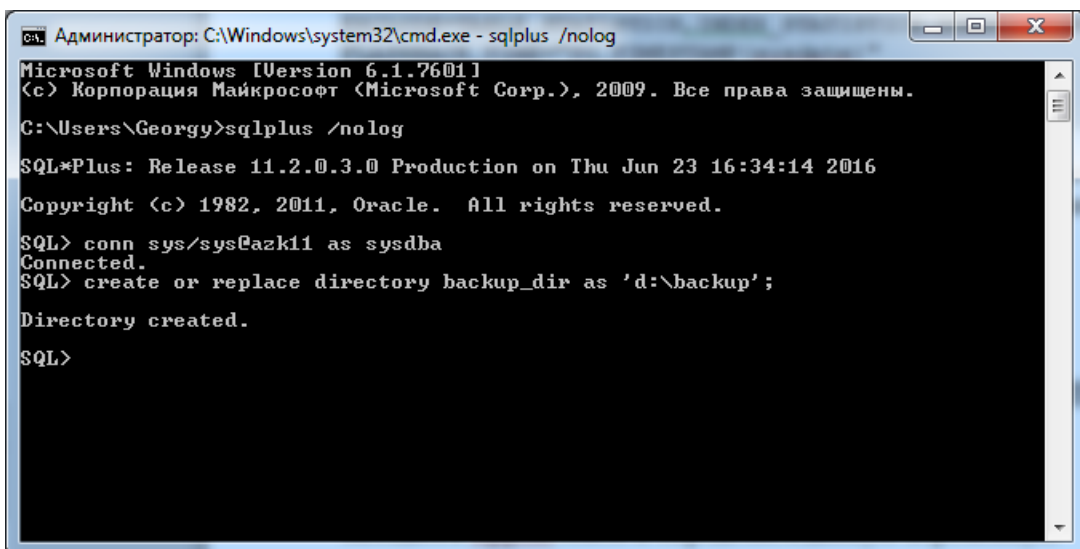


Рисунок 18 – Определение переменной backup\_dir

#### 4.2.4.2.1.2 Рекомендации для OS Linux

Чтобы выполнить резервное копирование на OS Linux, необходимо создать директорию, в которую будут сохраняться резервные копии:

```
# mkdir /expdir
```

Выполнить вход в консоль **sqlplus** от пользователя SYS и создать виртуальное пространство для директории:

```
# sqlplus SYS as SYSDBA
```

После входа в консоль **sqlplus**:

```
1 create directory expdir as '/expdir';
2 select * from dba_directories;
```

Для резервирования рекомендуется использовать отдельного пользователя. Пример

создания пользователя приведен ниже:

```
1 CREATE USER EXPORT
2 IDENTIFIED BY password
3 DEFAULT TABLESPACE USERS
4 TEMPORARY TABLESPACE TEMP
5 PROFILE "UNLIMITED PASSWORD EXPIRATION"
6 ACCOUNT UNLOCK;
7
8 GRANT CREATE SESSION TO EXPORT;
9 GRANT CREATE TABLE TO EXPORT;
10 ALTER USER EXPORT QUOTA UNLIMITED ON USERS;
11 GRANT EXP_FULL_DATABASE TO EXPORT;
12 GRANT READ, WRITE ON DIRECTORY EXPDIR TO EXPORT;
```

Содержание исполняемого файла, который будет запускать процесс резервирования, приведено ниже (ORACLE\_SID – экземпляр базы данных, данное значение может отличаться):

```
$ vi ora_expdp_full.sh
#!/bin/sh
STARTTIME=`date`
export ORACLE_SID=xe
export ORACLE_HOME=`cat /etc/oratab|grep ^${ORACLE_SID}:|cut -d':' -f2`
export EXPLOG=expdp_${ORACLE_SID}.log
export EXPDIR=/expdir
export PATH=$PATH:$ORACLE_HOME/bin
DATEFORMAT=`date +%Y%m%d`
STARTTIME=`date`
expdp export/password content=ALL directory=expdir dumpfile=expdp_`echo $ORACLE_SID`_%
U_`echo $DATEFORMAT`.dmp filesize=2G full=Y logfile=$EXPLOG nologfile=N parallel=2
ENDTIME=`date`
SUBJECT=`hostname -s`:${ORACLE_SID}:`tail -1 $EXPDIR/$EXPLOG`
echo -e "Start time:" $STARTTIME "\nEnd time:" $ENDTIME | mail -s "$SUBJECT"
dba@mydomain.com
```

Скрипт создаст резервную копию размером 2Гб, которая будет динамически

расширяться. Например, первый файл будет называться `expdp_oratst_01_20120503.dmp`, второй `expdp_oratst_02_20120503.dmp` и так далее. После создания вся информация о процессе резервирования будет сохраняться в `log`-файл. Необходимо изменить `ORACLE_SID` и `EXPDIR` переменные в скрипте на свои.

Для сохранения файла нажать **ESC**, ввести `iq` и нажать **Enter**. Скрипту дать права на запуск:

```
# chmod 700 ora_expdp_full.sh
```

Для автоматизации процесса резервирования этот скрипт можно добавить в планировщик:

```
# crontab -e
```

Вписать данную строку (путь к файлу указан для примера):

```
00 23 * * * /home/oracle/scripts/export/ora_expdp_full.sh 1>/dev/null 2>&1
```

**Внимание!** Показателем корректного резервного копирования будет отсутствие ошибок в логе. В противном случае, необходимо обратиться к официальной документации (<https://oracle-base.com/articles/10g/oracle-data-pump-10g>) и исходить из информативности ошибки.

#### 4.2.4.2.2 С помощью программ

**Recovery Manager (RMAN)** – это утилита Oracle, которая используется для управления операциями резервирования, копирования из резерва и восстановления базы данных Oracle. RMAN имеет мощный командный язык, который не зависит от ОС.

**Enterprise Manager Database Control Console** обеспечивают графический пользовательский интерфейс к большинству наиболее используемых возможностей RMAN.

Используя утилиту **Recovery Manager**, можно соединиться со следующими видами баз данных:

- База данных назначения.

Администратор для соединения с БД назначения должен иметь привилегию `SYSDBA`. Эта привилегия необходима для успешного соединения. При этом происходит подсоединение к экземпляру целевой БД, в которой необходимо



выполнить обычные операции RMAN.

- База данных каталога восстановления.

Это необязательная база данных, которая конфигурируется для репозитория RMAN. Соединение с базой данных каталога восстановления устанавливается для получения хранимой в нем информации, например, сведений о резервах или хранимых скриптах.

- Вспомогательная база данных. Такая БД может быть:
  - создана по команде **RMAN DUPLICATE**;
  - временной базой данных, используемой при восстановлении табличного пространства на момент времени в прошлом (*tablespace point-in-time recovery - TSP1TR*);
  - резервной базой данных, представляющей собой копию производственной базы данных и используемой при выходе из строя основной БД.

#### 4.2.4.2.2.1 Запуск RMAN

### Соединение с базой данных назначения без использования каталога восстановления

#### Локальное соединение

Для локального соединения с RMAN необходимо ввести в командной строке ОС:

- в UNIX:

```
# ORACLE_SID=DB01  
# rman target /
```

Где:

- *ORACLE\_SID* – экземпляр базы данных, к которой выполнится подключение;
- *target* – указывает на подключение к целевой базе.

- в Windows:

```
C :> SET ORACLE_SID=DB01  
C :> rman target /
```

Где:

- *ORACLE\_SID* – экземпляр базы данных, к которой выполнится подключение;
- *target* – указывает на подключение к целевой базе.

Если в команде старта не указываются имя и пароль, а только косая черта (*slash*), происходит соединение под пользователем SYS с аутентификацией на уровне ОС. Можно также указать необязательное ключевое слово NOCATALOG:

```
$ rman target / nocatlog
```

Где:

- *NOCATALOG* – режим по умолчанию, означающий использование RMAN без каталога восстановления.

### Удаленное соединение

Чтобы соединиться с другого сервера, используется псевдоним имени службы целевой базы данных в Oracle Net:

```
$ rman target sys/target_pwd@DB01
```

Команда **Backup** – это копия данных из вашей БД, используемая для воссоздания данных. В результате операции резервирования, выполняемой с использованием **RMAN**, могут быть созданы либо копии образов, либо резервные наборы. Для резервирования необходимо выполнить выполнить:

```
RMAN> BACKUP DATABASE;
```

Программа **RMAN** имеет свой встроенный язык, огромные возможности и гибкость в настройке – эти особенности делают **RMAN** мощным и незаменимым инструментом. Утилиту **RMAN** рекомендуется использовать как основное средство резервного копирования.

В официальной документации описано более подробно обо всех возможностях:  
<http://www.oracle.com/technetwork/database/features/availability/rman-overview-096633.html>.

### 4.2.4.3 Резервное копирование СУБД PostgreSQL

PostgreSQL – свободная кроссплатформенная объектно-реляционная СУБД. Но, так как для PostgreSQL в качестве сервера баз данных чаще всего используют ОС Linux, резервное копирования будет рассматриваться только для этой платформы.

#### 4.2.4.3.1 Резервное копирование с помощью командной строки

В комплект PostgreSQL входят утилиты ***pg\_dump***, ***pg\_dumpall*** и ***pg\_basebackup***, они позволяют делать резервные копии.

Запускать утилиты следует от пользователя **postgres**, для этого в системе Linux необходимо в терминале ввести:

```
# su postgres
```

##### 4.2.4.3.1.1 Резервное копирование с помощью утилиты Pg\_Dump

***pg\_dump*** — это программа для создания резервных копий базы данных PostgreSQL. Она создаёт целостные копии, даже если база параллельно используется. Программа ***pg\_dump*** не препятствует доступу других пользователей к базе данных (ни для чтения, ни для записи). Существует несколько режимов работы данной программой, отличающихся форматом полученного дампа, скоростью формирования дампа.

В этом разделе приведены основные возможности по созданию дампа. Более подробно можно почитать тут: <https://postgrespro.ru/docs/postgrespro/9.5/app-pgdump>.

Один из наиболее гибких форматов, позволяющим вручную выбирать и сортировать восстанавливаемые объекты. Вывод в этом формате по умолчанию сжимается.

```
pg_dump -h localhost -p 5432 -U postgres -Fc -v -d f_base_160629 -b -f /backup/f_base_160629.dmp
```

Выгрузить в формате каталога. При этом будет создан каталог, в котором для каждой таблицы и большого объекта будут созданы отдельные файлы, а также файл

оглавления в машинно-читаемом формате, понятном для программы восстановления БД из дампа **pg\_restore**. С полученной резервной копией можно работать штатными средствами ОС Unix/Windows, например, несжатую копию можно сжать посредством архиватора, например: **gzip**, **7z**.

```
pg_dump -h localhost -p 5432 -U postgres -Fd -b -v -d f_base_160629 -j 4 -f /backup/f_base_160629
```

**Внимание!** В этой команде используется ключ **-j**, позволяющий создавать **dump** в несколько потоков, количество которых равно количеству ядер процессоров сервера. **Самый быстрый способ создания дампа!**

Это поведение по умолчанию:

```
pg_dump -h localhost -p 5432 -U postgres -C -Fp -b -v -d f_base_160629 -f f_base_160629.txt
```

Дамп создается в виде архивного файла формата **tar**.

```
pg_dump -h localhost -p 5432 -U postgres -Ft -b -v -d f_base_160629 -f f_base_160629.tar
```

Создание резервной копии с последующим сжатием в **gz**.

```
pg_dump -h localhost -O -Fp -c -U postgres f_base_160629 | gzip -c > f_base_160629.gz
```

#### 4.2.4.3.1.2 Резервное копирование с помощью утилиты Pg\_DumpAll

Утилита **pg\_dumpall** позволяет выполнить резервную копию всего кластера базы данных.

```
pg_dumpall > outfile
```

### 4.2.4.3.1.3 Резервное копирование с помощью утилиты Pg\_BaseBackup

Резервное копирование **pg\_basebackup** подходит для случаев, когда нужно сделать резервную копию целиком всего кластера БД или настроить *hot standby* реплику (начиная с версии 9.3 входит в PostgreSQL). Пример резервирования с **backup** сервера в каталог */backup* (каталог должен быть предварительно создан):

```
pg_basebackup -x -h dbserver -U username -D /backup_dir
```

Резервное копирование **pg\_basebackup** со сжатием:

```
pg_basebackup -D - -Ft | bzip2 > backup.tar.bz2
```

### 4.2.4.3.1.4 Список наиболее часто используемых опций

Ниже приведен список наиболее часто используемых опций:

- **-h host** – хост, если не указан то используется *localhost* или значение из переменной окружения *PGHOST*.
- **-p port** – порт, если не указан то используется *5432* или значение из переменной окружения *PGPORT*.
- **-u** – пользователь, если не указан то используется текущий пользователь, также значение можно указать в переменной окружения *PGUSER*.
- **-a, -data-only** – дамп только данных, по-умолчанию сохраняются данные и схема.
- **-b** – включать в дамп большие объекты (*blobs*).
- **-s, -schema-only** – дамп только схемы.
- **-C, -create** – добавляет команду для создания БД.
- **-c** – добавляет команды для удаления (*drop*) объектов (таблиц, видов и т.д.).
- **-O** – не добавлять команды для установки владельца объекта (таблиц, видов и т.д.).
- **-F, -format {c|d|t|p}** – выходной формат дампа, *custom*, *directory*, *tar*, или *plain text*.
- **-t, -table=TABLE** – указываем определенную таблицу для дампа.

- **-v, -verbose** – вывод подробной информации.
- **-D, -attribute-inserts** – дамп используя команду *INSERT* с списком имен свойств.

#### 4.2.4.3.2 Резервирование с помощью пакетного файла (скрипта)

Для автоматизации действий, следует использовать планировщик задач и пакетный файл (скрипт). Любую из перечисленных выше команд можно добавить на ежедневное, еженедельное, ежемесячное выполнение с помощью планировщика **cron**.

CRON – классический планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определенное время. Регулярные действия описываются инструкциями, помещенными в файлы *crontab* и в специальные директории. Для этого необходимо ввести команду:

```
# crontab -e
```

Это откроет на редактирование таблицу задач планировщика. Чтобы добавить строку, необходимо нажать клавишу **a** и ввести примерно следующий код:

```
### backup databases #####  
00 3 * * * root pg_basebackup -D - -Ft | bzip2 > backup.tar.bz2
```

Для сохранения нажать **Esc**, затем ввести *iq* и нажать **Enter**.

После сохранения, планировщик будет выполнять ежедневное резервирование в 3 часа 00 минут.

#### 4.2.4.4 Проверка качества созданной резервной копии БД

Вне зависимости от СУБД необходимо раз в месяц (можно и чаще) выполнять проверку качества созданной резервной копии БД, для этого необходимо восстановить дамп. Если во время восстановления возникли ошибки, то ситуация явно вышла из-под контроля и необходимо провести ревизию системы резервного копирования данных.

#### 4.2.4.4.1 Резервное восстановление СУБД FireBird

Восстановление дампа осуществляется утилитой **gbak.exe (gbak.sh)**. Пример строки для запуска резервного восстановления дампа приведен ниже:

```
gbak.exe -c -v -user SYSDBA -pas masterkey c:\test\test.bak c:\test\db\firebird\test_new.fbd
```

Где:

- **-c** – ключ для создания БД.
- **-v** – вывод на экран лога восстановления БД из дампа.
- **-user** – пользователь БД.
- **-pas** – пароль пользователя БД.
- **c:\test\test.bak** – полный путь дампа.
- **c:\test\db\firebird\test\_new.fbd** – полный путь до восстановленной БД.

#### 4.2.4.4.2 Резервное восстановление СУБД Oracle

Для СУБД Oracle можно выделить два варианта восстановления дампа:

- с помощью утилиты **impdp**;
- с помощью утилиты **RMAN**.

##### 4.2.4.4.2.1 Резервное восстановление дампа с помощью утилиты impdp

Восстановление дампа осуществляется утилитой импорта **impdp**. Пример строки для запуска резервного восстановления дампа приведен ниже:

```
impdp system/Pa$$w0rd SCHEMAS=OracleUser directory=ExportImport  
dumpfile=DumpSCHEMAS.dmp logfile=ImportSCHEMAS.log
```

Где:

- **system/Pa\$\$w0rd** – это логин и пароль пользователя в СУБД;
- **SCHEMAS=OracleUser** – параметр, который указывает, что мы хотим импортировать конкретную схему (в нашем случае OracleUser);
- **directory=ExportImport** – параметр указывает директорию, в которой расположен файл дампа данных;

- **dumpfile=DumpSCHEMAS.dmp** – параметр для указания названия дампа файла;
- **logfile=ImportSCHEMAS.log** – параметр для указания названия лог файла импорта данных.

#### 4.2.4.4.2 Резервное восстановление дампа с помощью утилиты RMAN

Соединение с локальной/сетевой БД ранее описано в разделе [Запуск RMAN](#)<sup>[73]</sup>.

1. Для восстановления данных целевая БД должна находиться в состоянии NOMOUNT/ MOUNT/ OPEN в зависимости от характера восстановления, например:
  - NOMOUNT – для восстановления контрольных файлов БД (фактически – СУБД).
  - MOUNT – для восстановления БД целиком или табличного пространства SYSTEM.
  - OPEN – для восстановления табличных пространств, помимо SYSTEM (в этом случае перед процедурой восстановления само табличное пространство потребуется перевести в состояние OFFLINE).
2. Восстановление файлов (с данными и служебных) выполняется в **RMAN** командой **RESTORE**.

Восстановление данных выполняется либо в **RMAN**, либо в SQL\*Plus командами **RECOVER** при условии наличия восстановленных файлов. Рассмотрим некоторые примеры восстановления:

- Восстановление до момента сбоя («последнего момента»).

