



ООО «БФТ»

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

ined@bftcom.com
bftcom.com

Утвержден

БАРМ.00004-55 32 01-1-ЛУ

Система автоматизации финансово-экономических органов – Автоматизированный Центр Контроля процесса планирования и анализа бюджета АЦК-Планирование

Блок администрирования Подсистема администрирования Техническое описание

Автоматизированное рабочее место финансового органа

Руководство администратора

БАРМ.00004-55 32 01-1

Листов 230

© 2022 ООО «БФТ»

АННОТАЦИЯ

Приводится руководство администратора системы «АЦК-Планирование» по техническому описанию.

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

«Система автоматизации финансово-экономических органов – Автоматизированный Центр Контроля процесса планирования и анализа бюджета» («АЦК-Планирование») зарегистрирована в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам, Свидетельство № 2008610923 от 21 февраля 2008 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-Планирование» – 2.55.0.1. Последние изменения внесены 29.06.2022 г.

СОДЕРЖАНИЕ

1	Общие сведения о программе.....	10
1.1	Назначение программы.....	12
1.2	Системные требования.....	12
1.2.1	Требования к серверам.....	12
1.2.1.1	Общие требования.....	12
1.2.1.2	Сервер приложений системы АЦК.....	13
1.2.1.3	Сервер ЭП	14
1.2.1.4	Сервер базы данных АЦК.....	16
1.2.1.5	Совмещенный сервер приложений и сервер базы данных.....	17
1.2.1.6	Web-сервер	18
1.2.1.7	Сервер приложения для взаимодействия с ЕИС.....	19
1.2.2	Требования к клиентской части.....	19
1.2.2.1	Общие требования.....	19
1.2.2.2	Windows-клиент.....	20
1.2.2.3	Web-клиент	20
1.2.2.4	Требования к программному окружению при использовании криптографических функций.....	22
1.2.2.5	Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ.....	23
1.2.3	Требования к линиям связи.....	24
1.2.3.1	Общие требования к линиям связи.....	24
1.2.3.2	Линии связи между серверами системы.....	24
1.2.3.3	Линии связи между сервером приложений и Windows-клиентом.....	24
1.2.3.4	Линии связи между сервером приложений и WEB-клиентом.....	24
1.2.3.5	Линии связи для сервера ЭП (онлайн-соединение).....	25
1.2.4	Требования к безопасности.....	25
1.2.4.1	Демилитаризованная зона (DMZ).....	25
1.2.5	Требования к офисным приложениям.....	26
1.2.6	Ограничения.....	27
2	Структура программы.....	31
2.1	Сервер приложений.....	33
2.1.1	Компоненты сервера приложений.....	33
2.1.2	Утилиты для работы с данными. XML-скрипты.....	36
2.1.3	Утилиты для работы со структурой БД. SQL-скрипты.....	36

2.1.4	Файл свойств сервера приложений.....	38
2.1.4.1	Свойства базы данных.....	38
2.1.4.2	Свойства лога сервера приложений.....	41
2.1.4.3	Свойства управления заданиями сервера.....	44
2.1.4.4	Управление пользовательскими сессиями.....	45
2.1.4.5	Свойства сетевых параметров сервера.....	46
2.1.4.6	Настройки пула потоков.....	47
2.1.4.7	Свойства репликации.....	48
2.1.4.8	Логирование версий записей.....	48
2.1.4.9	OpenORB	49
2.1.4.10	Управление вложениями в документах.....	49
2.1.4.11	Свойства системы автоматического обновления.....	49
2.1.4.12	Настройка репликации нового сайта.....	52
2.1.4.13	Настройка параметров ЭП.....	52
2.1.4.14	Настройка параметров лицензии.....	53
2.1.4.15	Поддержка HTTP-соединения для Win-клиента.....	53
2.1.4.16	Настройка кластерного режима работы.....	54
2.1.4.17	Настройка взаимодействия с ЕИС.....	54
2.1.4.18	Настройка взаимодействия с системой Аванпост.....	55
2.1.4.19	Настройка хранилища ключей доверенных сертификатов.....	56
2.1.4.20	Настройка прокси-сервера по протоколу https.....	56
2.1.4.21	Кэширование профилей на сервере АЦК.....	57
2.1.4.22	Настройка сервиса помощи входа по сертификату (СПВС).....	57
2.1.4.23	Настройка сервиса передачи сообщений при формировании отчетов по Приказу 243н.....	58
2.2	Система лицензионной защиты.....	59
2.3	Клиентское приложение.....	59
2.3.1	Описание клиентского приложения.....	59
2.3.2	Параметры командной строки клиента.....	62
3	Основные понятия	65
3.1	Клиентские объекты, процессоры, провайдеры.....	66
3.2	Бюджет	68
3.3	Справочники	68
3.4	Документы. Редактор сценариев обработки документа.....	69
3.5	Репликация. Объекты репликации. Реплицируемые таблицы.....	69

3.6	Сайты. Типы соединений сайтов.....	71
3.7	Сообщения. Очереди сообщений.....	72
3.8	Планировщик. Задания планировщика. Расписание запуска заданий.....	74
3.8.1	Макросы для параметров begin_date и end_date.....	75
3.8.2	AssignExport.....	76
3.8.3	AutoControlProductPlanNotControlEIS.....	77
3.8.4	Backup DB	78
3.8.5	Backuper	79
3.8.6	CADDownloader.....	80
3.8.7	CertDownloader.....	80
3.8.8	Consistency check.....	81
3.8.9	ConsRepLauncher.....	81
3.8.10	ControlEventFactValueProcessor.....	82
3.8.11	ControlExecMessenger.....	82
3.8.12	ControlResultsInEISUploader.....	82
3.8.13	CreateNotifyMsg.....	83
3.8.14	CRLDownloader.....	84
3.8.15	CurdateSynchronizer.....	84
3.8.16	DataObjectChangeLogCleaner.....	85
3.8.17	DBCloneActivate.....	85
3.8.18	DBCloneBackuper.....	86
3.8.19	DBCloneCheck.....	87
3.8.20	DBCloneStatistical.....	88
3.8.21	DebitCardLauncher.....	88
3.8.22	DeleteNotifyMsg.....	89
3.8.23	DocRetentor.....	89
3.8.24	DocumentCleaner.....	89
3.8.25	EstimateImport.....	90
3.8.26	Exchange	91
3.8.27	FhdPlanStructTemplateImport.....	92
3.8.28	FORepLauncher.....	92
3.8.29	ImportBudgServiceFed2.....	92
3.8.30	ImportDocFromEis.....	94
3.8.31	ImportDocLauncher.....	95

3.8.32	ImportEhdDictionaries.....	96
3.8.33	ImportFinancingIndicatorsEHD	99
3.8.34	ImportRegProjCard.....	102
3.8.35	ImportTaskProjects.....	104
3.8.36	ImportUnitProcessor.....	105
3.8.37	KIExport	111
3.8.38	MailCleaner	112
3.8.39	MessageControlEvent.....	112
3.8.40	Messenger	113
3.8.41	MsgCleaner.....	114
3.8.42	MsgRepeater.....	114
3.8.43	NoteToChangePasswTask.....	115
3.8.44	NotifyMsgSender.....	116
3.8.45	NotificationSenderTask.....	116
3.8.46	OkpdlImport	116
3.8.47	OkvedImport.....	117
3.8.48	OosExport	117
3.8.49	PlanAssExport.....	118
3.8.50	PlanInExport.....	118
3.8.51	PlanInRefImport.....	119
3.8.52	PlanSrcExport.....	120
3.8.53	PlanXMLExport.....	120
3.8.54	PIBorExport.....	120
3.8.55	PIStateEventDocExport.....	121
3.8.56	PIStateProgDocExport.....	122
3.8.57	PIStateSubProgDocExport.....	122
3.8.58	ProgDocExport.....	123
3.8.59	PurposefulGrantCleanupTask.....	123
3.8.60	RegionalProjectCardEhdExportProcessor.....	124
3.8.61	RelmportFailedImportedDocs.....	125
3.8.62	Replication	125
3.8.63	ReportingCompanyRPPProcessor.....	126
3.8.64	RplGenerationStatusTask.....	127
3.8.65	RtRplRunner.....	127

3.8.66	SchExpDocs.....	128
3.8.67	Session cleaner.....	129
3.8.68	SQLTracer	129
3.8.69	SSLTrustManager.....	130
3.8.70	Statistics	130
3.8.71	STORAGE_COMPRESSOR.....	131
3.8.72	SyseventCleaner.....	131
3.8.73	SysInfoShower.....	131
3.8.74	TransactMonitorFlusher.....	132
3.8.75	Transport	133
3.8.76	UniAnalyticalReportDeleteCompleted.....	134
3.8.77	UniAnalyticalReportUpdate.....	134
3.8.78	UniAnalyticalTaskProcessor.....	134
3.8.79	UniContractProcessor.....	135
3.8.80	UpdateRegionalListRegNum.....	137
3.8.81	UploadOKSFromEIS.....	137
3.8.82	Validation	138
3.9	Выполнение XML-заданий.....	138
3.10	Xml для методики создания бюджета 2022 года.....	139
4	База данных	140
4.1	Основные принципы построения БД.....	141
4.2	Основные принципы работы с БД.....	142
4.2.1	SQLExecuter.....	143
4.2.1.1	Назначение SQLExecuter. Параметры запуска. Файл конфигурации	143
4.2.1.2	SQL-скрипт. Назначение, типы, нумерация, структура, синтаксис.....	144
4.2.1.3	Утилиты запуска sql-скриптов.....	146
4.2.2	XMLExecuter.....	147
4.2.2.1	Назначение, параметры запуска.....	147
4.2.2.2	XML-скрипт. Структура, назначение.....	149
4.2.2.3	XML-скрипты системы. Описание, зависимости запуска.....	150
4.2.2.4	Утилиты запуска XML-скриптов.....	153
4.2.3	Утилиты для работы с БД.....	154
4.2.3.1	Описание утилиты executer.cmd(sh).....	154
4.2.3.2	Утилита DBUpdate.cmd(sh).....	157

4.2.3.3	Утилита convert.cmd.....	158
4.2.3.4	Утилита comparer.cmd.....	158
4.2.3.5	Утилита update.cmd(sh).....	158
4.2.3.6	Создание новой БД.....	159
4.2.3.7	Резервное копирование/восстановление БД.....	160
5	Установка и настройка системы	161
5.1	Установка и настройка сервера приложений.....	162
5.1.1	Установка и настройка JRE.....	162
5.1.1.1	Настройка Java Cryptography Extension (JCE).....	164
5.1.2	Запуск и остановка сервера приложений. Файл конфигурации.....	165
5.2	Установка и настройка сервера БД.....	166
5.3	Подготовка БД к работе.....	166
5.3.1	Методика приведения БД к нужной версии.....	167
5.4	Установка, настройка и запуск клиентской части.....	167
5.5	Установка сборки отчетов.....	170
5.6	Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) учреждениях.....	170
5.6.1	Порядок предоставления справочной информации.....	174
5.6.2	Заполнение полей справочника «Организации» для интеграции с ЕИС..	175
5.6.3	Обработка приложенных документов при организации обмена с ЕИС....	179
5.6.4	Присоединение файлов к справочникам.....	180
5.6.5	Присоединение файлов к документу.....	182
5.7	Настройка соединения Win-клиента через шифрованный канал https.....	188
6	Обновление системы	191
6.1	Обновление сервера.....	192
6.2	Пакетное обновление БД.....	193
6.3	Обновление клиентской части.....	194
6.4	Обновление системы для работы с веб-клиентом.....	194
6.5	Автоматическое обновление клиентских компонент.....	195
6.5.1	Описание настроек системы для осуществления автоматического обновления системы.....	195
6.5.1.1	Настройки сервера приложений, влияющие на работу модуля обновления.....	195
6.5.1.2	Системные настройки.....	196

6.5.1.3	Настройки клиентской части.....	197
6.5.2	Обновление клиента вручную.....	198
6.5.3	Рекомендации по переходу.....	199
6.6	Обновление отчетов.....	200
6.6.1	Обновление специальных отчетов.....	200
7	Порядок технического обслуживания.....	202
7.1	Регламентно-профилактические мероприятия.....	203
7.1.1	Регламентно-профилактические мероприятия по ППО.....	203
7.1.1.1	Установка эталонной сборки.....	205
7.1.1.2	Параметры распределения JAVA памяти.....	206
7.1.1.3	Параметры настроек в конфигурационных файлах АЦК.....	207
7.1.1.4	Использование утилиты «jvisualvm».....	210
7.1.2	Регламентно-профилактические мероприятия по СПО.....	211
7.1.3	Мониторинг виртуальной инфраструктуры.....	212
7.1.4	Наличие тестового СП.....	212
7.1.5	Мониторинг структуры системы хранения данных.....	212
7.1.5.1	Оценка производительности СХД.....	213
7.1.5.2	Повышенный Service Time СХД.....	213
7.1.5.3	Утилизация CPU СХД.....	214
7.1.6	Проверка работоспособности.....	214
7.1.7	Проверочное восстановление резервной копии.....	216
8	Аванпост FAM.....	217
8.1	Сценарий взаимодействия с Клиентским приложением.....	218
8.2	Сценарий взаимодействия с Web-клиентом.....	220
9	Описание команд ОС Linux.....	221
10	Отслеживание изменений системных объектов.....	225
11	Устранение блокировок по коду цели.....	229



Общие сведения о программе

Прежде чем приступить к установке, настройке и эксплуатации программного обеспечения для Система автоматизации финансово-экономических органов – Автоматизированный Центр Контроля процесса планирования и анализа бюджета «АЦК-Планирование», распечатайте и прочитайте этот документ; в нем содержатся важные технические сведения о данной версии продукта.

Все ссылки на имена файлов даются относительно корневого каталога сборки. Кроме того, распечатайте и прочитайте следующие файлы, которые находятся в каталоге **./doc/**:

Таблица 1 – Файлы, рекомендуемые для ознакомления

Название файла	Описание
accounts.txt	Описание типов счетов.
backup_restore.txt	Описание резервного копирования базы данных и ее восстановления.
banks2xml.txt	Описание преобразования информации о банках РФ.
changelog_xl_template.txt	Описание процедуры обновления расчетных листов и шаблонов.
creditloan_account.txt	Правила получения счета в кредитно-ссудных договорах.
DBCcreator.txt	Правила создания новой БД.
dbmerge.txt	Описание утилиты, объединяющей данные из нескольких БД в одну.
doccleaner.txt	Описание настройки задания планировщика по физическому удалению документов из БД и архивированию документов, удаленных в системе.
doclauncher.txt	Описание настройки задания планировщика по автоматическому запуску в обработку импортированных документов.
docrule.txt	Правила работы с набором полей, описывающим организацию и счет.
forms.txt	Список стандартных форм.
methods.txt	Методы, доступные для документов.
msg.txt	Не используется .
planning_files.txt	Информация о том, как должны называться и где должны располагаться файлы связанные с подсистемами «Планирование» и «Межбюджет».
readme_DBC.txt	Описание проведения конвертации БД из одного типа в другой.
readme_RPL.txt	Описание настройки репликации при совместной работе с другой системой.
readme_SE.txt	Описание SQLExecuter (исполнителя SQL-скриптов).
readme_XE.txt	Описание XMLExecuter (исполнителя XML-скриптов).
schtasks.txt	Описание процедур планировщика.

Название файла	Описание
inker.txt	Описание приложения Inker, которое служит для создания, редактирования и исполнения сценариев динамического тестирования работоспособности и быстродействия сервера приложений АЦК-Планирование.
tfo_db.txt	Глава 1. Создание базы данных для ТФО. Глава 2. Принципы межсайтового взаимодействия серверов АЦК. Глава 3. Обмен документами между сайтами АЦК и ТФО
updater.txt	Описание процедуры автоматизированного обновления системы.
xml_dep.txt	Памятка об XML-скриптах.

Функциональные возможности системы «АЦК-Планирование» подробно изложены в соответствующей сопроводительной документации, которая входит в комплект поставки.

1.1 Назначение программы

Система «АЦК-Планирование» предназначена для автоматизации исполнения бюджета и управления бюджетным процессом в субъектах РФ и муниципальных образованиях.

Система используется для автоматизации всех структурных подразделений финансового органа (далее – ФО), включая его территориальные подразделения, распорядителей и получателей бюджетных средств (далее – РБС и ПБС соответственно).

Система предусматривает возможность информационного взаимодействия ФО с органами Федерального казначейства и учреждениями банков, в которых открыты счета ФО, РБС и ПБС.

1.2 Системные требования

1.2.1 Требования к серверам

1.2.1.1 Общие требования

Все серверы системы должны соответствовать требованиям ООО «БФТ», 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.

1.2.1.2 Сервер приложений системы АЦК

Сервер приложения – это сервер, на который устанавливается основная серверная часть систем. Данный сервер осуществляет обработку запросов клиентов 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:

Таблица 2 – 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 CentOS 6/7 MS Windows server 2008 R2 (64)/ 2012 R2 / 2019 Standart (1809) MS Windows 7 (64) (только для малых объектов с количеством ПБС до 25) MS Windows server 2012 R2 RedOS 7.1 / 7.2 Alt Linux 8.2 / 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 POCA «Кобальт» 7.3 (3.10)
	JDK (JRE)	Sun Java SE 8 update 40 и выше

1.2.1.3 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 системные требования к конфигурации и программному обеспечению сервера электронной подписи:

Таблица 3 – Системные требования к конфигурации сервера электронной подписи

Максимальное количество пользователей, одновременно выполняющих проверку ЭП для группы ЭД (более 50)*	Вид используемой ЭП	Рекомендуемая конфигурация		
		CPU	RAM	Ethernet
до 5	Усиленная, Усиленная (XML), Усиленная (со штампом времени)	2 Cores 3GHz Intel Xeon 64 bit	4GB	1Gbit
	Усиленная (с доказательствами подлинности)	4 Cores 3GHz Intel Xeon 64 bit		
до 10	Усиленная, 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, 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 20), одновременно выполняющих множественную проверку ЭП, рекомендуется развертывание дополнительного сервера ЭП.

Таблица 4 – Системные требования к программному обеспечению сервера электронной подписи

Тип сервера	Тип ПО	Программное окружение
Сервер ЭП	ОС	MS Windows server 2008 R2 (64)/ 2012 R2 Windows Server 2019 Standart (1809) <i>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения²⁷.</i>
		MS Windows 7 (64)(только для малых объектов с количеством ПБС до 25) Oracle Enterprise Linux 5.5 и выше RH Linux AS 5.5 и выше SUSE Linux 10 SP2/11 RedOS 7.1 / 7.2 <i>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения²⁷.</i>
		Alt Linux 8.2 / 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 <i>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения²⁷.</i> РОСА «Кобальт» 7.3 (3.10)
	JDK (JRE)	Sun Java SE 8 update 40 и выше
	ЭЦП	Одно из поддерживаемых СКЗИ: <ul style="list-style-type: none"> КриптоПро CSP 4.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»). Опционально: КриптоПро TSP Client, КриптоПро OCSP Client. <ul style="list-style-type: none"> ViPNet CSP версии 4.2 КриптоПро JCP 2.0.

Примечание. При использовании видов ЭП: Усиленная, Усиленная (XML), Усиленная (со штампом времени).

1.2.1.4 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:

Таблица 5 – Системные требования к конфигурации сервера СУБД

Тип сервера	Тип ПО	Программное окружение
Сервер СУБД	ОС в соответствии с требованиями к СУБД	<p>Рекомендуемые: Oracle Enterprise Linux 4/5/6/7 RH Linux AS 4/5/6/7 SUSE Linux 10/11 CentOS 6/7 MS Windows server 2008 R2 (64) / 2012 R2 / 2019 Standart (1809)</p> <p>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения²⁷.</p> <p>MS Windows 7 (64)(только для малых объектов с количеством ПБС до 25) RedOS 7.1 / 7.2 Alt Linux 8.2 Alt Linux 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 РОСА «Кобальт» 7.3 (3.10)</p>
	СУБД Oracle	Oracle 11g (рекомендуемая 11.2.0.4) Oracle 12c (рекомендуемая 12.1.0.2, 12.2 не поддерживается)
	СУБД Firebird	Firebird-2.5.2
	СУБД PostgreSQL	PostgreSQL 9, версия 9.6.11 и выше PostgreSQL 11, версия 11.8 и выше PostgreSQL 12, версия 12.7 и выше Рекомендуемая версия PostgreSQL: PostgreSQL 13, версия 13.3 и выше

1.2.1.5 Совмещенный сервер приложений и сервер базы данных

Совмещенный вариант сервера приложений и сервера БД возможен только при низкой нагрузке и малом количестве on-line пользователей до 50 человек. В таком варианте использования ОС MS Windows 2008/7 допустимо только для малых объектов с количеством on-line пользователей до 20 человек.

В таком варианте использование ОС MS Windows 2008/7 допустимо только для малых объектов с количеством on-line пользователей до 20.

1.2.1.6 Web-сервер

Данный сервер является промежуточным звеном между клиентами и сервером приложений, осуществляет обработку запросов клиентов и передачу этих запросов серверу приложений, а также получение данных от сервера приложений и передачу их клиентам. В качестве клиентского приложения используется web-браузер. При количестве пользователей больше 500 рекомендуется распределить web-сервер по разным физическим узлам. Для распределения конечных пользователей по узлам кластера web-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 web-сервера собственного сервера приложения без ресурсозатрат на мультиплексирование. Ниже приведены системные требования к конфигурации web-сервера:

Таблица 6 – Системные требования к конфигурации сервера приложений web-сервера

Тип сервера	Тип ПО	Программное окружение
Web-сервер	ОС	Oracle Enterprise Linux 5.5 и выше RH Linux 5.5 и выше SUSE Linux 10 SP2/11 CentOS 6/7 MS Windows 7(64) / 2008(64) / 2008 R2 / 2019 Standart (1809) * <i>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения²⁷.</i> RedOS 7.1 / 7.2 Alt Linux 8.2 / 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 ПОСА «Кобальт» 7.3 (3.10)
	JDK (JRE)	Sun Java SE 8 update 40 и выше
	Веб-сервер	Apache Tomcat 8.5 (Sun Java SE 8u40 и выше)

В случае необходимости организации подключений к web-серверу из сети Интернет, необходимо, чтобы web-сервер располагался в безопасной части сети – [«демилитаризованной зоне»](#)²⁵.

1.2.1.7 Сервер приложения для взаимодействия с ЕИС

Сервер приложений для взаимодействия с ЕИС – это сервер, на который вынесены функции для взаимодействия с ЕИС.

Таблица 7 – Системные требования к конфигурации сервера приложения для взаимодействия с ЕИС

Тип сервера	Тип ПО	Программное окружение
Сервер приложения для взаимодействия с ЕИС	ОС	Oracle Enterprise Linux 5.5 и выше Oracle Enterprise Linux 5/5 и выше RH Linux AS 5/5 и выше SUSE Linux 10 SP2/11 RedOS 7.2 Alt Linux 8.2 / 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 РОСА «Кобальт» 7.3 (3.10) клиентские ОС: MS Windows 7
	JDK (JRE)	Sun Java SE 8 update 40 и выше
	СКЗИ	КриптоПро JCP 2.0 (лицензия должна соответствовать установленной на сервере ОС (серверная или клиентская) согласно требованиям Компании «КРИПТО-ПРО»)

1.2.2 Требования к клиентской части

1.2.2.1 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).

1.2.2.2 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:

Таблица 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: MS Windows 7 / MS Windows 8 / MS Windows 10; RedOS 7.1, Alt Linux 8.2 / 9.1, Astra Linux 2.12.13, РОСА «Кобальт» 7.3 <i>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения ²⁷.</i> ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ²⁶ .	Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 20 Gb (свободно) Разрешение экрана: 1280x1024 или 1440x900 и выше Сеть: 100 Мбит/с и выше ОС: Windows 7/8/10 ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ²⁶ .

***Примечание.** 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4Gb.*

1.2.2.3 Web-клиент

Web-клиент – это вариант доступа к системам АЦК с помощью web-браузера, без

установки клиентского программного обеспечения на рабочую станцию. Системные требования к конфигурации компьютера и его оснащению:

Таблица 9 – Системные требования к конфигурации рабочего места с установленным на нем web-клиентом

Минимальные требования						Рекомендуемые требования					
Процессор: P-4 2.8 GHz Память: 1 Gb Диск: 1 Gb (свободно) Разрешение экрана: 1024x768 Сеть: 2 Мбит/с ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ²⁶ .						Процессор: Intel Core 2 Duo Память: 4 Gb Диск: 1 Gb (свободно) Разрешение экрана: 1280x1024 и выше Сеть: 10 Мбит/с и выше ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ²⁶ .					
ОС	Windows XP/Vista/7	Scientific Linux 7.2	RedOS 7.1	ALT Linux 8.2	Astra Linux (Orel) 2.12.21	ОС	Windows 7/8/10	Scientific Linux 7.2	RedOS 7.1	ALT Linux 8.2	Astra Linux (Orel) 2.12.21
Браузер	Mozilla Firefox 60.0, MS Internet Explorer 10.0, 11.0, Google Chrome 67.0, Яндекс. Браузер 19.4.2 (поддерживается ОС, начиная с WIN 7 версии), Chromium-gost 80.0.3987	Google Chrome 67.0	Firefox 68, Google-chrome 67	Google Chrome 76.0.3809.132 Яндекс. Браузер 17.7.0.1683	Яндекс. Браузер 19.10.3.302 Chromium-gost 72.0.3626	Браузер	Mozilla Firefox 60.0, Microsoft Edge, MS Internet Explorer 10.0, 11.0, Google Chrome 67.0, Яндекс. Браузер 19.4.2 (поддерживается ОС, начиная с WIN 7 версии), Chromium-gost 80.0.3987	Google Chrome 67.0	Firefox 68, Google-chrome 67	Google Chrome 76.0.3809.132 Яндекс. Браузер 17.7.0.1683	Яндекс. Браузер 19.10.3.302 Chromium-gost 72.0.3626

Примечание. 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4Gb.

1.2.2.4 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 10 – Требования к программному окружению при использовании криптографических функций

Минимальные требования					Рекомендуемые требования				
СКЗИ: КриптоПро CSP 4.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО») Опционально: КриптоПро TSP Client, КриптоПро OCSP Client					СКЗИ: ViPNet CSP 4.2				
Windows-клиент									
ОС	Windows Vista/7/8/10; RedOS 7.1 / 7.2				ОС	Windows Vista/7/10			
Web-клиент									
ОС	Windows Vista/7/8/10	Scientific Linux 7.2	RedOS 7.1	ALT Linux 8.2	ALT Linux 9.1	Astra Linux (Orel) 2.12.21	ПОСА «Кобальт» 7.3 (3.10)	Windows Vista/7/10	
Доп. ПО	Сервис ЭП АЦК 1.0.3.9	плагин nmsigner-1.0.9-1	СКЗИ: КриптоПро 4.0, плагин КриптоПро ЭП Browser plug-in 1.2.7 Браузер: Firefox версия 68, Google Chrome версия 67	Google Chrome 76.0.3809.132, КриптоПро ЭЦП Browser plug-in 2.0 Яндекс.Браузер 17.7.0.1683 КриптоПро ЭЦП Browser plug-in 1.2.7	Яндекс.Браузер 20.4.3.268 КриптоПро ЭЦП Browser plug-in 2.0	Яндекс.Браузер 19.10.3.302 Chromium-gost 72.0.3626 КриптоПро ЭЦП Browser plug-in 1.2.7,	Яндекс.Браузер 19.10.3.302 КриптоПро ЭЦП Browser plug-in 2.0	Сервис ЭП АЦК 1.0.4.3	

Внимание! 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4Gb.

Примечание. Сервис электронной подписи АЦК – программное окружение, обеспечивающее использование криптографических функций при работе в браузерах Google Chrome, Microsoft Edge, Mozilla Firefox и Internet Explorer под ОС Windows.

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

Внимание! При использовании функций наложения ЭП на прикрепленные к ЭД файлы (вложения) рекомендуемый размер подписываемого файла не должен превышать 20 Мб.

1.2.2.5 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 11 – 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 4.0, 5.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО») Опционально: КриптоПро TSP Client, КриптоПро OCSP Client Браузер: MS Internet Explorer 10.0, 11.0, Яндекс.Браузер 19.4.2, Chromium-gost 80.0.3987 ОС: Linux СКЗИ: КриптоПро CSP 4.0,5.0 (версия должна соответствовать установленной на сервере ОС согласно требованиям Компании «КРИПТО-ПРО»), Браузер: Яндекс.Браузер 19.10.3, Chromium-gost 72.0.3626	ОС: Windows 7 СКЗИ: ViPNet CSP 4.2 Браузер: MS Internet Explorer 10.0, 11.0

Примечание. 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4Gb.

1.2.3 Требования к линиям связи

1.2.3.1 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.

1.2.3.2 Линии связи между серверами системы

Все используемые линии связи должны поддерживать сетевой протокол TCP/IP, и обеспечивать работу со скоростью не ниже 1 Гбит/с. Реальная скорость передачи данных между серверами должна быть не ниже 20 Мбит/с. Подключение сервера приложения АЦК с сервером базы данных должно осуществляться со скоростью 1 Гбит/с. На объектах с количеством ПБС до 75 шт., количеством АРМ ПБС до 50 шт. и количеством платежей до 500 шт. в день допустимо использовать межсерверные соединения в 100 Мбит/с.

1.2.3.3 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.

1.2.3.4 Линии связи между сервером приложений и WEB-клиентом

Все используемые линии связи должны поддерживать сетевой протокол HTTP, и обеспечивать работу со скоростью не ниже 2 Мбит/с. В качестве типа подключения может использоваться xDSL-соединение или другой тип выделенной линии.

1.2.3.5 Линии связи для сервера ЭП (онлайн-соединение)

В случае применения электронной подписи со штампом времени или доказательствами подлинности, при обращении к внешним службам штампов времени и актуальных статусов сертификатов должны использоваться линии связи, обеспечивающие работу со скоростью не ниже 10 Мбит/с. В качестве типа подключения может использоваться ADSL- или оптоволоконное соединение.

1.2.4 Требования к безопасности

1.2.4.1 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.

