



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«БЮДЖЕТНЫЕ И ФИНАНСОВЫЕ ТЕХНОЛОГИИ»  
тел./факс: (495) 784-70-00; E-mail: ineed@bftcom.com  
www.bftcom.com

# «Система контроля задач по разработке программного обеспечения» («АЦК-Контроль»)

Руководство по эксплуатации

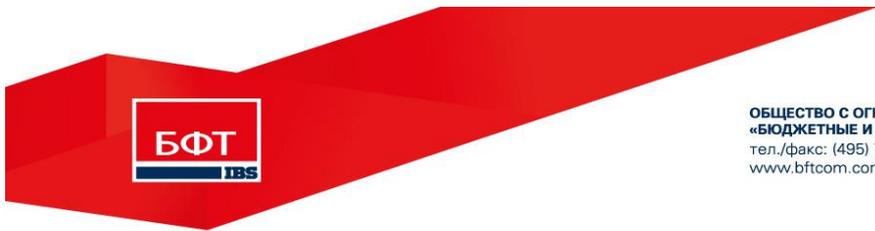
Листов 36

© 2016 ООО «Бюджетные и финансовые технологии»

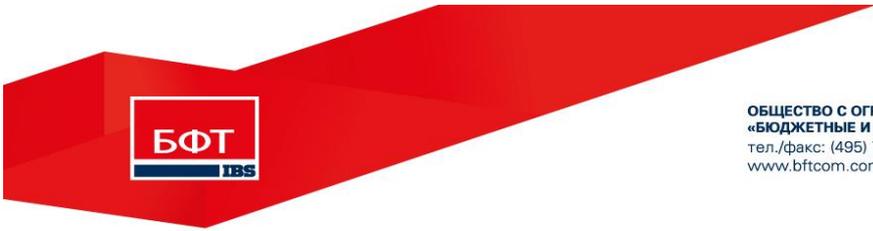


## Оглавление

1. Общие сведения о программе .....	5
1.1. Назначение программы .....	5
2. Структура программы .....	5
2.1. Архитектура системы .....	5
2.2. Компоненты системы .....	6
2.2.1. Серверные компоненты .....	7
2.2.2. Клиентские компоненты .....	8
2.3. Взаимодействие компонентов .....	9
2.3.1. Взаимодействие по протоколу HTTP .....	9
2.3.2. Взаимодействие по протоколу CORBA .....	10
2.4. Стандартная сборка системы .....	10
3. Системные требования .....	11
3.1. Требования к серверам .....	11
3.1.1. Общие требования .....	11
3.1.2. Сервер приложений системы АЦК .....	12
3.1.3. Сервер базы данных АЦК .....	12
3.1.4. Совмещенный сервер приложений и сервер базы данных .....	13
3.2. Требования к клиентской части .....	13
3.2.1. Общие требования .....	13
3.2.2. Windows-клиент .....	14
3.2.3. Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ .....	14
3.3. Требования к линиям связи .....	15
3.3.1. Общие требования к линиям связи .....	15
3.3.2. Линии связи между серверами системы .....	15



3.3.3. Линии связи между сервером приложений и Windows-клиентом .....	15
3.4. Требования к безопасности.....	15
3.4.1. Демилитаризованная зона (DMZ).....	15
4. Установка и настройка программы.....	16
4.1. Настройка операционной системы.....	16
4.2. Установка и настройка СУБД .....	17
4.2.1. Настройка СУБД «Oracle».....	17
4.3. Установка системы «АЦК-Контроль» .....	17
4.3.1. Установка в операционной системе «Linux».....	17
4.5.2. Установка в операционной системе «Windows» .....	18
4.5.3. Настройка запуска комплекса серверов приложений .....	19
4.5.4. Настройка базы данных .....	20
4.5.5. Установка и настройка сервера приложений .....	26
4.5.8. Настройка сжатия данных, передаваемых/отсылаемых между сервером и клиентом .....	30
4.6. Настройка обновления системы .....	30
4.7.1. Обновление структуры БД.....	31
4.7.2. Обновление сервера приложений .....	32
4.7.3. Обновление данных в БД .....	32
4.7.4. Обновление offline-клиентов .....	32
4.8. Настройка локальной сети.....	33
4.9. Настройка журнализации событий и отладочных режимов .....	33
5. Сопровождение системы.....	34
5.1. Задачи сопровождения системы .....	34
5.2. Резервное копирование и архивирование БД и журнального файла сервера приложений .....	34
5.3. Идентификация ошибок и их устранение .....	35



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«БЮДЖЕТНЫЕ И ФИНАНСОВЫЕ ТЕХНОЛОГИИ»  
тел./факс: (495) 784-70-00; E-mail: ineed@bftcom.com  
www.bftcom.com

5.4. Поддержка пользователей системы.....35

5.5. Другие функции администратора системы.....35



## **1. Общие сведения о программе**

### **1.1. Назначение программы**

Система контроля задач по разработке программного обеспечения предназначена для автоматизации процессов внутриведомственного взаимодействия. Объектами автоматизации является функциональная деятельность подразделений Заказчиков (получатели бюджетных средств, главные распорядители бюджетных средств), уполномоченного органа.