Некоторые примеры восстановления:

```
RMAN> RECOVER DATABASE;  
RMAN> RECOVER TABLESPACE users;  
RMAN> RECOVER DATAFILE 'd:\oracle\oradata\teacher\users01.dbf';  
RMAN> RESTORE CONTROLFILE;RMAN> RUN {> SET ARCHIVELOG DESTINATION TO 'd:  
\oracle\oradata\archive';  
3> RESTORE ARCHIVELOG ALL; }
```



- Восстановление пространств, закрытых на запись.

```
RMAN> SQL "ALTER TABLESPACE lookup_data OFFLINE";  
RMAN> RECOVER TABLESPACE lookup_data;  
RMAN> SQL "ALTER TABLESPACE lookup_data ONLINE";
```

- Восстановление до указанного момента в прошлом.

БД, работающую в режиме архивирования журнала, можно восстанавливать до определенного указанного момента с помощью фраз UNTIL {TIME ... | SCN ... | SEQUENCE ... THREAD...}.

```
Пример. RMAN> RESTORE DATABASE;      # восстановили файлы  
RMAN> RECOVER DATABASE UNTIL SCN 375831; # восстановили БД  
RMAN> ALTER DATABASE OPEN RESETLOGS;  # сбросили журнал
```

- Восстановление БД (вторая и третья строчки выше) можно выполнить и в SQL\*Plus:

```
SQL > RECOVER DATABASE UNTIL CANCEL;  
SQL> ALTER DATABASE OPEN RESETLOGS;
```

При таком восстановлении необходимо сбросить онлайн-журнал. После этого, как и при традиционном восстановлении со сбросом журналов (RESETLOGS), необходимо снять полную копию БД, так как с этого момента восстановление с более ранних резервных копий станет невозможным из-за того, что история журнальных записей прерывается.

Более подробно с описанием возможностей, функций и команд можно ознакомиться в официальной документации

<http://www.oracle.com/technetwork/database/features/availability/rman-overview-096633.html>.

#### 4.2.4.4.3 Резервное восстановление СУБД PostgreSQL

Дампы PostgreSQL восстанавливаются с помощью утилиты **pg\_restore**.

В данном разделе приведены основные возможности по созданию дампа. Более подробно можно почитать тут: <https://postgrespro.ru/docs/postgrespro/9.5/app-pgrestore>

**Внимание!** Для форматов **Custom**, **Directory** можно и нужно применять восстановление в несколько потоков.

Перед началом выполнения процедуры восстановления может понадобиться создание БД. Для этого необходимо выполнить команду:

```
CREATE DATABASE newdb
```

#### 4.2.4.4.3.1 Резервное восстановление дампа с использованием формата «tar»

Пример строки для запуска резервного восстановления дампа приведен ниже:

```
pg_restore -d newdb db.dump
```

Где:

- **-d** – ключ, подключиться к базе данных *newdb* и восстановить данные непосредственно в неё;
- **db.dump** – дамп БД.

#### 4.2.4.4.3.2 Резервное восстановление дампа с использованием формата «Custom»

Пример строки для запуска резервного восстановления дампа приведен ниже:

```
pg_restore -h localhost -U postgres -Fc -j 4 -e -v -d newdb db.dump
```

Где:

- **-v** – вывод подробной информации;
- **-d** – ключ, подключиться к базе данных *newdb* и восстановить данные непосредственно в неё;
- **-e** – завершать работу в случае возникновения ошибки при выполнении команд SQL в базе данных. По умолчанию процесс восстановления продолжается, а по его окончании выдаётся число ошибок;
- **-j** – запустить наиболее длительные операции *pg\_restore* (в частности, загрузку данных, создание индексов или ограничений) в нескольких параллельных заданиях. Это позволяет кардинально сократить время восстановления большой базы данных, когда сервер работает на многопроцессорном компьютере. Каждое задание выполняется в отдельном задании или потоке, в зависимости от операционной системы, и использует отдельное подключение к серверу. Оптимальное значение этого параметра зависит от аппаратной конфигурации сервера, клиента и сети. В частности, имеет значение количество

процессорных ядер и устройство дискового хранилища. В качестве начального значения можно выбрать число ядер на сервере, но и при увеличении этого значения во многих случаях восстановление будет быстрее. Конечно, при слишком больших значениях производительность упадет из-за перегрузки;

- **db.dump** – дамп БД.

#### 4.2.4.4.3.3 Резервное восстановление дампа с использованием формата «Directory»

Пример строки для запуска резервного восстановления дампа приведен ниже:

```
pg_restore -h localhost -U postgres -Fd -j 4 -e -v -d newdb db.dump
```

Где:

- **-v** – вывод подробной информации.
- **-d** – ключ, подключиться к базе данных **newdb** и восстановить данные непосредственно в неё.
- **-e** – завершать работу в случае возникновения ошибки при выполнении команд SQL в базе данных. По умолчанию процесс восстановления продолжается, а по его окончании выдаётся число ошибок.
- **-j** – запустить наиболее длительные операции **pg\_restore** (в частности, загрузку данных, создание индексов или ограничений) в нескольких параллельных заданиях. Это позволяет кардинально сократить время восстановления большой базы данных, когда сервер работает на многопроцессорном компьютере. Каждое задание выполняется в отдельном задании или потоке, в зависимости от операционной системы, и использует отдельное подключение к серверу. Оптимальное значение этого параметра зависит от аппаратной конфигурации сервера, клиента и сети. В частности, имеет значение количество процессорных ядер и устройство дискового хранилища. В качестве начального значения можно выбрать число ядер на сервере, но и при увеличении этого значения во многих случаях восстановление будет быстрее. Конечно, при слишком больших значениях производительность упадёт из-за перегрузки.
- **db.dump** – дамп БД.

### 4.3 Установка «Sentinel System Driver»

Установка драйвера аппаратного ключа **Sentinel System Driver** производится на машину, на которой будет работать сервер приложений в роли сервера лицензий.

Для установки надо выбирать машину, которая гарантировано будет включена при запуске других серверов приложений. Если машина с сервером лицензий будет выключена в момент запуска серверов приложений, то они будут запущены в демонстрационном режиме с ограниченной функциональностью.

Установка драйверов выполняется по инструкции производителя аппаратных ключей.

### 4.4 Установка «CryptoPro-CSP»

Криптопровайдер **CryptoPro-CSP** разработан ООО «Крипто-Про» в соответствии с криптографическим интерфейсом фирмы **Microsoft – Cryptographic Service Provider (CSP)**.

Криптопровайдер **CryptoPro-CSP** используется для ключей электронной подписи и имитозащиты данных, обеспечения целостности и подлинности информации не содержащей сведений, составляющих государственную тайну.

Установку **CryptoPro-CSP** необходимо выполнить на машинах, на которых предполагается заверять документы ЭП и на одном сервере приложений, который будет проверять ЭП полученных документов.

Документацию по установке можно получить из комплекта поставки **CryptoPro-CSP** или по адресу <http://www.cryptopro.ru/cryptopro/documentation/default.htm>.

## 4.5 Установка системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ»

### 4.5.1 Установка в операционной системе «Linux»

Для установки и настройки системы необходимо наличие установленных пакетов: **chkconfig**, **service**, **awk**.

Установка системы в операционной системе **Linux** может выполняться в ручном и автоматическом режиме.

#### 4.5.1.1 Установка системы в ручном режиме

Чтобы установить систему в ручном режиме, необходимо выполнить следующие действия:

1. Войти в систему под учетной записью с правами *root*.
2. Создать каталог для сервера приложений:

```
#mkdir /usr/local/GZ
```

3. Распаковать файлы сервера приложений из архива в дистрибутиве:

```
#unrar x -y <путь_к_архиву>/server.rar /usr/local/GZ
```

```
#unrar x -y <путь_к_архиву>/etc.rar /usr/local/GZ
```

4. Установить права доступа к файлам:

```
#chown -Rc root:root /etc.rar /usr/local/GZ
```

```
#chmod -Rc *.jar 755 #find /usr/local/GZ -iname *.dll -exec chmod -Rc 755 {} \;
```

```
#find /usr/local/GZ -iname start -exec chmod -Rc 755 {} \;
```

```
#find /usr/local/GZ -iname Server.properties -exec chmod -Rc 644 {} \;
```

5. Изменить параметры запуска сервера приложений в файле **Server.properties**.
6. Указать параметры доступа к БД:

```
azk.db.url=jdbc:firebirdsql:gzdb/3050:/var/db/basename.fdb
```

```
jdbc.drivers=org.firebirdsql.jdbc.FBDriver
```

```
azk.db.user=SYSDBA azk.db.password=masterkey
```

```
azk.db.accessmode=interbase
```

**Примечание.** Указание порта **3050** в переменной **azk.db.url** является необходимым условием для корректной работы задания планировщика **Backup**.

7. Указать настройки для проверки ЭП:

```
azk.sign.hostname=192.168.0.7  
azk.sign.port=2001  
azk.sign.user=singer  
azk.sign.pswd=
```

8. Указать настройки для ключа лицензий:

```
azk.license.typeinit=SERVER  
azk.license.path=192.168.0.7:2001
```

9. Если на сервере несколько сетевых интерфейсов, то указывается: *адрес:порт интерфейса (доступного клиенту)*. Формат:

```
iiop.alternateAddr.endpoint[номер]=[ip адрес]:[порт]  
iiop.alternateAddr.endpoint1=192.168.0.1:2001
```

10. Настроить автоматический запуск сервера приложений. Создать скрипт автозапуска. В каталоге **/etc/etc/rc.d/init.d** создать файл **gzapp**:

```
#touch /etc/etc/rc.d/init.d/gzapp
```

**Пример.** Файл автозапуска сервера приложений:

```
#!/bin/sh  
# gzapp - starts the goszakaz application server  
#  
# processname: java  
# pidfile: /var/run/gzapp.pid  
# Andrey Pavlovski  
# Source function library.  
. /etc/init.d/functions  
RETVAL=0  
PATH_GZ=/usr/local/GZ/cur  
PIDFILE=/var/run/gzapp.pid  
HOST=gzdb  
export SERVER_CONFIG=Server.properties
```

```
export SERVER_PORT=2001
export JAVA_HOME=/usr/java/jdk 1.8_40
export LANG=ru_RU.CP1251
export LANGUAGE=ru_RU.CP1251
start()
{
  if [ -s "$PIDFILE" ]; then
    action "pid file APP server GOSZAKAZ exist" "true"
  else
    cd $PATH_GZ
    echo "Waiting $HOST..."
    ping $HOST -c 1 >/dev/null
    R="$?"
    while [ "$R" != 0 ]; do
      ping $HOST -c 1 >/dev/null
      R="$?"
      sleep 10
    done
    sleep 30
    action "Start APP GOSZAKAZ" $PATH_GZ/server/bin/start
    sleep 3
    ps x|grep java| grep $PATH_GZ|awk '{print $1}>$PIDFILE
  fi
  RETVAL=$?
  return $RETVAL
}
stop()
{
  msg_stopping $"Appication server GOSZAKAZ"
  action "Stop APP server GOSZAKAZ" "kill `ps x|grep java| grep $PATH_GZ|awk '{print $1}'`"
  sleep 3
  action "Foce stop APP server GOSZAKAZ" "kill -9 `ps x|grep java| grep $PATH_GZ|awk '{print
$1}'`"
  rm -f $PIDFILE
  RETVAL=$?
}
```

```
    return $RETVAL
}
restart()
{
    stop
    sleep 3
    start
}
# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    status)
#   status --expect-user root -- acpid
        RETVAL=$?
        ;;
    *)
        msg_usage "${0##*/} {start|stop|restart|status}"
        RETVAL=1
esac
exit $RETVAL
```

11. Настроить вызов скрипта автозапуска:

```
#chkconfig gzapp on
```



### 4.5.1.2 Автоматическая установка системы

Для автоматической установки системы необходимо задать:

- размещение упакованных файлов дистрибутива системы: *[текущий каталог ./]*;
- каталог сервера системы: *[/usr/local/GZ]*;
- каталог лог-файлов: *[/var/log/GZ]*;
- путь к каталогу **JAVA-машины**: *[/usr/java/jdk1.8]*;
- адрес сервера БД: *[gzdb]*.

## 4.5.2 Установка в операционной системе «Windows»

### 4.5.2.1 Настройка ролей серверов приложений

Настройки выполняются в файле **Server.properties**:

#### 1. Сервер приложений в роли ЭП сервера.

Сервер может самостоятельно обращаться к средствам криптозащиты для проверки правильности ЭП, полученных документов. Для этого на машине с этим сервером приложений должен быть установлен пакет **CryptoPro-CSP**.

Сервер может осуществлять проверку документов переданных ему другими серверами приложений и возвращать им результат проверки.

Сервер приложений в роли сервера ЭП не настраивается. При организации кластера серверов приложений в файлах свойств серверов приложений, не предназначенных для проверки ЭП, необходимо в разделе настроек «Параметры ЭП» ввести следующие параметры (настройки выполняются в файле **Server.properties**):

```
azk.sign.hostname==<IP-адрес или сетевое имя>
```

```
azk.sign.port==<порт>
```

```
azk.sign.user=<пользователь>
```

```
azk.sign.pswd=<пароль пользователя>
```

*Примечание.* Для проверки ЭП в сервере приложений существует специальная учетная запись для осуществления проверки ЭП – **signer** с пустым паролем (пароль можно сменить).

## 2. Сервер приложений в роли сервера лицензий.

При наличии аппаратного ключа сервер приложений может проверить соответствие файла лицензии. При отрицательном результате сервер запустится в демонстрационном режиме.

Сервер может отвечать на запросы других серверов приложений о наличии лицензии и ее соответствии аппаратному ключу. Для настройки сервера приложений с аппаратным ключом в файле **Server.properties** выполняются следующие настройки:

```
azk.license.typeinit=FILE  
azk.license.path=/path_to_file_Licence/file.lic
```

Если организуется кластер серверов приложений, то в файлах свойств других серверов приложений, входящих в кластер, необходимо ввести следующие параметры:

```
azk.license.typeinit=SERVER  
azk.license.path=<app_server_ipaddress>:<app_server_port>
```

### **Пример:**

```
azk.license.path=162.168.0.2:2001
```

*Внимание!* Сервер в роли сервера лицензий должен запускаться первым из всех серверов приложений.

Альтернативные строки параметров следует закомментировать поставив знак «#» в начале строки.

## 4.5.2.2 Настройка файлового хранилища

### 4.5.2.2.1 Файловое хранилище «АЦК-Госзаказ»/«АЦК-Муниципальный заказ»

Осуществлением поддержки прикрепляемых файлов к документам занимается класс, поддерживающий интерфейс **AttachSupport**. Класс задается параметром **azk.docattach.class** в файле **Server.properties**.

По умолчанию (если не указано другое) используется старый класс **com.bssys.server.docflow.system.DocAttach**, неподдерживающий новый механизм хранения данных, основанный на хранилищах данных.

Альтернативная реализация аттачей поддерживает новый механизм хранения в файловой системе. Также она поддерживает хранение любых атрибутов файлов (заранее сконфигурированных в исходном коде путем наследования базового класса). Реализация лежит в классе **com.bssys.server.system.DocAttachEx**.

Механизм управления хранилищами данных основан на менеджере устройств, который управляет различными устройствами, указанными в файле конфигурации устройств. В файле указывается информация вида:

```
<STORAGES>
  <STORAGE name="File Storage" path="c:\test"
    class="com.bssys.server.system.storage.DateFolderFileDataStorage"
    read-only="false" default="true"/>
</STORAGES>
```

где:

- **name** – уникальное имя устройства, которое запоминается для каждого присоединенного к документу файла для того, чтобы в последствии система могла его найти;
- **class** – имя класса-обработчика данного устройства, класс реализует интерфейс **DataStorage**;
- **read-only** – признак того, что устройство только для чтения (нельзя удалять файлы и записывать новые);

- **default** – признак того, что устройство является активным, т.е. все новые файлы будут записываться на него. Активным может быть только одно устройство, все остальные параметры являются специфичными для каждого устройства. Имя файла с описанием устройств задается в параметре **azk.storage.conf**, например, **azk.storage.conf=k:\lazk2\storages.xml**.

#### 4.5.2.2 Совместное файловое хранилище «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» и «АЦК-Финансы»

Функционал позволяет двум взаимодействующим системам использовать одно файловое хранилище. Совместное файловое хранилище позволяет выгружать документы с прикрепленными файлами из одной системы в другую одним сообщением, которое содержит, помимо реквизитов документа, ссылку с наименованием хранилища, откуда выгружается документ и путь к прикрепленному файлу внутри этого хранилища.

Система-отправитель – система АЦК, выгружающая данные.

Система-получатель – система АЦК, принимающая данные.

На настройки необходимо выполнить следующие шаги:

1. Необходимо организовать доступ на просмотр для пользователей **Системы-получатель** к файлам хранилища **Системы-отправитель**, т.е. к каталогам прикрепленных к документам файлов, выгружаемым из **Системы-отправитель** в **Систему-получатель**.
2. В файле описания хранилищ (**storages.xml**) **Системы-получатель** необходимо описать все хранилища **Системы-отправитель** из которых могут выгружаться прикрепленные файлы. При этом необходимо учесть, что имя хранилища **Системы-отправитель** должно быть уникальным среди хранилищ **Системы-получатель**.

**Пример.** Настройки **storages.xml** системы «АЦК-Финансы» для взаимодействия с системой «АЦК-Госзаказ»/«АЦК-Муниципальный заказ»:

```
<GZDOCATTAACH external-name="DOCATTACH" path="\\server-gz\storages\fs_DocAttach">  
<DEVICE name="File Storage"  
caption="Файловая система"
```

```
path=".\\storages\\fs_DocAttach"  
class="com.bssys.server.system.storage.FileDataDevice"  
read-only="false"/>  
</GZDOCATTACH>
```

где:

- **external-name** – имя хранилища **Системы-отправитель**;
- **path** – путь к директории хранилища **Системы-отправитель**.

Настройки DEVICE для работы данного механизма не изменяются (остаются такими, как и были при настройке файлового хранилища раньше).