Рекомендуется использовать два межсетевых экрана, отделяющих DMZ от внешней и внутренней сети:

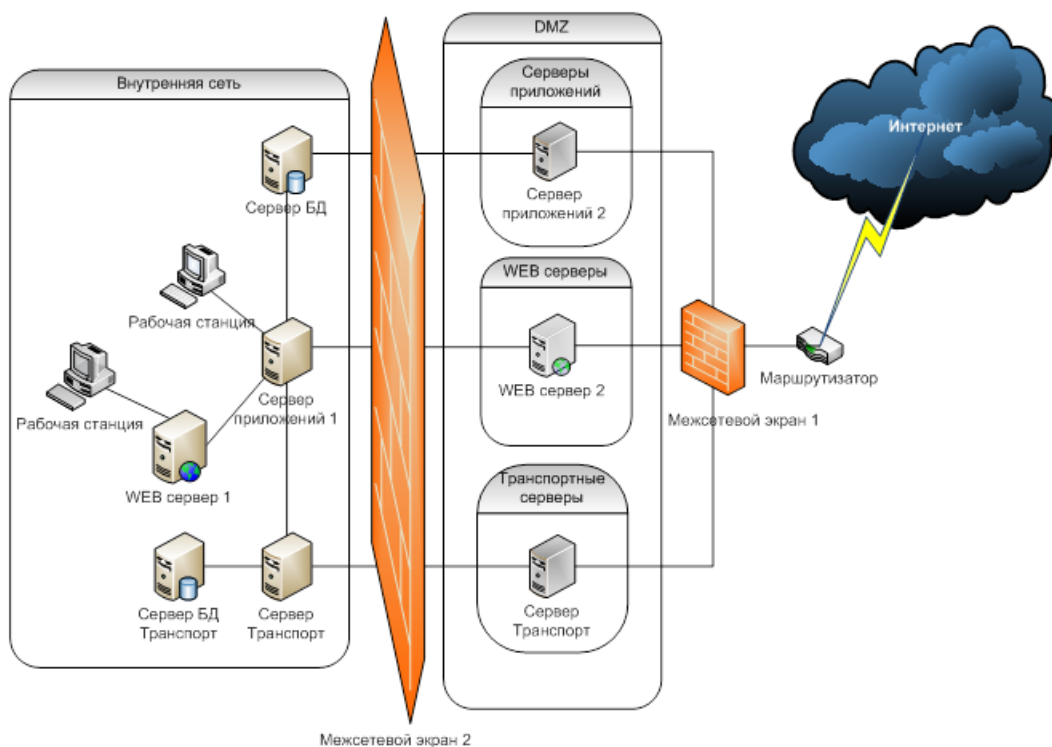


Рисунок 1 – Рекомендуемая схема сети

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.

1.2.5 Требования к офисным приложениям

Офисные приложения – набор приложений, предназначенных для обработки электронной документации на персональном компьютере.

Таблица 12 – Системные требования к офисным приложениям рабочего места с установленным на нем Windows-клиентом

ОС / продукт	Версия ОС	Libre Office	MS Office
Windows	7	LibreOffice 6.3.6.2	MS Office 2007/2010/2013/2016
	8		

ОС / продукт	Версия ОС	Libre Office	MS Office
	10		
Alt Linux	8.2		
	9.1	LibreOffice 6.3.6.2	
RedOS	7.1	LibreOffice 5.3.4.1	
Astra Linux	2.12.21		
РОСА «Кобальт»	7.3 (3.10)		

Таблица 13 – Системные требования к офисным приложениям рабочего места с установленным на нем Web-клиентом

ОС / продукт	Версия ОС	Libre Office	MS Office
Windows	7	LibreOffice 6.3.6.2	MS Office 2007/2010/2013/2016
	8		
	10		
Alt Linux	8.2		
	9.1	LibreOffice 6.3.6.2	
RedOS	7.1	LibreOffice 5.3.4.1	
Astra Linux	2.12.21		
РОСА «Кобальт»	7.3 (3.10)		

1.2.6 Ограничения

В разделе представлен перечень операционных систем, в работе которых имеются некоторые ограничения:

Таблица 14 – Перечень программного обеспечения с ограничениями

Наименование программного обеспечения	Примечание	Ограничения
RedOS 7.1	Операционная система	<ul style="list-style-type: none"> • Есть некритичные проблемы при работе со справочниками системы; • Есть некритичные проблемы при работе с электронными документами; • Есть некритичные проблемы при работе с АРМами; • Есть некритичные проблемы при работе с отчетами; • Есть некритичные проблемы с интерфейсом и работой форм.
RedOS 7.2	Операционная система	<p>Не работает ЭП вида «CAAdES-T».</p> <p>Для Сервера ЭП:</p> <ul style="list-style-type: none"> • Для АЦК-Финансы ниже 2.54.0.797 не работает ЭП для типа подписи «усиленный с доказательствами» с расширением подписи на сервере, другие типы подписи работают.
Astra Linux 2.12.29	Операционная система	<p>Для windows-клиента:</p> <ul style="list-style-type: none"> • Есть некритичные проблемы при работе со справочниками системы; • Есть некритичные проблемы при работе с электронными документами; • Есть некритичные проблемы при работе с АРМами; • Есть некритичные проблемы при работе с отчетами; • Есть некритичные проблемы с интерфейсом и работой форм; • Не работает ЭП для типа подписи «усиленный с доказательствами», другие типы подписи работают. <p>Для web-клиента:</p> <ul style="list-style-type: none"> • Есть некритичные проблемы при работе с отчетами; • Не работает ЭП для типа подписи «усиленный с доказательствами», другие типы подписи работают. <p>Для Сервера ЭП:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленный с доказательствами», другие типы подписи работают.
Astra Linux 2.12.43	Операционная система	<p>Не работает ЭП вида «CAAdES-T».</p> <p>Для windows-клиента:</p> <ul style="list-style-type: none"> • Некорректное отображение неактивных кнопок. <p>Для web-клиента:</p> <ul style="list-style-type: none"> • Есть некритичные проблемы при работе с отчетами. <p>Для Сервера ЭП:</p> <ul style="list-style-type: none"> • Для АЦК-Финансы ниже 2.54.0.797 не работает ЭП для типа подписи «усиленный с доказательствами» с расширением подписи на сервере, другие типы подписи работают.

Наименование программного обеспечения	Примечание	Ограничения
РОСА «Кобальт» 7.3	Операционная система	<p>Не работает ЭП вида «CAAdES-T».</p> <p>Для windows-клиента:</p> <ul style="list-style-type: none"> • Есть некритичные проблемы при работе со справочниками системы; • Есть некритичные проблемы при работе с электронными документами; • Есть некритичные проблемы при работе с АРМами; • Есть некритичные проблемы при работе с отчетами; • Есть некритичные проблемы с интерфейсом и работой форм; • Не работает ЭП видов «CAAdES-BES», «CAAdES-X Long Type 1». <p>Для web-клиента:</p> <ul style="list-style-type: none"> • Есть некритичные проблемы при работе с отчетами; • Не работает ЭП вида «усиленная с доказательствами подлинности» с расширением подписи на сервере для АЦК-Финансы ниже 2.54.0.797. <p>Для Сервера ЭП:</p> <ul style="list-style-type: none"> • Не работает ЭП вида «усиленная с доказательствами подлинности» с расширением подписи на сервере для АЦК-Финансы ниже 2.54.0.797.
Alt Linux 8.2	Операционная система	<p>Для win-клиента:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают; <p>Для Web-клиента:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают; <p>Для сервера ЭП:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают.
Alt Linux 9.1	Операционная система	<p>Для win-клиента:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают; <p>Для Web-клиента:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают; <p>Для сервера ЭП:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают.

Наименование программного обеспечения	Примечание	Ограничения
КриптоПро 4/5	Криптопровайдер	<p>Для windows-клиента:</p> <ul style="list-style-type: none">• Длительное время проверки ЭП формата CAdES-T на стороне криптопровайдера. <p>Для web-клиента:</p> <ul style="list-style-type: none">• Длительное время проверки ЭП формата CAdES-T на стороне криптопровайдера. <p>Для сервера ЭП:</p> <ul style="list-style-type: none">• Длительное время проверки ЭП формата CAdES-T на стороне криптопровайдера.
Windows Server 2019 Standart (1809)	Операционная система	<p>Для win-клиента:</p> <ul style="list-style-type: none">• Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают; <p>Для Web-клиента:</p> <ul style="list-style-type: none">• Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают; <p>Для сервера ЭП:</p> <ul style="list-style-type: none">• Не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают.



2

Структура программы

В состав информационной системы входят следующие элементы:

- система управления базой данных (СУБД);
- сервер приложений;
- система лицензионной защиты;
- клиентское приложение.

Такая схема обеспечивает равномерное распределение вычислительной нагрузки в локальной сети и хорошую масштабируемость системы в целом.

Для хранения, модификации и обработки информации используется СУБД производственного назначения, что гарантирует высокую скорость доступа к информации, ее целостность и непротиворечивость дает возможность создания резервных копий данных и восстановления их в случае необходимости.

Основная часть бизнес-логики обработки данных сконцентрирована на сервере приложений. Клиентское приложение предназначено для подготовки и отправки заданий на сервер приложений, а также представления данных, полученных от сервера приложений, в доступной для восприятия конечным пользователем форме.

Система лицензионной защиты предназначена для предотвращения несанкционированного использования и проверки подлинности модулей в процессе работы защищенной системы, а также для обеспечения использования электронных ключей защиты и связанных с ними файлов лицензии.

Сервер приложения предназначен для выполнения в среде виртуальной машины Java, построен на типовой схеме монитора объектных транзакций без запоминания промежуточных состояния. На сервере приложений сконцентрирована основная часть бизнес-логики системы, что позволяет централизованно хранить выполняемые процедуры, а также синхронизировать их работу в случае многопоточного доступа к данным.

Клиентское приложение представляет собой «толстый» клиент и веб-клиент, который предназначен для представления результатов работы сервера приложений конечному пользователю в удобочитаемом виде и подготовке заданий для выполнения сервером приложений.

2.1 Сервер приложений

Сервер приложений системы «АЦК-Планирование» написан с использованием языка программирования Java и предназначен для выполнения в среде виртуальной машины Java (JRE/JDK). Тем самым обеспечивается его независимость от используемой операционной системы и высокая надежность в эксплуатации.

Сервер приложений построен по типовой схеме монитора объектных транзакций (OMT) без запоминания промежуточного состояния.

На сервере приложений сконцентрирована основная часть бизнес-логики системы «АЦК-Планирование», что позволяет централизованно хранить выполняемые процедуры, а также синхронизировать их работу в случае многопоточного доступа к данным.

Доступ к данным, хранящимся в БД, осуществляется сервером приложений посредством драйверов JDBC, что обеспечивает независимость от используемой операционной системы и высокую надежность серверной части.

В связи с тем, что на производительность распределенной вычислительной системы в большой степени влияет пропускная способность локальной сети, рекомендовано размещать сервер приложений на компьютере вместе с сервером БД; либо соединять отдельные компьютеры сервера приложений и сервера БД высокоскоростным выделенным каналом.

Для повышения надежности и (или) производительности всей системы в целом несколько серверов приложений могут быть объединены в кластер серверов приложений, который имеет доступ к одному серверу БД.

Серверы приложений могут работать в режиме *симметричного* кластера. В случае *симметричного режима* все серверы приложений эквивалентны между собой.

Асимметричный режим подразумевает выполнение разными серверами разных задач (например, один занимается обслуживанием клиентских запросов, другой – обработкой входящих сообщений и взаимодействием с другими серверами).

2.1.1 Компоненты сервера приложений

Физически файлы, которые входят в состав сервера приложений, расположены в каталоге `./server/`. В каталоге `./` есть командные файлы, с помощью которых производится

запуск сервера приложений под управлением различных операционных систем.

Таблица 15 – Файлы сервера приложений

Название файла сервера приложений	Описание
./Server.properties	Файл настройки параметров сервера приложений.
./StartServer.bat	Командный файл для запуска сервера приложений под управлением ОС Windows.
./startsrv	Командный файл для запуска сервера приложений под управлением ОС Linux.
./server/AzkServer.jar	Сервер приложений АЦК-Планирование.
./server/core.jar	Утилиты для ядра.
./server/loader.jar	Загрузчик для остальных классов.
./server/Server.jar	Ядро сервера приложений АЦК-Планирование.
./server/Tools.jar	Вспомогательные классы и утилиты.
./server/bin/start	Вспомогательные командные файлы для запуска сервера приложений под управлением ОС Windows.
./server/bin/start.bat	
./server/bin/startserver	
./server/bin/startserver.bat	
./server/lib/cadessignverifyer.dll	Библиотека, реализующая функции проверки УЭП на ЭП-сервере, логику взаимодействия с криптопровайдером.
./server/lib/commons-collections-3.1.jar	Библиотека коллекций.
./server/lib/commons-email-1.2.jar	Библиотека классов для работы с электронной почтой.
./server/lib/commons-logging.jar	Подсистема логирования.
./server/lib/commons-pool-1.6.jar	API для создания пулов объектов.
./server/lib/concurrent.jar	Библиотека конкурентной (совместной) работы.
./server/lib/easymock.jar	Набор классов для тестирования.
./server/lib/easymock.src.jar	
./server/lib/gdiff.jar	Набор классов для построения бинарных диффов.
./server/lib/geronimo-jta_1.0.1B_spec-1.1.jar	Библиотека для файловых устройств хранения данных.
./server/lib/jaybird-full-2.0.1.jar	Драйвер JDBC для подключения к СУБД Firebird.
./server/lib/jdbf.jar	Библиотека для связи с базами данных.
./server/lib/JSQConnect.jar	Драйвер JDBC для подключения к СУБД MS SQL.
./server/lib/junit.jar	Библиотека автоматизированного тестирования.
./server/lib/libsxjdk.RedHat_3_kernel_2.4.so	Модуль лицензионной защиты, определяет доступ к ключам Sentinel.

Название файла сервера приложений	Описание
./server/lib/libsxjdk.RedHat_4_kernel-2.6.9-42.ELsmp.so	
./server/lib/rainbow.jar	
./server/lib/Sxjdk.dll	
./server/lib/log4j-1.2.15.jar	Общепринятая система логирования JAVA.
./server/lib/logkit.jar	Библиотека, реализующая логирование Java Apache Server.
./server/lib/mail.jar	Подсистема отправки и получения электронных сообщений.
./server/lib/ojdbc14.jar	Драйвер JDBC для подключения к СУБД.
./server/lib/avalon-framework.jar	Библиотеки CORBA для построения распределенных приложений.
./server/lib/openorb_orb-1.4.0.jar	
./server/lib/openorb_orb_omg-1.4.0.jar	
./server/lib/openorb_ssl-1.4.0.jar	
./server/lib/tools-1.4.0.jar	
./server/lib/truezip-6.jar	Библиотека для работы с zip-архивами.
./server/lib/xalan.jar	Библиотеки для работы с XML/XSL.
./server/lib/xercesImpl.jar	Библиотека Xerces для обработки документов в формате XML/XSL.
./server/lib/xml-apis.jar	Набор интерфейсов для XML/XSL.
./server/lib/xmlsec-1.3.0.jar	Библиотека для обеспечения безопасности XML/XSL.
./server/lib/ss-poi.jar	Не используются.
./server/lib/ss-poi-contrib.jar	
./server/lib/ss-poi-scratchpad.jar	
./server/lib/ss-poi-scr.jar	
./server/lib/serializer.jar	Сериализация данных в XML. Предназначена для работы приложения Java более поздних версий, чем Java 1.4.2.
./server/lib/log4j-1.2.15.jar	Библиотека, реализующая логирование работы сервера приложений.

Примечание. Для создания базы данных PostgreSQL автоматически создаются файлы: настройки сервера приложений *ServerXXX.properties* и файл для запуска сервера приложений *StartServerXXX.bat*, где XXX - это наименование создаваемой базы данных из *create_db_postgresql.properties* (*db_instance=XXX*).

2.1.2 Утилиты для работы с данными. XML-скрипты

После создания новой БД сервер приложения может ее использовать, но при этом его функциональность будет сильно ограничена. Для полноценного использования сервера приложений нужно заполнить соответствующей информацией системные справочники в БД.

Данные для системных таблиц, а также некоторых стандартизованных справочников находятся в каталоге `./xml/`. Здесь расположены файлы двух типов: xml-скрипты с данными `*.xml` и файлы-списки `*.lst`. **XML-скрипты** – текстовые файлы, которые содержат данные для внесения в БД, выполненные в удобочитаемом виде, в формате XML-документа. **Файлы-списки** – текстовые файлы, которые содержат перечни файлов с данными и определяют порядок загрузки последних в БД. Файлы обоих типов можно просматривать и редактировать с использованием любого текстового редактора.

Для исполнения XML-скриптов и файлов-списков существует утилита XMLExecuter. Используются следующие командные файлы для XMLExecuter:

- `./xml/xml.cmd` – для запуска под управлением ОС Windows;
- `./xml/xml.sh` – для запуска под управлением ОС Linux;
- `./xml/xsl.cmd` – для XSL-преобразований.

[XMLExecuter](#)¹⁴⁷ использует функциональность сервера приложений, посредством которого информация размещается в БД.

2.1.3 Утилиты для работы со структурой БД. SQL-скрипты

Сервер приложений «АЦК-Планирование» использует СУБД для хранения и обработки информации. Для нормальной работы необходимо создать новую БД или использовать существующую, восстановленную из резервной копии.

Все необходимые метаданные для создания и удаления объектов в БД находятся в SQL-скриптах. Они представляют собой текстовые файлы специального вида, которые содержат в себе команды на языке SQL в формализованном виде, пригодном для выполнения на различных типах СУБД. Их можно просматривать и редактировать с использованием любого текстового редактора.

В составе серверной части находится ряд специализированных утилит, предназначенных для упрощения действий обслуживающего персонала по администрированию базы данных.

Утилиты для работы с БД под управлением ОС Windows имеют следующее назначение:

Таблица 16 – Утилиты для работы с БД под управлением ОС Windows

Название утилиты	Описание
./sql/backup.cmd	Создание резервной копии БД.
./sql/comparer.cmd	Сравнение структуры эталонной и целевой БД.
./sql/change_site.cmd	Не используется.
./create_db.cmd	Создание БД начального уровня (создание новой БД, создание объектов БД по SQL-скриптам, заполнение системных справочников, создание бюджета и т.п.).
./create_db_interbase.cmd	Создание новой БД (Firebird).
./sql/DBUpdate.cmd	Обновление БД на версию системы.
./filloutdiff.cmd	Не используется.
./filloutdiff_tfo.cmd	Не используется.
./sql/fix_version.cmd	Установка для записей БД значений VERSION, равных null.
./integrity_check.cmd	Проверка целостности установки системы.
./makediff.cmd	Не используется
./sql/merge.cmd	Объединение данных из нескольких БД в одну.
./sql/perform.cmd	Создание в новой БД всех объектов инфраструктуры по метаданным из SQL-скриптов.
./sql/restore.cmd	Восстановление БД из резервной копии.
./sql/sql.cmd	Выполнение SQL-скриптов для БД.
./sql/update.cmd	Проведение модификации БД при переходе на новые версии.

Также есть командные файлы для запуска вышеперечисленных утилит под управлением ОС Linux. Это одноименные файлы, которые имеют расширение *.sh.

Примечание. Утилита **convert.sh** не используется.

Основной утилитой для работы со структурой БД является [SQLExecuter](#)¹⁴³.

2.1.4 Файл свойств сервера приложений

Параметры сервера приложений настраиваются в файле свойств (имя файла свойств: **Server_<имя_экземпляра>.properties**).

2.1.4.1 Свойства базы данных

1) Режим работы с базой данных:

Имя: azk.db.accessmode

Описание: Указывается СУБД, с которой будет работать сервер (oracle, mssql, interbase):

INTERBASE – сервер работает с СУБД Firebird или Interbase;

ORACLE – сервер работает с СУБД Oracle.

2) Класс драйвера:

Имя: azk.db.driver

Описание: Java-класс драйвера, используемый в «АЦК-Планирование» как средство подключения к БД.

На текущий момент «АЦК-Планирование» использует два вида СУБД:

1. Interbase (и его бесплатный клон Firebird). Для Interbase параметр должен быть равен: **org.firebirdsql.jdbc.FBDriver**.
2. Oracle. Для этой СУБД имя класса должно быть равно: **oracle.jdbc.driver.OracleDriver**.

3) Пользователь базы данных:

Имя: azk.db.user

Описание: Имя пользователя.

4) Пароль для пользователя базы данных:

Имя: azk.db.password

Описание: Пароль пользователя базы данных.

5) Путь к базе данных:

Имя: azk.db.url

Описание: URL (Universal Resource Locator) файла/схемы базы данных в сети или на локальной станции. В зависимости от выбранной СУБД формат пути к базе данных «АЦК-Планирование» различен.

1. Для СУБД Firebird/Interbase формат пути должен быть такой:

```
azk.db.url=jdbc:firebirdsql:<имя_хоста>[/<порт>]:<путь_к_файлу_базы_данных>
```

jdbc:firebirdsql – подключение выполняется с помощью jdbc и именно к Firebird (или Interbase);

<имя_хоста> – имя машины (или IP-адрес), на которой установлен сервер Firebird и находится база данных. (Для локальной машины – localhost или 127.0.0.1);

<порт> - порт, на котором ожидает соединения сервер Firebird; параметр является необязательным и по умолчанию, если его не указывать, равен 3050;

<путь_к_файлу_базы_данных> – абсолютный путь к файлу базы данных.

Внимание! Для ОС Windows в указании пути слешы должны быть прямыми (/).

Примеры путей:

```
azk.db.url=jdbc:firebirdsql:localhost:D:/DB/azk.fdb
```

```
azk.db.url=jdbc:firebirdsql:lnxstand.bftcom.com:/home/ott/azk2/9/lastBuild/db/azk29.fdb
```

```
azk.db.url=jdbc:firebirdsql:localhost/2055:C:/DB/azk.fdb
```

2. Для СУБД Oracle формат пути различается:

```
azk.db.url=jdbc:oracle:thin:@<имя_хоста>:<порт>:<имя_базы_данных>
```

<имя_хоста> – имя машины с установленным сервером Oracle;

<порт> – номер порта сервера Oracle (не зависимо от вида ОС -1521);

<имя_базы_данных> – имя базы данных, созданной на Oracle (например: azk).

Пример пути:

```
azk.db.url=jdbc:oracle:thin:@rh9test:1521:azk
```

Вместо **thin** для подключения может быть использован **oci8**-драйвер. Для этого необходимо наличие клиента Oracle на сервере приложений. В строке URL соединения задается:

```
azk.db.url=jdbc:oracle:oci8:@<tns_name>
```

oracle:oci8 – подключение к БД осуществляется через **oci8**-драйвер;

<tns_name> – имя, зарегистрированное в **tnsnames.ora** на сервере приложений.

Пример пути:

```
azk.db.url=jdbc:oracle:oci8:@azk
```

- 6) Роль пользователя базы данных, используемая при подключении:

Имя: azk.db.role

Описание: Параметр позволяет задать роль, которая будет использована при подключении

сервера приложений к СУБД. Необязательный параметр.

Значение по умолчанию: пустое значение.

7) Максимальное количество подключений к серверу базы данных:

Имя: azk.db.pool.size

Описание: Максимальное количество подключений к серверу баз данных, которое может использовать сервер приложений «АЦК-Планирование».

Значение по умолчанию: azk.db.pool.size =10

При интенсивной работе на «АЦК-Планирование», этого количества может не хватить. В таких случаях рекомендуется увеличить пул соединений на 50%.

8) Максимальное время соединения, при полностью занятом pool:

Имя: azk.db.pool.timeout

Описание: Время в миллисекундах, по истечении которого если ни одно соединение не освобождается, при максимальном количестве подключений к базе данных сервер возвращает исключение клиенту.

Значение по умолчанию: azk.db.pool.timeout=5000

9) Размер кэша:

Имя: azk.db.cachesize

Описание: Размер сессионного кэша откомпилированных запросов.

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

Значение по умолчанию: azk.db.cachesize=50

10)Время выполнения соединения:

Имя: azk.db.connectionlifetime

Описание: Время (в минутах), по истечении которого соединение будет закрыто. Параметр обеспечивает освобождение ресурсов в случае неиспользования сервера приложений, а также ротацию соединений. Необязательный параметр.

Значение по умолчанию: azk.db.connectionlifetime=20

При **azk.db.connectionlifetime = 0** соединения никогда не закрываются.

11)Максимальное время выполнения запроса:

Имя: azk.db.maxquerytimewarn

Описание: Время выполнения запроса (в миллисекундах), при превышении которого будет

выдаваться предупреждение в log. Работает только при включенной трассировке (**azk.db.traceenabled=true**). Необязательный параметр.

Значение по умолчанию: azk.db.maxquerytimewarn=2000

При **azk.db.maxquerytimewarn = 0** предупреждение никогда не выдается.

12) Вывод в лог SQL-трассировки запросов:

Имя: azk.db.traceenabled

Описание: Логический параметр, определяющий включение в лог вывода SQL-трассировки запросов, их параметров, времени выполнения. Необязательный параметр.

Значение по умолчанию: azk.db.traceenabled=true (выводить в лог).

13) Включение сбора SQL-статистики:

Имя: azk.db.statenabled

Описание: Логический параметр, определяющий включение сбора SQL-статистики. Статистика собирается в памяти сервера приложения. При выполнении задания SQLTracer эта информация записывается в лог (уровень вывода INFO). Необязательный параметр.

Значение по умолчанию: azk.db.statenabled=false

2.1.4.2 Свойства лога сервера приложений

Механизм логирования АЦК использует инструментарий Simple Logging Facade for Java (SLF4J). Для работы механизма в файле **Azk2Server.properties** необходимо указать ссылку на файл настроек **Azk2logback.xml**.

```
logback.configurationFile=Azk2logback.xml
```

Лог-AZK настраивается в xml файле **Azk2logback.xml**.

Подробное описание можно найти на сайте logback.qos.ch.

1) Параметры журнализации в консоль:

```
<appender name="stdout" class="ch.qos.logback.core.ConsoleAppender">
```

```
<filter class="ch.qos.logback.classic.filter.ThresholdFilter">
```

Настраивается уровень журнализации сервера приложений в стандартный поток вывода. Чем выше уровень журнализации, тем меньше сообщений попадает в журнал (лог-файл). Возможные значения уровня журнализации: *DEBUG, INFO, WARN, ERROR, FATAL, OFF*:

```
<level>DEBUG</level>
```

```
</filter>
```

```
<encoder>
```

Настраивается используемая кодировка для вывода на консоль. Возможные значения кодировки: Cp866, Cp1251, KOI8_R, ISO8859_5:

```
<charset>Cp866</charset>
```

Настраивается формат вывода:

```
<pattern>%d{dd.MM.yy HH:mm:ss.SSS},%t,%p,%c{0},%m%n</pattern>
```

```
</encoder>
```

```
</appender>
```

2) Параметры журнализации в файл:

```
<appender name="file" class="ch.qos.logback.core.rolling.RollingFileAppender">
```

```
<filter class="ch.qos.logback.classic.filter.ThresholdFilter">
```

Настраивается уровень журнализации сервера приложений:

```
<level>TRACE</level>
```

```
</filter>
```

```
<append>>true</append>
```

Настраивается путь к файлу и имя файла лога:

```
<File>azk 2.log</File>
```

```
<encoder>
```

Настраивается используемая кодировка для вывода в файл. Возможные значения кодировки: Cp866, Cp1251, KOI8_R, ISO8859_5, UTF-8:

```
<charset>UTF-8</charset>
```

Настраивается формат вывода:

```
<pattern>%d{dd.MM.yy HH:mm:ss.SSS},%t,%p,%c{0},%m%n</pattern>
```

```
</encoder>
```

```
<rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
```

Настраивается размер "револьверного" буфера архивов логов. При формировании нового архива лога старые файлы будут стираться. Если размер не указан, по умолчанию будет равен 7:

```
<maxIndex>5</maxIndex>
```

Настраивается путь к файлу и имя файла, используемое для сохранения архивов логов:

```
<FileNamePattern>log/azk2log.%i.log.zip
```

```
</FileNamePattern>
```

```
</rollingPolicy>
```

```
<triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
```

Настраивается максимальный размер файла лога в мегабайтах. После достижения этого размера, лог будет удален или запись будет продолжаться в новый файл, а старый файл будет заархивирован:

```
<MaxFileSize>100MB</MaxFileSize>
```

```
</triggeringPolicy>
```

```
</appender>
```

```
<root level="TRACE">
```

```
<appender-ref ref="stdout"/>
```

```
<appender-ref ref="file"/>
```

```
</root>
```

3) Параметры журнализации в файл конкретного класса (по умолчанию закомментированы):

```
<!--<appender name="classappender" class="ch.qos.logback.core.rolling.RollingFileAppender"-->
```

Настраивается путь к файлу и имя файла лога:

```
<!--<File>azk2class.log</File-->
```

```
<!--<encoder-->
```

Настраивается используемая кодировка для вывода в файл. Возможные значения кодировки: Cp866, Cp1251, KOI8_R, ISO8859_5:

```
<!--<charset>UTF-8</charset-->
```

Настраивается формат вывода:

```
<!--<pattern>%d{dd.MM.yy HH:mm:ss.SSS},%t,%p,%c{0},%m%n</pattern-->
```

```
<!--</encoder-->
```

```
<!--<rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy"-->
```

Настраивается размер "револьверного" буфера архивов логов. При формировании нового архива лога старые файлы будут стираться. Если размер не указан, по умолчанию будет равен 7:

```
<!--<maxIndex>5</maxIndex-->
```

Настраивается путь к файлу и имя файла, используемое для сохранения архивов логов:

```
<!--<FileNamePattern>/log/azk2classlog.%i.log.zip-->
```

```
<!--</FileNamePattern-->
```

```
<!--</rollingPolicy-->
```

```
<!--<triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy"-->
```

Настраивается максимальный размер файла лога в мегабайтах. После достижения этого размера, лог будет удален или запись будет продолжаться в новый файл, а старый файл будет заархивирован:

```
<!--<MaxFileSize>100MB</MaxFileSize-->
```

```
<!--</triggeringPolicy-->
```

```
<!--</appender-->
```

Настраивается класс уровня логирования:

```
<!--<logger name="com.bssys.server.invocation.Server" level="DEBUG"-->
```

```
<!--<appender-ref ref="classappender"/-->
```

```
<!--</logger-->
```

2.1.4.3 Свойства управления заданиями сервера

1) Время хранения результатов асинхронного задания:

Имя: azk.job.finishedtime

Описание: Время в минутах, по истечении которого, результаты завершенного асинхронного задания будут удалены с сервера приложений.

Значение по умолчанию: azk.job.finishedtime=15

2) Анализ асинхронных заданий:

Имя: azk.job.sweepinterval

Описание: Периодичность анализа асинхронных заданий на завершенность в минутах.

Значение по умолчанию: azk.job.sweepinterval=10000

2.1.4.4 Управление пользовательскими сессиями

1) Тайм-аут неактивных сессий:

Имя: `azk.session.finishedtime`

Описание: Время в минутах, по прошествии которого сессия, в которой не было активности пользователей, закрывается.

Значение по умолчанию: `azk.session.finishedtime=600`

2) Размер кэша сессий пользователей:

Имя: `azk.session.cachesize`

Описание: Максимальное количество сессий пользователей, которое сервер приложений удерживает в кэше.

Значение по умолчанию: `azk.session.cachesize=100`.

3) Проверка корректности сессии:

Имя: `azk.session.check`

Описание: Проверять ли корректность сессии при каждом вызове? Должно быть установлено в `true` при использовании кластеров серверов приложений иначе будут проблемы при административном принудительном закрытии сессий.

Значение по умолчанию: `azk.session.check=false`

4) Количество контекстов безопасности, которое сервер приложений удерживает в кэше:

Имя: `azk.security.cachesize`

Описание: Контекст безопасности содержит информацию о правах доступа пользователя к объектам АЦК-Планирование. Контекст используется всеми процедурами сервера приложения. Если не используется кластер серверов приложений, и у каждого пользователя по одной сессии, то параметр может быть равным кэшу сессий пользователей (`azk.session.cachesize`). В случае использования кластера параметры надо соотносить с количеством пользователей, приходящихся на 1 кластер, то есть поровну на все кластеры.