## **2. Структура программы**

### **2.1. Архитектура системы**

При построении системы использована трехуровневая схема.

Использование единой информационной базы данных обеспечивает централизованное хранение данных.

Табл. 1. Уровни системы «АЦК-Контроль»

Уровень	Наименование уровня	Реализация	Назначение	Объекты
	Хранилище данных	Сервер БД	Хранение данных. Обеспечение физической целостности данных.	Данные
	Прикладная логика	Сервер приложений (Сервер АПП)	Обработка прикладной логики, работа с документами и объектами системы. Обеспечение логической целостности данных.	Наборы данных
	Клиентский (Интерфейсный)	Win-клиент.	Взаимодействие с пользователем и с другими информационными системами.	Интерфейс

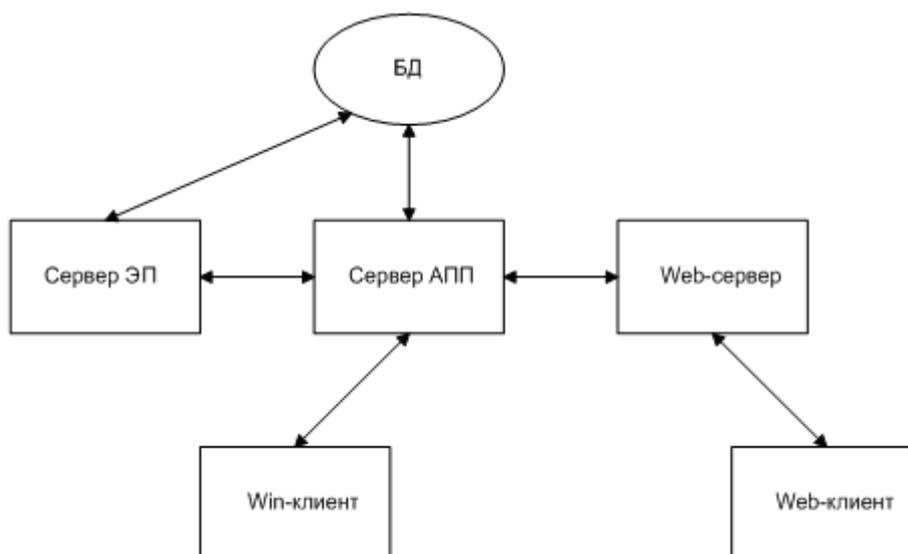


Схема обеспечивает:

- централизованное управление информационной базой;
- надежную целостность пользовательских данных;
- независимость отдельных компонентов системы;
- равномерное распределение вычислительной нагрузки в локальной сети;
- хорошую масштабируемость системы.

## 2.2. Компоненты системы

Система состоит из следующих взаимосвязанных компонентов:

- системы управления базами данных – СУБД;

- сервера приложений (серверов приложений может быть несколько);
- **offline-клиента**.

Использование **offline-клиента** является опциональным.

## 2.2.1. Серверные компоненты

### 2.2.1.1. СУБД

Система поддерживает работу с СУБД **Oracle 11g** (рекомендуемая **11.2.0.4**).

Данные системы хранятся в БД, непосредственный доступ к которой имеет только сервер (серверы) приложений. СУБД может работать как под управлением операционной системы **Windows 2003** (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

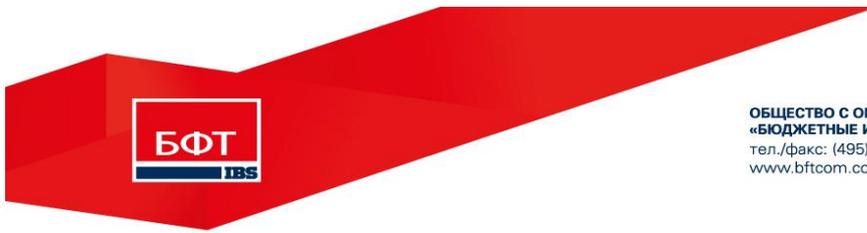
Сервер приложений является основным компонентом системы. Назначение сервера приложений заключается в обработке запросов пользователей и обработке данных. Функции сервера приложений:

- 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 7 Update 80** 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:

Табл. 2. Роли сервера приложений

Название сервера приложений	Роль сервера приложений
Сервер обработки запросов offline-клиентов	Взаимодействует с <b>offline-клиентами</b> .
Сервер взаимодействия со СКИБ	Взаимодействует с внешними системами

### 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.

В системе может использоваться клиентский компонент: **win-клиент**.