### 3. Выполнить файл **unload\_with\_doc.xml**:

```
<?xml version="1.0" encoding="windows-1251"?>  
<REFERENCE ref_name="SYSPARAM" action="synchronize_update">  
  <SYSPARAM NAME="unload.attaches.with.doc" CAPTION="Выгрузить ли сами файлы с  
документами">  
  <PARAM_VALUE><![CDATA[0]]></PARAM_VALUE>  
</SYSPARAM>  
</REFERENCE>
```

с параметрами:

- <![CDATA[0]]> – включить настройку;
- <![CDATA[1]]> – отключить настройку.

**Примечание.** После включения настройки, при выгрузке документа с прикрепленным файлом будет сформировано одно сообщение из **Системы-отправитель** в **Систему-получатель** в **xml-формате**, содержащее теги с реквизитами документа, описывающие бизнес-часть документа, а также теги, содержащие информацию с наименованием хранилища **Системы-отправитель** и путем к прикрепленному файлу внутри хранилища.

**Внимание!** Данный механизм работает лишь при условии, что есть одна и только одна **Система-отправитель** для текущей **Системы-получатель**.

### 4.5.3 Настройка запуска комплекса серверов приложений

Порядок запуска системы:

1. сервер БД;
2. сервер приложений с ролью сервера лицензий;
3. сервер приложений с ролью сервера ЭП;
4. остальные серверы приложений.

Порядок остановки системы:

1. серверы приложений;
2. сервер БД.

Предусмотрена возможность настройки скриптов для автоматического запуска системы. Для этого необходимо обеспечить соблюдение порядка загрузки и остановки серверов.

### 4.5.4 Настройка базы данных

Администратор БД должен быть ознакомлен с общими принципами работы конкретной СУБД. Для ознакомления с общими принципами работы СУБД необходимо изучить справочную документацию на СУБД. Администратор должен уметь устанавливать и настраивать СУБД.

Работа администратора БД системы подразумевает следующие действия:

- создание БД;
- первоначальное формирование структуры БД;
- обновление структуры БД;
- создание резервных и архивных копий БД;
- восстановление БД из резервных копий (при необходимости);
- оптимизация БД средствами СУБД (по данному вопросу следует обращаться к справочному руководству конкретной СУБД).

#### 4.5.4.1 Создание базы данных

При создании БД формируется первоначальная структура БД: *таблицы БД, хранимые процедуры, триггеры и т.д.*

##### 4.5.4.1.1 Создание базы данных под управлением СУБД Firebird

СУБД **Firebird** может работать под управлением ОС **Win32** и **Linux**. После установки СУБД на сервер БД настоятельно рекомендуется сменить используемый по умолчанию пароль администратора СУБД (в ОС **Win32** это учетная запись **SYSDBA** с паролем **masterkey**).

Чтобы создать БД (СУБД должна работать), необходимо выполнить следующие действия:

1. Открыть файл свойств сервера приложений **Server.Properties**.
2. В файле указать путь к создаваемой БД:

```
azk.db.url=jdbc:firebirdsql:localhost/3050:D:/base/test.fdb
```

*Примечание. При размещении СУБД на сервере под управлением Win32 в строке URL в указании места расположения файла БД в качестве разделителя следует использовать знак «\» (двойной обратный слеш), либо «/» (одинарный прямой слеш). Это связано с особенностями sql-утилиты, реализованной на Java и вызываемой скриптом первоначального формирования БД в ОС Win32. Далее в документе будет только указываться, реализована ли утилита (или приложение) при помощи Java. Соответственно следует учесть особенности указания расположения файлов. Для ОС Linux в качестве разделителя используется знак «/» (одинарный прямой слеш). Если порт 3050 не будет указан, то задание планировщика Backup работать не будет.*

3. В файле указать имя и пароль пользователя БД с неограниченными правами:

```
azk.db.sysuser=SYSUSER  
azk.db.syspassword=masterkey
```

4. Закрыть файл свойств сервера приложений, при этом необходимо сохранить изменения в файле.
5. В папке **/SQL** выполнить **sql-скрипт**.

*executer.cmd create\_user*

После выполнения sql-скрипта создается файл пустой БД (схема БД).

## 6. В папке **/SQL** выполнить **sql-скрипт**:

*perform.cmd <номер сайта>*

При создании БД с главным сайтом вместо *<номер сайта>* указывается значение 0. Если создается БД с подчиненным сайтом, то вместо *<номер сайта>* указывается любая цифра, кроме 0.

После выполнения **sql-скрипта** формируется структура БД.

Последовательность выполнения файлов с **sql-командами** определена файлом **/SQL/perform.lst**. Журнал обработки этих файлов (результат выполнения **sql-команд**) записывается в **order\_ib.log**.

*Примечание.* Для формирования БД необходимо и достаточно иметь каталоги **/SQL** и **./server** из поставляемой сборки на ПК, на котором будут запускаться формирующие скрипты.

## 7. Запустить сервер приложений и выполнить один из **xml-скриптов**:

*xml.cmd@all.lst – выполняется для БД с главным сайтом;*

*xml.cmd@slaveall.lst – выполняется для БД с подчиненным сайтом.*

Положительным выполнением скрипта является ответ **RPLRULE** (см. наличие **RplRule.xml** в списке **@all.lst** или **@slaveall.lst**). Если скрипт завершил работу с ошибкой, то необходимо перезапустить сервер приложений и повторить запуск скрипта **xml.cmd**.

*Внимание!* Предусмотрена возможность запуска скрипта с ПК, отличного от того, на котором установлен сервер приложений. Перед запуском скрипта необходимо выставить необходимые параметры в файле свойств **/XML/xml.properties**:

*azk.user=root (имя администратора системы по умолчанию);*

*azk.password=toor (пароль администратора системы по умолчанию);*

*azk.host=localhost (IP-адрес или сетевое имя сервера, где работает сервер приложений);*

*azk.port=2001 (порт, на котором сервер приложений принимает запросы, по умолчанию 2001).*



Перед началом работы с программой перезапустить сервер приложений. Завершение работы сервера приложений выполняется следующим образом:

1. закрывается консольное окно, либо сочетанием клавиш «**Ctrl+C**» для ОС **Win32**;
2. завершается работа **java-процесса** при помощи команды **kill** или **killall java** для ОС **Linux**.

Для формирования некоторых справочников стандартными и поставляемыми по умолчанию данными необходимо из папки **/XML/Example** запустить скрипт **xml.cmd** со следующими параметрами:

```
>xml.cmd @std.lst  
>xml.cmd @kbk.lst  
>xml.cmd @ex.lst
```

Можно посмотреть содержимое каждого **lst-файла** и **xml-файла**.

*Примечание.* Команда **xml.cmd @ex.lst**, помимо типовых данных, заполнит БД поставляемыми по умолчанию организациями. Организации можно в дальнейшем удалить или изменить.

#### 4.5.4.1.2 Создание схемы базы данных под управлением СУБД Oracle

СУБД **Oracle** может работать под управление ОС **Win32** и **Linux**. Вопрос установки и первоначальной настройки СУБД следует рассматривать в справочном руководстве **Oracle**.

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» в терминах СУБД **Oracle** использует в качестве базы для хранения данных специально созданную схему.

Для создания схемы БД необходимо выполнить следующие действия:

1. создать дополнительное табличное пространство (**Tablespace**);
2. задать имя табличного пространства;
3. файлу табличного пространства назначить начальное значение **25 Мб**, автоинкремент **8192 Кб**, безлимитное увеличение;

4. создать учетную запись пользователя и схему для хранения и доступа к данным системы (схема создается автоматически при создании объектов схемы под данным учетной записи пользователя);
5. создать учетную запись пользователя;
6. присвоить принадлежность к табличному пространству:

*Default = <Имя табличного пространства>*

*Temp = Temp*

присвоить роли:

*CONNECT, RESOURCE, SELECT\_CATALOG\_ROLE*

присвоить системные роли:

*UNLIMITED TABLESPACE*

присвоить объектную роль:

*SELECT ON "SYS"."V\_\$TRANSACTION"*

Также можно воспользоваться подготовленным скриптом `/bin/sql_createOra.cmd`, указав необходимые параметры в тексте скрипта.

#### 4.5.4.2 Загрузка и выгрузка произвольных данных

Существует возможность загрузки и выгрузки произвольных данных системы.

Загрузить можно справочник, например, *Организации*, *Справочник товаров*, *работ и услуг* и др. Выгрузить из системы можно справочники, элемент справочника, список документов, документ. Все операции загрузки и выгрузки осуществляются через взаимодействие с сервером приложений. Взаимодействие осуществляется путем обмена сообщениями в формате **XML**.

Для передачи и получения сообщений сервера приложений в дистрибутиве поставляется программа **ExecXML.exe**. В сообщении может содержаться задание серверу приложений.

Использование:

*ExecXML.exe budget\_id=[номер\_бюджета] [файл\_XML\_задания] > [результат\_обработки]*

**Пример:**

*ExecXML.exe budget\_id=1 task.xml > goods.xml*

Параметр action может принимать значения: **synchronize**, **perform**, **list**, **update**.

Таблица 20 – Формат XML-заданий

Значение параметра “action”	Описание
insert	Добавляет объекты в БД.
delete	Удаляет объекты из БД.
update	Обновляет объекты из БД.
synchronize	Объекты, отсутствующие в БД, добавляются. Изменения приводятся в соответствие с элементами в задании. Эквивалентно выполнению <b>insert</b> и <b>update</b> .
perform	Содержимое БД приводится в соответствие с данными в задании. Элементы добавляются, удаляются и изменяются по необходимости. Эквивалентно выполнению <b>insert</b> , <b>delete</b> и <b>update</b> .
list	Формирует список элементов.

Полный список сервер-процессоров и их команд, поддерживаемых текущей версией сервера приложений, можно найти в файле **serverprocessor.xml**.

#### 4.5.4.3 Создание резервной копии БД

Каждый тип СУБД использует свою процедуру для создания резервной копии БД и ее восстановления в случае необходимости. Для получения подробной информации следует обратиться к технической документации по СУБД **Oracle** и **Firebird**.

Резервное копирование в СУБД **Oracle** осуществляется утилитой **Oracle Recovery Manager (RMAN)**.

**RMAN** – это компонент БД **Oracle**, используемый для операций резервирования и восстановления.

Начиная с первого релиза **10g** СУБД **Oracle** предоставляет возможность определить специальную область на диске – флэш-область восстановления (**flash recovery area**), которая используется базой данных как резервное местоположение. По умолчанию

**RMAN** создает во **flash recovery area (FRA)** резервные копии всех типов: регулярных резервных копий, образов копий, журнальных архивных файлов.

Перед началом работы необходимо выполнить следующие действия:

1. Настроить местоположение и размер области **flash recovery area** (если она не была настроена во время создания базы данных). Для этого в файле настроек необходимо установить следующие параметры:

- **DB\_RECOVERY\_FILE\_DEST** = directory | disk group – путь к области восстановления;
- **DB\_RECOVERY\_FILE\_DEST\_SIZE** = integer [K | M | G] – размер области восстановления.

По возможности эта область должна располагаться отдельно от основной базы данных. Размер необходимо подобрать исходя из объемов сохраняемой информации (как правило в два-три раза больше размера базы данных). Следить за использованием **FRA** можно с помощью представления **V\$FLASH\_RECOVERY\_AREA\_USAGE**.

2. Настроить БД для работы в режиме архивирования логов, если это не было сделано при создании базы. Для этого необходимо выполнить следующие команды:

```
SHUTDOWN IMMEDIATE; -- остановить базу  
STARTUP MOUNT; -- стартовать в режиме mount  
ALTER DATABASE ARCHIVELOG; -- включить архивирование логов  
ALTER DATABASE OPEN; -- открыть базу для обычной работы
```

После выполнения указанных действий можно осуществлять резервное копирование без остановки работы базы данных.

Пользователям ОС **Linux** для удобства работы необходимо настроить интеграцию **SQL\*Plus** и **RMAN** с редактором командой строки:

1. Найти и установить пакет **rlwrap**:

```
# wget http://www.mirrorservice.org/sites/dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm  
# rpm -Uvh epel-release-6-8.noarch.rpm  
# yum install rlwrap
```

2. В файл **/home/oracle/.bash\_profile** добавить строки:

```
alias rlsqplus='rlwrap sqlplus'  
alias rlman='rlwrap rman'
```

Для того чтобы сразу подключиться к целевой базе данных, необходимо запускать **RMAN** при помощи команды **rman target/**.

Команда **show all** позволяет просмотреть конфигурацию **RMAN**. Если параметр не изменился, то в конце строки будет пометка **# default**.

Изменение значений параметров производится командой **configure**:

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

Сброс значения параметра осуществляется командой **clear**:

```
CONFIGURE CONTROLFILE AUTOBACKUP CLEAR;
```

**RMAN** автоматически удаляет ненужные, избыточные или устаревшие резервные копии, чтобы освободить место для новых копий. Для автоматического удаления необходимо правильно настроить политику удержания. Существуют две политики удержания – **recovery window** и **redundancy**, которые являются взаимозаменяемыми. В один момент времени работает только одна политика.

Политика **recovery window** устанавливает такой набор резервов, чтобы можно было восстановить базу на любой период в прошлом на количество дней, указанных в политике:

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

Политика **redundancy** (используется по умолчанию) устанавливает число резервных наборов, которые необходимо хранить:

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1;
```

Основные команды по созданию резервных наборов БД:

- резервирование всей БД:

```
BACKUP DATABASE;
```

- создание сжатого резервного набора всей БД:

```
BACKUP AS COMPRESSED BACKUPSET DATABASE;
```

- резервирование с использованием произвольного формата:

```
BACKUP DATABASE FORMAT 'd:\rman-backup\rman_%d_%U.bkp';
```

- резервирование отдельных табличных пространств:

```
BACKUP TABLESPACE system, users;
```

*Примечание.* Для больших баз данных может быть актуальным инкрементальное резервирование. В этом случае копируются не все блоки БД, а только те, которые были изменены с момента предыдущего резервирования.

- инкрементальное резервирование уровня 1 (перед ним надо сделать полное резервирование уровня 0):

```
BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

для обеспечения нормальной скорости при таком резервировании необходимо включить отслеживание измененных блоков командой в **sqlplus**:

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE  
'/opt/oracle/flash_recovery_area/change.log';
```

- просмотр всех резервных наборов:

```
LIST BACKUP;
```

- проверка соответствия записей о резервных наборах и журналах в контрольных файлах, самим файлам наборов и журналам:

```
CROSSCHECK BACKUP;
```

```
CROSSCHECK ARCHIVELOG ALL;
```

- удаление записей об отсутствующих наборах и журналах из контрольных файлов:

```
DELETE EXPIRED BACKUP;
```

```
DELETE EXPIRED ARCHIVELOG ALL;
```

- удаление устаревших копий (устаревшие копии определяются автоматически по политике удержания):

```
DELETE OBSOLETE;
```

Исходя из вышесказанного, типичный сценарий ежедневного полного резервирования может быть таким:

```
CONNECT TARGET /
```

```
BACKUP AS COMPRESSED BACKUPSET DATABASE;
```

```
CROSSCHECK BACKUP;
```

```
CROSSCHECK ARCHIVELOG ALL;  
DELETE NOPROMPT EXPIRED BACKUP;  
DELETE NOPROMPT EXPIRED ARCHIVELOG ALL;  
DELETE NOPROMPT OBSOLETE;  
EXIT
```

Данный скрипт можно запускать автоматически по расписанию, желательно в моменты наименьшей нагрузки, например, ночью.

Для СУБД **Firebird** нет специальных утилит, поставляемых в составе «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Это связано с тем, что стандартная утилита архивирования и восстановления БД, поставляемая с СУБД **Firebird**, достаточно проста в эксплуатации. В системе «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» автоматическая подготовка резервных копий БД выполняется с помощью задания планировщика **Backup DB**.

#### 4.5.4.4 Восстановление БД из резервных копий

При необходимости восстановления БД **Oracle** алгоритм восстановления и набор команд зависят от характера сбоя. Примеры действий по восстановлению в часто встречающихся ситуациях приведены ниже:

1. Потеря журнального файла. Если потеряна группа текущих журналов, то следует производить неполное восстановление базы данных. В случае потери одного файла из группы, восстановление производится простым копированием оставшегося файла. Если потеряна не текущая группа, то можно попытаться восстановить (очистить) ее командой:

```
ALTER DATABASE CLEAR LOGFILE GROUP 1;  
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 1;
```

2. Потеря несущественного файла данных (не **system** и не **undo**). В этом случае базу данных можно не останавливать. Необходимо выполнить следующие команды:

```
RESTORE DATAFILE 5; -- восстановление файла номер 5 из резервного набора  
RECOVER DATAFILE 5; -- накат логов до текущего состояния  
ALTER DATABASE DATAFILE 5 ONLINE; -- включение файла в работу
```

3. Потеря существенного файла данных (**system** или **undo**). Необходимо перевести базу данных в состояние **mount** и выполнить следующие команды:

```
SHUT IMMEDIATE; -- остановка БД  
STARTUP MOUNT; -- старт в режиме монтирования  
RESTORE DATAFILE 1; -- восстановление файла номер 1 из резервного набора  
RECOVER DATAFILE 1; -- накат логов до текущего состояния  
ALTER DATABASE OPEN; -- открытие БД для работы
```

Восстановление всей БД можно произвести командами:

```
SHUT IMMEDIATE; -- остановка  
БД STARTUP MOUNT; -- старт в режиме монтирования  
RESTORE DATABASE; -- восстановление  
БД RECOVER DATABASE; -- накат логов до текущего состояния  
ALTER DATABASE OPEN; -- открытие БД для работы
```

**RMAN** имеет возможность восстановления отдельных блоков файлов без остановки работы БД. Если известен файл и номер сбойного блока, то его можно восстановить следующей командой:

```
BLOCKRECOVER DATAFILE <FILE> BLOCK <BLOCK>;
```

где:

- <FILE> – номер файла;
- <BLOCK> – номер сбойного блока.

Если сбойных блоков много, то можно произвести восстановление по списку. Для этого выполняется валидация базы данных и заполняется представление **V\$DATABASE\_BLOCK\_CORRUPTION**, затем производится восстановление блоков:

```
BACKUP VALIDATE DATABASE;  
BLOCKRECOVER CORRUPTION LIST;
```

Для удобной работы с утилитой **RMAN** существуют следующие скрипты:

- **rman\_b0.sh**;
- **rman\_b1.sh**;



- **rman\_del.sh** (**rman\_b0.cmd**, **rman\_b1.cmd** и **rman\_del.cmd** под ОС **Windows** соответственно).

Автоматическая настройка резервного копирования производится с помощью встроенной в ОС **Linux** утилиты **crontab**. Для этого в файле **/etc/crontab** прописываются строчки следующего формата:

```
0 23 * * 1-6 su – oracle -c '/home/server/rman_b1.sh'  
0 22 * * 7 su – oracle -c '/home/server/rman_del.sh'  
0 23 * * 7 su – oracle -c '/home/server/rman_b0.sh'
```

где:

- 0 – минуты;
- 23 – часы;
- 1-6 – дни недели за которыми следует выполняемая команда (выполнение скрипта **rman\_\*.sh** от имени пользователя **oracle**).

Данные строки будут производить резервирование **уровня 1** с понедельника по субботу, а в воскресенье резервирование **уровня 0** и очистку устаревших резервных наборов, и архивных журналов.

После добавления строки необходимо выполнить команду **crontab** для обновления заданий планировщика.

*Примечание.* Для более полного изучения способов резервного копирования и восстановления данных обратитесь к справочному руководству по СУБД **Oracle**.

#### 4.5.5 Установка и настройка сервера приложений

Установку и первоначальную настройку сервера приложений можно разделить на следующие этапы:

1. копирование необходимых файлов и папок на физический сервер;
2. установка необходимых параметров в файле свойств;
3. дополнительная установка драйвера ключа защиты, копирование файла лицензии, установка ключа защиты и установка необходимых параметров в файле лицензий (для сервера лицензий);

4. дополнительная установка **Crypto Pro CSP** (для сервера проверки ЭП);
5. запуск сервера приложений и формирование данных в БД (выполнение **xml-команд**).

Для работы сервера приложений на физическом сервере необходимо создать отдельную папку (желательно, чтобы имя папки не содержало пробелов). В папку необходимо скопировать из сборки следующие файлы и папки:

- **/server**;
- **startsrv** (для ОС **Linux**);
- **StartServer.bat** (для ОС **Win32**);
- **Server.properties** (файл свойств).

При старте сервер приложений считывает значения необходимых параметров из файла свойств (**server.properties**). Считывание происходит только при старте сервера приложений.

Файл свойств представляет собой простой текстовый файл. Имя файла свойств можно изменить, но тогда в скрипте сервера приложений (**StartServer.bat** для **Win32**, **startsrv** для **Linux**) необходимо поменять значение переменной **SERVER\_CONFIG** на имя нужного файла свойств.

Если при старте сервер приложений не смог подключиться к БД, то он завершает свою работу. Если при старте сервер приложений не смог найти файл лицензии, либо не установлен аппаратный ключ защиты (либо если не найден сервер лицензий), а параметры пути лицензии были указаны, то сервер приложений завершает свою работу.

После запуска сервера приложений в созданной папке появится журнальный файл (**log-файл**), в котором будут вестись записи выполняемых сервером приложений команд и заданий.

В файле свойств (**Server.properties**) необходимо установить следующие параметры:

- в зависимости от используемой СУБД раскомментировать строки подключения к БД и ввести в них необходимые параметры;

- остальные параметры вводятся при необходимости (допускается использование их значений по умолчанию).

Для сервера лицензий также необходимо установить драйвер ключа защиты (поставляется отдельно от сборки), а также скопировать в папку установки сервера приложений файл лицензий.

Установка необходимых параметров в файле лицензий (для сервера лицензий) подробно описана в разделе [Параметры подключения лицензии](#)<sup>107</sup>.

Для функционирования сервера приложений в качестве сервера проверки ЭП необходимо установить на физический сервер ПО **Crypto Pro CSP**.

Установка необходимых параметров в для сервера проверки ЭП подробно описана в разделе [Параметры ЭП](#)<sup>117</sup>.

*Примечание. Для проверки ЭП в сервере приложений существует специальная учетная запись для осуществления проверки ЭП – **signer** с пустым паролем (пароль можно сменить).*

#### 4.5.5.1 Файл настроек сервера приложений

*Внимание! Неверное изменение файла настроек может привести к отказу сервера.*

##### 4.5.5.1.1 Параметры подключения лицензии

Для определения лицензии используются следующие параметры:

- **azk.license.typeinit** – тип источника лицензии. Может принимать следующие значения:
  - **FILE** – используется, когда для лицензия определяется через ключ;
  - **SERVER** – для определения лицензии через сервер лицензий.
- **azk.license.path** – путь к лицензии. В случае использовании типа источника **FILE** указывается путь к файлу лицензии. В случае использования типа источника **SERVER** указывается путь и порт сервера лицензий.

Если параметр не указан, то сервер будет работать в демонстрационном режиме.

#### 4.5.5.1.2 Параметры использования модулей

Для определения доступных модулей используются параметры:

- **azk.slave.modules** – параметр определяет доступные для **SLAVE-сайта** модули;
- **azk.modules.disabled** – параметр позволяет отключать на **MASTER-сайте** не используемые модули.

Таблица 21 – Перечень модулей

Лицензия	Описание
antidumping	Подсистема антидемпинговых мер.
auction	Доступ к ЭД «Решение о проведении аукциона».
auctionperunit	Подсистема проведения аукциона за единицу продукции.
auctionview	Подсистема просмотра хода проведения открытого аукциона в web (не электронная форма).
aumodule	Подсистема для работы в системе автономным учреждениям.
azkacsexchange	Подсистема интеграции с системой «АЦК-Бюджетный учет».
azkfin	Подсистема интеграции со СКИБ.
azkplancontrol	Подсистема контроля ЭД «План закупок» в системе «АЦК-Планирование» и возможность выгрузки проекта ЭД «План закупок» в систему «АЦК-Планирование».
bankGuarantee	Подсистема банковских гарантий.
beloblrep	Подсистема специальных отчетов Белгородской области.
c99_pre_advance	Предварительный контроль сведений, размещаемых в реестре контрактов
cavildoc	Интеграция с Порталом «За честные закупки».
claim	Подсистема исков и претензий.
cleanorders	Разрешает в решениях добавлять заказчиков вручную.
competitionperunit	Подсистема проведения конкурса за единицу продукции.
complaint	Подсистема жалоб.
ConGoodsSidWithoutOrder	Выгрузка в ЕИС в составе информации о контракте идентификатора объекта закупки без использования ЭД «Решение»/
concomparison	Автоматическая выгрузка ЭД «Контракт» в СКИБ/ЕИС в зависимости от внесенных в документ изменений.
confacts	Доступ к ЭД «Факт поставки».
consolidatedlimits	Доступ к сводному АРМ резервов лимитов/планов ФХД.
consoliddoc	Подсистема консолидированных закупок.

contract_constructor	Подсистема «Конструктор печатных форм контрактов»
contract223fl	Подсистема учета контрактов и договоров по 223-ФЗ.
contracts	Доступ к ЭД «Контракт».
ctattach	Подсистема шаблоны контрактов и договоров.
doublecontest	Подсистема проведения двухэтапного конкурса.
ekbneed	<p>При наличии лицензии:</p> <ul style="list-style-type: none"> <li>• доступен справочник <i>Отделы организации</i> и в справочнике <i>Отделы организации</i> отображается группа полей <b>Курируемая сфера деятельности</b>.</li> <li>• в ЭД «Потребность (44-ФЗ)» отображается группа полей <b>Курирующая организация</b>, поле <b>Периодичность закупки</b>, закладка <b>Лист согласования</b>, параметры системы по ЭД «Потребность». Также лицензия влияет на логику заполнения полей в строке спецификации ЭД «Потребность» (44-ФЗ).</li> </ul>
etpint	Подсистема интеграция с ЭТП.
etpint-micex	Подсистема интеграция с ЭТП «ММВБ».
etpint-rts	Подсистема интеграция с ЭТП «РТС-Тендер».
eshop_integration	Подсистема интеграции с электронным магазином «ОТС-Маркет»
finsupportrep	Доступ к АРМ ФО.
finsupportrep	Подсистема АРМ сотрудника ФО для контроля плана закупок.
fullreestr	Подсистема статистики по реестру контрактов.
infotenderrep	Предоставляет расширенный доступ к отчету Информация о торгах и других закупках.
intnetic	Подсистема интеграция с НЭТИС.
kb_integration	Интеграция с базой знаний
limitedcontest	Подсистема проведения конкурса с ограниченным участием.
misintegration	Интеграция с Единой медицинской информационной системой Ульяновской области
monitoringint	Подсистема интеграции с информационной системой «Мониторинг цен»
multibudget	Подсистема мультибюджетности.

	<ul style="list-style-type: none"> <li>• При наличии лицензии <b>multibudget</b> существует ограничение числа доступных для использования бюджетов с признаком <b>Бюджет автономного/бюджетного учреждения, юридического лица размещающего закупки по 223-ФЗ</b>. Если число активных бюджетов превышает допустимое число, система выдаёт ошибку: «Вход в систему невозможен, так как количество активных бюджетов превышает допустимое значение».</li> <li>• При отсутствии лицензии <b>multibudget</b> допустимо работать только в одном бюджете без признака <b>Бюджет автономного/бюджетного учреждения, юридического лица размещающего закупки по 223-ФЗ</b> и только в одном бюджете с данным признаком (т.е. при отсутствии лицензии допустимо работать только с двумя бюджетами, если бюджетов будет больше, то выдается сообщение об ошибке: LicenseException: Доступ запрещен. <i>Отсутствует лицензия на подсистему «Мультибюджетность» (multibudget)</i>).</li> </ul>
multibudgetlight	Подсистема мультибюджетности (сокращенная версия).
newplanviewer	Подсистема АРМ «Просмотра результатов размещения» (новый).
nnoblcat	Доступ к Региональному каталогу товаров, работ, услуг. Нижегородской области.
nnoblrep	Доступ к Универсальному отчету по плану закупок.
offerrequest	Подсистема проведения запроса предложений.
oosimport	Подсистема импорта данных из ЕИС в БД.
oosint	Подсистема интеграции с ЕИС.
optimized_price	Подсистема оптимизации начальной (максимальной) цены контракта при согласовании ЭД «Заявка на закупку».
plancontol	<p>Контроль системных параметров:</p> <ul style="list-style-type: none"> <li>• Превышение сумм плана над суммами доведенных лимитов;</li> <li>• Превышение сумм плана над суммами плана ФХД.</li> </ul> <p>В справочнике <i>Строки бюджетных и автономных учреждений</i> не доступны следующие колонки: <i>План за первый год, План за второй год, План за третий год.</i></p>
plancontrolslib	Подсистема контроля ЭД «План закупок» в системе «АЦК-Финансы» и возможность выгрузки документа в СКИБ для контроля лимитов/плана ФХД.
planning	Подсистема планирования (старое).
planning_procedures	Подсистема планирования (новое).
planviewer	Подсистема АРМ «Просмотр результатов размещения» (старый).
preselection_org	Подсистема подготовки предварительного отбора.
priceconfirmation	Доступ к отчету <i>Обоснование начальной (максимальной) цены контракта.</i>
PurchasePlanSmallGoods	Возможность формирования спецификации в планах для закупок малого объема.

	<p>При наличии лицензии становятся доступны системные параметры <b>Для закупок формируемых в соответствии с законодательством только в объеме финансирования, дополнительно указывать спецификацию в ЭД "План закупок" и Для закупок формируемых в соответствии с законодательством только в объеме финансирования, дополнительно указывать спецификацию в ЭД "План-график"</b>, а также функциональные возможности системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ», связанный с этими параметрами.</p>
purchases	Подсистема решений о проведении закупки.
purchases223fl	<p>Подсистема планирования и осуществления закупок отдельными юридическими лицами.</p> <p>При отсутствии лицензии:</p> <ul style="list-style-type: none"><li>• В справочнике <i>Бюджеты</i> признак <b>Бюджет автономного/бюджетного учреждения, юридического лица размещающего закупки по 223-ФЗ</b> не доступен для редактирования и по умолчанию выключен.</li><li>• Недоступен механизм интеграции с ЭТП «ОТС» и ЕИС-223-ФЗ. При попытке соединения система выводит сообщение об ошибке: <i>Доступ запрещен. Отсутствует лицензия на подсистему "&lt;Наименование на русском&gt;" (&lt;Наименование на английском&gt;).</i></li><li>• При обработке ЭД:<ul style="list-style-type: none"><li>○ «Решение о закупке у единственного источника»;</li><li>○ «Решение о проведении конкурса»;</li><li>○ «Решение о проведении запроса котировок»;</li><li>○ «Решение о проведении торгов на ЭТП»;</li><li>○ «Решение о проведении запроса предложений»,</li></ul>со статуса «Отложен» по действиям <b>Обработать, Подписать и обработать и На разработку документации</b> проверяется, что если в бюджете активирован признак <b>Бюджет автономного/бюджетного учреждения, юридического лица размещающего закупки по 223-ФЗ</b>, то системе доступна лицензия. Если лицензия отсутствует, то система выводит сообщение об ошибке: <i>Доступ запрещен. Отсутствует лицензия на подсистему "&lt;Наименование на русском&gt;" (&lt;Наименование на английском&gt;).</i></li><li>• При обработке ЭД «Контракт» со статуса «Отложен»/«Новый» по действиям <b>Направить на контроль и Подписать и обработать</b> проверяется, что если в бюджете активирован признак <b>Бюджет автономного/бюджетного учреждения, юридического лица размещающего закупки по 223-ФЗ</b>, то системе доступна лицензия. Если лицензия отсутствует, то система выводит сообщение об ошибке: <i>Доступ запрещен. Отсутствует лицензия на подсистему "&lt;Наименование на русском&gt;" (&lt;Наименование на английском&gt;);</i></li><li>• Недоступно редактирование поля <b>Ссылка на положение о закупках</b>, закладки <b>Контактная информация</b> справочника <i>Организации</i> в веб-клиенте.</li></ul> <p>Если лицензия доступна, то:</p>

	<ul style="list-style-type: none"> <li>• в справочнике <i>Бюджеты</i> доступен для редактирования признак <b>Бюджет автономного/бюджетного учреждения, юридического лица размещающего закупки по 223-ФЗ</b>;</li> <li>• доступен механизм интеграции с ЭТП «ОТС» и ЕИС-223ФЗ;</li> <li>• не осуществляются ограничения при обработке документов;</li> <li>• доступно редактирование поля <b>Ссылка на положение о закупках</b>, закладки <b>Контактная информация</b> справочника <i>Организации</i> в web-клиенте.</li> </ul>
ReqDocContractCreate	Подсистема формирования контракта, договора из заявки на закупку предоставляет возможность формирования ЭД «Контракт» и «Договор» из ЭД «Заявка на закупку».
repconexecution	Доступ к <i>Отчету об исполнении контракта</i> .
repmagnitogorsk	Подсистема отчетных формы города Магнитогорск.
reptemplate	Подсистема формирования конкурсной и аукционной документации.
requests	Доступ к ЭД «Заявка на закупку».
RequestForQuotation	Запрос цен
requesttemplate	Подсистема шаблонов заявок на закупку.
requirement	Подсистема сбора потребностей в закупке товаров, работ, услуг.
requirement44fz	Доступ к ЭД «Потребность (44-ФЗ)».
rqlots	Подсистема лотов в заявках.
rqplanfhndreserv	Предоставляет доступ к выгрузки в СКИБ заявок бюджетных и автономных учреждений.
rqqual	<p>Подсистема доступа к закладке <b>Требования</b> ...в ЭД «Заявка на закупку» и «Сводная заявка на закупку». На закладке доступен только блок с требованиями, т.е. блок с документами и сведениями для редактирования не доступен.</p> <p>При наличии лицензии в списке возможно сохранять только требования с типами:</p> <ul style="list-style-type: none"> <li>• <i>логические требования к продукции</i>;</li> <li>• <i>интервальные требования к продукции</i>;</li> <li>• <i>требования к продукции первого этапа конкурса</i>.</li> </ul> <p>При попытке сохранить требование с иным типом система выводит сообщение об ошибке: <i>Допустимо указывать только требования к продукции</i>.</p>
structural_units	Подсистема работы со структурными подразделениями.
subdivisionref	Подсистема справочник подразделений
summaryplanreport	Доступ к <b>Сводному отчету по планированию</b> и <b>Универсальному отчету по плану закупок</b> .
summaryrequestdoc	Доступ к ЭД «Сводные заявки на закупку».
suppliersportal	Подсистема интеграции с электронным магазином «Портал поставщиков»



togliattirep	Подсистема отчетов города Тольятти.
tomskoblrep	Подсистема специальных отчетов Томской области: <ul style="list-style-type: none"><li>• Отчеты. Томская область;</li><li>• Отчеты по планам. Томская обл.</li></ul>
web_client	Подсистема доступа к web-клиенту системы.
websiteplanview	Доступ к разделу <b>Планирование закупок</b> на web-сайте.
webstat	Подсистема статистика в web.
	Подсистема продления срока подачи заявок Заказчиков при проведении совместных закупок

#### 4.5.5.1.3 Параметры подключения к базе данных

Для подключения к базе данных используются следующие параметры:

- **azk.db.driver** – класс драйвера.

*Пример: azk.db.driver=oracle.jdbc.driver.OracleDriver.*

- **azk.db.url** – путь к базе данных.

*Пример: azk.db.url=jdbc:oracle:thin:@appserver.bssys.com:1521:AZK.*

- **azk.db.user** – пользователь базы данных.
- **azk.db.password** – пароль для пользователя базы данных.
- **azk.db.accessmode** – режим работы с базой данных (**oracle**, **mssql**, **interbase**).
- **azk.db.pool.size** – максимальное количество подключений к серверу базы данных.
- **azk.db.cachesize** – размер сессионного кэша откомпилированных запросов.
- **azk.db.sysuser** – имя пользователя БД с неограниченными правами.
- **azk.db.sypassword** – пароль пользователя БД с неограниченными правами.
- **azk.db.connectiontimeout** – максимальное время в секундах, в течении которого система будет пытаться установить соединение.
- **azk.db.connectiondelay** – задержка в секундах между неудачными попытками соединения (см. **azk.db.connectiontimeout**)

- **azk.integration.eacts\_attach.request\_timeout** – регулирует максимальное время в секундах, в течении которого система будет пытаться установить соединение, и передачу данных при приеме аттачей электронных актов.

**Внимание!** Для **Oracle** размер кэша очень критичен, т.к. у него ограничено возможное количество открытых одновременно курсоров.

#### 4.5.5.1.4 Параметры журнализации

Для журнализации используются следующие параметры:

- **azk.log.filename** – журнал сервера приложений.
- **azk.log.mode** – уровень журнализации сервера приложений. Чем он выше, тем меньше сообщений попадает в журнал. Допустимые значения от 0 (все) до 4 (только исключения).
- **azk.log.screenmode** – уровень журнализации сервера приложений в стандартный поток вывода.
- **azk.log.encodeScreen** – используемая кодировка для вывода на консоль (**Ср866, Ср1251, KOI8\_R, ISO8859\_5, UTF-8**).
- **azk.log.encode** – используемая кодировка для вывода в журнал (**Ср866, Ср1251, KOI8\_R, ISO8859\_5**).
- **azk.log.append** – перезаписывать лог или добавлять записи в существующий журнал.
- **azk.db.traceenabled** – включение вывода SQL-трассировки запросов, их параметров, времени выполнения в лог. Значение по умолчанию *false* (выключен), для активации ввести *true*.
- **azk.db.statenabled** – включение сбора SQL-статистики. Статистика собирается в памяти сервера приложения, при выполнении задания **SQLTracer** информация записывается в лог (уровень вывода **INFO**). Значение по умолчанию *false* (выключен), для активации ввести *true*.

*Примечание.* Механизм журнализации «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» использует инструментарий *Simple Logging Facade for Java (SLF4J)*. Для работы механизма в файле *Azk2Server.properties* указывается ссылка на файл настроек инструментария *Azk2logback.xml*.

#### 4.5.5.1.5 Параметры доступа

Для доступа к серверу приложений используются следующие параметры:

- **azk.db.pool.timeout** – время, по истечении которого, если ни одно соединение не освобождается, при максимальном количестве подключений к базе данных сервер возвращает исключение клиенту.
- **azk.job.finishedtime** – время в минутах, по истечении которого, результаты завершеного асинхронного задания будут удалены с сервера приложений.
- **azk.job.sweepinterval** – периодичность анализа асинхронных заданий на завершенность в минутах.
- **azk.msg.gateway** – сайт, которому направляются запросы на выделение адреса, а также перенаправляются сообщения с неизвестным получателем.
- **azk.msg.network** – набор диапазонов адресов, выделяемых данным сайтом по запросу.

*Пример:* `azk.msg.network=0, 4, 8-10, 11.`

- **azk.session.finishedtime** – время в минутах, по прошествии которого, сессия, в которой не было активности пользователей, закрывается.
- **azk.session.cleartime** – время в минутах, по прошествии которого, закрытая сессия удаляется.
- **azk.session.cachesize** – количество сессий пользователей, которое сервер приложений удерживает в кэше.
- **azk.session.check** – проверяется корректность сессии при каждом вызове (**true**, **false**), должно быть установлено в **true**, при использовании кластеров серверов приложений, иначе могут возникнуть проблемы при административном принудительном закрытии сессий.

- **azk.security.cachesize** – количество контекстов безопасности, которое сервер приложений удерживает в кэше.

#### 4.5.5.1.6 CORBA

Для **CORBA** используются следующие параметры:

- **iiop.port** – порт, на котором работает сервер приложений. Это значение лучше задавать из командной строки и иметь общий файл настроек для всех серверов кластера.
- **iiop.hostname** – имя хоста в идентификаторах объектов **CORBA**. В ситуациях, когда автоматическое определение невозможно, можно задать это значение здесь явно (для OS/400 нужно задать в качестве имени хоста его IP). В случае использования кластера, желательно задавать его с командной строки.
- **iiop.resolvehostname** – параметр задает определение сервером имен хостов по IP-адресу. В случае, если использовать **DNS** для определения имен невозможно, то можно установить значение параметра в значение **false**. В этом случае также рекомендуется установить значение параметра **iiop.hostname**, значение по умолчанию **true**.
- **iiop.alternateAddr.endpoint1** – параметр задает по какому IP-адресу и на каком порту клиент должен запрашивать список доступных сервисов, с которыми он будет в дальнейшем работать. Если клиент находится за туннелем, то нужно указывать адрес и порт туннеля, если на сервере несколько сетевых интерфейсов, то указывается адрес и порт интерфейса доступного клиенту, формат: **ip:port**.

**Пример:** *iiop.alternateAddr.endpoint1=192.168.0.1:2001.*

- **openorb.server.maxThreadPoolSize** – максимальное количество **CORBA**-потоков сервера приложений, обслуживающие клиентские запросы. Если к серверу обратится большее количество пользователей одновременно, то соответствующие обращения станут в очередь. Используется для ограничения/увеличения пиковой нагрузки на сервер приложений, в зависимости от мощности используемого сервера, по умолчанию **100**.

#### 4.5.5.1.7 Параметры репликации

Для репликации используются следующие параметры:

- **azk.rpl.user** – имя пользователя для доступа при репликации.
- **azk.msg.site\_id** – номер сайта репликации для сервера приложений. Это число, умноженное на 1000000000, добавляется к каждому создаваемому искусственному ключу.

#### 4.5.5.1.8 Параметры планировщика

Для планировщика используются следующие параметры:

- **azk.seheduler.maxRefreshInterval** – максимальный период обновления расписания в мс. Расписание обновляется перед каждым запуском задания, либо если установленное время прошло, то данным параметром от момента предыдущего запуска.
- **azk.scheduler.minRefreshInterval** – минимальный период между запусками заданий в мс, т.е. если выставить данный период в десять секунд, то задания не смогут выполняться на одном сервере чаще, чем один раз в десять секунд, соответственно погрешность вызова задания в указанное время тоже увеличивается. Параметр предназначен для снижения нагрузки на сервер приложений по отслеживанию расписания.
- **azk.session.sessioncleaner.iteration** – количество сессий, которые за раз могут быть закрыты заданием **Session cleaner**.

#### 4.5.5.1.9 Параметры ЭП

Для ЭП используются следующие параметры:

- **azk.sign.hostname** – адрес сервера, который осуществляет проверку ЭП. Если адрес не указан, то сервером выступает текущий сервер.
- **azk.sign.port** – порт сервера, который осуществляет проверку ЭП.

- **azk.sign.user** – учетная запись пользователя, под которой осуществляется проверка ЭП.
- **azk.sign.pswd** – пароль учетной записи пользователя, указанного в параметре **azk.sign.user**.

#### 4.5.5.1.10 Параметр настройки исходящих сообщений для ЕИС 223-ФЗ

При добавлении параметра **azk.oos.223fz.integration.create\_msg=true** в файл **Server.properties** вместо синхронной отправки в ЕИС документов создается (сохраняется в БД) исходящее сообщение для сайта ЕИС 223-ФЗ, содержащее файл-xml отправляемого документа.

В исходящем xml-сообщении передаются данные об извещениях и их лотах для закупок, для которых в рамках системы не предусмотрены лоты, но в соответствии с форматом интеграции с ЕИС по 223-ФЗ могут содержать лоты:

- **ORDERDOC.OOSKEY** – **guid** извещения;
- **CMPLOT.OOSKEY** – **guid** лота;
- **ORDERDOC.LOTOOSKEY** – **guid** лота извещения для безлотовых решений «АЦК-Госзаказ»/«АЦК-Муниципальный заказ».

#### 4.5.5.1.11 Настройка взаимодействия с системой Аванпост

Для настройки взаимодействия с системой Аванпост в файле **Server.properties** необходимо указать следующее:

```
# -----  
#! Настройка FAM!  
# -----  
#внутреннее название SSO сервиса  
azk.openid.provider.id=FAM  
#Все параметры ниже должны иметь постфикс со значением параметра provider.id  
#адрес хоста FAM  
azk.openid.provider.issuer.FAM=https://k8s-st-fam.bftcom.com
```

#путь обращения к FAM за авторизацией с учетом атрибутов необходимых для сверки при авторизации  
azk.openid.provider.basic\_auth\_uri.FAM=/oauth2/authorize?  
response\_type=code&scope=openid%20profile%20email  
#Client\_id из настроек приложения в FAM (пример: AvanPost FAM -> Приложения -> azk-finance -> Настройки -> Client ID)  
azk.openid.client\_id.FAM=d9d3e534-106d-4146-a92c-3bce69c43d0b  
#Секрет из настроек приложения в FAM (пример: AvanPost FAM -> Приложения -> azk-finance -> Настройки -> Смена секрета)  
azk.openid.client\_secret.FAM=12345678901234567890  
#Идентификатор приложения АЦК в FAM, формируется в FAM, должен стоять одинаковый в FAM и АЦК  
azk.openid.client\_id.FAM=b0540986-e0de-4fd3-a76d-4d04ba940731  
#пароль приложения АЦК в FAM, должен стоять одинаковый в FAM и АЦК  
azk.openid.client\_secret.FAM=Q!w2e3r4  
#адрес входа в веб клиент АЦК через SSO, должен стоять одинаковый в настройках FAM и АЦК  
azk.openid.redirect\_uri.FAM=http://172.30.3.14:9080/af/loginOpenId  
#локальный адрес и порт работы с windows-клиентом, настройка должна стоять одинаковой в FAM и АЦК  
azk.openid.thin\_redirect\_uri.FAM=https://localhost:50007/loginOpenId  
# Redirect\_uri из настроек приложения в FAM для web и delphi (thin) клиентов соответственно  
#(пример: AvanPost FAM -> Приложения -> azk-finance -> Настройки -> Redirect\_uris указываются через пробел)  
azk.openid.redirect\_uri.FAM=http://localhost:8081/azk/loginOpenId

#### 4.5.6 Установка и настройка online-клиентов

Установка online-клиентов выполняется при помощи файла программы, находящегося в каталоге */ClientX/full\_setup.exe*.

После установки необходимо выставить необходимые параметры в файле свойств online-клиента **order.ini** (альтернативные параметры должны быть закомментированы знаком «;» (точка с запятой)).

Если используется соединение по протоколу **CORBA**:

*ServerMode=CORBA (в секции [SERVER])*

*Server=localhost:2001 (в Секции [CORBA] – адрес:порт сервера приложений)*

Если используется соединение по протоколу **HTTP**:

*ServerMode=HTTP (в секции [SERVER])*

*URL=http://localhost:8080/proxy (вместо localhost:8080 – указать адрес и порт web-сервера Tomcat, обслуживающего Web-сайт «АЦК-Госзаказ»/«АЦК-Муниципальный заказ», параметр «/proxy» должен присутствовать обязательно)*

*ProxyServer= (указать прокси-сервер, через который осуществляется выход во внешнюю сеть, если есть)*

*ProxyPort= (указать порт прокси-сервера)*

*ProxyAuthentication= (для указания обязательности авторизации на прокси-сервере)*

*ProxyUser= (пользователь для выхода через прокси-сервер)*

*ProxyPassword= (пароль пользователя для выхода через прокси-сервер)*

Остальные параметры настраиваются опционально, т.е. при необходимости (назначения параметров описано в комментариях в самом файле свойств).

Запуск online-клиента можно осуществлять файлом **order.exe**.

В процессе администрирования системы у стартовой подсистемы online-клиента можно использовать вспомогательные параметры запуска **order.exe [options]**, где **[options]** может быть набором нескольких параметров (без учета регистра):

- **/INSTANCE** – позволяет запустить больше одного клиента на локальном компьютере.
- **/REGALL** – принудительная регистрация библиотек.
- **/NOREG** – не регистрировать библиотеки при запуске.



*Примечание.* Если с момента предыдущего запуска изменилась версия клиента, то автоматически очищается кэш и перерегистрируются библиотеки. Если указаны одновременно **/NOREG** и **/REGALL**, то перерегистрация библиотек не произойдет не только при этом запуске, но и при последующих запусках клиента этой версии.

- **/USER:Имя пользователя** – позволяет указать пользователя в командной строке, который будет подставляться в диалог входа в систему.
- **/PASSWORD:Пароль** – позволяет указать пароль пользователя в командной строке, для автоматического входа в систему.
- **/CACHE:Путь к директории** – месторасположение кеша.

*Примечание.* Имя директории в которой располагается кэш, состоит из трех частей:

- первая часть – параметр командной строки (необязателен), например, **/CACHE:separate;**
  - вторая часть – слово **Cache;**
  - третья часть – ID выбранного бюджета. Например, если **ID=3**, то директория будет называться **separate.Cache.3**.
- **/INI:Путь к ini-файлу** – месторасположение ini-файла клиента.

*Примечание.* Если параметр не задан, то поиск ini-файла осуществляется в директории запускающего файла.

- **/LOG:Путь к директории** – расположение директории для логов.

*Примечание.* Если переданная директория не существует, она будет создана. Если этот параметр не задан, то директория в которой располагается запускающий файл проверяется на доступность для записи. Если не доступна, то директория для логов создается рядом с директорией **Cache** с именем **LOG**.

Интерфейс online-клиента можно настраивать под конкретные потребности (ширина колонок в таблице, отображение конкретных колонок и т.д.), при этом данные о настройках хранятся в реестре ОС **Win32** и остаются актуальными даже после обновления версии клиента.

Допустимо устанавливать online-клиент на ПК под управлением ОС **Windows 98 SE**.

*Примечание.* Соединение *online*-клиента по HTTP-протоколу следует использовать, когда между хостом *online*-клиента и сервером приложений находится один сетевой шлюз или более (например, прокси-сервер, брандмауэр и т.д.).

Параметры ini-файла:

- **Interactive** – определяет будет ли выводиться при потере связи с сервером приложений диалог о повторной попытке соединения, значение по умолчанию равно **false**.
- **Server** – задает местонахождение сервера приложений. Задается в формате: *имя сервера:порт*;[*имя сервера:порт*].
- **ClientUpdatePath** – задает путь к папке, в которой лежит обновление клиента. По умолчанию значение равно **noupdate**, что означает не проводить поиск обновления.
- **SessionProvider** – задает местонахождение провайдера сессий. Если параметр не задан, то его значение формируется автоматически на основе параметра **Server**.
- **DataSetProvider** – задает местонахождение провайдера данных. Если параметр не задан, то его значение формируется автоматически на основе параметра **Server**.
- **TransactEngine** – задает местонахождение процессоров. Если параметр не задан, то его значение формируется автоматически на основе параметра **Server**.

*Примечание.* Если указан параметр **Server** и параметры **SessionProvider**, **DataSetProvider**, **TransactEngine**, то последние имеют более высокий приоритет. Для указания местонахождения нескольких серверов приложений значения параметров можно указать через символ «;» (точка с запятой).

**Пример.** `SessionProvider=corbaloc:iiop:1.2@server1:2001/SessionProvider;  
corbaloc:iiop:1.2@server2:2001/SessionProvider`

**Пример.** Простой ini-файл:

`Interactive=true Server=localhost:2001`

`ClientUpdatePath=noupdate`

**Пример.** ini-файл с заданием расширенных параметров:

```
Interactive=true
```

```
SessionProvider=corbaloc:iiop:1.2@localhost:2001/SessionProvider
```

```
DataSetProvider=corbalocsiiop:1.2@localhost:2001/DataSetProvider
```

```
TransactEngine=corbaloc:iiop:1.2@localhost:2001/TransactEngine
```

#### 4.5.7 Установка OLAP сервера

Порядок установки **OLAP** сервера:

1. Установить сервер **Tomcat** из сборки (в директорию **%TOMCAT%**).
2. Скопировать **/web/saikuUI.war** в **/webapps/**.
3. Скопировать **/web/olapreport.war** в **/webapps/**.
4. Создать в директории **/bin** скрипт **setenv.bat** со следующим содержанием:

```
@echo off
```

```
SET JAVA_TOOL_OPTIONS=-Dfile.encoding=UTF8
```

5. Запустить сервер **/bin/startup.bat** и дождаться распаковки приложений.
6. Остановить сервер.
7. Настроить подключение к БД и используемые схемы для **OLAP** в файле **/webapps/olapreport/WEB-INF/classes/saiku-datasources/schemaProperties**.
8. Снова запустить сервер.

#### 4.5.8 Настройка сжатия данных, передаваемых/отсылаемых между сервером и клиентом

Для сжатия данных, передаваемых/отсылаемых по сети между сервером и клиентом необходимо в файле **order.ini** указать следующие параметры:

```
ServerCompression=True
```

```
ClientCompression=True
```

## 4.6 Настройка системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ»

### 4.6.1 Настройка ролей организаций

Роль организации влияет на работу пользователей, относящихся к этой организации. Положение организации в иерархии организаций определяет видимость документов для пользователей. Пользователи видят документы своей организации и организаций в подчиненных группах списка.

### 4.6.2 Настройка ролей пользователей

Учетные записи пользователей системы делятся на *системные* и *пользовательские*.

*Системные* учетные записи необходимы для функционирования системы и ее настройки. Они создаются загрузкой данных при настройке системы.

*Пользовательские* учетные записи необходимы непосредственно для работы пользователей. Создаются администратором системы при создании новых учетных записей пользователей системы.

*Примечание.* Подробнее о ролях пользователей см. в документации [«БАРМ.00003-45 32 01-4 Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Блок администрирования. Подсистема администрирования системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Настройка ролей пользователей. Руководство администратора»](#).

### 4.6.3 Настройка пользовательского меню для Win-клиента

Описание настройки меню для win-клиента см. в документации [«БАРМ.00003-45 32 01-3 Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Блок администрирования. Подсистема администрирования системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Настройка и сервис системы. Руководство администратора»](#).

#### 4.6.4 Заполнение справочников системы

Справочники используются для заполнения документов. После заполнения в реквизитах документа, как правило, хранятся значения полей объекта справочника, а не ссылка на объект справочника. Поэтому после заполнения документа изменения в справочнике не отражаются на документе.

Доступ к справочникам осуществляется через пункт меню **Справочники**. В процессе эксплуатации системы все справочники необходимо поддерживать в актуальном состоянии.

#### 4.6.5 Выполнение XML-заданий

При необходимости выполнить какое-либо xml-задание, в том числе и размещенное в файле, (например, в каталоге */XML/...*) можно воспользоваться специальным сервисом меню **Сервис**→**XML-задания**. В данном диалоговом окне доступно вручную набрать xml-задание или открыть файл с таким заданием и выполнить его.

Необходимость может возникнуть при автоматическом формировании (заполнении, обновлении и изменении) справочников системы.

#### 4.6.6 Настройка планировщика заданий

Описание настройки работы планировщика заданий см. в документации [«БАРМ.00003-45 32 01-3 Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Блок администрирования. Подсистема администрирования системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Настройка и сервис системы. Руководство администратора»](#). При формировании данных в БД расписание планировщика заполняется типовыми заданиями, которые, при необходимости, можно изменить и дополнить.

## 4.6.7 Настройка подсистемы ЭП

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» позволяет работать с электронной подписью (ЭП) и сертифицированным в России криптографическим ПО **CryptoPro CSP** версии **3.0-3.6** или **ViPNet CSP 3.2**.

В системе документооборота ЭП предназначена для уникальной идентификации создателя документа и для гарантированной доставки именно этого документа в БД системы (не позволяет подменить документ при передаче в незащищенных сетях).

Описание настройки и использования средств ЭП в системе см. в документации [«БАРМ.00003-45 32 01-8 Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Блок администрирования. Подсистема администрирования системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Электронная подпись. Руководство администратора»](#).

При использовании кластера серверов приложений для оптимизации необходимо выделить и определить параметрами в файле свойств всех серверов приложений какой из серверов будет заниматься проверкой подлинности ЭП.

## 4.6.8 Подключение шаблонов печатных форм

Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» позволяет использовать при выводе на печать электронных документов использовать созданные администратором шаблоны печатных форм. Доступны для подключения шаблоны, написанные на технологии **BIRT**.

В качестве шаблона можно указывать не только \*.xlt-файлы, но и \*.rptdesign-**файлы шаблонов BIRT**, через специальные атрибуты также можно указать тип выходного файла и эмиттер для XLS-файлов.

Для шаблонов **BIRT** создается отдельная папка. Путь к ней задается в параметре **azk.report.birt.templatepath** в **Server.properties**:

```
# Путь к шаблонам печатных форм документов на BIRT  
azk.report.birt.templatepath=client/Reports/
```

**Внимание!** Шаблоны протоколов, ранее сделанные на **BIRT**, из папки в `module\bft.gz\reports` нужно переложить в новую папку!

Шаблоны печатных форм ЭД подключаются через системный справочник *DocumentReports*, который заполняется пролитием файла *documentreports\_birt.xml*.

*Примечание.* В файле *XML\Example\documentreports\_birt.xml* приведен пример пользовательского файла с данными справочника.

Возможность подключения для части документов осуществляется на основании лицензии:

- Заявка на закупку (класс 1),
- Сводная заявка на закупку (класс 2);
- Решение о размещении заказа (класс 3);
- Решение о проведении конкурса (класс 4);
- Решение о проведении запроса котировок (класс 5);
- Контракт (класс 8);
- Счет (класс 9);
- Конкурсная заявка поставщиков (класс 10);
- Котировочная заявка поставщиков (класс 11);
- Решение о проведении аукциона (класс 28);
- Аукционная заявка поставщиков (класс 29);
- Решение о проведении предварительного отбора (класс 30);
- Заявка поставщика на участие в предварительном отборе (класс 31);
- Договор (класс 32);
- Решение о проведении торгов на ЭТП (класс 36);
- Заявка на участие в открытом аукционе в электронной форме (класс 37);
- Сведения об исполнении контракта (класс 45);
- Решение о проведении запроса предложений (класс 49);
- Заявка на участие в запросе предложений (класс 50);
- Отказ заказчика в принятии банковской гарантии (класс 52);
- Закупка (класс 53);
- План закупок (класс 54);
- План-график (класс 55);
- Замечания к закупке (класс 56).

## 4.7 Настройка обновления системы

Обновлять систему рекомендуется после предоставления новой версии сборки компанией-разработчиком.

Администратору рекомендуется развернуть отдельно тестовый вариант системы и первоначально обновить именно его, а после успешного обновления тестового варианта обновлять рабочую систему.

Перед обновлением рекомендуется просмотреть краткие описания сделанных изменений в файле **changelog.txt**. Процедуру обновления системы можно разделить на следующие этапы:

1. обновление структуры БД;
2. обновление сервера приложений;
3. обновление данных в БД;
4. обновление online-клиентов;
5. обновление web-сайта.

Перед обновлением администратору необходимо выполнить следующие мероприятия:

- сделать резервную копию БД;
- сделать резервную копию обновляемой сборки.

После успешного обновления следует еще раз сделать резервную копию БД.

### 4.7.1 Обновление структуры БД

Обновление структуры БД можно разделить на следующие этапы:

1. подготовка файлов сборки для обновления структуры БД;
2. настройка необходимых параметров в файле свойств;
3. запуск обновляющего скрипта.



Для обновления структуры БД необходимо наличие подкаталогов сборки **/SQL** и **/server**, а также корректно установленную **JDK** и настроенную переменную окружения **JAVA\_HOME**.

Для обновления структуры БД используется тот же файл свойств (**/SQL/oracle.properties** для СУБД **Oracle**), что и при первоначальной установке.

Настройка необходимых параметров в файле свойств аналогична настройке при первоначальном формировании структуры БД.

Для старта процедуры обновления необходимо выполнить скрипт **/SQL/dbupdate.cmd**.

Данный скрипт при запуске проверяет какие были выполнены sql-команды (файлы **\*.sql**) в текущей (обновляемой) БД. Последовательность sql-файлов приведена в файлах **\*.lst**. При обнаружении невыполненных sql-команд скрипт спросит о том, выполнять ли их. Необходимо ответить утвердительно. Если неисполненных **sql**-команд не было найдено, то скрипт уведомит Вас об этом. Результаты выполнения sql-команд заносятся в файл **order\_ib.log**.

После утвердительного ответа скрипт выполнит обновление структуры БД. В случае неудачного выполнения какой либо sql-команды, необходимо в файле **order\_ib.log** выяснить на какой sql-команде произошел сбой. Затем найти sql-файл (файл с расширением **\*.sql**) с таким же именем, как и невыполненная sql-команда, но с символом «**\_**» (нижнее подчеркивание) вначале. Далее необходимо запустить откат сбойной sql-команды скриптом **sql.cmd** и в качестве параметра передать имя найденного sql-файла с символом подчеркивания впереди.

После откатывания сбойной sql-команды необходимо повторно запустить обновляющий структуру скрипт **dbupdate.cmd**.

***Примечание.** Каталог **/SQL/** содержит набор файлов с **sql**-командами. **SQL**-файлы без символа подчеркивания предназначены для прямого выполнения **sql**-команд, а файлы с символом подчеркивания в начале имени для отката выполненных (или не до конца выполненных) **sql**-команд **sql**-файла прямого действия.*

## 4.7.2 Обновление сервера приложений

Обновление сервера приложений включает в себя:

1. подготовку и замену файлов сборки для обновления сервера приложений;
2. настройку файла свойств сервера приложений.

Для обновления сервера приложений необходимо заменить файлы и папки старой сборки на соответствующие файлы и папки из новой сборки:

1. переместить в архивное место старое содержимое корневого каталога сервера приложений;
2. скопировать в корневой каталог сервера приложений необходимый набор файлов и папок.

Допускается оставить старый файл свойств (уже настроенный из старой сборки), если в нем не добавлялись/удалялись/изменялись параметры (см. **changelog.txt**).

## 4.7.3 Обновление данных в БД

Обновление данных в БД проводится аналогично первоначальному формированию данных.

*Примечание. Не нужно запускать выполнение xml-заданий из подкаталога сборки /XML/Example.*

Процедура обновления данных в БД:

1. настройка файла свойств скрипта, выполняющего xml-задания (**/XML/xml.properties**);
2. запуск сервера приложений, соединенного с обновляемой БД;
3. запуск **/XML/xml.cmd @all.lst** или **/XML/xml.cmd @slaveall.lst**;
4. перезапуск сервера приложений.

#### 4.7.4 Обновление online-клиентов

Описание настройки автоматического обновления online-клиентов см. в документации «[БАРМ.00003-45 32 01-3](#) Система «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Блок администрирования. Подсистема администрирования системы «АЦК-Госзаказ»/«АЦК-Муниципальный заказ». Настройка и сервис системы. Руководство администратора».

Если отключено автоматическое обновление online-клиента, то для обновления необходимо заменить файлы в каталоге клиента на новые из каталога */Client* новой сборки. Файлы дополнительных библиотек не меняются (о необходимости их сменить будет сообщено дополнительно).

#### 4.7.5 Обновление web-сайта

Обновление web-сайта заключается в выполнении следующих этапов:

1. Остановить web-сервер **Tomcat**.
2. Удалить директории с содержимым:

```
.../tomcat/work  
.../tomcat/webapps/ROOT  
.../tomcat/webapps/admin
```

3. Удалить файл **/tomcat/conf/Catalina/localhost/ROOT.xml**.
4. Скопировать архивные файлы **%BUILD%/website/ROOT.war**, **admin.war**, **static.war** web-сайта из новой сборки в папку **/tomcat/webapps**.
5. Запустить web-сервер **Tomcat** (**tomcat/bin/startup.sh** аналогично первоначальной установке). По сообщениям сервера убедиться в успешной распаковке **ROOT.war**, **admin.war**, **static.war** и присоединении к серверу приложений.
6. Настроить необходимые параметры в файле свойств **/tomcat/webapps/ROOT/WEB-INF/classes/web.properties** (аналогично первоначальной установке).
7. Убедиться в ответе сервера на ссылку [http://\[адрес WEB-сервера\]](http://[адрес WEB-сервера]).
8. Убедиться в ответе сервера на ссылку [http://\[адрес WEB-сервера\]/admin](http://[адрес WEB-сервера]/admin).

*Примечание.* Обновления web-клиента не требуется (разве что обновления, иницицированные разработчиком браузера с целью повышения безопасности операционной системы).

## 4.8 Настройка локальной сети

При построении сети с системой «АЦК-Госзаказ»/«АЦК-Муниципальный заказ» следует руководствоваться общими принципами построения корпоративных сетей с выходом во внешние сети.

## 4.9 Настройка журнализации событий и отладочных режимов

Можно выделить журнализацию работы пользователей и журнализацию работы сервера приложений.

*Журнализация работы пользователей* – журнал работы пользователей ведется для каждого документа и в целом для системы. Журнал обработки документа доступен всем пользователям, обладающим правами для просмотра этого документа. Общий журнал доступен только администратору.

*Журнализация работы серверов* – включение отладочных режимов или режимов с работы с более детальной журнализацией целесообразно при поиске причин неисправностей. При нормальной работе включение таких режимов нецелесообразно, так как ведет к замедлению работы серверных компонент и существенному увеличению объемов файлов журнализации.

Управление детализацией журнала работы сервера приложений в файле **server.properties**. Можно управлять уровнем журнализации сервера приложений (задается в **server.properties** в строке **log4j.rootLogger=DEBUG, stdout, file**) и потоками журнализации.

Чем выше уровень журнализации, тем меньше сообщений попадает в журнал.

Возможные значения уровня: **DEBUG, INFO, WARN, ERROR, FATAL, OFF**.

```
log4j.rootLogger=DEBUG, stdout, file
```

Журнализация в консоль:

```
log4j.appender.stdout=com.bssys.log.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{dd.MM.yy HH:mm:ss}, %p, %c{1}, %x, %m%n
```

Уровень журнализации сервера приложений в стандартный поток вывода.

Возможные значения см. в **log4j.rootLogger**:

```
log4j.appender.stdout.Threshold=INFO
```

Чтобы включить отладочный режим web-сервера, необходимо выполнить следующие действия:

1. Остановить сервер **Tomcat**.
2. Отредактировать файл **/TOMCAT/webapps/ROOT/WEB-INF/classes/web.properties**, файл должен содержать:

```
web.debug=yes
web.debug_mode=on
```

3. Перезапустить сервер **Tomcat**.
4. Убедиться в ответе сервера на ссылку <http://адрес WEB-сервера>, над полями ввода логина/пароля должна появиться ссылка **debug**.

## 4.10 Настройка оптимизации работы системы

Для оптимизации работы системы необходимо настроить совместную работу web-сервера с **Apache2**.

Чтобы настроить совместную работу web-сервера с **Apache2**, необходимо выполнить следующие настройки:

1. Проверить наличие в системе **Apache** (пакет **httpd 2.0.40**).
2. Проверить наличие установленного сервера **Tomcat**.
3. Установить пакет **mod\_jk2-2.0.4-2jpp.i386.rpm**.
4. Отредактировать **/etc/httpd/conf/httpd.conf**.

```
LoadModule deflate_module modules/mod_deflate.so
LoadModule jk2_module modules/mod_jk2.so
SetOutputFilter DEFLATE
DeflateFilterNote ratio
LogFormat "%r" %b (%{ratio}n) deflate
CustomLog logs/deflate_log deflate
```

5. Скопировать файл **workers2.properties** в директорию **/etc/httpd/conf/**.
6. Скопировать файл **jk2.properties** в директорию **/conf/**.
7. Перенастроить **Tomcat** на порт **8080**.
8. Отредактировать файл **/conf/server.xml** в строке:  

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector" ..
```

и на следующей строке исправить **port="80"** на **port="8080"**.
9. Отредактировать файл **/conf/server.xml** в строке:  

```
<Connector port="8081" protocol="HTTP/1.1"
```

и на следующей строке исправить **compression="on"**.
10. Перезапустить **Apache httpd** и **Tomcat**.
11. Убедиться в ответе сервера на **http://127.0.0.1/**.

## 4.11 Настройка механизма журналирования изменений системных параметров

Для включения ведения истории изменений необходимо выполнить скрипт **historymodify.xml**:

```
<?xml version="1.0" encoding="windows-1251" ?>
<REFERENCE ref_name="SYSPARAM" action="perform" NAME="historymodify.enabled">
<SYSPARAM CAPTION="Журналирование изменения справочников и документов">
<PARAM_VALUE><![CDATA[SYSPARAM=1]]></PARAM_VALUE>
</SYSPARAM> </REFERENCE>
```

В разделе **CDATA** указывается объект журналирования (**SYSPARAM**), а также режим журналирования:

- 0 (или пусто) – не запрашивать комментарий при сохранении измененного объекта;
- 1 – запрашивать комментарий, но он не является обязательным;
- 2 – запрашивать комментарий, выводить ошибку и не сохранять изменения в случае, если комментарий не введен.

После выполнения xhtml-скрипта журнал доступен по нажатию комбинации клавиш «Ctrl + F6».

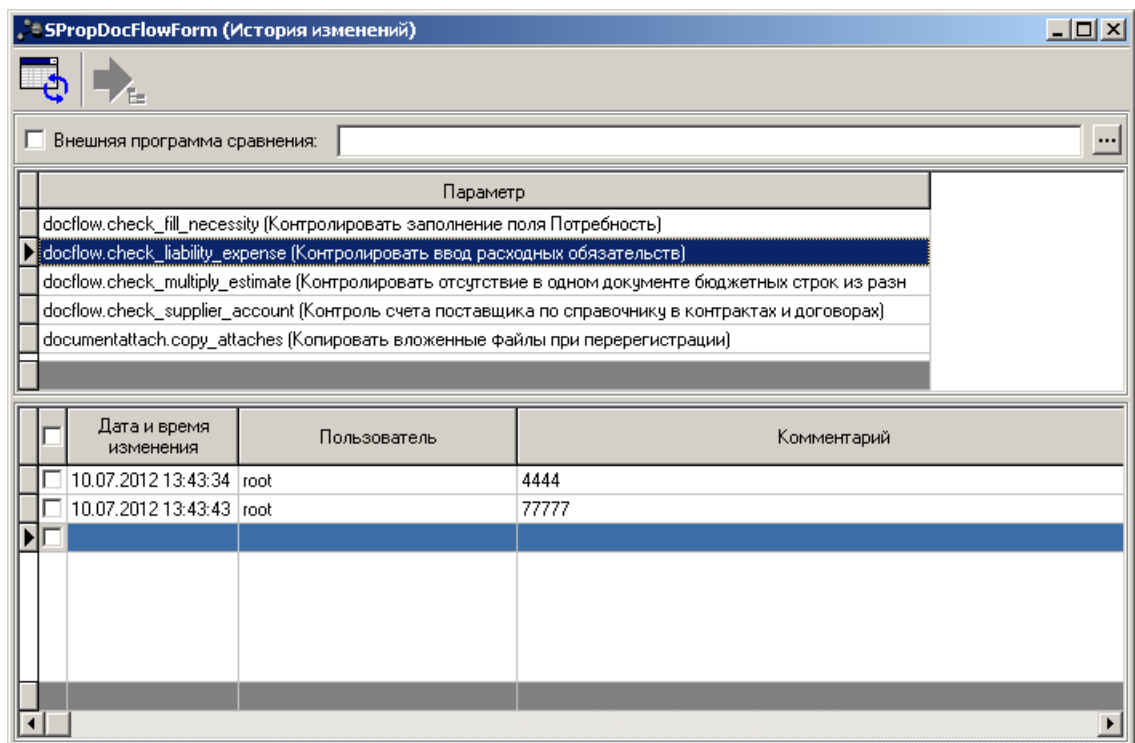


Рисунок 19 – Форма истории изменений документа



5

# Сопровождение системы



## 5.1 Задачи сопровождения системы

Целями сопровождения системы является поддержание ее в работоспособном и актуальном состоянии, исключении возможности потери данных.