Значение по умолчанию: `azk.security.cachesize=100`

5) Проверка доступа после каждого метода сервера приложений:

Имя: `azk.security.checkactions;`

Описание: Проверять ли права доступа при вызове каждого метода сервера приложений? Установка в `true` этого свойства немного замедляет работу системы, но увеличит ее безопасность.

Значение по умолчанию: azk.security.checkactions=false

2.1.4.5 Свойства сетевых параметров сервера

1) Порт сервера приложений:

Имя: iiop.port

Описание: Порт, на котором работает сервер приложений. Лучше задавать это значение из командной строки и иметь общий файл настроек для всех серверов кластера.

Значение по умолчанию: iiop.port=2001

2) Хост для идентификаторов объектов CORBA:

Имя: iiop.hostname

Описание: Что использовать как 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). Опять же в случае использования кластера желательно задавать его с командной строки, если это нужно.

Для сервера приложений под управлением ОС Windows этот параметр **должен** быть закомментирован, иначе подключение клиентов к серверу будет невозможным. Для систем класса Unix должен указываться хост, на котором запущен сервер CORBA. Если на хосте установлено несколько сетевых интерфейсов, то необходимо для CORBA определить порядок их прослушивания.

Пример:

```
iiop.alternateAddr.endpoint1=192.168.0.21:2001
```

```
iiop.alternateAddr.endpoint2=192.168.0.22:2001
```

Важно! Чтобы клиентское приложение работало на удаленной машине (под управлением ОС Windows 2003), нужно в файле свойств сервера (**Server.properties**) в переменной **iiop.hostname** указать имя сервера.

3) Использование SSL:

Имя: azk.UseSSL

Описание: При значении **true** используется зашифрованный канал SSL.

Значение по умолчанию: azk.UseSSL=true

4) Порт для подключения по SSL:

Имя: ssl_iop.port

Описание: Порт, используемый для шифрованного соединения клиентов и сервера по каналу SSL.

Значение по умолчанию: ssliop.port=2002

- 5) Максимальное количество CORBA потоков сервера приложений, обслуживающих клиентские запросы:

Имя: openorb.server.maxThreadPoolSize

Описание: Параметр используется для изменения (ограничения или увеличения) нагрузки на сервер приложений в зависимости от его мощности. Если к серверу приложений обратится большее количество пользователей, то обращения встанут в очередь.

Значение по умолчанию: openorb.server.maxThreadPoolSize=100

- 6) Параметр подключения веб-сервера при наличии на хосте нескольких сетевых интерфейсов:

Имя: java.rmi.server.hostname

Описание: Если на хосте несколько сетевых интерфейсов, возможно задание параметра подключения веб-сервера явно, указываются данные из файла web.properties файла настроек веб-сервера

Значение по умолчанию: java.rmi.server.hostname=<host:port интерфейс куда подключается web>

- 7) Параметр для использования сервера приложений только на одном системном сервере приложений:

Имя: azk.rpllog.service.disabled

Описание: При использовании более одного сервера приложений только на одном (системном) сервере приложений устанавливается значение **false**. На остальных серверах приложений устанавливается значение **true**.

Значение по умолчанию: azk.rpllog.service.disabled=false

При использовании более одного сервера приложений только на одном (системном) СП установить значение <azk.rpllog.service.disabled=false>

на остальных СП установить значение <azk.rpllog.service.disabled=true>

azk.rpllog.service.disabled=false

2.1.4.6 Настройки пула потоков

- 1) Максимальное количество потоков:

Имя: azk.pooledexecutor.maxpoolsize

Описание: Максимальное количество потоков должно быть ограничено, и должно зависеть от мощности сервера. Для отключения многопоточной работы пула нужно установить значение параметра в 0 (ноль).

Значение по умолчанию: `azk.pooledexecutor.maxpoolsize = 8`

2) Минимальное количество потоков:

Имя: `azk.pooledexecutor.minpoolsize`

Описание: Минимальное количество потоков следует установить равным количеству процессоров с учетом гипертрединга.

Значение по умолчанию: `azk.pooledexecutor.minpoolsize = 2`

3) Размер очереди команд:

Имя: `azk.pooledexecutor.queuecapacity`

Описание: Размер очереди команд лучше ставить большой.

Значение по умолчанию: `azk.pooledexecutor.queuecapacity = 200`

2.1.4.7 Свойства репликации

1) Максимальный размер пакета репликации:

Имя: `azk.rpl.maxpacketize`

Описание: Максимальный размер пакета репликации, при обмене серверов приложений «АЦК-Планирование». Используется для репликатора 2-ой версии.

Значение по умолчанию: `azk.rpl.maxpacketize = 200000`

2.1.4.8 Логирование версий записей

1) Количество дней, за которое сохраняется EraseLog:

Имя: `azk.eraselog.daysleft`

Описание: Количество дней хранения лога EraseLog.

Значение по умолчанию: `azk.eraselog.daysleft=30`

2.1.4.9 OpenORB

1) Параметры отладки OpenORB:

Имя: openorb.debug.level, openorb.debug.trace, javax.net.debug

Описание: По умолчанию параметры отладки OpenORB закомментированы:

- **openorb.debug.level** – уровень отладки OpenORB.
- **openorb.debug.trace** – уровень трассировки OpenORB.
- **javax.net.debug** – уровень сетевой отладки (для Java).

2.1.4.10 Управление вложениями в документах

1) Файл свойств хранилища данных:

Имя: azk.storage.conf

Описание: Файл настроек для хранилища файлов, вложенных в документы «АЦК-Планирование».

Пример:

```
azk.storage.conf=k:/azk2/storages.xml
```

2) Класс, управляющий хранением вложений:

Имя: azk.docattach.class

Описание: Имя класса в «АЦК-Планирование», выполняющего функции вложения и хранения файлов вложений.

Значение по умолчанию: azk.docattach.class=com.bssys.azkserver.udod.AzkDocAttachEx

2.1.4.11 Свойства системы автоматического обновления

1) Путь к клиенту:

Имя: azk.clientupdate.path

Описание: Относительный путь от рабочего каталога сервера к клиенту «АЦК-Планирование». Этот путь используется системой автоматического обновления для обновления клиентской части системы.

Значение по умолчанию: azk.clientupdate.path=client/

2) Управление системой автоматического обновления:

Имя: azk.update.active

Описание: логический параметр, задающий включение системы автоматического обновления:

- **0 (no)** – подсистема обновлений выключена, обновления не работают;
- **1 (yes)** – подсистема включена, и при каждом запуске сервера приложений выполняется проверка работоспособности подсистемы.

Значение по умолчанию: azk.update.active=0

3) Пути к каталогам сборки «АЦК-Планирование» для системы автоматического обновления: сервер, клиент, XML, SQL и корневой каталог соответственно.

Значения по умолчанию:

azk.update.server_files=server

azk.update.client_files=client

azk.update.xml_files=XML

azk.update.sql_files=SQL

azk.update.root_files=.

Пути можно указывать относительно корневого каталога (тот каталог, из которого запускается сервер, и в котором находятся StartServer.bat, startsrv и *.properties) либо абсолютные.

4) Имя общедоступной папки клиента:

Имя: azk.update.client_shares

Описание: Часто каталог с клиентом «АЦК-Планирование» может быть открыт для общего доступа другим пользователям сети. Параметр должен соответствовать сетевому имени данного каталога. Если каталог «расшарен» под несколькими именами, то эти имена можно перечислить через «;».

Значение по умолчанию: azk.update.client_shares=azkclient

Примеры:

azk.update.client_shares=azkclient; client2_8; share.

5) Путь к winupdate.exe (только для ОС Windows):

Имя: azk.update.winupdate

Описание: Путь относительно сервера приложений к клиентской утилите обновления winupdate.

Значение по умолчанию: azk.update.winupdate=client/winupdate.exe

6) Пользователь системы обновления и его пароль:

Имя: azk.update.user, azk.update.password

Описание: Пользователь сервера приложений, под которым будет происходить обновление, и от

имени которого будет производиться обновления, а также пароль этого пользователя.

azk.update.user=updater

azk.update.password=retadpu

7) Путь к лог-файлу системы обновления:

Имя: azk.update.log.filename

Описание: Во время процесса обновления ведется лог, где отражается ход процесса автоматического обновления «АЦК-Планирование», а также ошибки, которые при этом могут возникнуть.

***Внимание!** Файл лога обновления НЕ должен быть в корневом каталоге, иначе он будет перезаписан при обновлении!*

Значение по умолчанию: azk.update.log.filename=update/update.log

8) Уровень информативности вывода в лог:

Имя: azk.update.log.mode

Описание: Уровень лога обновления, записываемого в файл (от 0 – все записи, до 4 – только исключения).

Значение по умолчанию: azk.update.log.mode=0

9) Уровень информативности вывода на экран при выполнении обновления:

Имя: azk.update.log.screenmode

Описание: Уровень лога обновления, выводимого на консоль (от 0 – все записи, до 4 – только исключения).

Значение по умолчанию: azk.update.log.screenmode=1

10) Порт временного сервера приложений:

Имя: azk.update.port

Описание: Для исполнения новых XML-файлов, содержащихся в пакете обновления, система обновления поднимает временный сервер приложений. Порт этого сервера не должен совпадать с портами других серверов приложений «АЦК-Планирование» на обновляемой станции (а также не должен быть занят другими приложениями).

Значение по умолчанию: azk.update.port=3001

11) Максимальное количество попыток связи с сервером приложений:

Имя: azk.update.upserver.tries

Описание: Сколько раз пытаться связаться с временным сервером приложений. Если количество

попыток превысило этот параметр, то производится откат и обновление считается неудавшимся.

Значение по умолчанию: azk.update.upserver.tries=60

12) Период между попытками связи с временным сервером приложений (указывается в секундах):

Имя: azk.update.upserver.timeout

Значение по умолчанию: azk.update.upserver.timeout=5

15) Путь к GBAK (только для Firebird):

Имя: azk.update.backup.gbak

Описание: Путь к утилите Firebird, выполняющей backup/restore баз данных.

Значение по умолчанию: azk.update.backup.gbak=C:/fb/bin/gbak.exe

16) Тайм-аут для GBAK:

Имя: azk.update.backup.timeout

Описание: По истечении времени gbak объявляется зависшим и обновление откатывается.

Значение по умолчанию: azk.update.backup.timeout=3600

2.1.4.12 Настройка репликации нового сайта

Не используется.

2.1.4.13 Настройка параметров ЭП

1) Адрес сервера приложений, на котором осуществляется проверка ЭП:

Имя: azk.sign.hostname

Описание: Если адрес сервера приложений не указан, то проверка осуществляется на текущем сервере.

Значение по умолчанию: azk.sign.hostname=nazgul17

2) Порт сервера приложений, на котором осуществляется проверка ЭП:

Имя: azk.sign.port

Описание: Указывается порт сервера приложений, на котором осуществляется проверка ЭП.

Значение по умолчанию: azk.sign.port=2017

3) Имя пользователя, под которым выполняется проверка ЭП:

Имя: azk.sign.user

Описание: Указывается имя пользователя, который выполняет проверку ЭП.

Значение по умолчанию: azk.sign.user=root

4) Пароль пользователя, под которым выполняется проверка ЭП:

Имя: azk.sign.pswd

Описание: Указывается пароль пользователя, который выполняет проверку ЭП.

Значение по умолчанию: azk.sign.pswd=toor

5) Конфигурация birt

Имя: birt.engine.dir

Значение по умолчанию: birt.engine.dir=./server/lib/birt-report-engine/

2.1.4.14 Настройка параметров лицензии

1) Путь к файлу лицензии:

Имя: azk.license.name

Значение по умолчанию: azk.license.name=bft.lic

2.1.4.15 Поддержка HTTP-соединения для Win-клиента

1) Параметры встроенного томката:

Имя: azk.tomcat.embedded.port

Описание: Указывается порт 8082. После изменения значения настройки следует перезапустить сервер приложений.

Пример:

azk.tomcat.embedded.port = 8082

2) Параметры сервлета, принимающего команды от делфи-клиента

Имя: azk.servlet.exec.mapping

Описание: Указывается сервлет для обслуживания запросов с Win-клиента. После изменения значения настройки следует перезапустить сервер приложений.

Пример:

```
azk.servlet.exec.mapping = /exec
```

2.1.4.16 Настройка кластерного режима работы

1) Параметр отключения сервиса RPL\$ASSIGN_GENERATION:

Имя: azk.rpllog.service.disabled

Описание: для отключения сервиса на сервере приложений параметру необходимо присвоить значение `=true`.

Значение по умолчанию: сервис разрешен.

2.1.4.17 Настройка взаимодействия с ЕИС

Для настройки взаимодействия с ЕИС в файле **Azk2Server.properties** необходимо указать следующее:

```
# -----  
# Настройка взаимодействия с ЕИС  
# -----  
# Использовать ли проху  
# https.proxySet=true  
# Адрес проху  
# https.proxyHost=<proxy_host>  
# Порт проху  
# https.proxyPort=<proxy_port>  
# Имя пользователя для авторизации на проху  
# https.proxyUser=<proxy_user>  
# Пароль пользователя для авторизации на проху  
# https.proxyPassword=<proxy_password>  
# Путь к хранилищу доверенных сертификатов  
# javax.net.ssl.trustStore=<truststore_path>  
# Пароль к хранилищу доверенных сертификатов  
# javax.net.ssl.trustStorePassword=<truststore_password>  
--  
# Адрес сервера, который осуществляет отправку на ЕИС. Если не указано, то
```

```
сервером выступает текущий сервер
# azk.eis.sender.hostname=<hostname>
# Порт сервера, который осуществляет отправку на ЕИС
# azk.eis.sender.port=<port>
# Пользователь, под которым осуществляется отправка на ЕИС
# azk.eis.sender.user=<user>
# Пароль пользователя, под которым осуществляется отправка на ЕИС
# azk.eis.sender.password=<password>
```

2.1.4.18 Настройка взаимодействия с системой Аванпост

Для настройки взаимодействия с системой Аванпост в файле **Azk2Server.properties** необходимо указать следующее:

```
# -----
# ! Настройка FAM!
# -----
#внутреннее название SSO сервиса
azk.openid.provider.id=FAM
#адрес сервиса 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
#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
```

2.1.4.19 Настройка хранилища ключей доверенных сертификатов

Для настройки хранилища ключей доверенных сертификатов в файле **Azk2Server.properties** необходимо указать следующее:

```
# -----  
# Настройки хранилища ключей доверенных сертификатов  
# -----  
#javax.net.ssl.trustStoreType=HDImageStore # всегда такое значение  
javax.net.ssl.trustStore=C:\\trust.store # путь к хранилищу (путь к файлу, в котором  
сохраняется вновь создаваемый контейнер)  
javax.net.ssl.trustStorePassword=111 # пароль к хранилищу (пароль к контейнеру не  
должен быть пустым)  
#javax.net.ssl.supportGVO=true # всегда такое значение
```

2.1.4.20 Настройка прокси-сервера по протоколу https

В случае необходимости, если доступ в Интернет со стороны сервера приложений осуществляется не напрямую, а через прокси-сервер, произвести настройку прокси-сервера для доступа по протоколу https в соответствующем разделе файла **Azk2Server.properties**:

```
# -----  
# Настройка прокси для доступа по защищенному протоколу HTTPS  
# (как правило, должны задаваться  
# те же настройки что и для HTTP)  
# -----  
https.proxySet=true  
https.proxyHost=proxy.bftcom.com  
https.proxyPort=8080  
https.proxyUser=user  
https.proxyPassword=pass
```


2.1.4.21 Кэширование профилей на сервере АЦК

Для кэширования профилей на сервере АЦК настраиваются параметры:

1) Включение и выключение кэширования:

Имя: azk.profile.cache.enabled

Описание: допустимые значения true/false. Не рекомендуется выбирать значение false.

Значение по умолчанию: true.

2) Включение и выключение асинхронной записи измененного профиля:

Имя: azk.profile.cache.asyncSave

Описание: допустимые значения true/false.

Значение по умолчанию: true.

3) Максимальный размер кэша:

Имя: azk.profile.cache.maxSize

Описание: Влияет на количество кэшируемых профилей. Рекомендуется устанавливать размер кэша не менее, чем число активных пользователей

Значение по умолчанию: 10000.

4) Включение и выключение сжатие кэша (интернирование строк ключей и значений):

Имя: azk.profile.cache.compressed

Описание: допустимые значения true/false.

Значение по умолчанию: true.

5) Период проверки актуальности закэшированного профиля в секундах:

Имя: azk.profile.cache.checkInterval

Значение по умолчанию: 300.

2.1.4.22 Настройка сервиса помощи входа по сертификату (СПВС)

Для настройки сервиса помощи входа по сертификату (СПВС) в файле **Azk2Server.properties** необходимо указать следующее:

1) Адрес сервиса идентификации СБ

Имя: azk.integration.security_service.identification.uri

Описание: Настройка для авторизации пользователя по jwt токену.

Значение по умолчанию: azk.integration.security_service.identification.uri=localhost:8080/api

2.1.4.23 Настройка сервиса передачи сообщений при формировании отчетов по Приказу 243н

Для настройки сервиса передачи сообщений в подсистему АРМ «Оператор» при формировании отчетов по Приказу 243н в файле **Azk2Server.properties** необходимо указать путь к файлу с настройками сервиса в параметре **azk.server.msgchecker.config**.

Пример:

```
azk.server.msgchecker.config=../config/msgcheckerconfig.xml
```

Внимание! Файл с настройками сервиса должен быть в кодировке UTF-8.

Описание формата:

- Атрибуты **msgchecker**:
 - **active** – атрибут активности сервиса;
 - **threadcount** – количество потоков работы сервиса;
 - **timeout** – период опроса таблицы.
- Элементы **msg** содержат данные о том, какие сайты и какие задания для этих сайтов должны выполняться, чтобы сообщения были отправлены:
 - **site_id** – ID сайта;
 - **task_name** – имя задания в расписании планировщика.

Пример структуры файла:

```
<msgchecker active="true" threadcount="4" timeout="60">  
<msg site_id="111" task_name="243n-rpl"/>  
  <msg site_id="101" task_name="243n-reports"/>  
</msgchecker>
```

2.2 Система лицензионной защиты

В системе лицензионной защиты используются электронные ключи защиты и связанные с ними файлы лицензий.

Электронные ключи защиты и файлы лицензии устанавливаются на компьютере, на котором установлен сервер приложений. Электронный ключ защиты связывает между собой зарегистрированного клиента и экземпляр системы «АЦК-Планирование». Файл лицензии содержит:

- атрибуты, которые связывают его с электронным ключом защиты;
- список подсистем и модулей, разрешенных к использованию;
- список дополнительных параметров, определяющих условия использования системы.

2.3 Клиентское приложение

2.3.1 Описание клиентского приложения

Клиентское приложение представляет собой «толстый клиент», который предназначен для представления результатов работы сервера приложений конечному пользователю в удобочитаемом виде и подготовке заданий для выполнения сервером приложений. Клиентское приложение спроектировано для исполнения только в графической среде ОС Windows.

Для увеличения скорости работы приложения, а также для уменьшения трафика по сети и нагрузки на сервер приложений и сервер БД, клиентское приложение использует кэш информации некоторых, наиболее часто используемых справочников. Кэш с данными хранится локально в каталоге пользователя. Он считывается в структуры клиентского приложения при запуске. Информация сохраняется в кэше при закрытии приложения.

Работа по обработке пользовательского интерфейса в клиентской части проводится в одном потоке. Клиентская часть не предусматривает многопоточной обработки.

Для хранения данных в клиентской части реализован компонент DataStore, предназначенный для хранения локальных данных и для взаимодействия с серверными

провайдерами.

Файлы клиентской части расположены в каталоге **.client/**. Здесь находится основной запускаемый модуль, динамические библиотеки с клиентскими объектами и несколько утилит.

Таблица 17 – Каталоги и файлы клиентского приложения

Файлы клиентского приложения	Описание
maincontroller.exe	Основной запускаемый модуль клиентского приложения.
Accounting.ocx	Динамическая библиотека, содержащая клиентские объекты для работы бухгалтерии.
Analysis.ocx	Не используется.
Asset.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с документами по активам.
Azk2DocFlow.ocx	Динамическая библиотека, содержащая клиентские объекты для просмотра связанных документов, бюджетных и бухгалтерских проводок.
Azk2Imp.ini	
Azk2Kernel.ocx	Динамическая библиотека, содержащая клиентские объекты ядра «АЦК-Планирование».
.Tools/banks2xml.exe	Утилиты для преобразования информации о банках РФ (БД Bankseek) из формата *.dbf в формат XML-скрипта, пригодный для выполнения на сервере приложений.
JWinCryptoLib64	Библиотека для проверки подписи.
BFTlib.ocx	Не используется.
cadessigner.dll	Библиотека для проверки подписи.
CapInvest.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с документами по капитальному строительству.
Credits.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с документами по кредитам и ссудам.
DocFlow.ocx	Динамическая библиотека, содержащая клиентские объекты общего назначения для документооборота.
ExpressSpreadSheetPack.bpl	Не используется.
Expense.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с расходными документами.
Income.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с доходными документами.
InterBudget.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с подсистемой мультибюджетности.
kernel.ocx	Динамическая библиотека, содержащая клиентские объекты ядра программных продуктов серии «АЦК-Планирование».

Файлы клиентского приложения	Описание
OlePack.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с динамическими документами.
Payables.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с модулем по учету специальных средств.
Payment.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с платежными документами.
plandoc.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с документами по планированию бюджета.
planning.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с объектами планирования бюджета.
PlanningTools.ocx	Не используется.
profiles.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с профилями отчетов.
reciprocal.ocx	Не используется.
Reporter.ocx	Динамическая библиотека, содержащая клиентские объекты для построения динамических отчетов.
Reports.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с отчетами.
Servntf.dll	Динамическая библиотека времени выполнения; обеспечивает соединение клиентского приложения с сервером приложений посредством CORBA. Позволяет отправлять XML-задание, получать результат его выполнения, в случае асинхронного выполнения – получать результат выполнения задачи. Клиентский модуль при подключении случайным образом выбирает сервер кластера из списка, чем достигается равномерность загрузки серверов внутри кластера.
Sign.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с ЭП.
Sources.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с документами по источникам.
SpreadSheet.ocx	Не используется.
SpreadSheetPack.bpl	Не используется.
StdRef.ocx	Динамическая библиотека, содержащая клиентские объекты для работы со справочниками.
sysprop.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с системными настройками.
turbopack.bpl	
userprop.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с пользовательскими настройками.
updater.exe	Модули для автоматического обновления.
winupdate.exe	

Файлы клиентского приложения	Описание
az_lib_rt.bpl, AnalysisPack.bpl, Azk2ClientPack.bpl, Azk2ExpensePack.bpl, clientpack.bpl, ehlib.bpl, kerpac.bpl, ReportTools.bpl, RuntimeControls.bpl	Динамические библиотеки времени выполнения, в которых содержится реализация дополнительных компонентов среды программирования Delphi7.
Azk2CInt.ini	Файл конфигурации клиентского приложения.
killcache.bat	Командный файл для очищения кэша данных клиентского приложения.
./Help/Help.chm	Файл контекстной справки (вызываемой по клавише F1) по работе пользователя с объектами системы.
./Tools/execxml.exe	Утилита выполнения XML-команд в интерактивном режиме и из командной строки.

2.3.2 Параметры командной строки клиента

Для запуска клиента с различными дополнительными опциями предусмотрены параметры командной строки, которые надо указывать после исполняемого файла клиента, т.е.: **maincontroller.exe <имя_параметра>**.

Список параметров клиента и их значения:

- **/NOREG** – параметр запрещает регистрацию библиотек и компонентов OCX после установки новой версии клиентского приложения. Библиотеки должны автоматически перерегистрироваться, если указать этот параметр, библиотеки регистрироваться не будут.
- **/REGALL** – принудительная регистрация библиотек, даже если версия клиентского приложения не поменялась.
- **/PASSWORD:<пароль>** – ввод пароля пользователя с консоли, чтобы не указывать его в форме подключения к серверу. Если этот параметр используется без параметра **/USER** (см. ниже), указывается имя пользователя, который последний входил в систему, если пароли не совпадают, на экран выводится сообщение об ошибке.
- **/USER:<имя_пользователя>** – ввод имени пользователя с консоли. Если этот параметр указан, в форме подключения к системе будет заполнено поле имени пользователя. Если использовать этот параметр вместе с

параметром **/PASSWORD**, сервер будет искать введенное в этом параметре, имя пользователя с таким паролем, в случае неудачи на экран выводится сообщение об ошибке.

- **/PREFIX:<имя_префикса>** – задается имя префикса для совместного запуска двух или более клиентов одновременно. Для каждого клиента необходимо указать уникальный префикс, под этим префиксом будет храниться конфигурационная информация в реестре.

Внимание! Не рекомендуется запускать несколько клиентов разных версий, т.к. версии библиотек клиентов также различаются друг от друга, что может привести к возникновению ошибок.

- **/CACHE:<имя_префикса_для_кэша>** – этот параметр задает имя префикса для клиентского кэша, в котором хранится содержимое кэшируемых справочников. Параметр может быть полезен, когда на одной машине запущено несколько клиентов, работающих с разными базами данных. В этом случае если указать этот параметр, то содержимое кэшируемых справочников не будет вызывать конфликты у клиентов.
- **/INSTANCE:<имя_экземпляра>** – параметр предназначен для запуска нескольких клиентов системы «АЦК-Планирование» на одной машине, независимо от префикса реестра (параметра **/PREFIX**).
- **/INI:<имя_файла>** – по умолчанию клиент системы «АЦК-Планирование» использует ini-файл **AZK2CInt_ini**, если необходимо указать другой файл, с альтернативными параметрами подключения, то это можно сделать с помощью параметра.

Пример:

```
maincontroller.exe /ini:Azk2CInt_fb.ini
```

- **/REGNAME:<имя_раздела_реестра>** – параметр служит для использования альтернативной ветви системного реестра. По умолчанию системная информация АЦК-Планирование хранится в ветви реестра: **HKEY_CURRENT_USER\Software\BFT\AZK2**. Если указать этот параметр равным какому-нибудь значению, это значение будет добавляться к номеру версии.

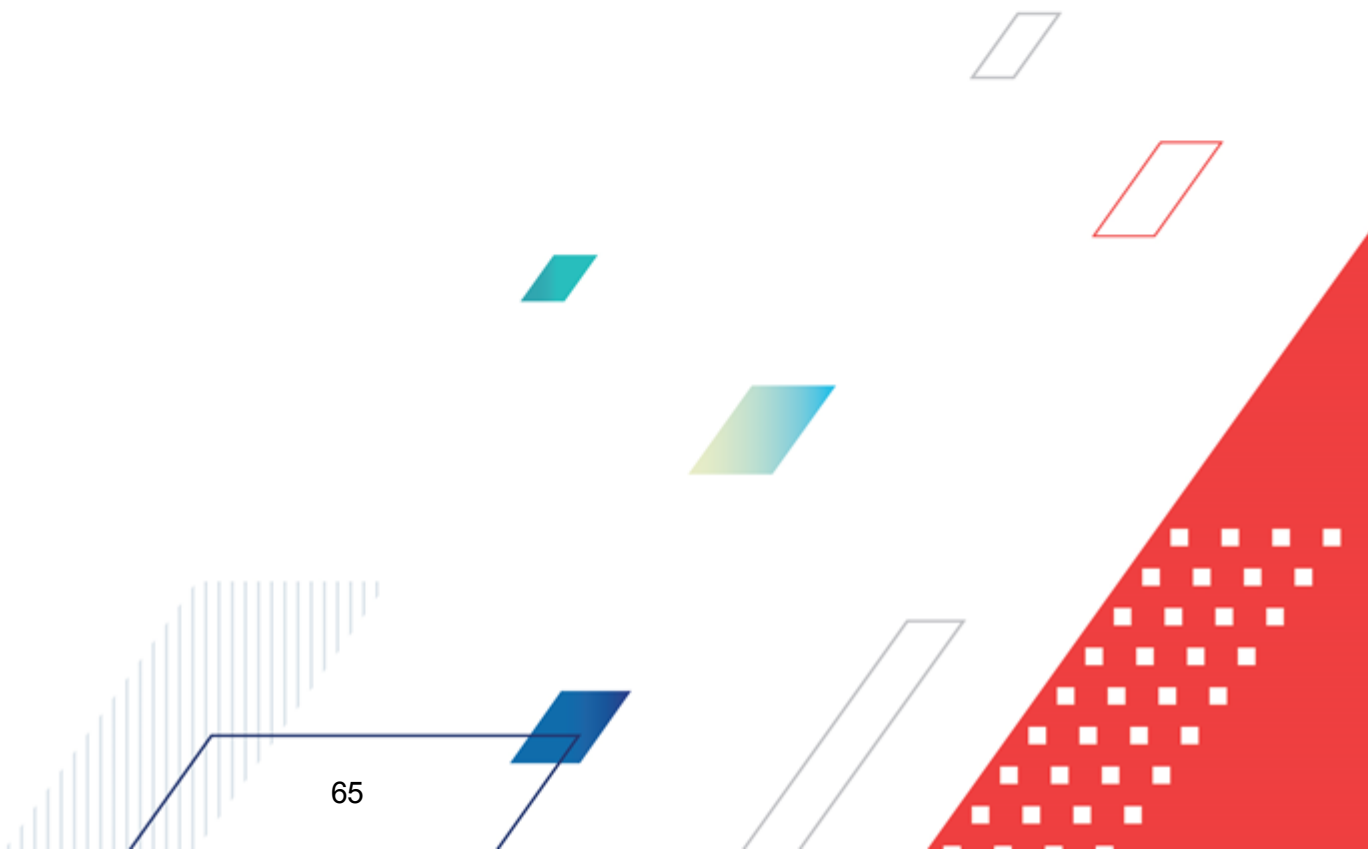
Пример:

Конфигурация клиента версии 2.52 хранится в `HKEY_CURRENT_USER\Software\BFT\AZK2\2.52`, если запустить клиент таким образом: `maincontroller.exe /regname:123`, то информация будет храниться в `HKEY_CURRENT_USER\Software\BFT\AZK2\2.52_123`.

- `/CAPTION:<заголовок_окна>` – параметр задает наименование заголовка окна. Наименование выводится в начале заголовка окна и позволяет определить к какому серверу приложений подключен запущенный пользователем win-клиент. В наименовании заголовка окна указываются буквы и цифры длиной 100 символов.



Основные понятия



Для более полного восприятия состава и структуры системы «АЦК-Планирование» необходимо рассмотреть некоторые концептуальные понятия.

3.1 Клиентские объекты, процессоры, провайдеры

На клиентской части для ведения интерактивного диалога основной единицей взаимодействия с пользователем является *клиентский объект*. Клиентские объекты находятся в файлах динамически подгружаемых библиотек *.osx и фактически представляют собой формы с расположенными на них элементами управления. Перед использованием, клиентские объекты должны быть зарегистрированы в реестре ОС Windows. Для создания экземпляров клиентских объектов и отображения их на рабочем столе используется основной контроллер `.client/maincontroller.exe`.