**Win-клиент** обладает полной функциональностью и предназначен для исполнения функций по сопровождению сотрудников отделов сопровождения и уполномоченных представителей заказчика.

### 2.2.2.1. Offline-клиент

**Offline-клиент** представляет собой **Windows-приложение**.

Главное назначение **offline-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. **Offline-клиент** связан только с сервером приложений и не имеет прямого доступа к БД.

## 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=2100;`
- в файле **control.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:2100/
```

где:

- 2100 – 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-Контроль» 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:

Табл. 3. 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 offline-клиента. В каталоге отсутствуют необходимые для работы offline-клиента дополнительные программные библиотеки.
Server	Содержит скомпилированные файлы сервера приложений.
SQL	Содержит скрипты для первоначального формирования структуры БД, а также для ее обновления (подкаталог <i>/log</i> ). Подкаталог <i>/ServiceScript</i> содержит сервисные скрипты. Их применение возможно только после предварительной проверки на тестовом стенде.
XML	Содержит xml-скрипты для первоначального формирования данных в БД, а также для их обновления.
Startsrv	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.
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> .

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

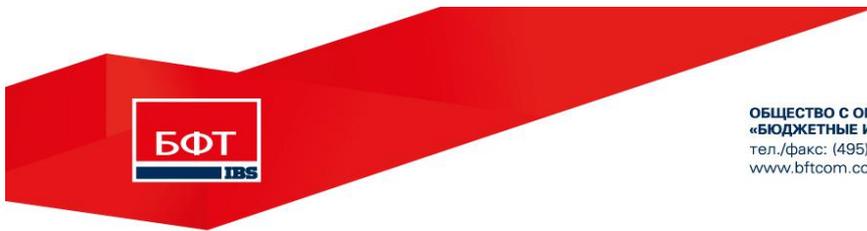
*Примечание. При необходимости администратору могут быть высланы недостающие в папке */Client* дополнительные программные библиотеки, если он решит запускать offline-клиент без предварительной установки.*

## 3. Системные требования

### 3.1. Требования к серверам

#### 3.1.1. Общие требования

Все серверы системы должны соответствовать требованиям ООО «Бюджетные и финансовые технологии», 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 ПО, которое не имеет отношения 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.

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

На серверы могут устанавливаться операционные системы, указанные в разделах ниже.

По предварительному согласованию с разработчиком системы АЦК возможна установка других ОС.

### 3.1.2. Сервер приложений системы АЦК

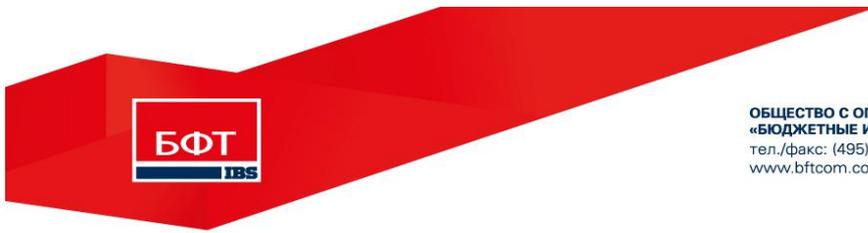
**Сервер приложения** – это сервер, на который устанавливается основная серверная часть систем. Данный сервер осуществляет обработку запросов клиентов (АРМ ФО и АРМ РБС) и передачу этих запросов базе данных системы, а также получение данных из базы и передача их клиенту. Сервер приложения выступает промежуточным звеном между базой данных и клиентами, обеспечивая тем самым защиту данных и распределение нагрузки. Ниже приведены системные требования к конфигурации сервера приложений АЦК:

Табл. 4. Системные требования к конфигурации сервера приложений АЦК

Тип сервера	Тип ПО	Программное окружение
Сервер приложений АЦК	ОС	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 ПБС до 25)/2012 R2(64)
	JDK (JRE)	Java SE 7 update 51 и выше

### 3.1.3. Сервер базы данных АЦК

**Сервер базы данных АЦК** – это сервер, на который устанавливается система управления базами данных (СУБД). Данный сервер осуществляет хранение и обработку



данных системы. К этому серверу предъявляются особые требования по производительности и надежности. Ниже приведены системные требования к конфигурации сервера СУБД:

Табл. 6. Системные требования к конфигурации сервера СУБД

Тип сервера	Тип ПО	Программное окружение
Сервер СУБД	ОС в соответствии с требованиями к СУБД	Рекомендуемые: Oracle Enterprise Linux 4/5/6/7 RH Linux AS 4/5/6/7 SUSE Linux 10/11  MS Windows 2003/2008/7 (только для малых объектов с количеством ПБС до 25)
	СУБД Oracle	Oracle 11g (рекомендуемая 11.2.0.4)