В задачи сопровождения системы входит:

- резервное копирование и архивирование БД, файловые УХЗ и журнального файла сервера приложений;
- идентификация ошибок и их устранение;
- поддержка пользователей системы;
- другие функции администратора системы.

## 5.2 Резервное копирование и архивирование БД и журнального файла сервера приложений

Рекомендуется ежедневно делать резервную копию БД, файловых УХЗ, а также журнального файла сервера приложений (всех журнальных файлов серверов приложений в кластере).

Также необходимо делать резервные копии БД и log-файла перед началом обновления (каких-либо других операций по настройке/перенастройке серверов приложений и СУБД).

Рекомендуется хранить на отдельных (надежных) носителях информации данные резервные копии за каждый день как минимум один месяц (в идеале – хранить всегда каждую резервную копию).

Создание резервных копий БД подробно описано в разделе [Создание резервной копии БД](#)<sup>[99]</sup>.

Создание резервной копии log-файла можно разделить на следующие этапы:

1. копирование в отдельный каталог log-файла;
2. архивирование его и перенос на отдельный носитель.

Журнальный (log) файл содержит все операции, выполненные сервером приложений с момента последнего старта. Существует возможность настроить ведение журнального

файла так, что он будет сохранять выполненные операции сервера приложений за все время работы, вне зависимости от перезапуска сервера приложений.

Однако рекомендуется оставить режим перезаписывания log-файла при перезапуске сервера приложений. Но следует учесть, что перед повторным стартом необходимо скопировать log-файл в отдельный каталог, т.к. сервер приложений его полностью очистит от старых записей.

### 5.3 Идентификация ошибок и их устранение

Ошибки, возникающие при работе с системой, можно разделить на следующие:

- ошибки, инициируемые СУБД;
- ошибки, инициируемые сервером приложений;
- ошибки, инициируемые online-клиентом;
- ошибки, инициируемые web-сервером;
- ошибки, инициируемые web-сайтом;
- ошибки, инициируемые web-клиентом (браузером).

Сервер приложений записывает все возникающие ошибки и их возможное описание в log-файл.

Часть ошибок это попытка доступа к функциям системы, не имея на это прав, некорректная работа той или иной части системы.

Помимо log-файла ошибки также выводятся в стандартных диалоговых окнах online-клиента, при работе с которым возникла эта ошибка.

При возникновении ошибки следует прочесть ее описание и предпринять меры по ее устранению (если это требуется).

Если не удастся устранить или даже выявить причину ошибки, то следует обращаться за помощью в отдел сопровождения разработчика. При этом необходимо предоставить log-файл (либо полную копию, либо кусок с сообщениями об ошибке), описать ситуацию, при которой возникла эта ошибка и, если потребуется, выслать архивную копию БД, при работе с которой ошибка проявилась.

## 5.4 Поддержка пользователей системы

Для нормального функционирования системы желательно организовать поддержку конечных пользователей системы для следующих задач:

- установки и обновления клиентских приложений у конечных пользователей;
- консультирования по вопросам интерфейса системы;
- консультирования по вопросам документооборота в системе;
- консультирования по возникающим нестандартным ситуациям (ошибкам);
- обобщения информации об ошибках и передачи ее в отдел сопровождения разработчика.

Для этих целей можно организовать общение с пользователями по телефону и/или через интернет-службы обмена мгновенными сообщениями (**ICQ**, **IRC** и т.д.).

## 5.5 Другие функции администратора системы

Помимо вышеописанных функций администратор системы должен:

- постоянно поддерживать в актуальном состоянии служебные системные справочники;
- производить обновления системы и смежного ПО;
- организовывать защиту серверов от внешних и внутренних сетевых атак, а также от проникновения вирусов на них;
- прочие функции.



6

# Порядок технического обслуживания

Регламентированное техническое обслуживание информационной системы (далее ИС) АЦК включает в себя:

- Регламентно-профилактические мероприятия, включая мониторинг системы;
- Проверку работоспособности;
- Проверочное восстановление резервной копии.

Регламентированное техническое обслуживание включает в себя работы по техническому обслуживанию:

- Прикладного программного обеспечения (далее ППО), в состав которого входят сервера приложений (далее СП);
- Системного программного обеспечения (далее СПО), в состав которого входит операционная система (далее ОС), компоненты java, web-сервера tomcat или/i Apache, балансировщик nginx и пр.
- Системы управления базами данных (далее СУБД).

Все регламентные работы регистрируются в специальном журнале регистрации работ.

Регламентированное техническое обслуживание оборудования, требующее приостановки сервисов обслуживаемых систем, должно выполняться в согласованное время (технологическая пауза), выделяемое для выполнения работ, требующих остановки сервиса.

## 6.1 Регламентно-профилактические мероприятия

Регламентно-профилактические мероприятия выполняются на периодической основе и призваны минимизировать риски возникновения нештатных ситуаций функционирования ППО, СПО и СУБД.

### 6.1.1 Регламентно-профилактические мероприятия по ППО

Регламентно-профилактические мероприятия по ППО включают в себя следующие действия, выполняемые с определенной периодичностью (в таблице приведены шаблонные наименования файлов свойств, конфигураций и запускающих файлов):

Таблица 22 – Действия, выполняемые в рамках регламентно-профилактических мероприятий по ППО

Мероприятия	Периодичность выполнения
1. Проверка содержимого конфигурационных файлов СП, включающие в себя файлы: <ul style="list-style-type: none"> <li>• <i>StartServer.bat/StartServer.sh</i> (в зависимости от ОС);</li> <li>• <i>Azk2Server.properties, Server.properties</i>;</li> <li>• <i>Azk2logback.xml</i>;</li> <li>• <i>lconflserver.xml</i> (для веб-сервера).</li> </ul>	До и после выполнения процедуры обновления сборки системы\изменения количества серверов приложений.
2. Проверка стартового файла СП <i>start.bat/start.sh</i> . <i>Примечание. Необходимо обратить внимание на наличие основных java параметров <b>XXMX</b>, <b>XXMS</b>, сборщик мусора <b>UseG1GC</b>, интервал его запуска <b>MaxGCPauseMillis</b>.</i>	До и после выполнения процедуры обновления сборки системы\изменения количества серверов приложений.
3. Выполнение резервирования конфигурационных файлов АЦК (файл свойств, настройки логов, настройки Tomcat, настройки storage, запускающие файлы).	До и после выполнения процедуры обновления\реконфигурации серверов\гипервизоров\СХД.
4. Выполнение резервирование сборки АЦК.	До и после выполнения процедуры обновления\переконфигурации серверов\гипервизоров\СХД.
5. Резервирование прикрепленных файлов (attach).	Ежедневное резервирование дельты, резервирование и хранение должно осуществляться на отдельные дисковые массивы отдельных серверов\СХД.
6. Сравнение файлов используемой сборки с эталонной (сравнение файлов можно проводить с помощью различных утилит, например, winmerge).	До и после выполнения процедуры обновления сборки системы\ изменения количества серверов приложений.
7. Полное обновление сборки веб-сервера путем удаления содержимого папок <b>webapps</b> , <b>work</b> , <b>temp</b> и копирование <i>azk.war</i> в каталог <b>webapps</b> с последующим запуском <i>tomcat</i> .	После проведения процедуры обновления сборки системы.
8. Мониторинг виртуальной Java машины (далее JVM) для выявления случаев утечки памяти (например, с помощью утилиты <i>jvisualvm</i> ).	Периодически, при возникновении проблем с производительностью. Постоянный мониторинг опасен тем, что дополнительно нагружает систему, замедляя работу.
9. Мониторинг логов сервера приложений на наличие ошибок типа: <i>Java heap space</i> , <i>Out of memory</i> , sql-ошибок и пр. Мониторинг логов веб-сервера на наличие ошибок.	Постоянно.
10. Мониторинг и контроль выполнения заданий планировщиков.	Постоянно.
11. Мониторинг и контроль за «системными событиями».	Постоянно.

Мероприятия	Периодичность выполнения
12. Использование эталонных копий ПО.	1 раз в месяц и каждый раз при внесении изменений в эталонные копии (проведение процедуры обновления сборки системы).
13. Мониторинг межсистемного обмена сообщениями.	Постоянно.

### 6.1.1.1 Установка эталонной сборки

Установку эталонной сборки Системы условно можно разделить по типу используемого клиента:

- Установка серверной части Системы;
- Установка толстого клиента;
- Установка тонкого клиента.

Установка серверной части и толстого (win) клиента:

1. Выполнить резервирование сборки на физически отдельном диске.
2. Загрузить с облачного ресурса <https://bftcloud.bftcom.com> требуемую сборку. В зависимости от используемых продуктов, догрузить связанные модули, например, СОФИТ, модуль отчетов.
3. Переименовать каталог старой сборки.
4. Создать каталог новой сборки.
5. В каталог новой сборки последовательно разархивировать загруженные модули, начиная непосредственно с серверной части Системы.
6. В конфигурационные файлы *Server.properties* \ *Azk2Server.properties* эталонной сборки перенести настройки, используемые в предыдущей версии.
7. Настроить уровень журнализации в *Azk2logback.xml* (Возможные значения уровня журнализации: *ALL*, *TRACE*, *DEBUG*, *INFO*, *WARN*, *ERROR*, *OFF*).
8. Для <%GZ%> в конфигурационном файле *Server.properties* указать сервер лицензии. Для остальных АЦК – поместить в корневой каталог файл лицензии *bft.lic*.
9. Добавить параметры распределения java-памяти в исполняемые файлы

`\server\bin\start.sh` (ОС Linux) или `\server\bin\start.bat` (ОС Windows).

10. В каталог новой сборки разархивировать клиентскую часть (папка `client`). Для `<%GZ%>` данная папка не упакована, необходимо скопировать.
11. Настроить файл конфигурации толстого клиента `Azk2CInt.ini` по аналогии с предыдущей версии.
12. Загрузить с облачного ресурса <https://bftcloud.bftcom.com> файлы папки VCL, переместить их в папку `client`.

Установка тонкого (web) клиента:

1. Выполнить последовательно пункты по установке серверной части ([пп. 1-9](#)<sup>143</sup>).
2. Выполнить резервирование файлов запуска Tomcat (**startup**, **shutdown**, **setenv**).
3. Очистить каталоги Tomcat: **webapps**, **work**, **temp**.
4. Скопировать `azk.war` в каталог **webapps**.
5. Проверить существование файла `tomcat\bin\setenv.bat` (`tomcat\bin\catalina.bat` для `<%GZ%>`) (ОС Windows) или `tomcat\bin\setenv.sh` (`tomcat\bin\catalina.sh` для `<%GZ%>`) (ОС Linux) с параметрами распределения java памяти. При необходимости отредактировать в соответствии с системными параметрами сервера.
6. Запустить Tomcat.

### 6.1.1.2 Параметры распределения JAVA памяти

**Внимание!** По всем изменениям параметров рекомендуется предварительная консультация с техническим сопровождением.

Подробнее об основных параметрах:

- 1 **-Xmx** – количество максимально возможной используемой для СП/web сервера JAVA памяти.

В данном параметре необходимо указывать объем оперативной памяти (далее ОП) сервера, выделенный под сервер приложения. Следует учитывать, что для работы операционной системы также необходима память.



*Например, если объем ОП сервера 16 ГБ, то непосредственно для работы системы нужно оставить 2 ГБ, параметр **-Xmx** будет выглядеть как **-Xmx14G**.*

- 2 **-Xms** – количество начальной выделяемой JAVA памяти. Рекомендуется выставлять величину, равную **Xmx**.
- 3 **-XX:+UseG1GC** – активирует использование сборщика мусора G1GC.
4. **-XX:MaxGCPauseMillis=100** – длительность времени в миллисекундах между вызовом сборщика мусора. Для G1GC исполняет роль максимально установленного промежутка.
5. В зависимости от количества процессоров сервера, возможно использование настроек по параллелизму сбора мусора, таких, как:
  - **-XX:ParallelGCThreads=n** значение потоков сборщика мусора;
  - **-XX:ConcGCThreads=n** значение потоков для пометки мусора;Настройка параллелизма сборки мусора. Рекомендовано количество ядер минус два и пополам для первого параметра, и ещё пополам для второго.

*Например, на сервере имеется 24 ядра, тогда настройки будут выглядеть следующим образом:*

**-XX:ParallelGCThreads=11** (пример расчета:  $(24-2)/2$ )

**-XX:ConcGCThreads=5** (пример расчета:  $(24-2)/2/2$ )

- 6 **-Djava.awt.headless** – доступность графической консоли. Значение *true* означает, что графическая консоль недоступна.
- 7 **-Djava.net.preferIPv4Stack=true** – работа только с протоколом IP4.

### 6.1.1.3 Параметры настроек в конфигурационных файлах АЦК

**Внимание!** По всем изменениям параметров рекомендуется предварительная консультация с техническим сопровождением.

Основные параметры настроек конфигурационных файлов АЦК:

- **azk.db.pool.timeout** – время (в миллисекундах), по истечении которого если ни одно соединение не становится доступным в пуле коннектов к СУБД, сервер возвращает исключение клиенту. Необязательный параметр. Значение по

умолчанию – 5000.

- **azk.db.pool.size** – максимальное количество подключений к серверу базы данных от данного сервера приложений. Необязательный параметр. Значение по умолчанию – 10.
- **azk.db.cachesize** – размер кэша соединения для откомпилированных запросов. Суммарное максимальное количество откомпилированных запросов определяется произведением этого параметра на размер пула (**azk.db.pool.size**). Для СУБД Oracle размер кэша важен, т.к. у него ограничено возможное количество открытых одновременно курсоров. Необязательный параметр. Значение по умолчанию – 50.
- **azk.db.connectionlifetime** – время соединения (в минутах), по истечении которого соединение будет закрыто. Данный параметр обеспечивает освобождение ресурсов в случае неиспользования сервера приложений, а также ротацию коннектов. Необязательный параметр. Значение по умолчанию – 20.
- **azk.db.fetchsize** – задает количество строк, которое JDBC-драйвер запрашивает у сервера БД за один раз при чтении результата запроса. Чем больше значение, тем быстрее выполняется чтение (за счет уменьшения количества сетевых запросов), но тем больше памяти будет зарезервировано под буферы как на стороне сервера БД, так и на стороне СП. Оптимальное по времени и памяти значение зависит как от запроса, так и от количества возвращаемых им строк, поэтому данный параметр рекомендуется применять с осторожностью, возможно только для разработки/тестирования. Значение по умолчанию зависит от JDBC-драйвера, например, для Oracle это 10. Если параметр не указан или равен 0, будет использоваться значение по умолчанию. Большие значения (>1000) значительно ускоряют работу с удаленными БД, особенно это заметно на загрузке серверных кэшируемых справочников и загрузке справочников в кэш Win-клиента при его старте, а также любых других запросах, возвращающих большое количество строк.

#### Настройки пула:

Минимальное количество потоков устанавливается равным количеству процессоров с учетом гипертрединга. Максимальное количество потоков должно быть ограничено и зависеть от мощности сервера. Размер очереди команд лучше ставить большим. Для отключения многопоточной работы пула необходимо установить в параметре **maxpoolsize** значение 0 (ноль).

- **azk.poolexecutor.queuecapacity** – размер очереди команд.
- **azk.context.subcontextpool.size** – размер очереди в пуле.
- **openorb.server.maxThreadPoolSize** – максимальное количество CORBA потоков сервера приложений, обслуживающие клиентские запросы. Если одновременно к серверу обратятся большее количество пользователей, соответствующие обращения встанут в очередь. Используется для ограничения/увеличения пиковой нагрузки на сервер приложений, в зависимости от мощности используемого сервера. Значение по умолчанию – 100.
- **iiop.port** – порт, на котором работает сервер приложений. Предпочтительно задавать это значение из командной строки и иметь общий файл настроек для всех серверов кластера.
- **iiop.hostname** – используется как имя хоста в идентификаторах объектов CORBA. В ситуациях, когда автоматическое определение невозможно, задается этот параметр (для OS/400 необходимо задать в качестве имени хоста, его IP). В случае использования кластера, рекомендуется задавать его с командной строки, при необходимости.
- **iiop.alternateAddr.endpoint1** – если на хосте установлено несколько сетевых интерфейсов, для CORBA необходимо определить порядок их прослушивания.

#### Рекомендации по настройкам параметров конфигурационного файла СП при работе с СУБД PostgreSQL:

*Примечание. Данные рекомендации приведены в качестве примера. Определение необходимых значений в параметрах является индивидуальным. Необходима обязательная консультация с техническим сопровождением.*

#### СП интенсивной работы пользователей:

1. **azk.db.pool.timeout=10** (рекомендовано от 10 до 15 секунд).
2. **azk.db.pool.size=50** (возможно указание меньшего значения) – количество возможных подключений от Системы к базе. Необходимо внимательно изменять значение данного параметра, т.к. под каждое соединение будет выделяться память.
3. **azk.db.cachesize=200** (200-300) – количество запросов на стороне сервера БД кэш запросов. Чем он больше в случае с PG, тем медленнее обрабатывается новый запрос в рамках подключения.