Условно можно определить три основных типа клиентских объектов:

- **список** – позволяет представить информацию в табличной форме, в виде набора записей; для упрощения навигации по списку и удобства использования есть возможность фильтрации записей по сложному условию и контекстный поиск записи по полям таблицы;
- **справочник** – то же, что и список; справочник представляет собой набор данных, которые редко изменяются в процессе повседневной работы, но постоянно используются для создания и обработки документов;
- **редактор** – представляет запись из списка в удобном для восприятия виде, позволяет редактировать информацию и выполнять различные действия над записью.

Могут встречаться и другие формы сложных диалоговых окон, в зависимости от решаемых задач. К таким клиентским объектам можно отнести формы просмотра, квитанции, сложных справочников и т.д.

Чтобы клиентское приложение могло использовать клиентский объект, последний должен быть зарегистрирован в системной таблице CLIENTOBJECT. Для внесения данных о клиентских объектах в БД обычно используется XML-скрипт `.xml/clientobject.xml`. Для вызова клиентских объектов из основного контроллера используется панель системного

меню. Для внесения данных о системном меню в БД обычно используется XML-скрипт `./xml/menu.xml`.

Доступ к клиентским объектам, а также видимость пунктов системного меню, носит ролевой характер и ограничивается на уровне системных настроек.

Основной единицей обработки информации на сервере приложений «АЦК-Планирование» является *процессор* (processor). **Серверный процессор** (serverprocessor) – это программный модуль сервера приложений, представляющий собой откомпилированный java-класс, который выполняет запросы клиентских объектов. Его основное назначение – совершение действий над одним объектом или группой объектов с учетом переданных ему параметров (атрибутов). Объектом в среде сервера приложений «АЦК-Планирование» может являться любая сущность, которая носит абстрактный или объектный характер (например: документ, запись в справочнике и т.п.). Для определения того, что должен выполнить процессор над объектом, вводится атрибут *действия* (action). Команды, которые передаются для выполнения процессорам, представляются в виде XML-документов; используется следующий формат:

```
<PROCESSOR_NAME action="action_name" att1="att1" . />
```

Чтобы сервер приложений мог обращаться к серверному процессору, последний должен быть зарегистрирован в системных таблицах SERVERPROCESSOR и SERVERACTION. Для внесения данных о серверных процессорах обычно используется XML-скрипт `/xml/serverprocessors.xml`. Информацию о серверных процессорах можно посмотреть и при необходимости отредактировать, используя клиентское приложение.

После окончания работы процессор возвращает результат в формате XML-документа. Данные могут быть использованы клиентскими объектами для отображения результатов для конечного пользователя, либо использованы клиентским приложением для внутренних нужд.

Доступность серверного процессора, а также область его видимости ограничивается на уровне системных настроек.

Для заполнения таблиц в клиентских объектах со стороны серверной части используются *провайдеры* (provider). **Серверные провайдеры** (serverproviders) – программный модуль сервера приложений, представляющий собой откомпилированный java-класс. Основное назначение провайдера – сбор информации в БД и представление ее в удобной для табличного отображения форме с учетом фильтров и дополнительных условий.

Чтобы сервер приложений мог обращаться к серверному провайдеру, последний должен быть зарегистрирован в системной таблице SERVERPROVIDER. Для внесения данных о серверных провайдерах обычно используется XML-скрипт `./xml/serverprovider.xml`.

Доступность серверного провайдера, а также область его видимости ограничивается на уровне системных настроек.

3.2 Бюджет

В связи с тем, что основное назначение АЦК-Планирование – это формирование бюджетной росписи, а также контроль ее исполнения, понятие *бюджет* имеет ключевое значение.

АЦК-Планирование позволяет вести в одной базе данных несколько независимых бюджетов на разные периоды исполнения и легко переключаться между ними при регистрации пользователя в системе.

В зависимости от выбранного бюджета становятся доступными одни данные из БД и недоступны другие.

3.3 Справочники

Информация в базе данных носит связанный характер и построена по принципу подчиненности одних данных другим. Таким образом, достигается много преимуществ при накоплении информации и ее последующей обработке.

Для ускорения ввода однотипных данных, которые, как правило, имеют ограничения по уникальности в таблице баз данных, используются справочники.

В справочниках находится различная по своему назначению информация. В системных справочниках хранится информация необходимая для внутреннего использования подсистемами серверной и клиентской части.

3.4 Документы. Редактор сценариев обработки документа

АЦК-Планирование по своей концепции является системой электронного документооборота. Во внутреннем представлении любой документ представляется в виде объекта, реализации конкретного java-класса.

После создания в системе документ переходит на определенный начальный статус. Над документом в зависимости от его статуса могут выполняться различные действия. В зависимости от результата выполнения действия документ может сменить свой статус. Таким образом, жизненный цикл документа может быть представлен в виде дерева последовательного прохождения от начального статуса до конечного статуса – «*обработка завершена*». На конечном статусе дальнейшая обработка документа невозможна. Далее документ возможно лишь вернуть на предыдущий статус. Для этого предусмотрена операция возврата.

Для визуального построения дерева сценариев существует редактор в клиентской части, в котором в наглядной форме представлены порядок обработки документа при вызове различных методов (**Справочники→Документооборот→Классы документов**).

Примечание. Описание работы Редактора базовых сценариев см. в документации «БАРМ.00004-55 32 01-3 Система «АЦК-Планирование». АРМ ФО. Блок администрирования. Подсистема администрирования. Настройка и сервис системы. Руководство администратора».

3.5 Репликация. Объекты репликации. Реплицируемые таблицы

Под процессом *репликации* понимается приведение нескольких баз данных с одинаковой структурой в одно и то же непротиворечивое (не вызывающее нарушений целостности) состояние, сопровождающееся взаимным внесением изменений. При выполнении процесса БД-издатель выступает в качестве источника изменений, которые принимаются в БД-подписчике, заинтересованной в изменениях. Рассмотрим процесс репликации данных в контексте «АЦК-Планирование».

В основе репликации лежит понятие *объекта репликации*. **Объект репликации** – это набор правил отбора записей из таблиц БД-издателя, которые могут быть представлены в виде абстрактной единицы информации, которую необходимо синхронизировать с таблицами из БД-подписчика. В объекте репликации указывается, какие БД будут выступать издателем и подписчиком изменений.

Чтобы таблицу можно было включить в состав объекта репликации, ее необходимо прописать в БД. Для хранения информации о таблицах служит таблица RPLTABLE, в полях которой необходимо определить наименование реплицируемой таблицы и поля, по которым будет выполняться отбор записей.

В «АЦК-Планирование» поддерживаются два типа объектов репликации:

- **однаправленная репликация** – в этом случае изменения в БД-издателе применяются на БД-подписчике;
- **двунаправленная репликация** – то же, что и однаправленная, только обмен данными происходит с учетом внесенных в БД-подписчика изменений, которые распространяются на БД-издателя.

Репликация данных подразумевает использование внутренних подсистем сервера, которые работают асинхронно и независимо друг от друга. К ним относятся:

- **служба репликации** – обеспечивает просмотр содержимого БД-издателя и подготовку репликационных сообщений (пакетов);
- **служба обмена сообщениями** – обеспечивает своевременную доставку информационных сообщений между БД-издателем и БД-подписчиком;
- **служба обработки сообщений** – обеспечивает своевременную обработку сообщений в зависимости от очередности поступивших сообщений и их приоритета и записи результатов работы в БД-подписчика.

В результате применения правил отбора записей, которые указаны в репликационных объектах, мы получаем совокупность изменений, которым были подвержены все реплицируемые таблицы с момента последнего просмотра БД-источника. Изменения оформляются службой репликации в виде XML-документов (пакетов) и помещаются в очередь сообщений. Служба обмена сообщениями просматривает очередь в поисках новых сообщений, которые отправляет на сервер приложений БД-подписчика. Пакет с изменениями будет рассмотрен службой сообщений и, после последующей обработки,

информация будет размещена в БД-подписчика. В случае двунаправленной репликации последним этапом репликационного взаимодействия будет подготовка и отправка репликационного пакета от БД-подписчика для БД-издателя.

Запуск вышеперечисленных служб осуществляется с помощью подсистемы *Планировщик задач*. Расписание запуска заданий планировщиком в интуитивно понятном диалоге можно настроить из клиентского приложения.

Предусмотрена возможность назначать дополнительные обработчики при подготовке и применении репликационных пакетов. Таким образом, процесс репликации может быть расширен дополнительными функциями, такими как взаимодействие с информационными системами третьих фирм.

3.6 Сайты. Типы соединений сайтов

Сайт – это абстрактное понятие, под которым подразумевается платформа, в состав которой входит сервер (кластер серверов) приложений и база данных. Благодаря этому понятию «АЦК-Планирование» может представлять собой распределенную информационную систему. Процесс репликации построен на использовании понятия *сайтов*.

Сайт в распределенной вычислительной системе должен иметь натуральный уникальный идентификатор. В зависимости от номера сайта будет выполняться идентификация записей в таблицах БД. Идентификатор новой записи (ID) рассчитывается со смещением равным $\langle \text{site_number} \rangle * 1000000000$. Таким образом, в процессе обмена информацией идентификатор добавляемой записи, как основной признак уникальности записи, не будет вызывать исключений нарушения целостности БД.

Номер сайта – это атрибут базы данных, который указывается при создании БД и больше не изменяется на протяжении всего процесса эксплуатации сайта.

В настоящее время существуют следующие типы соединения сайтов в процессе обмена информационными сообщениями:

- **Прямое соединение серверов приложений** – выполняется с использованием CORBA посредством протокола TCP/IP.

- **Обмен файлами** – выполняется посредством обмена файлами-пакетами (используется шифрование). Доставка файлов может быть реализована любым способом (например, T-Mail или каталоги общего доступа в локальной сети).
- **Электронная почта SMTP/POP3** – выполняется посредством обмена электронными письмами (шифрование) с использованием почтовых серверов.
- **ДБО BS-Client v.3** – выполняется с использованием CORBA посредством протокола TCP/IP, используется для связи с сервером WinTransport.
- **Обмен с веб-сервисом** – выполняется посредством POST-запроса на указанный url-адрес.

3.7 Сообщения. Очереди сообщений

В состав сервера приложений «АЦК-Планирование» входит служба обработки сообщений.

Сообщение в среде сервера приложений представляют собой XML-скрипт, который описывает команду в формате, пригодном для исполнения на сервере приложений. Физически сообщения хранятся в базе данных в таблице MSG. При сохранении сообщения в БД дополнительно указываются следующие основные параметры: сайт-отправитель, сайт-получатель, наименование очереди, порядковый номер, приоритет и статус, на котором сообщение находится.

Сообщения по своему назначению, формируют независимые очереди. В очереди сообщениям присваивается сквозная нумерация, в зависимости от которой далее сообщения будут обрабатываться на сайте-получателе. Сообщение после создания может принимать следующие статусы:

- 0 – «новый»,
- 1 – «отправлен»,
- 2 – «получен»,
- 3 – «выполнен»,
- 4 – «завершен».

В системе предусмотрена стандартная функция поиска в теле сообщений заданных сочетаний символов и возможность автозамены заданного сочетания символов на другое сочетание символов. Форма поиска вызывается сочетанием клавиш **Ctrl+F**.

В процессе обмена и просмотра сообщений принимают участие две подсистемы: служба доставки сообщений и служба обработки сообщений. Первая отвечает за своевременную доставку сообщения адресату и обратную доставку отчетов о состоянии отправленных сообщений (квитков). Вторая просматривает очереди сообщений в поисках новых сообщений или пытается обработать сообщения, выполнение которых было прервано исключительной ситуацией со стороны БД-подписчика.

Рассмотрим порядок работы службы доставки сообщений:

- 1) на сайте-отправителе просматриваются сообщения в статусе «*новый (0)*»; после отправки им присваивается статус «*отправлен (1)*»;
- 2) по прибытию на сайт-получатель сообщения сохраняются в статусе «*новый (0)*»;
- 3) сайт-получатель подготавливает квитки о доставленных сообщениях;
- 4) по квиткам на сайте-отправителе доставленные сообщения переходят в статус «*получен (2)*»;
- 5) после успешного прохождения обработки на сайте-получателе сообщения переходят в статус «*выполнен (3)*», и отправляется ответный квиток на сайт-отправитель;
- 6) после отправки квитка, сообщение на сайте-получателе переходит в статус «*завершен (4)*»;
- 7) после получения квитка на сайте-отправителе сообщение переходит в статус «*завершен (4)*».

Порядок работы службы обработки сообщений таков:

- 1) сканируется список сообщений в статусе «*новый (0)*», в которых в поле получателя указан текущий сайт; сортировка по очереди и порядковому номеру;
- 2) для каждой очереди ищется сообщение, порядковый номер которого является следующим относительно последнего исполненного сообщения;
- 3) сообщение исполняется; в случае удачного завершения проставляется статус «*выполнен (3)*» и запоминается номер этого сообщения, как последнего исполненного в перечне очередей; если выполнение закончилось с ошибкой, то

устанавливается статус «получен (2)» и заполняется поле с диагностикой об ошибке.

Если в процессе обработки сообщения возникла исключительная ситуация, то выполнение очереди на время приостанавливается. Администратор системы должен выяснить причину сбоя и, если это требуется, произвести корректировку данных вручную. Сбой легко обнаружить по наличию сообщений в статусе «получен (2)» в сообщениях, получателем которых является текущий сайт. В случае отсутствия административного вмешательства, через некоторое время службой обработки сообщений будет предпринята повторная попытка исполнить «ошибочное» сообщение. Если все пройдет без ошибок, то сообщения в очереди продолжат свое последовательное исполнение в штатном режиме. Если нет, то процесс исполнения сбойного документа повторится через некоторое время вновь.

3.8 Планировщик. Задания планировщика. Расписание запуска заданий

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 осуществляется из клиентского приложения посредством пункта меню **Справочники**→**Планировщик**.

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.

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

Задания планировщика хранятся в таблице SCHEDULE, которая обычно заполняется из системного файла `./xml/schedule.xml`.

В расписании перечислены задания, параметры заданий, время последнего запуска,

результаты последнего запуска заданий. Расписание хранится в таблице SCHPLAN. Существует несколько моделей задания времени запуска, к таким относятся:

- **Один раз** – требуется установить дату и время запуска, а также при необходимости повтора задания ежегодно, включить контроль каждый год.
- **Периодически** – требуется указать количество дней, то есть промежуток, через которое задание повторится в назначенное время.
- **Ежедневно** – задание запускается каждый день в назначенное время.
- **Еженедельно** – задание запускается в назначенное время в определенные дни недели.

Использование фоновых процессов влечет повышение нагрузки на сервер приложений АЦК-Планирование. Поэтому все неиспользуемые службы на сервере приложений АЦК-Планирование рекомендуется удалить из расписания планировщика.

Примечание. Описание настройки расписания запуска заданий планировщика приведено в документации «[БАРМ.00004-55 32 01-3 Система «АЦК-Планирование»](#). АРМ ФО. Блок администрирования. Подсистема администрирования. Настройка и сервис системы. Руководство администратора».

3.8.1 Макросы для параметров `begin_date` и `end_date`

В параметрах `begin_date` и `end_date` заданий планировщиков могут использоваться следующие макросы:

- `%РабочаяДата%` – текущая рабочая дата;
- `%НачалоГода%` – начало текущего года;
- `%КонецГода%` – конец текущего года;
- `%НачалоКвартала%` – начало квартала;
- `%КонецКвартала%` – конец квартала;
- `%НачалоМесяца%` – начало текущего месяца;
- `%КонецМесяца%` – конец текущего месяца;
- `%НачалоНедели%` – начало текущей недели;

- %КонецНедели% – конец текущей недели;
- %НачалоПрошГода% – начало прошлого года;
- %КонецПрошГода% – конец прошлого года;
- %НачалоПрошМесяца% – начало прошлого месяца;
- %КонецПрошМесяца% – конец прошлого месяца;
- %НачалоПрошНедели% – начало прошлой недели;
- %КонецПрошНедели% – конец прошлой недели;
- %ПрошРабочаяДата% – прошедшая рабочая дата.

Примечание. Прошедшая рабочая дата не может быть субботой или воскресеньем.

3.8.2 AssignExport

XML для запуска:

```
<ASSIGNEXPORT action="run" budget_id="" site_id=""/>
```

Параметры: budget_id.

Описание: Задание планировщика *AssignExport* предназначено для выгрузки данных по ассигнованиям в стороннюю систему.

Процедура предназначена для выгрузки на веб-сервис сторонней системы xml-сообщения, сформированного согласно установленным параметрам планировщика.

Принимает параметры:

- **budget_id** – ID бюджета, для которого делается выгрузка. Обязательный параметр.
- **site_id** – указывается способ отправки данных: соединение с web-сервисом, выгрузка сформированной xml в указанную в адресе сайта папку. В настройках Сайта указывается тип соединения сайта:
 - «Обмен файлами» (id=2) с обязательным параметром **dst**;
 - «Обмен с веб-сервисом» (id=11) с обязательным параметром **url**;

3.8.3 AutoControlProductPlanNotControlEIS

Задание планировщика работает только для ЭД «Планирование закупок» на статусе «*есть обеспечение*», если скрытый признак **Не отражать изменения на ЕИС** активен.

XML для запуска:

```
<GENERICSTASK action="run"  
class_name="com.bssys.azkserver.expense.AutoControlProductPlanNotControlEIS" />
```

Параметры: fin_year, budget_id.

Описание: Автоматический контроль Планирования закупок, не требующих отражения на ЕИС.

Принимает параметры:

- **budget_id** – ID бюджета-отправителя. Возможно указание нескольких параметров через запятую. Если параметр не задан, задание планировщика выполняется для всех бюджетов системы.
- **fin_year** – финансовый год. Возможно указание нескольких параметров через запятую. Если параметр не задан, выполняется задание планировщика для всех финансовых лет системы, либо только для отдельных бюджетов, если задан параметр **budget_id**. Если параметр **fin_year** задан, но при этом также задано конкретное значение параметра **budget_id**, то параметр **fin_year** при выполнении задания планировщика игнорируется (т.е. выполняется только для конкретных бюджетов, указанных в параметре **budget_id**).

Задание планировщика выполняет автоматическую обработку ЭД «Планирование закупок» со статуса «*есть обеспечение*» до статуса «*контроль пройден*», независимо от состояния бюджетного параметра **Позволять автоматически обрабатывать Планирование закупок до статуса Контроль пройден/Не пройден контроль** (пункт меню **Сервис**→**Бюджетные параметры**, группа настроек **Закупки**, подгруппа настроек **Автоматический контроль документов**→**Планирование закупок**).

При этом ЭД «Уведомление о соответствии контролируемой информации»

автоматически не создается, независимо от состояния бюджетного параметра **Автоматическое формирование ЭД Уведомление о соответствии контролируемой информации** (пункт меню **Сервис**→**Бюджетные параметры**, группа настроек **Закупки**, подгруппа настроек **Автоматический контроль документов**→**Планирование закупок**). А также не осуществляется отправка во внешнюю систему сообщения о смене статуса ЭД.

*Примечание. Если скрытый признак **Не отражать изменения на ЕИС не активный**, обработка ЭД «Планирование закупок» планировщиком не осуществляется.*

3.8.4 Backup DB

XML для запуска:

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

Параметры: cmd, cmdZip, path, name.

Описание: Задание планировщика *Backup DB* предназначено для автоматической подготовки резервной копии базы данных.

Рекомендуемый период запуска: 3 часа.

Процедура предназначена для автоматизированной подготовки резервных копий БД на Firebird для сервера приложений «АЦК-Планирование».

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=/backups; 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_ГГГГММДД_ЧЧММСС.fbk,

где:

- **base** – имя файла, определяется по параметру name;
- **X.X.X.X** – версия билда;
- **ГГГГММДД** – текущая дата;
- **ЧЧММСС** – текущее время.

Пример:

Tomsk_2.55.0.1_20101217_121153.fbk

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

3.8.5 Vbackup

XML для запуска:

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

Параметры: backuppath, backupname, -checkrestore.

Описание: Задание планировщика *Vbackup* предназначено для резервного копирования БД.

Рекомендуемый период запуска: периодически, 3 часа.

Процедура используется для резервного копирования зеркал (используется только для БД, работающих под управлением СУБД Firebird). Используется только при работе с подсистемой *Зеркалирование*.

Принимает следующие параметры:

- **backuppath=<path>** – путь, по которому будут размещаться резервные копии зеркал;
- **backupname=<name>** – имя файла резервной копии зеркала;
- **checkrestore** – параметр для включения выполнения тестового восстановления.

Пример

```
backuppath=./backup; backupname=b  
backuppath=./backup; backupname=b; -checkrestore
```

3.8.6 CADDownloader

XML для запуска:

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

Описание: Задание планировщика *CADDownloader* предназначено для загрузки и актуализации перечня аккредитованных удостоверяющих центров.

Параметр: URL=<http://e-trust.gosuslugi.ru/CA/DownloadTSL?schemaVersion=0>.

3.8.7 CertDownloader

XML для запуска:

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

Параметры: URL=.

Описание: Загрузка и актуализация сертификатов.

Процесс предназначен для автоматической загрузки сертификатов в систему «АЦК-Планирование» и привязки к пользователям.

Принимает параметры:

- **URL=** – путь к папке с сертификатами для импорта в систему «АЦК-Планирование». Указывается сетевой ресурс или локальный путь к папке импорта на сервере.

Примечание. Если сервер приложений работает под управлением ОС Linux, в параметре указывается локальный каталог или сетевой диск.

3.8.8 Consistency check

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *Consistency check* предназначено для проверки целостности системы.

Рекомендуемый период запуска: выкл.

Внутренняя проверка целостности системной информации в БД.

3.8.9 ConsRepLauncher

XML для запуска:

```
<GENERICSCHTASK class_name="com.bssys.azkserver.system.ConsRepLauncher" action="run"/>;
```

Параметры: ProcessClasses.

Описание: Задание планировщика *ConsRepLauncher* предназначено для автоматического формирования отчетов нижестоящих.

Рекомендуемый период запуска: периодически, 30 дней.

Процедура используется для автоматического формирования отчетов нижестоящих из шаблонов отчетов подведомственных. Используется только при работе с подсистемой Консолидированный бюджет.

Принимает параметр:

- **ProcessClasses=<num>** – номер класса документов, из которых будут формироваться отчеты подведомственных.

3.8.10 ControlEventFactValueProcessor

XML для запуска:

```
<CONTROL_EVENT_FACT_VALUE_PROCESSOR action="run"  
class_name="com.bssys.azk.server.processors.ControlEventFactValueProcessor"/>
```

Параметры: нет;

Описание: Задание планировщика *ControlEventFactValueProcessor* предназначено для:

- запроса и заполнения информации о фактическом значении по источнику данных и о факте выполнения ЭД «Контрольное событие»;
- автоматической обработки ЭД «Контрольное событие».

Рекомендуемый период запуска: не используется.

3.8.11 ControlExecMessenger

XML для запуска:

```
<CONTROLEXECMESSAGERTASK action="run"  
class_name="com.bssys.azk.server.planning.controlexec.ControlExecMessenger"/>
```

Параметры: нет;

Описание: Задание планировщика *ControlExecMessenger* предназначено для использования механизма оповещений АРМ Руководителя;

Рекомендуемый период запуска: не используется.

3.8.12 ControlResultsInEISUploader

Задание планировщика *ControlResultsInEISUploader* предназначено для ЭД «Планирование закупок», у которых активирован бюджетный параметр **Автоматическая обработка и отправка результатов контроля для ЭД «Планирование закупок»**.

XML для запуска:

```
<GENERICSCHTASK action="run"  
class_name="com.bssys.azk.server.expense.ControlResultsInEISUploader"/>
```

Параметры: budget_id.

Описание: Автоматическая отправка результатов контроля в ЕИС.

Принимает параметры:

- **budget_id** – ID бюджета, для которого делается выгрузка. Есть возможность множественного выбора, а так же значения **all** (для всех бюджетов).

Задание планировщика выполняет:

- Автоматическую обработку ЭД «Уведомление о соответствии контролируемой информации»/«Протокол несоответствия» со статуса «*черновик*» до статуса «*подготовлен*», если в качестве родительского ЭД «Планирование закупок».
- Отправку в ЕИС ЭД «Уведомление о соответствии контролируемой информации»/«Протокол несоответствия» со статуса «*не принят ЕИС*» (по действию пользователя **Отправить в ЕИС/ГЗ**).
- Запрос статуса для ЭД «Уведомление о соответствии контролируемой информации»/«Протокол несоответствия» находящихся на статусе «*в обработке*» (по действию пользователя **Запрос статуса обработки**).

Если ЭД «Уведомление о соответствии контролируемой информации»/«Протокол несоответствия» имеет в качестве родительского ЭД «Планирование закупок», то выбирается пользователь, указанный в настройке бюджетных параметров для бюджета указанного в ЭД «Планирование закупок».

Если при обработке или отправке ЭД возникли ошибки, такие документы пропускаются и продолжается выполнение задания.

3.8.13 CreateNotifyMsg

XML для запуска:

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

Параметры: нет;

Описание: Задание планировщика *CreateNotifyMsg* предназначено для создания

оповещений при отсутствии смены статуса.

Рекомендуемый период запуска: не используется.

3.8.14 CRLDownloader

XML для запуска:

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

Параметры: -reload.

Описание: Загрузка и установка списка отзыва сертификатов.

При выполнении задания по URL-адресам точек распространения списков отзыва зарегистрированных в системе сертификатов ключей подписи (справочник *Точки распространения списков отзыва*) автоматически скачиваются списки отзыва и устанавливаются в хранилище сертификатов ОС сервера ЭП. Поиск отозванного сертификата осуществляется следующим образом: если по набору значений *Серийный номер сертификата + Серийный номер сертификата издателя* сертификат не найден, то поиск сертификата осуществляется по набору *Серийный номер сертификата + Идентификатор ключа центра сертификатов*.

Принимает параметр:

- **-reload** – обновление списков отзыва при каждом запуске задания, необязательный параметр.

3.8.15 CurdateSynchronizer

XML для запуска:

```
<GENERICSCHTASK action="run" class_name="com.bssys.azkserver.system.CurdateSynchronizer"/>
```

Параметры: нет.

Описание: Задание планировщика *CurdateSynchronizer* предназначено для автоматической смены операционного дня.

Рекомендуемый период запуска: 1 раз в сутки, в 00:00:00.

Процедура автоматически изменяет системную дату **Операционный день** для сервера приложений «АЦК-Планирование».

3.8.16 DataObjectChangeLogCleaner

XML для запуска:

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

Параметры: PERIOD, OBJECTS.

Описание: Задание планировщика *DataObjectChangeLogCleaner* предназначено для очистки журнала изменений объектов системы.

Принимает параметры:

- **PERIOD** – период очистки журнал, по умолчанию 8.
- **OBJECTS** – список объектов для которых производить очистку.

Например, **obj_name1[:days]**, **obj_name2**, где

Obj_name1 – имя объекта, **days** – период.

3.8.17 DBCloneActivate

XML для запуска:

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

Параметры: clone.

Описание: Задание планировщика *DBCloneActivate* предназначено для активирования клона.

Рекомендуемый период запуска: по усмотрению.

Процедура предназначена для активизации зеркал (клонов) по расписанию. Задание будет применено ко всем зеркалам (клонам) из списка, заданного в параметре **clone** и находящимся в состояниях: невалиден, неактивен и ошибка.

Принимает параметр:

- **clone=<someclone(s)>** – список зеркал (клонов).

Пример:

CLONE=clone_name_1,clone_name_2 – задание будет применено к зеркалам (клонам) clone_name_1 и clone_name_2.

3.8.18 DBCloneBackuper

XML для запуска:

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

Параметры: clone, backuppath, backupname, -checkrestore.

Описание: Задание планировщика *DBCloneBackuper* предназначено для запуска резервного копирования на клоне.

Рекомендуемый период запуска: 1 день (при создании клона на основе копии рабочей БД задание можно не использовать).

Процедура предназначена для резервного копирования клонов. Используется только при работе с подсистемой *Зеркалирование*.

Принимает параметры:

- **clone=<name>** – название исходной БД, для которой создается копия. Обязательный для заполнения параметр.
- **backuppath=<path>** – путь к архиву, по которому будет размещаться резервная копия клона БД;
- **backupname=<name>** – префикс в имени резервной копии клона БД;
- **-checkrestore** – параметр для включения выполнения тестового восстановления резервной копии клона БД.

Примеры:

```
clone=clone1; backuppath=./clnbackup; backupname=cln1  
clone=clone1; backuppath=./clnbackup; backupname=cln1; -checkrestore  
clone=clone2
```

3.8.19 DBCloneCheck

XML для запуска:

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

Параметры: CLONE, FULLCHECK, READONLY.

Описание: Задание планировщика *DBCloneCheck* предназначено для проверки соответствия главной базы и зеркала.

Рекомендуемый период запуска: периодически, 1 час.

Процедура предназначена для проверки данных на клоне. Используется только при работе с подсистемой *Зеркалирование*.

Принимает параметры:

- **CLONE=<name>** – названия клонов, при синхронизации которых сравниваются строки таблиц только по полю **VERSION**.
- **FULLCHECK** – работа параметра определяется его значениями:
 - **true** (по умолчанию) – проверка данных на соответствие будет осуществлена по всем полям таблиц.
 - **false** – проверка данных на соответствие будет осуществлена только по полям **ID** и **VERSION**.
- **READONLY** – работа параметра определяется его значениями:
 - **true** (по умолчанию) – используется для определения варианта поведения в случае, если проверка выявит расхождение данных, несовместимое с нормальным функционированием механизма инкрементального обновления. При значении **true** клон не переводится в состояние **ошибка**. В журнал системных событий будет добавлена запись об ошибке с описанием. Значение **true** используется только в целях отладки.
 - **false** – используется для определения варианта поведения в случае, если проверка выявит расхождение данных, несовместимое с нормальным функционированием механизма инкрементального обновления. При значении **false** клон переводится в состояние **ошибка**. В журнал системных событий добавляется запись об ошибке с описанием.

Пример использования:

```
FULLCHECK=false – проверка данных таблиц по полям ID и VERSION;  
CLONE=clone_name_1,clone_name_2 – задание применяется к зеркалам (клонам)  
clone_name_1 и clone_name_2;
```

READONLY=false – в случае неуспешной проверки зеркала (клон) переходит в состояние ошибка.

3.8.20 DBCloneStatistical

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *DBCloneStatistical* предназначено для сбора статистики на всех активных клонах.

Рекомендуемый период запуска: 1 день или после обновления коннектов зеркалирования или продолжительной работы коннектов зеркалирования.

Примечание. Обновление осуществляется для коннектов зеркалирования, которые находятся в статусах «невалиден» (клон включен и нуждается в обновлении) и «неактивен» (клон выключен).

Процедура предназначена для сбора статистики на всех активных коннектах зеркалирования. Используется только при работе с подсистемой *Зеркалирование*. Задание при запуске собирает статистику производительности всех активных зеркал и заносит ее в системные таблицы БД, осуществляет пересбор индексов на активных клонах БД для оптимизации выполнения сложных запросов.