### 3.1.4. Совмещенный сервер приложений и сервер базы данных

Совмещенный вариант сервера приложений и сервера БД возможен только при низкой нагрузке и малом количестве off-line пользователей до 50 человек.

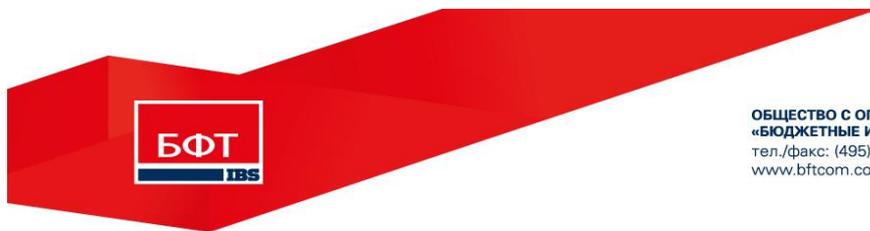
В таком варианте использование ОС MS Windows 2000/XP/2003/2008/Vista/7 допустимо только для малых объектов с количеством off-line пользователей до 20.

## 3.2. Требования к клиентской части

### 3.2.1. Общие требования

Все рабочие места с установленной на них клиентской частью 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. Windows-клиент

**Windows-клиент** – это приложение, являющееся клиентской частью системы АЦК и предоставляющее пользователю доступ к данным и функциям системы посредством графического интерфейса ОС семейства Windows. Ниже приведены системные требования к конфигурации рабочего места с установленным на нем Windows-клиентом:

**Табл. 8. Системные требования к конфигурации рабочего места с установленным на нем Windows-клиентом**

Минимальные требования	Рекомендуемые требования
Процессор: Intel P-4 2.8 GHz Память: 2 Gb Диск: 10Gb (свободно) Разрешение экрана: 1024x768 Сеть: 10 Мбит/с (для АРМ ФО 100 Мбит/с) ОС: Windows XP/Vista/7 ПО: MS Excel 2000/2003/XP/2007/2010	Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 20 Gb (свободно) Разрешение экрана: 1280x1024 и выше Сеть: 100 Мбит/с и выше) ОС: Windows XP/7/8 ПО: MS Excel 2000/2003/XP/2007/2010/2013

### 3.2.3. 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:

**Табл. 11. Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ**

При использовании СКЗИ КриптоПро	При использовании СКЗИ ViPNet
ОС: Windows XP/Vista/7/8 СКЗИ: КриптоПро CSP 3.0-3.9 (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 17.0, MS Internet Explorer 10.0, 11.0	ОС: Windows XP/Vista/7 СКЗИ: ViPNet CSP 3.2 Браузер: MS Internet Explorer 10.0, 11.0

### **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 должно осуществляться со скоростью 1 Гбит. На объектах с количеством ПБС до 75 шт. и количеством платежей до 500 шт. в день допустимо использовать межсерверные соединения в 100 Мбит.

#### **3.3.3. Линии связи между сервером приложений и Windows-клиентом**

Все используемые линии связи должны поддерживать сетевой протокол TCP/IP, и обеспечивать работу со скоростью не ниже 100 Мбит/с для АРМ ФО и не ниже 10 Мбит/с для АРМ РБС. Реальная скорость передачи данных от любого рабочего места до сервера, в момент пиковых нагрузок, должна быть не ниже 1 Мбит/с для АРМ ФО и не ниже 20 Кбит/с для АРМ РБС.

### **3.4. Требования к безопасности**

#### **3.4.1. Демилитаризованная зона (DMZ)**

В случае, когда необходим доступ пользователей к серверу приложений АЦК или веб-серверу системы АЦК из общественной сети Интернет, необходимо обеспечение безопасности и защиты 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.

Внешняя сеть оказывается между маршрутизатором провайдера и первым межсетевым экраном, в то время как демилитаризованная зона размещается между

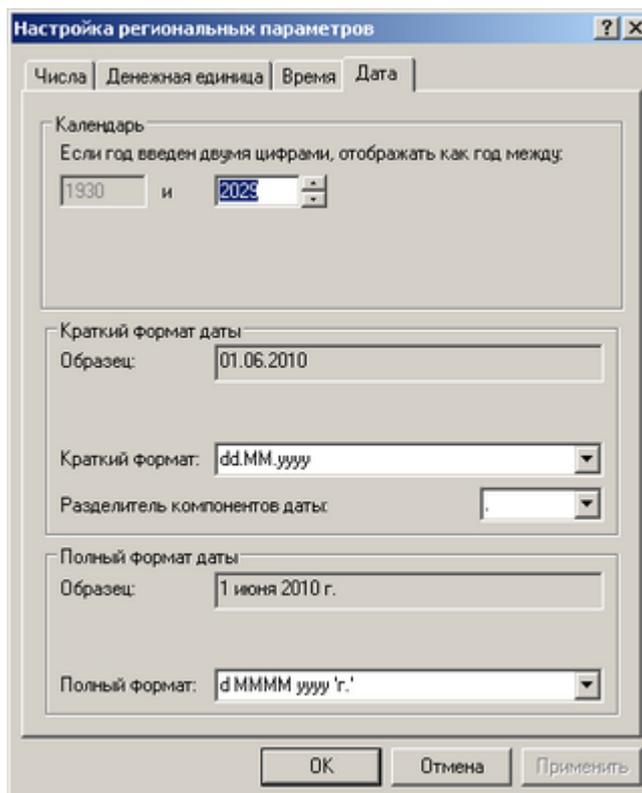
межсетевыми экранами №1 и №2. Архитектура с двумя межсетевыми экранами требует, чтобы межсетевой экран №1 мог обрабатывать достаточный объем трафика, если системы в DMZ будут работать с большим объемом трафика. Межсетевой экран №2 может быть менее производительной системой, так как он обрабатывает только внутренний трафик.

## 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 окно *Regional Options*.
2. Перейти на закладку **Date** и в списочном поле **Short date format** выбрать формат даты *dd.MM.yyyy*.



**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**.

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

### 4.2.1. 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**.

## 4.3. Установка системы «АЦК-Контроль»

### 4.3.1. 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/Control
```