4. **azk.db.connectionlifetime =1** (1-2) —период действия неактивного соединения от АЦК к БД. В случае с PG и пользователями рекомендуется установить значение 1 или 2.
5. **azk.pooledexecutor.queuecapacity =200** (200-300).
6. **azk.context.subcontextpool.size=50** (50-100).
7. **openorb.server.maxThreadPoolSize=500** (500-600) – количество потоков на объекты orb. Влияет на то, сколько объектов при обмене с клиентами может параллельно обрабатываться. Значение по умолчанию – 100. В случае, если на СП одновременно работает 200 пользователей, будет образовываться очередь. Чтобы этого не происходило, необходимо установить значение, равное количеству пользователей, подключающихся к СП.

СП для интенсивной работы по обработке реплик и/или других сервисных заданий (в том числе на выгрузку данных вовне):

1. **azk.db.pool.size=100** (100-200);
2. **azk.db.cachesize=50** (50-100);
3. **azk.db.connectionlifetime=5** (5-10);
4. **azk.pooledexecutor.queuecapacity=100** (100-200).

Не зависящие от типа нагрузки на СП параметры конфигурационного файла:

1. **azk.pooledexecutor.minpoolsize** и **azk.pooledexecutor.maxpoolsize**. Эти параметры связаны с количеством потоков java в параллельной обработке.
2. **azk.pooledexecutor.minpoolsize** – устанавливается равным количеству ядер. Допускается установить в два раза больше чем количество ядер.
3. **azk.pooledexecutor.maxpoolsize** – вычисляется эмпирически. Допускается установить в 4 раза больше, чем количество ядер.

#### 6.1.1.4 Использование утилиты «jvisualvm»

Рассматриваемая бесплатная утилита проста в использовании. Минусом является описываемая выше нагрузка на систему.

Для запуска утилиты необходимо в файл запуска (**start.sh/start.bat** для толстого клиента, **setenv.sh/setenv.bat** для тонкого) внести следующие изменения:

```
-Dcom.sun.management.jmxremote
```

```
-Dcom.sun.management.jmxremote.port=9000
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Djava.rmi.server.hostname=<IP сервера, где стоит утилита>
```

После запуска утилиты в левом окне выбирается PID для наблюдения. После этого, необходимо перейти в правое окно и наблюдать на вкладке **Monitor**, вкладку **Heap**, где отражается использование памяти JVM в реальном режиме времени.

## 6.1.2 Регламентно-профилактические мероприятия по СУБД

Регламентно-профилактические мероприятия по СУБД включают в себя следующие действия, выполняемые с определенной периодичностью:

Таблица 23 – Действия, выполняемые в рамках регламентно-профилактических мероприятий по СУБД

№	Мероприятия	Периодичность выполнения
1.	Сбор статистики СУБД.	Ежедневно.
2.	Сохранение конфигурационных файлов (Для Oracle – <b>init.ora</b> , <b>listener.ora</b> , <b>tnsnames.ora</b> . Для PG – <b>postgresql.conf</b> , <b>pg_hba.conf</b> ) на физическом отдельном диске.	Периодически и перед выполнением обновления/реконфигурации сборки/серверов/гипервизоров/СХД.
3.	Резервное копирование БД	Полная резервная копия СУБД – 1 раз в неделю. Инкрементальная копия – раз в день. Для Oracle\PG – возможно использование связки master-slave; Для СУБД Firebird – ежедневно; Для СУБД Oracle: <ul style="list-style-type: none"> <li>• архивирование журнала транзакции для восстановления на момент времени;</li> <li>• полная резервная копия СУБД (full backup) – 1 раз в неделю;</li> <li>• инкрементальная LVL0 – раз в неделю.</li> </ul> Инкрементальная копия LVL1 CUMULATIV после создания LVL0 – раз в день.
4.	Проверка резервных копий БД – контрольное восстановление из резервной копии БД.	1 раз в неделю.
5.	Мониторинг логов СУБД на наличие ошибок.	Постоянно.

№	Мероприятия	Периодичность выполнения
6.	Проверка наличия невалидных объектов/отключенных триггеров, констрейнтов в БД.	Ежедневно.
7.	Мониторинг количества активных сессии, заблокированных сессий.	Постоянно.
8.	Мониторинг дискового пространства, загрузки CPU, RAM.	Постоянно.
9.	Мониторинг наполненности таблицы RPLLOG. Выполнение запроса: <i>select 'all', count(1) from rpllog union all select 'nines', count(1) from rpllog where generation = 999999999999999;</i>	Периодически, несколько раз в день.
10.	Сверка с эталонной БД.	Периодически: 1 раз в квартал.
11.	Проверка полноты применения sql-скриптов после обновления.	после выполнения процедуры обновления сборки системы.

### 6.1.2.1 Рекомендации по параметрам СУБД

#### 6.1.2.1.1 СУБД ORACLE

Распределение памяти в пропорции: SGA – 90%, PGA – 10%.

SGA (system global area) – это часть общей памяти, которую разделяют между собой все серверные процесс (включая фоновые).

PGA – специфичная для процессов часть памяти известна как программная глобальная область (program global area).

Необходимо выполнить:

```
SQL>alter system set optimizer_index_cost_adj=1 scope=spfile sid='*';
SQL>alter system set open_cursors=5000 scope=spfile sid='*';
SQL>alter system set undo_retention=10800 scope=spfile sid='*';
SQL>alter system set db_securefile=always scope=spfile sid='*';
SQL>alter system set audit_trail=none scope=spfile sid='*';
SQL>alter system set db_ultra_safe=data_only scope=spfile sid='*';
SQL>alter system set recyclebin=off scope=spfile sid='*';
SQL>alter system set deferred_segment_creation = false scope=spfile sid='*';
```

### 6.1.2.1.2 СУБД PostgreSQL

Начальные параметры конфигурации СУБД PostgreSQL следующие:

1. **max\_connections** должно быть менее 1000.
2. **maintenance\_work\_mem** – рекомендуется устанавливать значение 50-75% от размера самой большой таблицы или индекса, но чтобы памяти хватило для работы системы и приложений. Рекомендуется устанавливать значения больше чем **work\_mem**.
3. **shared\_buffers** – рекомендуемое начальное значение – 25% от объема памяти.
4. **work\_mem** – рекомендуемое начальное значение в 2-4% объема оперативной памяти.
5. **max\_worker\_process** – задается равное количеству процессоров сервера СУБД.
6. **random\_page\_cost=1.2**.
7. **seq\_page\_cost=1.2**.
8. **effective\_cache\_size** – для начальной установки рекомендуется следующие размеры: не более 1/3 оперативной памяти, далее можно увеличивать пока хватает дискового пространства (800ГБ и более).
9. **default\_statistics\_target =300**.
10. **autovacuum\_vacuum\_scale\_factor = 0.01**.
11. **autovacuum\_analyze\_scale\_factor = 0.005**.
12. **vacuum\_cost\_limit=1000**.
13. **autovacuum\_vacuum\_cost\_limit=1000**.
14. **log\_line\_prefix=%m [%p]: [%d-1] [%v] [%x] user=%u,db=%d,client=%h**.
15. **superuser\_reserved\_connections=3**.
16. **max\_locks\_per\_transaction = 256**.
17. **max\_pred\_locks\_per\_transaction = 2000**.
18. **idle\_in\_transaction\_session\_timeout=300**.

### 6.1.2.1.3 СУБД Firebird

Начальные параметры конфигурации для данного вида СУБД рекомендуется использовать установленные по умолчанию. В дальнейшем, по результатам мониторинга за состоянием работы СУБД возможно использование оптимизации параметров.

*Примечание.* Оптимизированные файлы конфигурации под определенную версию ОС, СУБД, разрядность СУБД, выделенную память и пр. располагаются на сайте <https://ib-aid.com/en/optimized-firebird-configuration/>.

### 6.1.3 Регламентно-профилактические мероприятия по СПО

Регламентно-профилактические мероприятия по СПО включают в себя следующие действия, выполняемые с определенной периодичностью:

Таблица 24 – Действия, выполняемые в рамках регламентно-профилактических мероприятий по СПО

№	Мероприятия	Периодичность выполнения
1.	Проверка содержимого конфигурационных файлов веб-сервера, балансировщика (nginx/apache) включающие в себя файлы: <ul style="list-style-type: none"> <li>• <i>server.xml</i>,</li> <li>• <i>nginx.conf/httpd.conf/vhttpd2.conf/apache.conf</i>.</li> </ul>	До и после выполнения процедуры обновления сборки системы/изменения количества серверов приложений.
2.	Резервирование конфигурационных файлов (для nginx – <b>nginx.conf</b> , для apache – <b>httpd.conf/vhttpd2.conf/apache.conf</b> , для tomcat – <b>server.xml</b> ) на физическом отдельном диске.	Периодически и перед выполнением процедуры обновления/реконфигурации сборки/серверов/гипервизоров/СХД.
3.	Мониторинг загруженности компонентов JVM.	Постоянно.
4.	Мониторинг балансировщиков (Nginx: <b>connections accepted per sec, connections reading, connections waiting, connections writing, connects active, Number of processes nginx, requests per sec</b> ).	Постоянно.
5.	Проверка наличия обновлений для ОС и своевременная установка.	Периодически, раз в неделю.
6.	Сервисное обслуживание ОС, включающее в себя: чистку кэша\cookies, временных файлов ОС, неиспользуемых\ошибочных ключей реестра. Для ОС Linux проверку <b>sysctl.conf, limits.conf, security.java</b> .	Периодически, раз в месяц.
7.	Проверка журналов событий ОС на наличие ошибок.	Периодически, раз в день.
8.	Мониторинг и контроль за доступностью ресурсов (дисковые хранилища, CPU, RAM).	Постоянный.
9.	Профилактическое обслуживание ИТ-системы (анализ журналов, диагностика, тонкая настройка).	Периодически, 1 раз в месяц.
10.	Мониторинг системных журналов, содержащих информацию об успешности авторизации для выявления угроз безопасности, хакерских атак методом перебора и др..	Постоянно.



### 6.1.4 Мониторинг виртуальной инфраструктуры

Мониторинг виртуальной инфраструктуры необходим при обслуживании виртуальной инфраструктуры, поскольку вследствие различных причин нагрузка на один из виртуальных серверов может возрасти, что приведет к падению производительности всех гостевых систем хоста. В то время как обычная загрузка большинства физических серверов составляет 5-10 процентов, загрузка хоста, поддерживающего виртуальные машины после консолидации, составляет 60-80 процентов и это делает необходимым постоянный мониторинг его активности. В большинстве случаев для мониторинга виртуальных серверов «VMware» используется продукт «Virtual Center», дающий общую информацию о загрузке аппаратных ресурсов серверов. Однако, когда требуется детальный отчет по всем видам ресурсов, его информации недостаточно и целесообразно использовать специализированные продукты. Также для мониторинга виртуальной инфраструктуры можно использовать среду мониторинга «Zabbix».

***Внимание!** Мониторинг виртуальной инфраструктуры должен выполняться регулярно, на постоянной основе.*

### 6.1.5 Наличие тестового СП

Рекомендуется иметь тестовый СП, подключенный к реальной БД, для проверки реконфигураций файлов свойств и настроек ОС. Аналогично, рекомендуется иметь тестовый веб-сервер, подключенный к реальной БД, для проверки реконфигураций файлов свойств и настроек ОС.

### 6.1.6 Мониторинг структуры системы хранения данных

Необходимо постоянно выполнять мониторинг системы хранения данных (далее СХД). Для этого используется универсальная система мониторинга типа «Nagios», «Icinga», «Zabbix», которая будет собирать информацию по протоколу **SNMP** или используется узкоспециализированное ПО от производителей самих СХД.

### 6.1.6.1 Оценка производительности СХД

Для оценки производительности СХД используют три основные метрики:

1. **Service Time**, часто именуемый **latency** или **response time**, измеряется в миллисекундах и обозначает:
  - при чтении: время с момента получения СХД задания на чтение блока информации до отправки запрошенной информации.
  - при записи: время с момента получения записываемого блока информации до подтверждения о его успешной записи.
2. **IO/s** – количество операций ввода вывода в секунду.
3. **MB/s** – количество переданных мегабайт в секунду.

Параметры **IO/s** и **MB/s** тесно связаны между собой размером блока данных, т.е. один мегабайт информации можно записать блоками по 4к и получить 256 операций ввода-вывода, или блоками 64к и получить 16 IO.

### 6.1.6.2 Повышенный Service Time СХД

Для каждого СХД есть максимальное значение **Service Time**, которое соответствует максимальной производительности, т.е незначительное увеличение нагрузки приведет к существенному повышению **Service Time**, вызвав тем самым деградацию требовательных к задержкам приложений. В общих случаях для **All Flash** СХД приемлемым считается **Service time** меньше *1ms*, а для классических СХД до *20ms*. Порог приемлемого **Service time** зависит от числа контроллеров, скорости дисков и модели самой СХД и может отличаться от приведенных значений.

Необходимо учитывать, до какого уровня задержек дисковой подсистемы сохраняется нормальная работоспособность приложения и всегда иметь необходимый запас.

### 6.1.6.3 Утилизация CPU СХД

Утилизация **CPU** контроллеров СХД в общих случаях не должна превышать 70%. Если она постоянно выше 70%, это свидетельствует об отсутствии запаса производительности СХД.

### 6.1.7 Проверка работоспособности

Проверка работоспособности включает в себя контрольное выполнение критических функций после каждого обновления ИС, изменений конфигурации аппаратного обеспечения, а также ежедневная проверка работоспособности системы перед началом рабочего дня.

В число способов проверки работоспособности перед началом дня входит:

- визуальный контроль;
- контроль журнальных файлов ППО и СПО;
- контроль по данным средств мониторинга;
- контроль путем тестовых или диагностических операций с помощью ППО и СПО (без изменения данных).

По результатам проверки работоспособности для каждого ППО и СПО может быть выявлено одно из следующих состояния:

- функционирует в штатном режиме;
- функционирует со снижением производительности или с ограничением функциональности, вследствие чего возможны ограничения качества исполнения производственных процессов;
- функционирует со сбоями или ограничениями, препятствующими исполнению производственных процессов.

В число способов проверки работоспособности после проведения обновления ИС

входит:

- проверка обновления тестовой среды:
  - анализ журналов установки обновлений ИС;
  - по возможности исправление ошибок установки обновлений и подготовка рекомендаций для обновления промышленной среды, включая исправления ошибок установки обновления на тестовой среде;
- при наличии сервера электронной подписи:
  - завершение работы сервера приложений (далее СП) с электронной подписью в последнюю очередь, для <%GZ%> в последнюю очередь – СП лицензии;
  - запустить в первую очередь, для <%GZ%> сервер лицензий, для остальных систем АЦК – СП с электронной подписью.
- проверка отсутствия ошибок при запуске СП.
- проверка версии в окне аутентификации после проведения процедуры обновления (Web/Delphi).
- проверка входа в Систему под пользователем. При наличии электронной подписи, проверка входа в Систему под пользователем с тестовым сертификатом (Web/Delphi).
- проверка работоспособности балансировщика и вход в Систему с его использованием.

Настройка и изменение конфигураций аппаратного обеспечения включают в себя:

- настройку аппаратного обеспечения, в том числе для оптимизации производительности соответствующих информационных систем (подсистем);
- изменение конфигураций аппаратного обеспечения.

После настройки и изменения конфигураций аппаратного обеспечения проверяется работоспособность аппаратного обеспечения системы (подсистемы) и в случае их неработоспособности, восстановлена существовавшая ранее работоспособная конфигурация. Проверка работоспособности выполняется по аналогии с ежедневной проверкой работоспособности.

### 6.1.8 Проверочное восстановление резервной копии

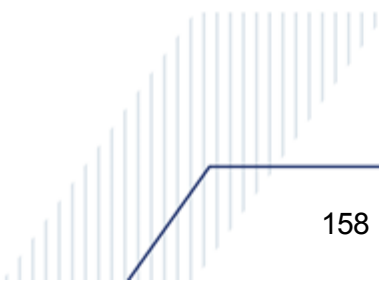
Проверка резервных копий бывает двух видов:

- **Проверка целостности резервной копии** – сверяются контрольные суммы блоков данных резервной копии.
- **Проверка восстановления из резервной копии** – в специальной изолированной среде моделируется процесс восстановления системы из резервной копии, и проверяется, будет ли она работать корректно после восстановления.

*Примечание.* Проверочное восстановление из резервной копии необходимо проводить минимум 1 раз в месяц.



# Аванпост FAM



Avanpost FAM – система единой аутентификации сотрудников в корпоративных ресурсах организации. Система позволяет настроить прозрачную и многофакторную аутентификацию в мобильных и веб-приложениях, приложениях с толстым клиентом, SaaS-сервисах и терминальных решениях.

Функциональные возможности системы позволяют осуществлять поддержку аутентификации по технологии единого входа (SSO, Single Sign-On) при интеграции с Avanpost FAM:

- обеспечение переадресации клиентского приложения в Avanpost FAM для аутентификации по технологии единого входа (SSO);
- получение клиентским приложением от Avanpost FAM одноразового кода доступа, предъявление полученного одноразового кода доступа системе АЦК-Финансы (система);
- передача системой одноразового кода доступа, предъявленного клиентским приложением, в Avanpost FAM;
- получение от Avanpost FAM ключа доступа (access token) и ключа обновления доступа (refresh token) в случае успешной проверки одноразового кода доступа на стороне Avanpost FAM, разрешение доступа клиентскому приложению к системе;
- проверка срока действия ключа доступа (access token) при обращении клиентского приложения к системе;
- предъявление ключа обновления доступа (refresh token) в Avanpost FAM при истечении срока действия ключа доступа (access token) для получения нового ключа доступа (access token) клиентского приложения.

## НАШИ КОНТАКТЫ

### Звоните:

(495) 784-70-00

### Пишите:

bft@bftcom.com

### Будьте с нами online:

[www.bftcom.com](http://www.bftcom.com)

### Приезжайте:

129085, г. Москва,  
ул. Годовикова, д. 9, стр. 17

### Дружите с нами в социальных сетях:



[vk.com/bftcom](https://vk.com/bftcom)



[t.me/ExpertBFT\\_bot](https://t.me/ExpertBFT_bot)