3.8.21 DebitCardLauncher

XML для запуска:

```
<GENERICSCHTASK class_name="com.bssys.azk.server.system.DebitCardLauncher" action="run"/>
```

Параметры: нет.

Описание: Задание планировщика *DebitCardLauncher* предназначено для блокировки дебетовых карт по окончании срока действия.

Рекомендуемый период запуска: не используется.

3.8.22 DeleteNotifyMsg

XML для запуска:

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

Параметры: нет;

Описание: Задание планировщика *DeleteNotifyMsg* предназначено для удаления оповещений.

Рекомендуемый период запуска: не используется.

3.8.23 DocRetentor

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *DocRetentor* предназначено для выполнения проверок над документами.

Рекомендуемый период запуска: периодически, 3 часа.

Процедура предназначена для запуска проверки условий нарушения сроков, в течение которых документы могут находиться в определенном статусе.

3.8.24 DocumentCleaner

XML для запуска:

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

Параметры: path, delete, nosave.

Описание: Задание планировщика *DocumentCleaner* предназначено для архивирования удаленных документов.

Рекомендуемый период запуска: 1 раз в сутки, в нерабочее время.

Процесс, предназначенный для физического удаления из БД и архивирования

документов в статусе «удален» в системе «АЦК-Планирование», даже при наличии ссылки на вложение, отсутствующее в хранилище. Принимает следующие параметры:

- **path=<path>** – путь для файла архива без завершающего «\». Если требуется архивное сохранение, то по этому пути будет создан файл с маской «doc_yyyymm-dd hh-mm-ss.xml», где **yyyymm-dd hh-mm-ss** – текущие дата и время.
- **delete=<true/false>** – удалять или нет документы из базы;
- **nosave=<true/false>** – сохранять или не сохранять документы в архив;
- **format=<true/false>** – форматирование архива;
- **packet_size=n** – commit после n удалений.
- **BUDGET_ID=?** – ID бюджета, в котором обрабатываются удаленные документы. В параметре **BUDGET_ID** можно указать несколько ID бюджетов (через запятую).

Примечание. Имя параметра является регистрозависимым, вводится только прописными буквами.

Пример использования параметров:

```
{path=./arc; delete={true | false}; nosave={true | false}} [; format=true] [; packet_size=n] [; BUDGET_ID=?]
```

3.8.25 EstimateImport

XML для запуска:

```
<ESTIMATE_IMPORT action="import"/>
```

Параметры: source_budget_id, target_budget_id, import_org_role, save_blank, update_estimate.

Описание: Задание планировщика *EstimateImport* предназначено для импорта бланков расходов из бюджета в бюджет.

Процедура предназначена для автоматизации импорта данных справочника *Бланки расходов*.

Принимает параметры:

- **source_budget_id** – ID бюджета, из которого производится импорт.
- **target_budget_id** – ID бюджета, в который производится импорт.
- **import_org_role** – импорт настройки организационных ролей для справочника «Бланки расходов».
- **save_blank** – настройка «Сохранять шаблоны бланков расходов». Если настройка включена, то при импорте бланков расходов будут переноситься и шаблоны бланков расходов.
- **update_estimate** – настройка «Обновлять бланки расходов». Если настройка включена, то при повторном копировании бланков расходов обновлять записи в целевом бюджете, если они были скопированы ранее.

Пример использования:

```
source_budget_id=<id_бюджета_источника>; target_budget_id=<id_бюджета_цели>;  
import_org_role=1; save_blank=1; update_estimate=1
```

3.8.26 Exchange

XML для запуска:

```
<COMMONPARAMSTASK action="run">  
<TASK>  
<RPLTASK action="run"/>  
<TRANSPORTTASK action="run"/>  
<MESSENGERTASK action="run"/>  
</TASK>  
</COMMONPARAMSTASK>
```

Параметры: site_id.

Описание: Задание планировщика *Exchange* предназначено для репликации и обработки сообщений.

Рекомендуемый период запуска: 20 мин.

Процедура включает в себя последовательное выполнение подготовки

репликационных пакетов, доставку сообщений между сайтами, а также обработку новых поступивших сообщений. Рекомендуется использовать процедуру вместо отдельного вызова планировщиком процедур **Replication** и **Messenger**.

Период запуска необходимо устанавливать в зависимости от интенсивности поступления новых сообщений и потребностей объекта в оперативности обработки полученной информации, а так же интенсивности нагрузки на сервер приложений «АЦК-Планирование».

Примечание. Если в задании планировщика `site_id` не указан или указан некорректно, то:

- задание планировщика не выполняется;
- записывается ошибка в результат работы (системное событие).

Формат указания сайта: `site_id= номер сайта`.

3.8.27 FhdPlanStructTemplateImport

XML для запуска:

```
<FHDPLANSTRUCT_TEMPLATE_IMPORT_PROCESSOR action="import"/>
```

Описание: Задание планировщика `FhdPlanStructTemplateImport` предназначено для импорта шаблона ПФХД в нижестоящие бюджеты.

3.8.28 FORepLauncher

XML для запуска:

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

Описание: Задание планировщика `FORepLauncher` предназначено для автоматического формирования отчетов ФО.

3.8.29 ImportBudgServiceFed2

XML для запуска:

```
<IMPORT_BUDGSERVICE_FED2 action="import"/>
```

Описание: планировщик *ImportBudgServiceFed2* предназначен для загрузки данных из Общероссийского перечня услуг, оказываемых физическим лицам.

Задание позволяет устанавливать в параметрах возможность фильтрации по:

- коду вида деятельности:
 - если =34,35 – загрузка происходит только по виду деятельности = 34 и 35;
 - если <>34,35 – загрузка происходит по всем видам деятельности кроме 34 и 35.
- статусу записи:
 - если статус записи в параметрах не указан, то загружаются все записи;
 - если статус записи в параметрах указан в качестве значения = 1, то загружаются только записи с статусом «действующая»;
 - если статус записи в параметрах указан в качестве значения = 2, то загружаются только записи со статусом «архивная».
- признаку **Обновлять существующие записи:**
 - если значение не указано, то принято загрузка происходит с не активным признаком;
 - если значение указано в качестве = 1, то загрузка происходит с активным признаком.
- Указание директории, в которой сохраняются логи. Формат имени файла с логом: dd_mm_yyyy_ppp.log, где:
 - dd_mm_yyyy - день, месяц, год;
 - ppp – порядковый номер. Если за один день (dd_mm_gggg_) задание выгружается несколько раз, то производится порядковая нумерация начиная с 001.

Примечание. В части контроля соответствия вида деятельности коду услуги (работы) при загрузке услуг ОБПУ планировщиком:

- при включенном системном параметре **Отключить контроль соответствия вида деятельности коду услуги (работы)** (Системные параметры → Планирование → Бюджетные услуги (работы) → Перечни услуг(работ) → закладка **Отраслевые перечни услуг (работ)**) сообщение об отключении контроля не выдается и задание в дальнейшем выполняется;
- при отключенном системном параметре **Отключить контроль соответствия вида деятельности коду услуги (работы)** ошибка выполнения задания планировщика отображается.

3.8.30 ImportDocFromEis

XML для запуска:

```
<GENERICSCHTASK action="run" class_name="com.bftcom.server.eisinteraction.ImportDocFromEis"/>
```

Параметры: finYear, budget_id, dir, site.

Описание: Задание планировщика *ImportDocFromEis* предназначено для автоматической загрузки выгруженных файлов из ЕИС .

Примечание. Загрузка xml-файлов также может осуществляться из zip-архива, выгруженного из ЕИС.

Процесс, предназначенный для загрузки в систему «АЦК-Планирование» файлов, выгруженных из ЕИС. Принимает следующие параметры:

- **finYear** – указывается год бюджета, в котором осуществляется загрузка данных. Обязательный параметр.
- **budget_id** – указывается бюджет, в котором осуществляется загрузка документов в систему «АЦК-Планирование». Обязательный параметр.
- **dir** – указывается путь директории, где находятся файлы или zip-архив с файлами из ЕИС. Обязательный параметр.
- **site** – указывается номер сайта. Обязательный параметр.

Пример использования:

finYear=год_бюджета; budget_id=id_бюджета; site=номер сайта; dir=директория для загрузки файлов.

3.8.31 ImportDocLauncher

XML для запуска:

```
<GENERICSCHTASK action="run"  
class_name="com.bssys.azk.server.system.ImportDocLauncher"/>
```

Параметры: OnlyWhenControlsOn, ProcessClasses, WorkDate.

Описание: Задание планировщика *ImportDocLauncher* предназначено для автоматического запуска в обработку импортированных документов.

Рекомендуемый период запуска: 30 мин.

Процесс обрабатывает импортированные в сервер приложений «АЦК-Планирование» документы. Принимает следующие параметры:

- **OnlyWhenControlsOn=<0/1>** – если данный параметр включен (=1), то запуск документов в обработку не будет производиться при отключенных контролях на кассовый план по всем типам бланкам расходов. Если хотя бы по одному типу бланков расходов контроль выключен, запуск не будет произведен, будет выдаваться сообщение о выключенных контролях. Если параметр равен 0, либо не указан – запускается всегда.
- **ProcessClasses=<num>,<num>** – перечисляются классы документов, которые необходимо обработать.
- **WorkDate =**
 1. **CalendarDate** – дата, с которой выполняется задание планировщика, равна дате текущей системной дате.
 2. **OperDate** – дата, с которой выполняется задание планировщика, равна дате операционного дня.
 3. **<ГГГГ-ММ-ДД>** – дата, с которой выполняется задание планировщика, равна заданной дате.

Отбираются все документы перечисленных классов, находящиеся в статусе «импортирован (23)». Каждому документу посылается уведомление на обработку по дереву

сценариев *action=process*. При отсутствии исключительных ситуаций считается, что запуск прошел успешно. Если возникла исключительная ситуация, то текст ошибки помещается в поле REMARK документа (если оно присутствует) и он переводится в статус «*ошибка импорта (42)*». Следует отметить, что указания номера класса в параметре недостаточно, для автоматического запуска. Класс документ помимо этого должен еще иметь специфическую обработку в статусах «*импортирован*» и «*ошибка импорта*». В данный момент в дереве сценариев автоматический запуск поддерживается для следующих классов документов: 3,5,11,24,75,116,125,126.

Пример использования:

```
OnlyWhenControlsOn=1; ProcessClasses=11,48;
```

```
OnlyWhenControlsOn=1; ProcessClasses=3,5,11,24,31,48,65,75,77,116,120,125
```

или, что то же самое:

```
OnlyWhenControlsOn=1; ProcessClasses=3,5,11,24,31,48,65,75,77,116,120,125;
```

```
WorkDate=OperDate;
```

```
OnlyWhenControlsOn=1; ProcessClasses=3,5,11,24,31,48,65,75,77,116,120,125;
```

```
WorkDate=CalendarDate;
```

```
OnlyWhenControlsOn=1; ProcessClasses=3,5,11,24,31,48,65,75,77,116,120,125; WorkDate=2008-12-28.
```

3.8.32 ImportEhdDictionaries

XML для запуска:

```
<IMPORT_EHD_DICT class_name="com.bftcom.server.importer.planning.ImportEhdDictProcessor" action="import"/>
```

Описание: Загрузка и импорт справочников из ЕХД.

Параметры: TABLES, BUDGETS, PPO.

Рекомендуемый период запуска: 1 раз в сутки.

Процесс выполняет загрузку информации по справочникам из ЕХД (Единого хранилища данных) с помощью REST запроса.

Источником данных справочника является ЕХД. Значения параметров по умолчанию URL REST запроса к ЕХД определяются настройкой **Параметры связи с ЕХД**.

(Сервис→Системные параметры→Планирование→Региональные проекты).

Принимает параметры:

- **TABLES** – указывается список таблиц для загрузки, записываются через запятую, предусмотрены следующие параметры запросов:
 - **obsprkvr** – запрашивает информацию по таблице *obsprkvr* (справочник *Классификатор видов расходов*);
 - **obsprkadm** – запрашивает информацию по таблице *obsprkadm* (справочник *Классификатор администраторов поступлений и выбытий*);
 - **obsprkdr** – запрашивает информацию по таблице *obsprkdr* (справочник *Дополнительный код расхода*);
 - **obsprkdf** – запрашивает информацию по таблице *obsprkdf* (справочник *Дополнительный функциональный код*);
 - **obsprkde** – запрашивает информацию по таблице *obsprkde* (справочник *Дополнительный экономический код*);
 - **obsprsp** – запрашивает информацию по таблице *obsprsp* (справочник *Целевые назначения*);
 - **obsprkcsr** – запрашивает информацию по таблице *obsprkcsr* (справочник *Классификатор целевых статей расходов*);
 - **obsprkfsr** – запрашивает информацию по таблице *obsprkfsr* (справочник *Дополнительный экономический код*);
 - **ORGxx** – запрашивает информацию по таблице *ORGxx*, где *xx* - последние две цифры года (справочник *Организации*).
- **BUDGETS** – указывается ID бюджетов для загрузки.
- **PPO** – указывается ID публично-правового образования при загрузке справочника организаций.

Пример использования:

TABLES=obsprkvr,obsprsp;BUDGETS=1401000005499;PPO=0

Для формирования запроса необходимо в адресной строке после адреса страницы ввести «?», а затем один или несколько параметров запроса.

Например: <http://srv-obvp-app.bft.local:8086/obconf/wsna/GetTableContent?TABLES=obsprsp>
где:

URL: <http://srv-obvp-app.bft.local>

Порт: 8086

Путь: /obconf/wsna/GetTableContent

Параметр: TABLES=obsprsp – указывает наименование таблицы ЕХД для запроса. Запрашивает информацию по таблице «obsprsp» (справочник «Целевые назначения»).

Атрибутный состав:

После ответа от сервиса формируется набор с данными:

Таблица 18 – Структура данных для таблиц: obsprkvr, obsprkfsr, obsprkadm, obsprkdr, obsprkdf, obsprkde, obsprsp, obsprkcsr

Наименование поля в API-интерфейсе	Формат	Длина	Обязательность
id	int8	-	да
version	int8	-	да
code	varchar	100	да
date_from	timestamp	-	нет
date_to	timestamp	-	нет
finyear	int4	-	да
lv	int4	-	да
name	varchar	2000	да
parent_code	varchar	100	нет
subjectcode	varchar	100	да
parent_id	int8	-	нет

Таблица 19 – Структура данных таблицы ORGxx:

Наименование поля в API-интерфейсе	Формат	Длина	Обязательность
Текущая дата	timestamp	-	нет
Id Учреждения	numeric	-	да
Сокращенное наименование	varchar	2000	да
Краткое наименование	varchar	2000	да
Полное наименование	varchar	2000	нет
ИНН	varchar	255	нет
КПП	varchar	255	нет
ППО	varchar	255	нет
ОГРН	varchar	2000	нет

Наименование поля в API-интерфейсе	Формат	Длина	Обязательность
Наименование ГРБС(РБС)	varchar	2000	нет
ФИО руководителя	varchar	2000	нет
Должность руководителя	varchar	2000	нет
НаименовОбособлеПодразде	varchar	2000	нет
ОКАТО	varchar	2000	нет
ОКТМО	varchar	2000	нет
ОКФС	varchar	2000	нет
ОКОПФ	varchar	255	нет
ОКПО	varchar	255	нет
Субъект РФ	varchar	2000	нет
Субъект РФ - Код КЛАДР	varchar	2000	нет
Район	varchar	2000	нет
Район - Код КЛАДР	varchar	2000	нет
Город	varchar	2000	нет
Город - Код КЛАДР	varchar	2000	нет
Населенный пункт	varchar	2000	нет
НаселенныйПункт-КодКЛАДР	varchar	2000	нет
Улица	varchar	2000	нет
Улица - Код КЛАДР	varchar	2000	нет
Номер дома	varchar	255	нет
Номер офиса (квартиры)	varchar	255	нет
Почтовый индекс	varchar	2000	нет
Сайт учреждения	varchar	2000	нет
Телефон/факс	varchar	255	нет
Адрес электронной почты	varchar	2000	нет
Тип учреждения	varchar	255	нет
ИД пакета	varchar	20	нет

3.8.33 ImportFinancingIndicatorsEHD

XML для запуска:

```
<IMPORT_FINANCING_INDICATORS_EHD action="import"
class_name="com.bftcom.server.importer.planning.ImportFinancingIndicatorsEHDProcessor"/>
```

Описание: Загрузка и импорт для справочника Финансовые показатели (ЕХД).

Параметры: BUDGET_ID="*<id бюджета>*", table="NORBIT_EXPENSE".

Рекомендуемый период запуска: 1 раз в сутки.

Процесс выполняет загрузку справочника *Финансовые показатели (ЕХД)* информацией по суммам из Единого хранилища данных (далее ЕХД) с помощью REST запроса.

Источником данных справочника является ЕХД. Значения параметров по умолчанию URL REST запроса к ЕХД определяются настройкой **Параметры связи с ЕХД (Сервис→Системные параметры→Планирование→Региональные проекты)**.

Принимает параметры:

- **BUDGET_ID="*<id бюджета>*"** – определяет с каким бюджетом будут загружаться все входящие данные запроса.
- **table="NORBIT_EXPENSE"** – указывает наименование таблицы ЕХД для запроса.

Для формирования запроса необходимо в адресной строке после адреса страницы ввести «?», а затем один или несколько параметров запроса.

Например: [http://srv-obvp-app.bft.local:8086/obconf/wsna/GetTableContent?
table="NORBIT_EXPENSE"](http://srv-obvp-app.bft.local:8086/obconf/wsna/GetTableContent?table=)

где:

URL: <http://srv-obvp-app.bft.local>

Порт: 8086

Путь: /obconf/wsna/GetTableContent

Параметр: table="NORBIT_EXPENSE" — указывает наименование таблицы ЕХД для запроса. Запрашивает информацию по таблице NORBIT_EXPENSE с импортированными данными по расходной части бюджета.

Атрибутный состав:

После ответа от сервиса формируется набор с данными:

Таблица 20 – Описание атрибутивного состава запроса

Наименование поля в API-интерфейсе	Формат	Длина	Описание (от источника данных)	Поле таблицы справочника «Финансовые показатели (ЕХД)»	Логика заполнения
–	–	–	–	Бюджет	Формируется ссылка на справочник <i>Бюджеты</i> по параметру BUDGET_ID
ondate	timestamp	–	На дату	На дату	Заполняется датой в формате ДД.ММ.ГГГГ
budgetRecipientAccount	varchar	100	Код из справочника <i>Организации</i> .	Организация	Производится поиск ссылки в справочнике <i>Организации</i> по полю Код организации .
chiefManagerBudgetFunds	varchar	100	Код из справочника <i>Ведомственный классификатор расходов</i> .	КВСП	Производится поиск ссылки в справочнике <i>Ведомственный классификатор расходов</i> по комбинации полей Бюджет (из настройки)+ Код .
budgetExpendituresSectSubSect	varchar	100	Код из справочника <i>Функциональный классификатор расходов</i> .	КФСР	
targetArticleClassifierCode	varchar	100	Код из справочника <i>Классификатор целевой статьи расходов</i> .	КЦСП	
budgetExpendituresClass	varchar	100	Код из справочника <i>Классификатор вида расходов</i> .	КВП	
budgetExpendituresAddFuncCode	varchar	100	Код из справочника <i>Дополнительный функциональный код</i> .	Доп.ФК	
budgetExpendituresAddEcoCode	varchar	100	Код из справочника <i>Дополнительный экономический код</i> .	Доп.ЭК	
budgetExpendituresAddCode	varchar	100	Код из справочника <i>Дополнительный код расхода</i> .	Доп.КР	

Наименование поля в API-интерфейсе	Формат	Длина	Описание (от источника данных)	Поле таблицы справочника «Финансовые показатели (ЕХД)»	Логика заполнения
amountAssignCur	numeric	–	Ассигнования (текущий финансовый год)	Ассигнования (текущий финансовый год)	Заполняется значением одноименного поля, если атрибут = <пусто>, то заполняется значением 0.
amountAssignFirst	numeric	–	Ассигнования (первый год планового периода)	Ассигнования (первый год планового периода)	
amountAssignSecond	numeric	–	Ассигнования (второй год планового периода)	Ассигнования (второй год планового периода)	
amountLimit	numeric	–	Лимиты (текущий финансовый год)	Лимиты (текущий финансовый год)	
amountFund	numeric	–	Финансирование	Финансирование	
amountCashExec	numeric	–	Кассовое исполнение	Кассовое исполнение	

Обновление записей справочника *Финансовые показатели (ЕХД)*:

- Проверяются строки на уникальность по полям: **На дату, КВСР, Организация, КЦСР, КФСР, КВР, Доп, ФК, Доп, ЭК, Доп.КР**, если хотя бы одно из полей отличается от имеющейся записи, то формируется новая запись. Если найдена запись с аналогичными параметрами, то поля сумм в ней обновляются.
- Если значение суммовых полей в теле ответа равно <пусто> или <null>, то в значении суммы записывается 0.
- Если поля КБК не указаны, то строка КБК заполняется нулями.
- Если соответствующая запись КБК не найдена в справочниках КБК, т.е. создать ссылку невозможно, то она пропускается, и информация о пропущенных записях записывается в лог.

3.8.34 ImportRegProjCard

XML для запуска:

```
<IMPORT_REG_PROJ_CARD action="import"/>
```

Параметры: url, rproj_code, fproj_code, org_caption, date_begin, date_end, update_exsisted_records, debug, log_path.

Описание: Задание планировщика *ImportDocLauncher* предназначено для загрузки данных в ЭД «Карточка регионального проекта».

Принимает следующие параметры:

- **url** – адрес для доступа к данным ГИИС ЭБ (обязательный параметр).
- **rproj_code** – код регионального проекта (необязательный параметр):
 - если = 44,45, то загрузка происходит только по региональным проектам с кодами 44 и 45;
 - если <>A3,D5, то загрузка происходит по всем региональным проектам кроме 44 и 45.
- **fproj_code** – код федерального проекта (необязательный параметр):
 - если = A3,D5, то загрузка происходит только по федеральным проектам *Цифровая культура* и *Цифровые технологии*;
 - если <>A3,D5, то загрузка происходит по всем федеральным проектам кроме *Цифровая культура* и *Цифровые технологии*.
- **org_caption** – наименование организации ответственного исполнителя
- **date_begin** – дата начала (необязательный параметр).
- **date_end** – дата окончания (необязательный параметр).
- **update_exsisted_records = 0/1** – обновлять существующие записи (необязательный). Если значение не указано, то признак **Обновить существующие записи** при загрузке данных в ЭД «Карточка регионального проекта» выключен. Если указано значение 1, то признак **Обновить существующие записи** при загрузке данных в ЭД «Карточка регионального проекта» включен.
- **debug = 0/1** – Если указано значение 1, в лог выводится подробная информация о процессе загрузки данных в ЭД «Карточка регионального проекта» (необязательный)
- **log_path** – указание директории, в которой сохраняются логи. Формат имени файла с логом: dd_mm_yyyu_ppp.log, где:

- dd_mm_yyyy - день, месяц, год;
- rrr – порядковый номер. Если за один день (dd_mm_gggg_) задание выгружается несколько раз, то производится порядковая нумерация начиная с 001.
- **dispstatus_id** – если параметр не заполнен, то обновление выполняется только по ЭД на статусе «новый». Если параметр заполнен, то обновление ЭД производится на любом его статусе. Доступна возможность множественного указания статусов ЭД.

3.8.35 ImportTaskProjects

XML для запуска:

```
<IMPORT_TASK_PROJECTS action="import"/>
```

Параметры: url, filtercode.

Описание: Загрузка целей и задач рег. проектов

Процесс загружает данные в справочник *Цели и задачи региональных проектов*.

Принимает следующие параметры:

- **URL** – указывается путь для сервиса API. Возможность выбора значения по умолчанию: <http://budget.gov.ru/epbs/registry/7710168360-NATIONALPROJECT/data?blocks=tasks>
- **filtercode** – включается дополнительный фильтр для запроса:
filtercode=<значение>

Общие требования к загрузке:

- Создаваемые записи формируются с:
 - Тип = *Цель национального проекта/Задача национального проекта*
 - Ведомственная принадлежность – запись с ID=0 (НЕ УКАЗАНО)
 - Публично-правовое образование – запись с ID=0. Если отсутствует, то запись из бюджета с уровнем бюджета 02 и максимальным значением финансового года.
 - Вышестоящая цель или задача – null.

- При первичном создании записи через задание заполняется системное поле для хранения recordid с API.
- Обновление выполняется по скрытому системному полю recordid_нас.
- При обновлении проверяются значения полей name и realizationdate в системе с данными по API. Если значения не отличаются, то запись не обновляется (поле VERSION не увеличивается).

3.8.36 ImportUnitProcessor

XML для запуска:

```
<IMPORT_UNIT action="import" class_name="com.bftcom.server.importer.planning.ImportUnitProcessor"/>
```

Описание: Загрузка и импорт справочника «Единицы измерения».

Процесс выполняет загрузку справочника *Единицы измерения* информацией из API-интерфейса с открытых данных единого портала бюджетной системы.

Источником данных единиц измерения является сервис на сайте Электронного бюджета: <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data>.

Для формирования запроса необходимо в адресной строке после адреса страницы ввести «?», а затем один или несколько параметров запроса.

Например: <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?pageSize=10>.

Для API предусмотрены следующие параметры запросов:

Параметры:

- Параметр **pageSize** – задает число отображаемых элементов на одной странице. Минимум на одной странице может быть отображен 1 элемент, максимум – 1000. По умолчанию выводится 10 элементов. Чтобы изменить количество элементов на одной странице нужно ввести, например: <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?pageSize=100>, количество отображаемых элементов на одной странице станет 100.
- Параметр **pageNum** – задает номер отображаемой страницы. Если в запросе указать <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?pageNum=2> в

ответе отобразится 2 страница. При совместном использовании параметров **pageNum** и **offset** приоритетным будет параметр **offset**, так как позволяет более точно задать позицию.

- Параметр **offset** – задает номер элемента, с которого нужно начать вывод данных. Если в запросе указать <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?offset=100>, вывод данных начнется со 101-го элемента. При совместном использовании параметров **pageNum** и **offset** приоритетным будет параметр **offset**, так как позволяет более точно задать позицию.

Если в одном запросе требуется объединить несколько параметров, то используется символ: «&».

Пример: <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?pageSize=1000&offset=100>.
Количество отображаемых элементов на одной странице будет 1000, вывод данных начнется с 101-го элемента.

- Параметр **sortField** отвечает за сортировку данных по определенному атрибуту. По умолчанию сортировка осуществляется по атрибуту **Code**. Сортировка возможна по всем атрибутам. Например: <http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?sortField=code> – сортировка по атрибуту **Code**. Для изменения порядка сортировки используется параметр **sortDir**:
<http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?sortField=code&sortDir=desc> – сортировка по убыванию по атрибуту **Code**,
<http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?sortField=code&sortDir=asc> – сортировка по возрастанию по атрибуту **Code**.

В параметрах планировщика предусмотрен обязательный для заполнения параметр **NAME_IN_LOAD**, при установлении которого в поле **NAME** записи справочника *Единицы измерения* подтягивается значение поля **localsymbol** (условное обозначение ЕИ) или **localname** (буквенное обозначение ЕИ). Параметр **NAME_IN_LOAD** может принимать значение *localsymbol* или *localname*. (**NAME_IN_LOAD**=*localsymbol*).

Фильтры:

Для сервиса предусмотрен параметр **filter**. При помощи параметра осуществляется фильтрация по:

Таблица 21 – Описание параметра filter

№	Наименование параметра	Формат атрибута	Пример
1	Код (атрибут Code)	Длина атрибута равна 3 или 4 знакам	<p>Например, если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filtercode=020, то в ответ получим только те записи, у которой код равен «020».</p> <p>Для того чтобы вывести в отчет те записи, у которых в коде вторая цифра равна 2 нужно вместо оставшихся цифр ввести символ "_". Запрос будет выглядеть следующим образом: http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filtercode=_2. Данный вид сортировки данных возможен и в других фильтрах, содержащих цифровые значения.</p>
2	Наименование (атрибут Name)	-	<p>Например, если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filtername=метр, то в ответ получим только те записи, у которых в наименовании содержится «метр».</p>
3	Раздел (Section)	<p>Атрибут может принимать значения</p> <ul style="list-style-type: none"> «1» – Международные единицы измерения, включенные в ЕСКК. «2» – Национальные единицы измерения, включенные в ЕСКК. 	<p>Например, если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filtersection=1, то в ответ получим только те записи, у которой значение в поле Раздел равно «1».</p>

№	Наименование параметра	Формат атрибута	Пример
4	Группа (Subsection)	Атрибут может принимать значения <ul style="list-style-type: none"> • «1» – Единицы длины. • «2» – Единицы площади. • «3» – Единицы объема. • «4» – Единицы массы. • «5» – Технические единицы. • «6» – Единицы времени • «7» – Экономические единицы. 	Например, если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filterSubsection=1 , то в ответ получим только те запись, у которой значение в поле Группа равно «1».
5	Статус записи (атрибут Status)	Атрибут может принимать значения: <ul style="list-style-type: none"> • ACTIVE – актуальная запись; • ARCHIVE – архивная запись. 	Например, если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filterstatus=ACTIVE , то в ответ получим только актуальные записи
6	Дата загрузки (выгрузки) на ЕПБС (атрибут loaddate, filedate)	Дата: ДД.ММ.ГГГГ	Запрос позволит вывести записи, загруженные (выгруженные) с определенной даты, по определенную дату или за период дат. Например, если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filterminloaddate=27.01.2016 , то результатом запроса будут записи, загруженные (обновленные) на ЕПБС с 27.01.2016. Если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filtermaxloaddate=05.02.2016 , то результатом запроса будут записи ЕПБС, загруженные (обновленные) на ЕПБС по 05.02.2016. Если указать в запросе http://budget.gov.ru/epbs/registry/7710568760-OKEI/data?filterminloaddate=27.01.2016&filtermaxloaddate=05.02.2016 , то результатом запроса будут записи ЕПБС, загруженные (обновленные) на ЕПБС в диапазоне указанных дат.