3. Распаковать файлы сервера приложений из архива в дистрибутиве:

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

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

4. Установить права доступа к файлам:

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

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

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

```
#find /usr/local/Control -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:oracle:thin:@itasrv:1521:ora
```

```
azk.db.user=control
```

```
azk.db.password=control
```

```
azk.db.accessmode=oracle
```

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
```

#### 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/Control]*;
- каталог лог-файлов: *[/var/log/Control]*;
- путь к каталогу **JAVA**-машины: *[/usr/java/jdk1.7]*;
- адрес сервера БД: *[controldb]*.

#### 4.5.2. Установка в операционной системе «Windows»

##### 4.5.2.1. Настройка файлового хранилища

##### 4.5.2.2.1. Файловое хранилище «АЦК-Контроль»

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:|azk2|storages.xml**.

#### 4.5.3. Настройка запуска комплекса серверов приложений

Порядок запуска системы:

1. сервер БД;
2. 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 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-Контроль» 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.

#### 4.5.4.3. 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**.

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 rlsqlplus='rlwrap sqlplus'  
alias rlrman='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.

#### 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. 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).

#### 4.5.5.1. Файл настроек сервера приложений

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

##### 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.syspassword** – 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**).

*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 (**Cp866**, **Cp1251**, **KOI8\_R**, **ISO8859\_5**).
- **azk.log.encode** – используемая кодировка для вывода в журнал (**Cp866**, **Cp1251**, **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*.

#### 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.scheduler.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.

#### 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 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 offline-клиентов;

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 **JRK** 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) в начале. 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 offline-клиентов

Описание настройки автоматического обновления offline-клиентов см. в документации «».

Если отключено автоматическое обновление offline-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).

## 4.8. Настройка локальной сети

При построении сети с системой «АЦК-Контроль» 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
```

## 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](#).

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 offline-клиентом;

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 offline-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 нештатным 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;



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«БЮДЖЕТНЫЕ И ФИНАНСОВЫЕ ТЕХНОЛОГИИ»  
тел./факс: (495) 784-70-00; E-mail: ineed@bftcom.com  
www.bftcom.com

- производить обновления системы и смежного ПО;
- организовывать защиту серверов от внешних и внутренних сетевых атак, а также от проникновения вирусов на них;
- прочие функции.