Атрибутный состав:

После ответа от сервиса формируется набор с данными:

Таблица 22 – Описание атрибутивного состава запроса

Наименование поля в API-интерфейсе	Формат	Длина	Обязательность	Примечание
guid	guid	=36	да	Глобально-уникальный идентификатор записи
startdate	datetime2		да	Дата начала действия записи
enddate	datetime2		нет	Дата окончания действия записи
status	string	<=7	да	Статус записи
code	string	=3 или 4	да	Код
section	string	=1	да	Раздел
subsection	string	=1	да	Группа
name	string	<=250	да	Наименование
localsymbol	string	<=50	да	Условное обозначение (национальное)
internatsymbol	string	<=50	нет	Условное обозначение (международное)
localname	string	<=50	да	Буквенный код (национальный)
internatname	string	<=50	нет	Буквенный код (международный)
comments	string	<=250	нет	Комментарии
clarification	string	<=8000	нет	Пояснение
lastchangenumb	string	<=3	да	Номер последнего изменения
lastchangetype	string	=1	да	Тип последнего изменения
changedate	date2		да	Дата принятия изменения
introductiondate	date2		да	Дата введения изменения
loaddate	datetime2		да	Дата загрузки на ЕПБС
filedate	datetime2		да	Дата файла, дата выгрузки из сводного реестра

Данные загружаются в справочник *Единицы измерения* по следующему алгоритму:

Проверяется соответствие поля **code** полю **OKEI**:

- Если да, то проверяется условие:

В записи справочника поле **UNIT_GROUP_ID** значение равно 0 (группа «НЕ

УКАЗАНО»):

- Если да, то проверяется условие:

соответствие полей **Наименование**, **Актуальность**, **Описание**:

- если да, справочник *Единицы измерения* не обновляется в соответствии с записью классификатора ОКЕИ.
- если нет, запись справочника *Единицы измерения* обновляется в соответствии с записью классификатора ОКЕИ:
 - при изменении основного наименования единицы измерения:
 - если в списке дополнительных наименований отсутствует запись, совпадающая с новым наименованием основной единицы измерения, то осуществляется изменение старого наименования основной единицы на новое наименование в списке дополнительных наименований;
 - если в списке дополнительных наименований присутствует запись, совпадающая с новым наименованием основной единицы измерения, изменения не осуществляются.
 - при изменении актуальности основной единицы измерения:
 - если актуальность основной записи меняется с ARCHIVE на ACTIVE, то признак актуальности активируется для записи с таким же наименованием в списке дополнительных наименований;
 - если актуальность основной записи меняется с ACTIVE на ARCHIVE, то признак актуальности снимается у всех записей в списке дополнительных наименований.
- Если нет, запись в справочник *Единицы измерения* изменяется в соответствии с записью классификатора ОКЕИ; осуществляется заполнение поля **Наименование** значениями наименования единицы измерения, признак актуальности выставляется в зависимости от признака **IS_ACTUAL** (актуален, если **status= ACTIVE**, и не актуален, если **status= ARCHIVE**).
- Если нет, запись в справочник *Единицы измерения* изменяется в соответствии с записью классификатора ОКЕИ.

Таблица 23 – Описание полей запроса «ImportUnitProcessor»

Наименование поля	Атрибут	Значение для загрузки
Наименование	NAME	В зависимости от выставления параметра NAME_IN_LOAD в задании планировщика: <ul style="list-style-type: none">• если NAME_IN_LOAD=localsymbol, то localsymbol,• если NAME_IN_LOAD=localname, то localname
Описание	CAPTION	name
Код ОКЕИ	OKEI	code
Актуальность	IS_ACTUAL	= 1, если status = ACTIVE = 0, если status = ARCHIVE
Группа	UNIT_GROUP	"НЕ УКАЗАНО"
Список дополнительных наименований		
Наименование	NAME	=NAME
Актуальность	IS_ACTUAL	=IS_ACTUAL

3.8.37 KExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *KExport* предназначено для экспорта справочника «Классификатор вида источников».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки справочника *Классификатор вида источников* для портала «Открытый бюджет» в формате xml-файлов – SPR_W_SOURCE.xml.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.38 MailCleaner

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *MailCleaner* предназначено для удаления заархивированных сообщений.

Рекомендуемый период запуска: 1 раз в сутки, в нерабочее время.

Процедура обеспечивает удаление архивных почтовых сообщений.

3.8.39 MessageControlEvent

XML для запуска:

```
<MESSAGECONTROLEVENT class_name="com.bssys.azk server.processors.MessageControlEventProcessor" action="run"/>
```

Параметры: type, action.

Описание: Задание планировщика *MessageControlEvent* предназначено для рассылки уведомлений об истечении срока и смене статуса ЭД «Контрольное событие».

Принимает следующие параметры:

- **type** – параметр используется для отправки сообщений пользователям об истечении срока ЭД «Контрольное событие». Возможные значения:
 - 1 – пользователю отправляется сообщение на электронную почту.
 - 0 – пользователю отправляется системное почтовое сообщение (Сервис→Почтовые сообщения).
 - Если параметр не выбран, то отправляется и системное почтовое сообщение, и уведомление на электронную почту.
- **action** – параметр используется для отправки сообщений пользователям об

истечении срока ЭД «Контрольное событие». Возможные значения:

- 0 – пользователю не отправляется сообщение об изменении статуса ЭД «Контрольное событие».
- 1 (или параметр не выбран) – пользователю осуществляется отправка сообщения об изменении статуса ЭД «Контрольное событие».

3.8.40 Messenger

XML для запуска:

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

Параметры: ProcessClasses, WorkDate.

Описание: Задание планировщика *Messenger* предназначено для обработки очереди сообщений.

Рекомендуемый период запуска: 15 мин.

Процесс обрабатывает импортированные в сервер приложений «АЦК-Планирование» документы.

Имеет предустановленные параметры:

- **ProcessClasses=<10>,<31>,<116>,<184>** – перечисляются классы документов, для которых соответствующие проводки формируются датой сервера БД.
- **WorkDate=CalendarDate** – дата бюджетных проводок для документов из параметра **ProcessClasses** равна текущей системной дате.

Примечание. Предустановленные параметры нельзя изменить.

Процедура обеспечивает исполнение новых сообщений на сервере приложений. Сообщения могут быть сгенерированы самим сервером приложений (для внутреннего использования). Период запуска необходимо устанавливать в зависимости от интенсивности поступления новых сообщений и потребностей объекта в оперативности обработки полученной информации.

3.8.41 MsgCleaner

XML для запуска:

```
<MSGCLEANERTASK action="run"/>;
```

Параметры: path, format, daysAgo, delete, nosave.

Описание: Задание планировщика *MsgCleaner* предназначено для удаления сообщений в статусе «*обработан*» с возможностью выгрузки в файл.

Рекомендуемый период запуска: 1 раз в сутки, в нерабочее время.

Процесс, предназначенный для физического удаления из БД обработанных процедурой «Messenger» информационных сообщений в статусе «*обработан*».

Принимает следующие параметры:

- **path=<path>** – путь для выгрузки сообщений.
- **format=<true/false>** – форматировать выгруженный XML/не форматировать.
- **daysAgo=<num>** – для скольких дней оставлять сообщения.
- **delete=<true/false>** – удалять после выгрузки/не удалять.
- **nosave=<true/false>** – не выгружать/выгружать.

Пример использования параметров:

```
path=./completedMessegas; format=true; daysAgo=3; delete=true; nosave=false
```

Удаление сообщений (только в статусе «*обработан*») из БД с сохранением в файл (форматированный XML) с именем *msg_yyy-mm-dd hh-mm-ss.xml*, где **yyy-mm-dd hh-mm-ss.xml** – текущие дата и время, в каталог *completedMessegas*, находящийся в текущем каталоге (относительно сервера приложений), в базе данных выполняется сохранение сообщений для последних 3-х дней после чего они удаляются.

3.8.42 MsgRepeater

XML для запуска:

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

Параметры: CHECK_FAILED, ProcessClasses, WorkDate.

Описание: Задание планировщика *MsgRepeater* предназначено для повторной обработки сообщений, обработанных с ошибкой.

Рекомендуемый период запуска: 1 час.

Имеет предустановленные параметры:

- **CHECK_FAILED** – если параметр указан, дополнительно просматриваются и обрабатываются сообщения в статусе «получен», если параметр не указан, просматриваются и обрабатываются только сообщения в статусе «новый».
- **ProcessClasses=<10>,<31>,<116>,<184>** – классы документов, для которых соответствующие проводки формируются датой сервера БД.
- **WorkDate=CalendarDate** – дата бюджетных проводок для документов из параметра **ProcessClasses** равна текущей системной дате.

Примечание. Предустановленные параметры нельзя изменить.

Процедура обеспечивает исполнение сообщений на сервере приложений в статусе «новый», а также повторную обработку сообщений в статусе «получен». Сообщения могут быть сгенерированы самим сервером приложений (для внутреннего использования).

3.8.43 NoteToChangePasswTask

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *NoteToChangePasswTask* предназначено для рассылки уведомлений об истечении срока действия пароля.

Рекомендуемый период запуска: 1 раз в сутки.

Процедура обеспечивает выполнение рассылки об истечении срока действия паролей пользователей.

3.8.44 NotifyMsgSender

XML для запуска:

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

Параметры: нет;

Описание: Задание планировщика *NotifyMsgSender* предназначено для отправки оповещений по электронной почте.

3.8.45 NotificationSenderTask

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *NotificationSenderTask* предназначено для рассылки оповещений.

Рекомендуемый период запуска: по усмотрению.

3.8.46 OkpdImport

XML для запуска:

```
<OKPD_IMPORT_PROCESSOR action="import"/>
```

Параметры: log_path, output_path, ftp_url, login, password.

Описание: Импорт ОКПД из ftp.

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для загрузки справочника ОКПД в формате xml-файлов.

Принимает следующие параметры:

- **log_path** – указывается путь для файлов лога;
- **output_path** – указывается путь для обработанных файлов;

- **ftp_url** – указывается адрес ftp с обязательным указанием порта, значение по умолчанию *ftp://ftp.zakupki.gov.ru:21/fcs_nsi/nsiOKPD2*;
- **login** – указывается логин от ресурса ftp, значение по умолчанию *free*;
- **password** – указывается пароль от ресурса ftp, значение по умолчанию *free*.

3.8.47 OkvedImport

XML для запуска:

```
<OKVED_IMPORT_PROCESSOR action="import"/>
```

Описание: Импорт ОКВЭД из ftp.

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для:

- актуализации справочника с частотой, устанавливаемой администратором системы;
- установки параметров, для определения бюджета(ов) для загрузки в систему. Определение бюджета(ов) для загрузки происходит по полю *budget_id*, возможно указание нескольких бюджетов через запятую для автоматической загрузки в несколько бюджетов.
- указания директории для логирования ошибок, возникающих при загрузке.
- указания адреса ресурса (FTP), с которого производится загрузка справочника. По умолчанию *ftp://ftp.zakupki.gov.ru/fcs_nsi/nsiOKVED2*. Если адрес указан не верно, выводится информативная ошибка в задании планировщика, загрузка не осуществляется.
- указания логина и пароля от ресурса FTP. По умолчанию значение и логина, и пароля = *free*. Если не верно указан логин или пароль, выводится информативная ошибка в задании планировщика, загрузка не осуществляется.

3.8.48 OosExport

XML для запуска:

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

Описание: Задание планировщика *OosExport* предназначено для выгрузки данных на веб сервис ЕИС.

3.8.49 PlanAssExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PlanAssExport* предназначено для экспорта расходной части бюджета.

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД расходной части бюджета для портала «Открытый бюджет» в формате xml-файлов – *PLAN_ASS.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Список видов ЭД расходной части бюджета, по которым будет осуществляться выгрузка данных для портала «Открытый бюджет» задается системным параметром **Открытый бюджет** (пункт меню **Сервис**→**Системные параметры**, группа настроек **Планирование**).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.50 PlanInExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PlanInExport* предназначено для экспорта доходной части бюджета.

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД «Прогноз по доходам» и «Планирование изменений прогноза по доходам» для портала «Открытый бюджет» в формате xml-файлов – *PLAN_IN.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.51 PlanInRefImport

XML для запуска:

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

Описание: Задание планировщика *PlanInRefImport* предназначено для загрузки доходной части бюджета из системы АЦК-Финансы для специальных отчетов.

Примечание. В части загрузки сумм «КП-Доходы всего» за месяц предусмотрена возможность загрузки в справочник «Отчетные данные и иные сведения для прогноза доходов» без признака **нарастающая сумма**. Блок xml-задания:

```
<column_name>value_inc_cash_all_m_<номер загружаемого месяца></column_name>  
<comments>КП – доходы всего <наименование загружаемого месяца></comments>
```

Примечание. Загрузка данных с типом «Остаток зачисления» происходит без признака **нарастающая сумма**.

Для избежания повторной загрузки предусмотрена возможность переноса загруженного файла по указанному пути.

Примечание. При настройке задания планировщика необходимо указать параметр **output** – путь к папке, куда будут перемещены обработанные файлы.

3.8.52 PlanSrcExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PlanSrcExport* предназначено для экспорта данных по источникам.

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк «План по источникам» и «Изменения плана по источникам» для портала «Открытый бюджет» в формате xml-файлов – *PLAN_SRC.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.53 PlanXMLEXPOR

XML для запуска:

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

Описание: Задание планировщика *PlanXMLEXPOR* предназначено для выгрузки из системы АЦК-Планирование.

3.8.54 PIBORExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PIBorExport* предназначено для экспорта справочника «Цели и задачи».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк справочника *Цели и задачи* для портала «Открытый бюджет» в формате xml-файлов – *SPR_PL_BOR.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.55 PIStateEventDocExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PIStateEventDocExport* предназначено для экспорта ЭД «ВЦП/АЦП/Основное мероприятие».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД «ВЦП/АЦП/Основное мероприятие» для портала «Открытый бюджет» в формате xml-файлов – *PIStateEventDocExport.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.56 PISStateProgDocExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PISStateProgDocExport* предназначено для экспорта ЭД «Государственная (муниципальная) программа».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД «Государственная (муниципальная) программа» для портала «Открытый бюджет» в формате xml-файлов – *PISStateProgDocExport.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.57 PISStateSubProgDocExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PISStateSubProgDocExport* предназначено для экспорта ЭД «Подпрограмма».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД «Подпрограмма» для портала «Открытый бюджет» в формате xml-файлов – *PISStateSubProgDocExport.xml*.

Принимает следующие параметры:

- **budget_id** – Id-идентификатор бюджета, за который требуется выгрузить данные.

- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.58 ProgDocExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *ProgDocExport* предназначено для экспорта ЭД «Программа».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД «Программа» для портала «Открытый бюджет» в формате xml-файлов – *PL_PROGDOC.xml*.

Принимает следующие параметры:

- **budget_id** – id-идентификатор бюджета, за который требуется выгрузить данные.
- **directory** – директория выгрузки xml-файла в файловой системе (на сервере).

Пример указания параметров:

```
budget_id=1000000102;directory=c:\temp
```

3.8.59 PurposefulGrantCleanupTask

XML для запуска:

```
<GENERICSCHTASK class_name="com.bssys.azk server.purposefulgrant.PurposefulGrantCleanupTask" action="run"/>
```

Описание: Задание планировщика *PurposefulGrantCleanupTask* предназначено для очистки резервных проводок по коду цели.

3.8.60 RegionalProjectCardEhdExportProcessor

XML для запуска:

```
<GENERICSCHTASK action="run"  
class_name="com.bssys.azk.server.planning.project.export.regprojcard.ehd.RegionalProjectCardEhd  
ExportProcessor"/>
```

Описание: ЭД Карточка рег. проекта: экспорт в ЕХД.

Параметры: SITE_ID, PROJ_CODE, EHD_PURCHASES, EHD_AMOUNTS.

Рекомендуемый период запуска: 1 раз в сутки.

Процесс выполняет выгрузку информации о плановых и фактических бюджетных показателях в структурированный xml-файл для последующей его передачи на сервис ЕХД (Единое хранилище данных).

Источником данных xml-файла является ЭД «Карточка регионального проекта», суммы исполнения по источникам финансовых средств (запрос информации из ЕХД), сведения о закупках (запрос информации из ЕХД).

При выполнении задания формируется запрос в ЕХД для сбора информации по закупкам и информации об исполнении бюджета. Также запрашивается актуальная информация по уточненным суммам бюджета из «АЦК-Планирование». После получения ответа от ЕХД информация структурируется в xml-файл и отражается в репликационном сообщении. Xml-файл передается в ЕХД. Со стороны ЕХД происходит обработка и вставка данных в таблицы, к которым в дальнейшем можно обращаться из внешней системы.

Принимает параметры:

- **SITE_ID** – номер сайта получателя.
- **PROJ_CODE** – код регионального проекта.
- **EHD_AMOUNTS=<0/1>** – если параметр включен (=1), загружаются остатки из ЕХД.
- **EHD_PURCHASES=<0/1>** – если параметр включен (=1), загружаются данные по закупкам из ЕХД.

Пример использования:

```
SITE_ID=666;PROJ_CODE=A1-73;EHD_PURCHASES=0;EHD_AMOUNTS=0 – выгрузка
```

регионального проекта с кодом А1-73 без загрузки из ЕХД.

Если при выполнении задания планировщика сервис ЕХД недоступен для запроса информации, то возникает ошибка вида: «Сервис "Хранилище данных" недоступен», информация об ошибке фиксируется в лог-файл. Доступ к сервису ЕХД определяется настройкой **Параметры** **связи** **с** **ЕХД** (**Сервис**→**Системные параметры**→**Планирование**→**Региональные проекты**).

3.8.61 RelreportFailedImportedDocs

XML для запуска:

```
<GENERICSCHTASK action="run"  
class_name="com.bftcom.server.importer.RelreportFailedImportedDocsTask"/>
```

Параметры: budget_id, class.

Описание: Повторный импорт документов в статусе «ошибка импорта».

Принимает следующие параметры:

- **budget_id** – ID Бюджета, обязательный параметр.
- **class** – класс документа, обязательный параметр.

Процедура находит в указанном бюджете для указанного класса документы в статусе «ошибка импорта» и выполняет над такими документами действие **Импортировать повторно**.

3.8.62 Replication

XML для запуска:

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

Параметры: SITE.

Описание: Задание планировщика *Replication* предназначено для подготовки репликационных пакетов.

Рекомендуемый период запуска: 20 мин.

Процедура обеспечивает подготовку репликационных пакетов для обмена информацией между сайтами АЦК-Планирование. Исходными данными для формирования репликаций являются объекты репликации. В объектах репликации указывается: таблицы БД, информацию которых необходимо синхронизировать, способ подготовки этой информации (процессор), а также дополнительные условия для отбора информации из таблиц. Период запуска необходимо устанавливать в зависимости от загрузки сервера приложений. При интенсивной работе его нужно уменьшать. Чрезмерное уменьшение этого параметра вызывает повышение нагрузки на сервер приложения; объем сообщений при этом уменьшается.

Принимает необязательный параметр:

- **SITE=<number site>** – номер сайта, для которого подготавливаются репликационные пакеты. В задании допустимо указание только одного номера сайта. Таким образом, для каждого сайта может быть создано отдельное задание и определен различный период запуска.

Например:

Пусть номер сайта =200, тогда для задания репликации только для этого сайта нужно определить параметр:

SITE=200

3.8.63 ReportingCompanyRPProcessor

XML для запуска:

```
<REPORTING_COMPANY_RP_PROCESSOR action="run"  
class_name="com.bssys.azk.server.processors.ReportingCompanyRPProcessor"/>
```

Описание: Задание планировщика *ReportingCompanyRPProcessor* предназначено для автоматического создания отчетной кампании по утвержденным ЭД «Карточка регионального проекта».

При выполнении задания планировщика создается запись в справочнике *Отчетная кампания* с типом *Исполнение региональных проектов*. Поля записи заполняются по следующему принципу:

- **Дата** – поле заполняется автоматически датой, равной дате выполнения задания.

планировщика.

- **Примечание** – поле заполняется значением «Исполнение региональных проектов по состоянию на «ДД.ММ.ГГГ создания отчетной кампании».
- **Тип** – поле заполняется значением «Исполнение региональных проектов».

Примечание. Если в момент выполнения задания планировщика на дату в справочнике уже присутствует отчетная кампания, то возникает ошибка выполнения задания планировщика.

В момент выполнения задания планировщика осуществляется:

1. Автоматическое создание записи справочника *Отчетные кампании*.
2. Автоматическое создание ЭД «Исполнение регионального проекта» и ЭД «Исполнение события регионального проекта» по всем ЭД «Карточка регионального проекта» на статусах «*утверждено*» и «*изменение*», у которых дата утверждения меньше или равна дате создаваемой отчетной кампании.

3.8.64 RplGenerationStatusTask

XML для запуска:

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

Описание: Статистика значений в RplLog.

3.8.65 RtRplRunner

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *RtRplRunner* предназначено для запуска realtime-репликации.

Рекомендуемый период запуска: 1 минута или определяется актуальностью

данных, хранящихся в клоне (чем чаще запускается процедура, тем чаще осуществляется синхронизация клона и основной БД).

Процедура предназначена для запуска репликации между зеркалом и основной БД. Используется только при работе с подсистемой *Зеркалирование*.

3.8.66 SchExpDocs

XML для запуска:

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

Параметры: -matchtext; -export_code, -max_documents

Описание: Задание планировщика *SchExpDocs* предназначено для выгрузки документов с ЭП в электронный архив

Рекомендуемый период запуска: зависит от необходимости выгрузки документов с ЭП.

Процедура предназначена для автоматической выгрузки документов с ЭП. Может принимать необязательные параметры:

- **-matchtext** – выгрузка только тех документов, у которых текст в БД соответствует тексту подписываемых данных. Если параметр не указан, то выгружаются все документы, определенные параметром **export_code**.
- **-export_code** – код варианта выгрузки из справочника *Документы с ЭП, выгружаемые по расписанию*. Если параметр не указан, то выгружаются все документы из справочника *Документы с ЭП, выгружаемые по расписанию*.
- **-max_documents** - максимальное количество выгружаемых документов.

Например:

```
max_documents=1000
```

При запуске задания с таким параметром система выгружает не более 1000 документов.

При совместном использовании параметров **max_documents** и **except_exported** возможна выгрузка частями большого количества документов, которое по техническим причинам не может быть выгружено за один раз. Например:

```
[except_exported]; max_documents=количество;] [export_code=код1,код2,..кодN]
```


Выгрузка документов с ЭП выполняется в директорию, указанную в системном параметре **Директория для выгрузки ЭД с ЭП** (пункт меню **Сервис**→**Системные параметры**→**ЭП**, закладка **Общие**).

*Примечание. Подробное описание настройки системного параметра **Директория для выгрузки ЭД с ЭП** см. в документации «[БАРМ.00004-55 32 01-3 Система «АЦК-Планирование»](#). АРМ ФО. Блок администрирования. Подсистема администрирования. Настройка и сервис системы. Руководство администратора».*

3.8.67 Session cleaner

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *Session cleaner* предназначено для очистки неактивных сессий.

Рекомендуемый период запуска: 30 мин.

Процедура закрывает сессии пользователей, по которым длительное время не было активности.

3.8.68 SQLTracer

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *SQLTracer* предназначено для дампа результатов трассировки SQL.

Рекомендуемый период запуска: однократно, ручной запуск процедуры.

Процедура осуществляет сбор статистики по использованию SQL запросов и записывает ее в память сервера приложения. Информация необходима для последующего

анализа департаментом разработки, служит для отладочных целей.

Задание желательно использовать в крайних случаях для отладочных целей, т.к. при его активации система значительно «притормаживает» и файлы с логом разрастаются.

3.8.69 SSLTrustManager

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *SSLTrustManager* предназначено для обновления списка активных SSL сертификатов.

Рекомендуемый период запуска: 1 час.

Период запуска можно увеличить в зависимости от частоты изменения сертификатов. Если сертификаты не изменяются – службу можно отключить вообще.

3.8.70 Statistics

XML для запуска:

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

Параметры: -stat; -sweep.

Описание: Задание планировщика *Statistics* предназначено для сбора статистики.

Рекомендуемый период запуска: 1 раз в сутки, в нерабочее время.

Процедура предназначена для сбора статистики БД и удаления информации из журнала удаленных записей БД.

- **-stat** – запуск сбора статистики БД;
- **-sweep** – запуск процедуры удаления информации из журнала удаленных записей БД.

Если задание принимает оба параметра, то при его запуске одновременно осуществляется сбор статистики и удаление информации из журнала удаленных записей БД.

3.8.71 STORAGE_COMPRESSOR

XML для запуска:

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

Параметры: нет.

Описание: Задание планировщика *STORAGE_COMPRESSOR* предназначено для архивирования данных в устройствах хранения.

Рекомендуемый период запуска: 1 раз в сутки, в нерабочее время.

Процедура предназначена для сжатия файлов, выгруженных с помощью устройств хранения.

3.8.72 SyseventCleaner

XML для запуска:

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

Параметры: MSGTYPE.

Описание: Задание планировщика *SyseventCleaner* предназначено для очистки журнала системных событий.

При запуске строки расписания с заданием выполняется очистка записей в журнале **Системные события**.

Если в настройке строки расписания указан параметр **MSGTYPE**, то производится очистка записей журнала только с указанным одним или несколькими типами событий:

- **MSGTYPE=0** – уведомление;
- **MSGTYPE=1** – предупреждение;
- **MSGTYPE=2** – ошибка.

3.8.73 SysInfoShower

XML для запуска:

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

Параметры: Нет.

Описание: Задание планировщика SysInfoShower предназначено для получения системной информации.

Рекомендуемый период запуска: 1 час.

Процедура выводит в лог сервера приложений сообщение о среде выполнения JVM, а так же другую справочную информацию о состоянии сервера приложений.

Для поиска ошибок и на период опытной эксплуатации период запуска можно уменьшить.

3.8.74 TransactMonitorFlusher

XML для запуска:

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

Параметры: -flush; -sweep; -clear.

Описание: Задание планировщика *TransactMonitorFlusher* предназначено для выгрузки данных о трафике в БД.

Рекомендуемый период запуска: 1 раз в неделю.

Процедура выгружает данные о трафике в базе данных в соответствии с установленными параметрами:

- **-flush** – запись данных о трафике в БД;
- **-sweep** – удаление данных о трафике из БД.
- **-clear** – удаление данных о трафике из памяти СП.

Пример использования параметров:

```
<ROW ID="1" EXAMPLE="-flush" COMMENT="Запись данных о трафике в БД"/>
```

```
<ROW ID="2" EXAMPLE="-sweep" COMMENT="Удаление данных о трафике из БД"/>
```

```
<ROW ID="3" EXAMPLE="-clear" COMMENT="Удаление данных о трафике из памяти СП"/>
```

3.8.75 Transport

XML для запуска:

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

Параметры: site_id, period_from, period_end.

Примечание. Параметры **period_from** и **period_end** используются при репликационном взаимодействии с внешней системой управления закупками.

Описание: Задание планировщика *Transport* предназначено для обмена сообщениями между серверами приложений АЦК-Планирование.

Рекомендуемый период запуска: 15 мин, периодически – для репликационного взаимодействия с внешней системой управления закупками.

Процедура обеспечивает обмен сообщениями между серверами приложений АЦК-Планирование (сайтами). Способ и параметры соединения необходимо указать для каждого сайта отдельно в справочнике сайтов. Период запуска службы следует выбирать в зависимости от потребностей в скорости доставки сообщений получателю.

Принимает параметры:

- **site_id** – запускает передачу сообщений между текущим и вторым сайтом АЦК.

Примечание. Если в задании планировщика **site_id** не указан или указан некорректно, то:

- задание планировщика не выполняется;
- записывается ошибка в результат работы (системное событие).

Формат указания сайта: **site_id= номер сайта.**

- **timeout** – время в секундах. По истечении указанного времени, если нет ответа от внешней системы о статусе выполнения задания, задание планировщика завершает свою работу с выводом информации об этом в поле **Результат последнего выполнения.**
- **period_from** – время начала выполнения задания планировщика по отправке сообщений во внешнюю систему управления закупками в статусе «отправлен», с которого будет повторяться отправка сообщений, пока сообщения не будут доставлены. Формат задания параметра **ЧЧ:ММ:СС.**
- **period_end** – время окончания выполнения задания планировщика по отправке

сообщений во внешнюю систему управления закупками в статусе «отправлен» до которого будет повторяться отправка сообщений, пока сообщения не будут доставлены. Формат задания параметра ЧЧ:ММ:СС.

Примечание. Если время запуска не попадает в период, заданный в параметрах, задание планировщика завершится без отправки сообщений.

Пример указания параметров в расписании планировщика:

```
SITE_ID=0;PERIOD_FROM=00:00:00;PERIOD_TO=05:00:00
```

3.8.76 UniAnalyticalReportDeleteCompleted

XML для запуска:

```
<UNIANALYTICALTASKMANAGER action="clear"
class_name="com.bssys.azk server.reports.kernel.uni.processing.analyticalReport.UniAnalyticalTask
ManagerProcessor"/>
```

Описание: планировщик *UniAnalyticalReportDeleteCompleted* предназначен для удаления выполненных заданий на построение аналитического отчета.

3.8.77 UniAnalyticalReportUpdate

XML для запуска:

```
<UNIANALYTICALTASKMANAGER action="run"
class_name="com.bssys.azk server.reports.kernel.uni.processing.analyticalReport.UniAnalyticalTask
ManagerProcessor"/>
```

Описание: Задание планировщика *UniAnalyticalReportUpdate* предназначено для добавления заданий на построение аналитического отчета.

3.8.78 UniAnalyticalTaskProcessor

XML для запуска:

```
<UNIANALYTICALTASK action="run"
```

```
class_name="com.bssys.azk server.reports.kernel.uni.processing.analyticalReport.UniAnalyticalTaskProcessor"/>
```

Описание: Задание планировщика *UniAnalyticalTaskProcessor* предназначено для построения аналитического отчета.

3.8.79 UniContractProcessor

XML для запуска:

```
<UNICONTRACT_EXPORT action="run" class_name="com.bssys.azk server.reports.common.impexp.exportData.xml.openBudget.processor.UniContractProcessor"/>
```

Параметры: finYear, budget, budget_id, useChildBudget, dest, dir, user, password, wsdlLoc, begin_date, end_date, status, with_pay_docs, dop_kd.

Описание: Задание планировщика *UniContractProcessor* предназначено для выгрузки данных из отчета *Универсальный отчет по договорам*.

Принимает параметры:

- **finYear** – год бюджета, для которого делается выгрузка. Обязательный параметр.
- **budget** – наименование бюджета, для которого требуется выгрузить данные. Не обязательный параметр..
- **budget_id** – ID бюджета, для которого делается выгрузка. Обязательный параметр, если не указаны параметры **finYear** и **budget**. Если параметр **budget_id** заполнен, параметры **finYear** и **budget** не обязательны для заполнения. Формируются отдельные файлы для выгрузок из разных бюджетов.
- **useChildBudget** – необязательный параметр, принимаемые значения: 1 – выгрузка осуществляется для бюджета, определенного по параметру **budget_id** или **finYear** и **budget**, и всем его нижестоящим бюджетам; 0 (или параметр не выбран) – выгрузка осуществляется только для бюджета, определенного по параметру **budget_id** или **finYear** и **budget**; 2 – выгрузка осуществляется поочередно по всем бюджетам.

Например,

задание

"dest=3;site=99;queue=VP;budget_id=1201000001639;begin_date=01.12.2018;end_date=31.12.2019;useChildBudget=*2*;divide_period=60" равносильно последовательному запуску 218 заданий планировщика, где в качестве **budget_id** поочередно подставляются id нижестоящих бюджетов.

Формируются отдельные файлы для выгрузок из разных бюджетов.

- **dest** – назначение выгрузки файла. Возможные значения: 1 – передать файл веб-сервису, 2 – сохранить файл в файловой системе, 3 – передать сообщением. Обязательный параметр.
- **dir** – директория выгрузки в файловой системе. Обязательный параметр, если dest=2.
- **user** – логин для входа на веб-сервис.
- **password** – пароль для входа на веб-сервис.
- **wsdlLoc** – адрес веб-сервиса, на который производится выгрузка.
- **begin_date** – дата начала выгружаемых данных. Необязательный параметр. По умолчанию начало года.
- **end_date** – дата окончания выгружаемых данных. Обязательный параметр. По умолчанию текущая дата.

Примечание. В параметрах **begin_date** и **end_date** могут использоваться макросы. Список возможных макросов приведен в разделе [Макросы для параметров begin_date и end_date](#)⁷⁵.

- **status** – статус документа. Может принимать значения: 6, 10, 28.
- **with_pay_docs** – параметр позволяет выгрузить данные по документам исполнения. Возможные значения:
 - 0 (или параметр не выбран) – выгрузка осуществляется без документов, подтверждающих исполнение договора;
 - 1 – выгружается информация по документам, подтверждающим исполнение договора.
- **dop_kd** - необязательный параметр, позволяет настраивать выгружаемые записи по признаку ненулевого значения одного из дополнительных кодов классификации.

Возможные значения: *kdf* – в выгрузку попадают только те строки, в которых значение поля **Доп.ФК** отлично от значения *Не указано*; *kdr* – в выгрузку попадают только те строки, в которых значение поля **Доп.КР** отлично от значения *Не указано*; *kde* - в выгрузку попадают только те строки, в которых значение поля **Доп.ЭК** отлично от значения *Не указано*.

3.8.80 UpdateRegionalListRegNum

XML для запуска:

```
<REGIONALLIST_REGNUM_PROCESSOR action="import"/>
```

Параметры: log_path, name_subject, reg_numbe

Описание: планировщик *UpdateRegionalListRegNum* предназначен для загрузки и обновления реестровых номеров из регионального перечня ЭБ.

Принимает параметры:

- **log_path** – указание директории, в которой сохраняются логи. Формат имени файла с логом: dd_mm_yyyy_ppp.log, где:

dd_mm_yyyy - день, месяц, год;

ppp – порядковый номер. Если за один день (dd_mm_gggg_) задание выгружается несколько раз, то производится порядковая нумерация начиная с 001.

- **name_subject** – указывается значение для фильтрации данных.
- **reg_numbe** – уникальный номер РЗ ЭБ.

3.8.81 UploadOKSFromEIS

- **Параметры:** site, dir, url.

Описание: Задание планировщика выполняется на сервере приложений, используемого для взаимодействия с ЕИС.

Принимает параметры:

- **site** - номер сайта ЕИС. Обязательный параметр.

- **dir** - каталог для сохранения архивов, скаченных с ЕИС. Обязательный параметр.
- **url** - путь к ftp. Обязательный параметр.

Задание планировщика выполняет:

- с **ftp**, указанного в параметре **url** под пользователем **free** и паролем **free** по пути **fcs_nsi/nsiKOKS** скачивается архив и сохраняется в папке, указанной в параметре **dir**.
- на основании **xml**, содержащейся в архиве создаются записи в справочнике **Объекты капитального строительства**, заполняя необходимые поля.

Примечание. Задание планировщика доступно в списке справочника только при пролитии дополнительной **xml**.

3.8.82 Validation

XML для запуска:

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

Параметры: -ignorechecksum.

Описание: Задание планировщика *Validation* предназначено для проверки БД.

Рекомендуемый период запуска: 1 раз в сутки, в нерабочее время.

Процедура предназначена для проверки целостности БД.

Если задание принимает параметр **-ignorechecksum**, то при проверке БД игнорируется ошибка контрольных сумм.

3.9 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.

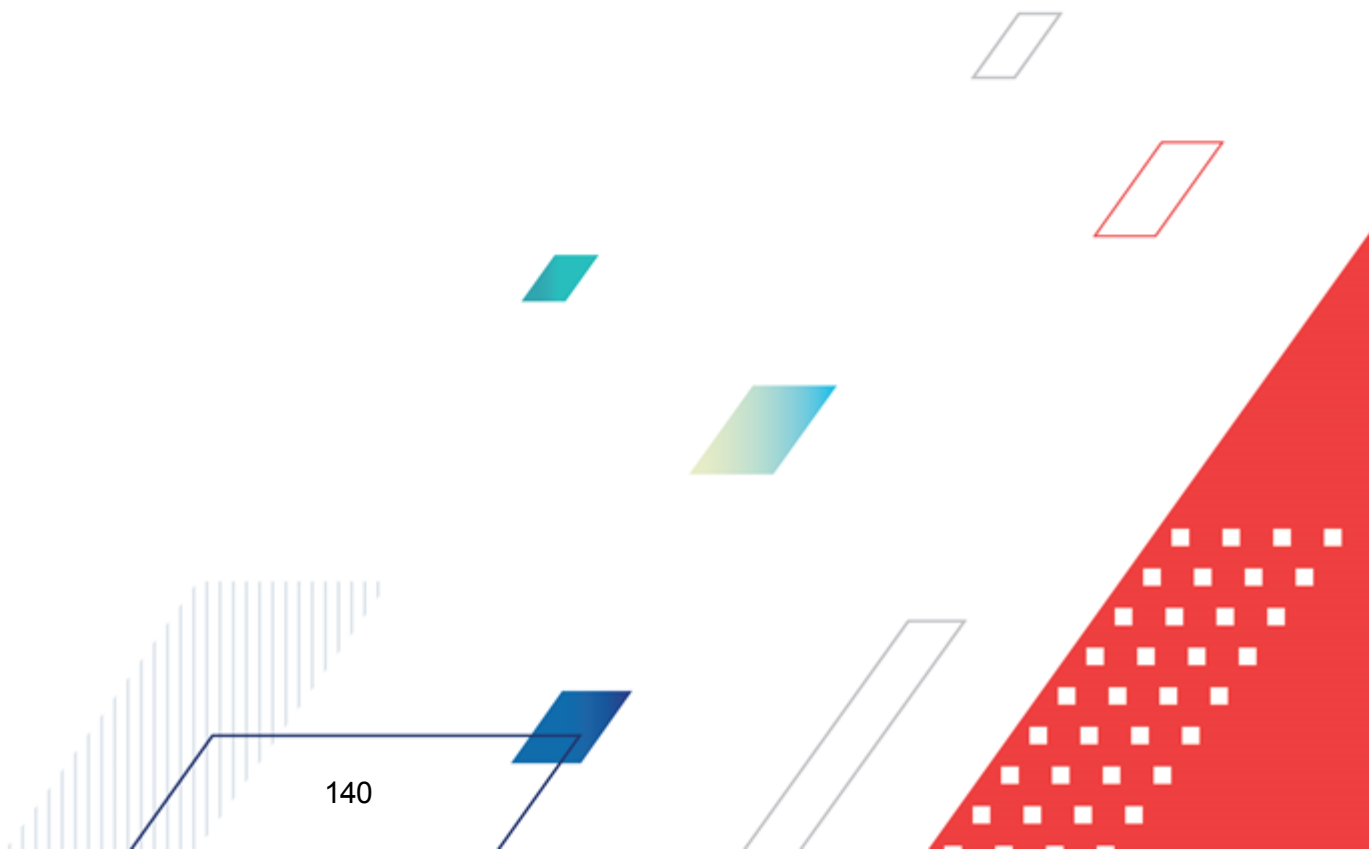
3.10 Xml для методики создания бюджета 2022 года

В сборке системы «АЦК-Планирование», начиная с версии 2.53.0, содержится папка XML\newyear\2022\ с перечнем xml, скорректированных для пролития по бюджетам 2022 года:

- good_to_fin.xml
- good_to_pl.xml
- LN_fin_to_pl.xml
- ln_ref_edit_pl.xml
- pl_repl.xml
- pl_rpl_source_to_fin.xml
- pl_rpl_source_to_pl.xml
- ref_edit_disable_pl.xml
- ref_edit_disable_source.xml
- ref_edit_disable_triu.xml
- ref_edit_enable_pl.xml
- ref_edit_enable_source.xml
- ref_edit_enable_triu.xml
- ref_edit_pl.xml
- rpl_plan.xml
- pl_rpl_orgcount_to_pl.xml
- ref_edit_orgcount_pl.xml



База данных



База данных – это совокупность данных, организованная с определенной целью. Здесь под *организованной совокупностью данных* понимается, что указанная совокупность включает данные, которые сохраняются и имеют определенный формат, к ним может быть обеспечен доступ, и они могут быть представлены потребителю информации в приемлемом виде.

В случае использования базы данных в целях «АЦК-Планирование», можно сказать, что используется база данных уровня предприятия, потому что в ней охватывается деятельность предприятия в целом.

Программный комплекс, обеспечивающий функционирование базы данных, называют *системой управления базой данных (СУБД)*. Она играет роль хранилища данных, отвечает за их сохранность, безопасность, целостность, взаимное соответствие и обеспечивает доступ к данным.

4.1 Основные принципы построения БД

В настоящее время широкое использование получила реляционная модель построения БД. Она предусматривает организацию данных исключительно в виде таблиц. Таблица представляет собой набор полей и множество записей. Пересечение поля и строки определяет ячейку. Поле характеризуется типом данных; запись представляет собой данные.

Над наборами данных можно выполнять различные действия. Результатом выполнения запросов к БД тоже являются таблицы. Единственным стандартизированным средством, которое позволяет выполнять операции, является язык SQL. Язык SQL одновременно является и языком определения данных и языком управления данными.

Особое внимание следует уделить понятиям:

- **Первичный ключ** – столбец или подмножество столбцов, которое уникально, т.е. единственным способом определяет строки в этой таблице;
- **Внешний ключ** – столбец или подмножество столбцов одной таблицы, который может служить в качестве первичного ключа для другой таблицы. Говорят также, что внешний ключ одной таблицы является «ссылкой» на первичный ключ другой таблицы.

В состав БД кроме *таблиц* и *ссылок* могут входить и другие типы объектов. К таким типам относятся: *триггер*, *ограничение*, *функция*, *процедура* и др. Наличие тех или иных типов объектов зависит от конкретной реализации СУБД. Более полную информацию об использовании конкретной СУБД можно получить из соответствующей литературы, которая входит в комплект поставки сервера или из популярной технической литературы.

4.2 Основные принципы работы с БД

С принципами создания и обслуживания БД под управлением конкретной СУБД можно ознакомиться в соответствующих технических описаниях.

В связи с тем, что «АЦК-Планирование» разрабатывается с учетом хранения данных на различных типах СУБД, SQL-скрипты, используемые для создания объектов БД, написаны с использованием специального внутреннего формата. Все SQL-выражения оптимизированы для использования на конкретной СУБД, поэтому работают максимально быстро и имеют эффективные алгоритмы и структуры данных.

В тех случаях, когда выполняются действия по изменению структуры БД, строго рекомендуется оставить одно соединение с БД, а остальные закрыть. Это относится к неиспользуемым в текущий момент клиентским приложениям, которые работают с БД напрямую посредством API, или подключенным к БД серверам приложений «АЦК-Планирование».

Не рекомендуется исправлять информацию в БД, потому что это может катастрофически повлиять на вид представляемой информации и ход последующих вычислений. Контроль целостности БД не позволяет отслеживать нюансов в хранении информации, а основные алгоритмы обработки информации расположены на сервере приложений. Возникшая ошибочная ситуация может стать критической, и восстановление информации может занять много ресурсов или быть вообще невозможной. В случае возникновения критической ошибки рекомендуется обращаться в службу поддержки, где можно получить консультацию и разъяснения последовательности дальнейших действий.

Для манипулирования с данными в БД необходимо использовать клиентское приложение, которое, используя функциональные возможности сервера приложений, будет корректно выполнять все действия над информацией.

К простому клиентскому приложению можно отнести утилиту XMLExecuter. Его функциональности достаточно для выполнения простых XML-выражений на сервере приложений в командном или интерактивном режиме.

4.2.1 SQLExecuter

В состав серверной части входит SQLExecuter. **SQLExecuter** – это специализированная утилита, которая предназначена для исполнения SQL-скриптов на СУБД, тип которой поддерживается в процессе разработки.

4.2.1.1 Назначение SQLExecuter. Параметры запуска. Файл конфигурации

Наличие SQLExecuter обусловлено наличием некоторых, иногда концептуальных, различий между возможностями различных СУБД. Производитель программных продуктов, стараясь придать СУБД лучшие технические характеристики, вносили в них свои особенности, которые разработчикам приходится учитывать в повседневной деятельности. SQLExecuter предназначен для максимального сглаживания вышеуказанных различий в процессе разработки и эксплуатации БД на различных типах СУБД.

Экземпляр SQLExecuter представляет собой реализацию java-класса **com.bssys.tools.sqlxecuter.Executer**. Для создания экземпляра класса и передачи ему параметров для исполнения применяется командная строка. Это позволяет, используя особенности командного процессора операционной системы, создавать командные файлы по обработке структуры БД различной сложности.

Параметры SQLExecuter содержатся в файле свойств сервера приложений. Путь к файлу свойств сервера приложений указывается в параметре SQLExecuter **SERVER_CONFIG**.

Значение по умолчанию:

```
SERVER_CONFIG=..\Azk2Server.properties
```

4.2.1.2 SQL-скрипт. Назначение, типы, нумерация, структура, синтаксис

Для формирования объектов в БД необходимы метаданные, т.е. данные об информации, которую предполагается хранить в БД. Для хранения метаданных используются SQL-скрипты, в которых используется функциональность языка SQL по определению структуры БД.

SQL-скрипты представлены в виде файлов, которые расположены в каталоге `./sql/` и имеют следующий формат наименования:

```
[_]<number>_<author>_<caption>.sql
```

где:

_ (знак подчеркивания) – признак скрипта, который используется для отмены действий SQL-скрипта со сходным наименованием только без `_` в начале имени;

<number> – порядковый номер SQL-скрипта; для создания корректной структуры БД необходимо выполнять скрипты последовательно по возрастанию номеров;

<author> – условное обозначение разработчика, создавшего скрипт;

<caption> – наименование файла, кратко описывающий его назначение.

По своему назначению SQL-скрипты можно разделить на 2 типа:

- 1) скрипт, формирующий структуру БД; могут использоваться как для создания объектов БД так и для их уничтожения;
- 2) сервисный скрипт, используемый для исправления ошибочных состояний БД; используется в основном в аварийных ситуациях или в тех случаях, когда корректировка информации в БД может использовать значительные ресурсы.

Для исполнения SQL-скриптов используется функциональность `SQLExecuter`. В связи с тем, что SQL-скрипты используются для выполнения в СУБД различных типов, метаданные, которые используются в них, имеют специальный формат. Для упрощения наименования часто повторяемых функций, имеющих в различных типах СУБД одинаковое или сходное назначение, введен обобщенный синтаксис.

- 1) SQL-выражение должно начинаться со следующей строки после признака команды `<-->` (двойной знак минус):

```
--
```


insert into

- 2) Чтобы выполнить SQL-выражение без анализа и преобразования (только для конкретного типа БД) необходимо после признака команды указать через символ # (решетка) наименование типа БД. В настоящее время поддерживаются следующие типы: ORACLE, INTERBASE и MSSQL:

--#INTERBASE

insert into

- 3) При составлении SQL-скриптов рекомендуется использовать строки комментария. Это упростит задачу разработки и сопровождения БД. Для задания комментария используются следующие обозначения:

- // - определение однострочного комментария; действует до признака конца строки;
- — — за признаком начала команды после знака пробела можно размещать комментарий, который действует до признака конца строки;
- /*, */ - позволяет выделить многострочный комментарий.

Примеры использования комментариев в теле SQL-скрипта:

- Комментарий

// Комментарий

*create table XXX (
KCSR_CODE NUMERIC (15, 0),*

KVR_CODE NUMERIC (15, 0),

*/*Комментарий*

Внимание! Необходимо учитывать, что комментарии внутри SQL-выражений не удаляются.

**/*

KCSR_CODE NUMERIC (15, 0),

KVR_CODE NUMERIC (15, 0),

);

Обобщенный синтаксис для некоторых выражений.

1. **SYSDATE** – является значением текущего дня и времени;
2. Создание и удаление генератора (аналогичен ORACLE):

create sequence sequence_name maxvalue 999999999;

3. Использование генератора:

- **GenCurr(gen_name)** – взять текущее значение генератора;
 - **GenNext(gen_name)** – взять значение генератора после инкремента;
4. Удаление колонки из таблицы (аналогичен ORACLE или MSSQL):

Внимание! По одной колонке на SQL-выражение.

```
alter table table_name drop column column_name
```

5. **GetSiteOffset** – функция для возвращения префикса сайта для генерации ID:

```
GetSiteOffset+GenNext(gen_name)
```

6. Создание и удаление индекса (аналогичен ORACLE и FIREBIRD):

```
drop index idx_some_index
```

7. Изменение свойств *nullable* и *default*:

```
ALTER TABLE table_name MODIFY COLUMN column_name column_type <nullable_value>  
[DEFAULT default_value];
```

```
<nullable_value> ::= NOT NULL | NULL
```

Чтобы **удалить** default, нужно написать **null** в качестве **default_value**.

Чтобы **поменять** default, нужно сначала удалить его, а потом создать новый.

Тип DATE: FB – DATE, ORACLE – DATE, MSSQL – DATETIME;

Тип TIME: FB – TIME, ORACLE – DATE, MSSQL – SMALLDATETIME;

Тип TIMESTAMP: FB – TIMESTAMP, ORACLE – DATE, MSSQL – DATETIME;

Тип CLOB: FB – BLOB, ORACLE – CLOB, MSSQL – TEXT;

Тип NLOB: FB – BLOB, ORACLE – NLOB, MSSQL – NTEXT;

Тип BLOB: FB – BLOB, ORACLE – BLOB, MSSQL – IMAGE;

Команда разрыва и повторного соединения с СУБД; рекомендуется использовать только с СУБД Firebird:

```
reconnect
```

4.2.1.3 Утилиты запуска sql-скриптов

Для запуска утилит, которые исполняют SQL-скрипты на БД, используются следующие командные файлы:

Таблица 24 – Командные файлы, предназначенные для запуска утилит, исполняющих SQL-скрипты на БД

Название командного файла	Назначение
sql.cmd	Выполнение одного или нескольких SQL-скриптов на БД. Количество SQL-скриптов, указываемых в командной строке, ограничивается особенностями операционной системы (10 шт. для ОС Windows). Формат для вызова: <i>sql.cmd <sql_script> [<sql_script>] [...]</i>
DBUpdate.cmd	Обновление БД до версии системы.
comparer.cmd	Сравнение эталонной и целевой БД.
perform.cmd	Создание в новой БД всех объектов инфраструктуры по метаданным из SQL-скриптов. Для выполнения обязательным является указания номера сайта для вновь создаваемой БД. Формат использования: <i>perform.cmd <site_number></i>

Вышеперечисленные файлы используют параметр, который указывает на путь к SQLExecuter.

4.2.2 XMLExecuter

В состав серверной части «АЦК-Планирование» входит XMLExecuter. **XMLExecuter** – это специализированная утилита, которая предназначена для выполнения XML-команд на сервере приложений.

4.2.2.1 Назначение, параметры запуска

XMLExecuter представляет собой простейшее консольное клиентское приложение. Его основное назначение – исполнение XML-скриптов на сервере приложений «АЦК-Планирование» в пакетном или интерактивном режиме.

Экземпляр XMLExecuter представляет собой реализацию java-класса **com.bssys.tools.xml.Executer**. Для создания экземпляра этого класса и передачи ему параметров для исполнения используется командная строка, что позволяет, используя особенности командного процессора операционной системы, создавать различные по сложности командные файлы по наполнению данными БД.

Для XMLExecuter определены следующие параметры командной строки:

- **-config "<path/properties_file>"** – определяет файл конфигурации;
- **-curdate "<work_date>"** – устанавливает рабочую дату;
- **-budget_id "<budget_identity>"** – устанавливает идентификатор бюджета;
- **-user "<user_name>"** – определяет имя пользователя, под правами которого выполняется соединение;
- **-password "<user_password>"** – задает пароль пользователя для соединения;
- **-host "<host_name>"** – определяет имя компьютера, на котором расположен сервер приложений «АЦК-Планирование»;
- **-port "<port_number>"** – определяет номер порта, используемый сервером приложений для работы с клиентскими приложениями.

Все параметры, кроме **-config**, являются необязательными и используются для построения командных файлов.

Файл конфигурации XMLExecuter содержит основные параметры, которые указываются в виде пары **<parameter>=<value>**. Каждый параметр должен начинаться с начала новой строки. В этом файле можно использовать строки комментария, которые позволяют размещать любую информацию, в том числе описание используемых параметров. Строка комментария начинается с символа «#» и заканчивается в текущей строке. Рассмотрим параметры, которые могут быть использованы XMLExecuter:

- Рабочая дата (необязательный, по умолчанию текущий день):

azk.curdate=

- Бюджет (необязательный, по умолчанию «0»):

azk.budget_id=

- Пользователь (обязательный):

azk.user=root

- Пароль (обязательный):

azk.password=toor

- Расположение сервера приложений (обязательный):

`azk.host=localhost`

- Порт сервера приложений (обязательный):

`azk.port=2001`

- Файл для логов (необязательный):

`azk.log.filename=xml.log`

- Кодировка символов, используемая для вывода лога в консоль:

`azk.log.encodeScreen=Cp866`

- Кодировка символов, используемая для вывода лога в log-файл:

`azk.log.encode=Cp1251`

- Параметр перезаписи log-файла при запуске XMLExecuter (0 – лог очищается при запуске; 1 – в лог добавляются новые записи при запуске):

`azk.log.append=0`

- Уровень журнализации лога при выводе в log-файл:

`azk.log.mode=0`

- Уровень журнализации лога при выводе в консоль:

`azk.log.screenmode=0`

Внимание! Если наименование файла лога не указано, лог формироваться НЕ БУДЕТ.

4.2.2.2 XML-скрипт. Структура, назначение

XML-скрипт по своей структуре является документом, выполненным в формате XML. Структура документа соответствует стандарту, принятому на формат документа. Полное описание стандарта можно получить из технической литературы.

Основными понятиями, которые встречаются при построении XML-документа, являются *тег* и *атрибут*. Первый из них описывает абстрактный объект, второй – свойства объекта. В примере представлено описание объекта (BANK), которое имеет свойство (ID) равное 1:

```
<BANK ID="1" />
```

По своему назначению, в контексте использования на сервере приложений «АЦК-

Планирование», XML-скрипт играет роль команды представленной в формате XML-документа.

4.2.2.3 XML-скрипты системы. Описание, зависимости запуска

Для штатного функционирования сервера приложений, при котором он использует все функции необходимо заполнить информацией все системные справочники в БД.

Вся необходимая для системных справочников информация находится в XML-скриптах системы. Они находятся в каталоге `./xml/`.

Чтобы при сохранении информации в БД не возникали ошибки и не появлялись сообщения об исключительных ситуациях, используется заранее определенная последовательность применения скриптов. При необходимости просмотра последовательности можно обратиться к содержимому файла-списка `@azk2_sys.lst`. Исполнение файла-списка сервером приложений гарантирует полную функциональность системы и готовность «АЦК-Планирование» к работе. Рассмотрим назначения входящих в его перечень файлов:

Таблица 25 – XML-скрипты, входящие в файл-список `@azk2_sys.lst`

Название XML-скрипта	Назначение
reference.xml	Перечень справочников системы.
serverprocessor.xml	Перечень серверных процессоров.
serverprovider.xml	Перечень серверных провайдеров.
clientobject.xml	Перечень клиентских объектов.
func_res.xml	Общий функционал системы.
dbconstraint.xml	Перечень интерпретации в формат, пригодный для восприятия конечным пользователем возможных исключительных ситуаций, которые могут быть выведены на экран в процессе работы с БД.
servererror.xml	Перечень ошибок, информация о которых может быть выведена сервером приложений.
msglinktype.xml	Перечень типов соединения серверов приложения.
numgenerator.xml	Перечень генераторов номеров.
metaobject.xml	Перечень метаданных объектов данных.
menu.xml	Перечень меню.
@docreaction.lst	Список XML-скриптов с указанием информации о порядке прохождения дерева сценариев каждым классом документов.

Название XML-скрипта	Назначение
sysopertype.xml	Перечень типов системных операций.
acctype.xml	Перечень типов счетов.
finotype.xml	Перечень типов финансирования.
paydocauthor.xml	Перечень лиц, которые могут создавать платежные документы.
payground.xml	Перечень оснований платежа.
paytype.xml	Перечень типов платежа.
accplan.xml	План счетов.
@sysaccsetup.lst	Список XML-скриптов с указанием информации о проводках, используемых каждым классом документов.
repkind.xml	Перечень форм отчетов.
doccategory.xml	Перечень модулей. Нужен для включения функционала, реализованного в рамках общих модулей.
sysconf.xml	Перечень системных параметров.
sysparam.xml	Стандартная предустановка параметров системы.
rtpl_config.xml	Перечень настройки репликаций между клоном и основной базой.
client_resource.xml	Перечень описаний клиентских ресурсов.
profiles.xml	Перечень профилей отчетов.
sysright.xml	Стандартные права пользователей в системе.
creditcalc.xml	Правила расчета кредита.
schtask.xml	Перечень заданий для исполнения планировщиком задач.
;inguarantee.xml	Не используется.
funcat.xml	Список функциональных категорий системы.
plantermkind.xml	Перечень типов периодов планирования.
expviewerscols70n.xml	Состав колонок форм просмотра бюджета расходов, доходов, источников и операций БУ/АУ.
ci_calcmethod.xml	Не используется.
orgacctype.xml	Перечень типов счетов организаций.
orgrole.xml	Перечень ролей организации.
orgstatus.xml	Перечень статусов организаций.
bankoper.xml	Перечень типов банковских операций.
;banks.xml	Не используется.
;symbcash.xml	Не используется.
territory.xml	Перечень территорий.
budgcarry.xml	Перечень бюджетных проводок.
finsource.xml	Перечень источников финансирования.

Название XML-скрипта	Назначение
estkind.xml	Перечень типов бланков расходов.
orgbookcapt.xml	Перечень заголовков привязок бухгалтерских книг к организациям.
docexchange.xml	Перечень привязок классов документов к сайтам.
rplbudget.xml	Шаблоны объектов репликации сайтов исполняемых бюджетов.
budglevel.xml	Перечень уровней бюджетов.
extstatus.xml	Перечень внешних статусов документов системы.
operkind.xml	Перечень видов операций по ЛС.
taskaudit.xml	Настройки правил журнализации обращений к процессорам системы.
placeordermethod.xml	Перечень способов размещения заказов.
limdoc_multiestimate.xml	Настройка возможности создания многострочных документов с разными бланками расходов.
TypeExpSchedule.xml	Настройка доведения бюджетных данных за счет связанных кредитов.
SymSecrExpShedule.xml	Не используется.
primarydoctype.xml	Список типов первичных документов.
;clndtype.xml	Не используется.
clnddaytype.xml	Перечень типов дат.
;clndday.xml	Не используется.
fgroups.xml	Настройка групп полей для подписания документов ЭП.
stockexchange	Справочник бирж.
sc_issueform	Справочник форм выпуска ценных бумаг.
sc_kind	Справочник наименований и видов ценных бумаг.
sc_placementkind	Справочник видов размещения ценных бумаг.
sc_redemptionorder	Справочник порядка погашения ценных бумаг.
@azk2_tfo.lst	Не используется.
;Calc@calc.lst	Не используется.
;Calc@spreadsheet.lst	Не используется.
@module.lst	Добавляет категорию по умолчанию в справочник типов прикрепляемых файлов.
Importer\@importer.lst	Перечень объектов системы, необходимых для загрузки справочника ППО.
;anydoctype.xml	Не используется.
grantinvestmenttype.xml	Соответствие КВФО типу субсидии.
pfgsource.xml	Источники средств целевых назначений по уровням бюджета.
rolegisgmp.xml	Роли для модуля ГИС ГМП.
doctypecode.xml	Справочник типов услуг для потребителей услуг.

Название XML-скрипта	Назначение
pl_servererror.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_serverprocessor.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_serverprovider.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_dbconstraint.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_reference.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_numgenerator.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_clientobject.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
planinrefkind.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_import_reference_params.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_import_errors.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
pl_direct_group_template.xml	XML-скрипт с указанием информации об объектах, используемых в системе «АЦК-Планирование».
unidoctype.xml	Типы универсальных документов.
ln_ref_edit_pl.xml	XML-скрипт для настройки редактирования справочников, используемых в системе «АЦК-Планирование».
ref_edit_pl.xml	XML-скрипт для настройки редактирования справочников, используемых в системе «АЦК-Планирование».

4.2.2.4 Утилиты запуска XML-скриптов

Рассмотрим примеры использования утилиты запуска XML-скриптов. Для упрощения запуска приложения рекомендуется использовать командный файл, например:

```
rem "xml.cmd"
@echo off
:start
set SERVER_PROPS="%_HOME%/XML/xml.properties"
set JAVA_PARAMS=-Dconfig.log=com.bssys.azkserver.system.LogConfigurator
call %_HOME%\server\bin\start com.bssys.tools.xml.Executer -config %SERVER_PROPS% %*
```

С помощью командного файла можно исполнить от одного до девяти XML-скриптов или XML-списков (списков XML-скриптов или списков XML-списков).

```
xml.cmd @azk_sys.lst -ignore  
xml.cmd serverprocessors.xml serverprovider.xml @azk_data.lst
```

Отличительной особенностью XML-списков является наличие в конце имени файла подстроки .lst. Списки состоят из перечня файлов (по одному наименованию файла в строке). Пустые строки допустимы. Например, рассмотрим содержимое файла @azk_sys.lst:

```
serverprocessors.xml  
serverprovider.xml  
servererror.xml
```

Для работы в интерактивном режиме (ручной ввод XML-команд) необходимо запустить командный файл без указания файлов для исполнения в командной строке.

```
xml.cmd  
xml.cmd -budget_id "1"
```

4.2.3 Утилиты для работы с БД

В процессе эксплуатации СУБД администратору системы необходимо выполнять совокупность действий, направленных на поддержание БД в работоспособном состоянии, обеспечения ее безопасности и уменьшения потерь информации после сбоев.

Чтобы упростить процесс эксплуатации БД, спроектированы и разработаны следующие утилиты командной строки.

4.2.3.1 Описание утилиты executer.cmd(sh)

SQLExecuter – консольное java-приложение предназначенное для исполнения SQL-скриптов, при помощи которых осуществляется первичное наполнение СУБД (Oracle, Interbase или MSSQL) данными, а также модификация данных в СУБД на этапе обновления ПО.

SQLExecuter обеспечивает единый для всех поддерживаемых СУБД синтаксис SQL, автоматически модифицируя SQL-скрипты в зависимости от особенностей SQL-диалекта используемой СУБД.

В состав программы входят java-классы, реализующие логику функционирования программы (`com.bssys.tools.sqlxecuter.Executer` – базовый java-класс).

`Executer.sh` – основной файл сценария Linux SHELL, предназначенный для передачи управляющих параметров в класс `com.bssys.tools.sqlxecuter.Executer`.

`Executer.cmd` – основной файл сценария OS Windows, предназначенный для передачи управляющих параметров в класс `com.bssys.tools.sqlxecuter.Executer`;

Перед запуском программы необходимо настроить сервер приложений, отредактировав его файл конфигурации. Путь к файлу свойств сервера приложений можно настроить, открыв основной файл сценария `executer.sh` для ОС Linux (`executer.cmd` для ОС Windows), и установив в нем значение параметра `SERVER_CONFIG`. В `SERVER_CONFIG` допустимо использовать относительные пути. `SERVER_CONFIG` равен по умолчанию значению `../Azk2Server.properties`.

При настройке файла свойств сервера приложений особое внимание стоит уделить следующим параметрам:

- `azk.db.driver` – класс драйвера, который будет использоваться для доступа к СУБД;
- `azk.db.url` – путь к базе данных или схеме данных (по этому пути будет создана новая база данных или схема данных);
- `azk.db.user` – пользователь базы данных;
- `azk.db.password` – пароль пользователя базы данных.
- `azk.db.sysuser` – системный пользователь СУБД;
- `azk.db.syspassword` – пароль системного пользователя СУБД.

В `SQLExecuter` допустимо использовать следующие параметры:

- `ignore` – не прерывать работу в случае ошибки при выполнении скриптов;
- `pathlist <путь>` – дополнительные пути, для поиска SQL-скриптов где искать скрипты
- `site <номер сайта>` – задает номер сайта (используется только совместно с командой `perform_all`)

- *config* - путь к файлу конфигурации сервера приложений (включая имя файла)

Команды:

- *perform <file1 ... fileN>* - обработать один или несколько SQL-скриптов перечисленных после команды (допустимы SQL-файлы и LST-файлы). Запускается под ОС Linux *sh ./executer.sh perform <file1 .. fileN>*, под ОС Windows *executer.cmd perform <file1 .. fileN>*. На основе перечисленного в аргументах команды списка файлов (<file1 .. fileN>) создается список SQL-скриптов, которые затем обрабатываются последовательно (в порядке упоминания в командной строке и LST-файлах).
- *triggers* – создать триггеры в БД. Запускается под ОС Linux *sh ./executer.sh triggers*, под ОС Windows *executer.cmd triggers*.
- *triggersdrop* - удалить триггеры из БД. Запускается под ОС Linux *sh ./executer.sh triggersdrop*, под ОС Windows *executer.cmd triggersdrop*.
- *setsite <site_number>* – установить номер сайта равным <site_number> (если номер сайта < 0, то таблица SystemSite будет удалена). Запускается под ОС Linux *sh ./executer.sh setsite <site_number>*, под ОС Windows *executer.cmd setsite <site_number>*.
- *export* – экспорт описания данных БД в XML-файл. Запускается под ОС Linux *sh ./executer.sh export > <file>*, под ОС Windows *executer.cmd export > <file>*.
- *compare <scheme1.xml> <scheme2.xml>* – сравнение двух XML-описаний данных БД. В процессе работы проверяются следующие параметры схем БД. Для схемы данных: список таблиц. Для каждой таблицы: список полей, список констрейнтов, список индексов. Для каждого поля таблицы: типы полей, размеры полей, наименования полей, признак обязательности заполнения поля. Для каждого констрейнта: список полей, наименование констрейнта, ссылка на таблицу, поле в таблице (по ссылке), наименование констрейнта, на который ссылается данный констрейнт, тип констрейнта. Для каждого индекса: список полей индекса, наименование индекса, тип индекса. Запускается под ОС Linux *sh ./executer.sh compare <file1> <file2>*, под ОС Windows *executer.cmd compare <file1> <file2>*
- *update <file1 ... fileN>* - провести обновление БД, используя SQL-файлы и LST-

файлы, перечисленные в `<file1 ... fileN>`. После старта приложения на экран выводится список обнаруженных новых SQL-скриптов и пользователю предлагается подтвердить намерение обновить БД (Do you want to execute them? (Y/N) [N]:), в случае ответа «Y» запускается процесс обновления структуры БД, в случае другого ответа работа приложения прекращается. Запускается под ОС Linux `sh ./executer.sh update <file1 ... fileN>`, под OS Windows `executer.cmd update <file1 ... fileN>`.

- `perform_all <file1 ... fileN>` – установить номер сайта (если указан параметр `-site`), обработать все файлы из списка `<file1 ... fileN>`, после чего создать в БД триггеры. Запускается под ОС Linux `sh ./executer.sh perform_all <file1 ... fileN>`, под OS Windows `executer.cmd perform_all <file1 ... fileN>`.
- `create_user` – создать новую базу данных (схему данных). Перед запуском убедитесь, что: каталог, указанный в параметре `azk.db.url` существует; файл или схема, указанная в параметре `azk.db.url` не существует. В процессе работы будет создана новая база данных (файл или набор файлов). Запускается под ОС Linux `sh ./executer.sh create_user`, под OS Windows `executer.cmd create_user`.
- `drop_user` – удалить существующую базу данных (схему данных). После старта приложения, пользователю предлагается подтвердить намерение удалить базу данных (Type yes to confirm:), в случае ответа «yes» база данных будет удалена, в случае другого ответа база данных удалена не будет и работа приложения будет завершена. Запускается под ОС Linux `sh ./executer.sh drop_user`, под OS Windows `executer.cmd drop_user`.

4.2.3.2 Утилита DBUpdate.cmd(sh)

Утилита `DBUpdate.cmd(sh)` вызывает утилиту `executer.cmd(sh)` с параметрами `pathlist`, `ignore` и командой `update`. В качестве листа скриптов в команде `update` используется файл `perform.lst`, содержащий список всех sql скриптов.

При запуске утилиты считывается таблица `DBUpdate` в базе, в которой хранятся номера выполненных скриптов, и файл `perform.lst`. Если в `perform.lst` есть номерные скрипты, которые отсутствуют в таблице `Dbupdate`, то на экран выведется список не выполненных скриптов с предложением их пролить.

4.2.3.3 Утилита `convert.cmd`

Для конвертации баз данных, с помощью утилиты `convert.cmd`, необходимо установить пути к файлу параметров исходной базы, из которой будут «выливаться» данные, параметр `SOURCE_CONFIG`, и путь к файлу параметров, в которую будут «заливаться» данные, параметр `TARGET_CONFIG`. При запуске утилиты выполняется консольное java-приложение. В состав программы входят java-классы, реализующие логику функционирования программы (`com.bssys.tools.dbc.Main`- базовый java-класс). В файле параметров, указанном в `TARGET_CONFIG`, должен быть прописан путь к «пустой» базе данных. Создание «пустой» базы данных описано в разделе [Создание новой БД](#)¹⁵⁹.

4.2.3.4 Утилита `comparer.cmd`

В утилите `comparer.cmd` прописываются следующие параметры:

- Параметры базы для сравнения – целевой базы (путь к базе, пользователь базы данных, пароль для пользователя базы данных).
- Параметры эталонной базы (путь до базы, пользователь базы данных, пароль для пользователя базы данных).
- Прописывается путь до конфигурационного файла, необходимый для конфигурирования лога.
- Устанавливаются параметры java.

Утилита `comparer.cmd` вызывает `\server\bin\start` в качестве параметра передается java-класс `com.bssys.tools.dbcompare.DBComparer` и различные конфигурационные параметры для эталонной и целевой баз.

4.2.3.5 Утилита `update.cmd(sh)`

Утилита `update.cmd(sh)` используется для обновления БД и вызывает утилиту `executer.cmd(sh)` с командами `triggersdrop`, `perform`, `triggers`. При выполнении команды `triggersdrop` происходит отключение триггеров. В качестве листа скриптов в команде `perform` используются файлы `update_rollback.lst`, `update_perform.lst`, содержащие список всех sql скриптов. При выполнении команды `triggers` происходит включение триггеров.

Если при выполнении утилиты происходят ошибки, то они не игнорируются и работа утилиты завершается (параметр, отвечающий за игнорирование ошибок – NOIGNORE, если true, ошибки не игнорируются).

4.2.3.6 Создание новой БД

Для создания новой БД начального уровня (т.е. полностью работоспособную настроенную базу с минимально настроенными пользовательскими справочниками) используется командный файл `./create_db_interbase.cmd` (для СУБД Firebird) или `./create_db_oracle.cmd` (для СУБД Oracle).

Для успешного выполнения данной утилиты, необходимо иметь доступ к установленному в локальной сети серверу баз данных и предустановленную на локальной машине виртуальную машину Java (JRE).

Перед запуском утилиты необходимо предварительно заполнить:

- параметр **azk.license.name** в файле свойств временного сервера приложений **Server.properties.template**;
- параметры создания новой базы в файле настроек **./create_db_interbase.properties** (для СУБД Firebird) или **./create_db_oracle.properties** (для СУБД Oracle). Назначение каждого параметра подробно описано в комментариях внутри файла;
- значение атрибута **setEndDateOfLatestApprovedLine** тега **REFERENCE** в XML-скриптах с данными справочников кодов бюджетной классификации (БК) в папке **\Xml\Template**. Атрибуту устанавливается одно из двух значений:
 - **setEndDateOfLatestApprovedLine = "1"** – при выполнении XML-скрипта кодам БК, которые есть в базе данных, но отсутствуют в XML-скрипте, устанавливается дата окончания действия последней утвержденной строки кода БК (если она не установлена) – 31.12. XX (XX – год бюджета, в котором заведен код).
 - **setEndDateOfLatestApprovedLine = "0"** – кодам БК, отсутствующим в выполняемом XML-скрипте, дата окончания действия не устанавливается.

После заполнения всех параметров запускается файл `./create_db_interbase.cmd` (для СУБД Firebird) или `./create_db_oracle.cmd` (для СУБД Oracle), который последовательно выполнит следующие действия:

- подключится к серверу БД;
- создаст базу данных (для СУБД Firebird) или схему (для СУБД Oracle);
- выполнит все необходимые SQL-скрипты;
- запустит сервер приложения;
- выполнит все необходимые XML-скрипты;
- зарегистрирует сайты «АЦК-Планирование» по указанным параметрам;
- заведет бюджет и справочники;
- создаст файл свойств (*.properties) и командный файл для запуска данного экземпляра со следующими именами: **StartServer_<db_instance>.bat** и **Server_<db_instance>.properties**.

В ходе выполнения утилиты могут возникнуть ошибки:

1. При попытке создать экземпляр «АЦК-Планирование» с именем (параметр **db_instance**), который уже существует, процесс создания файлов прекращается.
2. Если создается БД, которая уже существует, то в логе формируется сообщение о невозможности выполнения процедуры. Такое же сообщение сформируется в логе, создание базы данных не удалось (недостаточно прав, недостаточно места на диске и т.п.).

4.2.3.7 Резервное копирование/восстановление БД

Информация о резервном копировании/восстановлении для СУБД Oracle, Firebird, PostgreSQL приведена в документе «[БАРМ.00004-55 32 01-8](#)»



5

Установка и настройка системы

Под установкой и настройкой системы «АЦК-Планирование» понимается совокупность действий, которые необходимо выполнить для формирования операционного окружения среды выполнения, физического размещения файлов, запуска и мониторинга приложений, оптимизации работы всей системы в конечном итоге.

5.1 Установка и настройка сервера приложений

Для успешного запуска сервера приложений «АЦК-Планирование» необходимо наличие среды выполнения Java (JRE). В настоящее время разработка/тестирование сервера приложений ведется на JDK/JRE 1.8. Минимальным требованием является использование JRE версии 1.8. Использование JRE снимает ограничение на используемую операционную систему. Рассмотрим работу только в ОС Windows и ОС Linux.

Для настройки сервера приложений используется конфигурационный файл. В нем находится перечень параметров, которые влияют на работу различных служб сервера приложений; в том числе параметры подключения к базе данных.

5.1.1 Установка и настройка JRE

Последнюю версию программного обеспечение среды исполнения Java для используемой серверной платформы можно скачать с сайта производителя <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Настоятельно не рекомендуется использовать версии JRE, которые публикуются на сайте для alpha- и beta-тестирования.

Для платформы ОС Windows инсталляционный пакет JRE поставляется в виде исполняемого файла с расширением *.exe. По имени этого файла, как правило, легко можно определить его версию, платформу и минимальный тип процессора. Например, файл **jre-8u45-windows-i586.exe**, есть не что иное, как JRE Java версии 1.8, ОС Windows и требуемый процессор не ниже Pentium. Для установки JRE запускаем этот файл и действуем согласно выводимым инструкциям. Как правило, параметров по умолчанию вполне достаточно. Однако рекомендуем устанавливать JRE в **C:\Program Files\Java\jre8**. Наименование каталога инсталляции JRE добавить в переменные среды (в контекстном меню **Мой компьютер** выбрать пункт **Свойства**, на закладке **Дополнительно** нажать

кнопку **Переменные среды**). В форме переменных сред присвоить путь к каталогу инсталляции JRE переменной окружения `JAVA_HOME`:

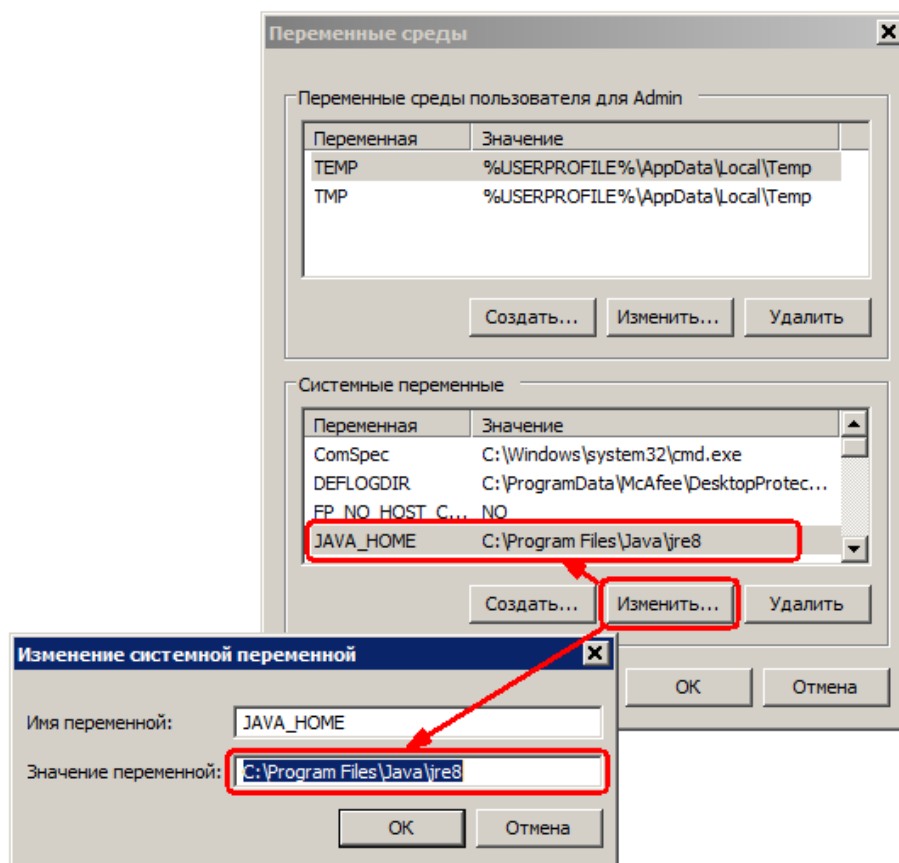


Рисунок 2 – Редактирование переменных сред

После окончания установки, обычно требуется перезагрузка компьютера.

Для платформы Linux инсталляционный пакет JRE поставляется в виде пакета `*.rpm`. Для установки JRE необходимо войти в систему с правами администратора (`root`) и выполнить команду:

```
#rpm -Uvh <file_jre.rpm>
```

Наименование каталога, в который была установлена JRE, необходимо присвоить переменной окружения `JAVA_HOME` и добавить в переменную окружения `PATH`. Это можно сделать глобально для всех пользователей, прописав следующие строки в конец файла `/etc/profile`:

```
export JAVA_HOME=/usr/java/jre1.8.0_01  
export PATH=$PATH:$JAVA_HOME
```

Перезапуск системы после установки JRE в ОС Linux не требуется. Изменения в файле `/etc/profile` будут актуальны для вновь зарегистрированных в системе пользователей; для использования изменений в текущем сеансе нужно непосредственно исполнить этот файл командой:

```
# ./etc/profile
```

Для проверки того, что виртуальная машина JAVA установлена и доступна для использования, можно использовать команду:

```
#java -version
```

В ответ должна быть отображена информация об установленной версии JRE либо, в случае неправильных настроек, сообщение о попытке запуска неизвестного приложения.

5.1.1.1 Настройка Java Cryptography Extension (JCE)

Настройка Java Cryptography Extension (JCE) к JRE осуществляется следующим образом:

1. Скачать add-on к JRE «Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy» <http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html>.
2. В форме переменных сред присвоить путь к каталогу инсталляции JDK8 переменной окружения `JAVA_HOME`, выполняется аналогично [Установка и настройка JRE](#)¹⁶².
3. Установить add-on к JRE, скачанный в п.1. Для этого необходимо:
 - распаковать архив;
 - скопировать 4 файла из него (заменяя существующие) по пути `%JAVA_HOME%\jre\lib\security`.

Примечание. Если ранее на данном сервере и в данной копии JDK было установлено и настроено программное обеспечение Mendelson, то данный add-on уже установлен.

5.1.2 Запуск и остановка сервера приложений. Файл конфигурации

Для запуска [сервера приложений](#)³³ «АЦК-Планирование» используются командные файлы, которые были написаны для каждой из используемых операционных систем.

Для запуска сервера приложений в среде ОС Windows используется командный файл **./StartServer.bat**. Формат команды следующий:

```
StartServer.bat [<port> <file.properties>]
```

где **port** – номер порта для прослушивания сервером приложений,

file.properties – наименование файла конфигурации сервера приложений.

Задание параметров командной строки не является обязательным. Если параметры не заданы явно, то будут использованы значения параметров, заданные по умолчанию ("**port**"=2001, "**file.properties**"=Server.properties). Установить другие значения по умолчанию можно путем изменения командного файла **./StartServer.bat**.

Корректное завершение работы сервера приложения осуществляется с использованием комбинации клавиш **<Ctrl+C>**. При простом закрытии окна сервера приложения возможны проблемы в работе.

Для запуска сервера приложений под управлением ОС Linux используется командный файл **./startserver**, который расположен в корневом каталоге сборки. Формат команды следующий:

```
$startserver
```

Основные параметры для запуска сервера приложений:

- путь к файлу конфигурации (**SERVER_CONFIG=./Azk2Server.properties**);
- номер порта для прослушивания клиентских запросов – (**SERVER_PORT=2001**).

Результатом выполнения этой команды будет запущенный в фоновом потоке сервер приложений «АЦК-Планирование».

Примечание. Для запуска сервера приложений не в фоновом режиме, необходимо закомментировать строку "export CONSOLE_MODE=SILENT" в файле **StartServer**.

Для того чтобы увидеть этот поток, можно использовать системную команду:

```
$ps | grep java
```

По этой команде будет выведен список потоков, которые были инициализированы виртуальной машиной Java. В первом столбце выведенной информации будет отображаться PID. Это значение можно использовать для уничтожения потока сервера приложений командой kill:

```
$kill <pid>
```

Альтернативный способ остановки сервера приложений – это использование команды **killall**. По следующей команде будут закрыты все потоки, запущенные под управлением виртуальной машины Java:

```
$killall java
```

Не рекомендуется запускать сервер приложений с правами администратора ОС Linux (root).

Файл конфигурации сервера приложений содержит все необходимые для нормальной работы параметры. Для параметров, указанных в этом файле, значений по умолчанию не существует. Параметры в этом файле описываются в виде списка в формате *<name>=<value>*. Описание каждого параметра должно начинаться с новой строки и не содержать комментариев. Каждый параметр в файле имеет подробный комментарий и начинающийся с символа решетки «#».

5.2 Установка и настройка сервера БД

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

5.3 Подготовка БД к работе

Перед использованием БД сервером приложений в ней должна быть создана соответствующая структура. В ряде случаев БД может быть восстановлена из резервной копии или конвертирована из БД другого типа.

5.3.1 Методика приведения БД к нужной версии

Для того чтобы привести БД к нужной версии сборки необходимо определить, какие SQL- и XML-скрипты были изменены между установленной и целевой сборками.

Если вы планируете повышать версию стенда, то необходимо вначале применить на БД скрипты, которые отсутствовали с момента последней сборки. Запустить сервер приложений и исполнить новые версии XML-скриптов. После этого БД поднята до нужной версии сборки и готова к использованию.

Если вам необходимо понизить версию БД, что возникает крайне редко, то необходимо вначале откатить на БД скрипты, которые появились с момента целевой сборки. Запустить сервер приложений целевой версии и исполнить прежние версии XML-скриптов. На БД был произведен откат до нужной версии сборки, и она готова к использованию. По возникшим вопросам и ошибкам, возникшим в ходе обратного преобразования, необходимо консультироваться со службой поддержки в каждом частном случае.

5.4 Установка, настройка и запуск клиентской части

Клиентская часть «АЦК-Планирование» написана в среде разработки Delphi XE4. В связи с этим она требует для своей работы наличие файлов библиотек динамической компоновки. Эти файлы не входят в состав сборок, поэтому необходимо дополнительно установить компоненты Delphi XE4.

Для клиентского приложения, откомпилированного в среде разработки приложений Delphi XE4, необходимо использовать следующие файлы библиотек: **dbrtl180.bpl**, **dsnapi180.bpl**, **IndyCore180.bpl**, **IndyProtocols180.bpl**, **IndySystem180.bpl**, **inet180.bpl**, **soaprtl180.bpl**, **vcl180.bpl**, **vcldb180.bpl**, **vclsmpl180.bpl**, **vclx180.bpl**, **xmrtl180.bpl**.

Для установления соединения с сервером приложений «АЦК-Планирование» нужно настроить файл **Azk2CInt.ini**, находящийся в корневом каталоге **./client/**.

Параметры, которые необходимо задать в файле свойств online-клиента (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:8082/ehes (вместо localhost:8082 – указать адрес и порт web-сервера Tomcat, обслуживающего Web-сайт АЦК-Планирование); параметр «/ehes» 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; доступные значения TRUE и FALSE);

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

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

Остальные параметры **ini-файла** настраиваются опционально, т.е. при необходимости:

- **Interactive** – определяет будет ли выводиться при потере связи с сервером приложений диалог о повторной попытке соединения, значение по умолчанию равно **True**.
- **ServerCompression** – включение компрессии данных возвращаемых с сервера: данные приходят в сжатом виде и распаковываются на клиенте, значение по умолчанию равно **True**.
- **ClientCompression** – включение компрессии данных отсылаемых с клиента: данные отсылаются в сжатом виде и распаковываются на сервере, значение по умолчанию равно **True**.
- **UpdaterDisable** – параметр для запрета запуска службы обновления, значение по умолчанию равно **false**.
- **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=corbaloc:iiop:1.2@localhost:2001/DataSetProvider
```

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

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.).

Для запуска клиента необходимо запустить файл: **maincontroller.exe**.

В случае использования локальной сети рекомендуется файлы клиентской части размещать на сервере сети. Библиотеки динамической компоновки рекомендовано скопировать в каталог клиентской части, чтобы они были доступны для всех пользователей. Каталог с клиентской частью необходимо открыть для общего использования в режиме доступа «только чтение».

На клиентских машинах, на которых планируется использовать «АЦК-

Планирование», необходимо подключить сетевой диск, который указывает на предоставленный для всеобщего использования каталог клиентской части «АЦК-Планирование». Для удобства использования на рабочем столе компьютера клиента необходимо поместить ярлык, который бы использовался для запуска клиентского приложения. При необходимости, в ярлыке можно указать дополнительные ключи запуска клиентского приложения (например, для запуска нескольких клиентских приложений на одном рабочем столе).

5.5 Установка сборки отчетов

Отчеты системы «АЦК-Планирование» поставляются в виде отдельных сборок, сквозная нумерация которых не зависит от нумерации сборок самой «АЦК-Планирование».

Установка сборки отчетов производится путем копирования отчетной сборки на действующий стенд в корневой каталог. При этом клиентская и серверная части должны быть остановлены.

После копирования необходимо запустить сервер приложения и выполнить XML-скрипт `.xml/xml.cmd @azk2_rep_pl.lst`. Выполнение скрипта должно закончиться без ошибок.

После перезапуска сервера приложений «АЦК-Планирование» отчеты готовы к использованию в клиентских приложениях.

Если используется медленный канал связи клиентского приложения с сервером приложений, то файлы сборок отчетов необходимо копировать на каждый клиентский компьютер.

5.6 Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) учреждениях

Для осуществления выгрузки данных из системы на Официальный сайт для размещения информации о государственных (муниципальных) учреждениях (Официальный сайт ГМУ) предварительно необходимо осуществить настройку выгружаемых данных.

В параметрах пользователя указывается каталог для выгрузки в поле **Каталог экспорта на сайт bus.gov.ru** (пункт меню **Сервис**→**Параметры пользователя**, группа настроек **Выгрузка на bus.gov.ru**).

Экспорт информации по учреждениям осуществляется через пункт меню **Отчеты**→**Организация работы с федеральным сайтом**→**Выгрузка данных по учреждениям**:

Выгрузка информации по учреждениям

Кем сформировано:

Для кого: Все подведомственные

Общая информация об учреждении

Информация о государственном (муниципальном) задании

Информация о государственном задании в соответствии с Постановлением Правительства РФ от 26 июня 2015 г. № 640

Информация о плане финансово-хозяйственной деятельности

Информация о плане финансово-хозяйственной деятельности, начиная с 2017 года

Информация о плане финансово-хозяйственной деятельности, начиная с 2020 года

Параметры выгрузки

Выгружать строки с пустым значением в поле 'Код строки'

Информация об операциях с целевыми средствами из бюджета

Информация о бюджетных обязательствах и их исполнении

Выгружать прикрепленные документы

Проверять данные перед экспортом

Выгружать данные на сайт

Адрес сайта: Указать другой сайт

https://private.bus.gov.ru/gmu-integration-web/services/upload

Пароль:

Проверить Начать экспорт

Рисунок 3 – Окно экспорта данных по учреждениям

В форме отчета заполняются поля:

- **Кем сформировано** – учреждение, осуществляющее выгрузку данных.
- **Для кого** – учреждение, для которого осуществляется выгрузка данных. При незаполнении поля и выключенном параметре **Все подведомственные** файл формируется по организации, указанной в поле **Кем сформировано**.
- **Все подведомственные** – если параметр включен, то осуществляется контроль, чтобы в списке содержались все подведомственные организации.

*Примечание. Если организация выгружает информацию о себе, она выбирает себя в полях **Кем сформировано** и **Для кого**.*

- **Общая информация об учреждении** – при включении параметра выгружается общая информация об учреждении.
- **Информация о государственном (муниципальном) задании** – при включении параметра выгружается информация о государственном (муниципальном) задании на оказание государственных (муниципальных) услуг (выполнение работ) и его исполнении. Формирование информации о показателях качества осуществляется с учетом настройки системы «АЦК-Планирование» **Отображать показатели качества согласно услуге (работе) (Сервис→Системные параметры→Планирование→Бюджетные услуги (работы)→Государственное (муниципальное) задание)**. При выключенной настройке выгрузка производится по каждому показателю качества с суммированием значений для одного наименования, так же, как для показателей объема.

Примечание. На ЕИС выгружаются данные из ЭД «Государственное (муниципальное) задание» на статусе «утверждено».

- **Информация о государственном задании в соответствии с Постановлением Правительства РФ от 26 июня 2015 г. №640** – при включении параметра выгружается информация о государственном задании по форме в соответствии с Постановлением Правительства РФ от 26 июня 2015 г. № 640 «О порядке формирования государственного задания на оказание государственных услуг (выполнение работ) в отношении федеральных государственных учреждений и финансового обеспечения выполнения государственного задания».
- **Информация о плане финансово-хозяйственной деятельности** – при включении параметра выгружается информация о плане финансово-хозяйственной деятельности.

Примечание. На ЕИС выгружаются данные из ЭД «План ФХД» на статусе «утверждено».

Внимание! Выгрузка данных ЭД «План ФХД» на ЕИС производится только по тем записям документа, у которых для заведенного показателя указано значение поля **Код информации Плана ФХД** из справочника «Информация о плане ФХД для выгрузки на сайт ГМУ».

При указании иных кодов, кроме представленных, выгрузка по ним производиться не будет.

- **Информация о плане финансово-хозяйственной деятельности, начиная с 2017 года** – при включении параметра выгружается информация о плане финансово-хозяйственной деятельности. По умолчанию параметр активен для бюджетов 2017 года и позднее.

- **Информация о плане финансово-хозяйственной деятельности, начиная с 2020 года** – при включении параметра выгружается информация о плане финансово-хозяйственной деятельности в новом формате. По умолчанию параметр активен для бюджетов 2020 года и позднее.
- **Выгружать строки с пустым значением в поле «Код строки»** – при включении параметра выгружаются строки с пустым значением в поле **Код строки**. Доступен к установке только при установке признака **Информация о плане финансово-хозяйственной деятельности, начиная с 2020 года**.
- **Информация об операциях с целевыми средствами из бюджета** – при включении параметра выгружается информация об операциях с целевыми средствами из бюджета.

Примечание. На ЕИС выгружаются данные из ЭД «Сведения об операциях с целевыми субсидиями» на статусе «утверждено».

- **Информация о бюджетных обязательствах и их исполнении** – при включении параметра выгружается информация о бюджетных обязательствах и их исполнении.

Примечание. Выгрузка производится из справочника «Данные для расчета планируемых расходов» по бюджетным обязательствам с типом «План».

- **Выгружать прикрепленные документы** – при включении параметра прикрепленные документы будут выгружены на сайт.
- **Проверять данные перед экспортом** – при включении параметра производится проверка данных перед экспортом.
- **Выгружать данные на сайт** – при включении параметра данные будут выгружены на сайт.
- **Логин** – логин пользователя, осуществляющего выгрузку данных на сайт.
- **Пароль** – пароль пользователя, осуществляющего выгрузку данных на сайт.

*Примечание. Поля **Логин** и **Пароль** доступны для заполнения, если включен параметр **Выгружать данные на сайт**.*

*Примечание. Не выгружаются строки, дата окончания которых проставлена и менее даты утверждения документа. Подготовленные строки xml сортируются не в порядке следования на закладках ПФХД, а в порядке значения поля **Код строки**. Реализована предварительная проверка на заполнение поля Код ОГС в справочнике «Организации» для значений, указанных в полях **Кем сформировано**, **Для кого сформировано**, **Орган, осуществляющий функции и полномочия учредителя**.*

После заполнения полей нажимается кнопка **Начать экспорт**.

*Примечание. При нажатии на кнопку **Проверить** осуществляется контроль на правильность заполнения полей справочника «Организации» (при включенном параметре **Общая информация об учреждении**).*

5.6.1 Порядок предоставления справочной информации

Для каждой организации формируется xml-документ. Документ размещается в папке с названием организации. Организации, имеющие вышестоящую организацию – ГРБС (РБС), размещаются в подчиненных папках.

Формирование XML-документов производится в формате UTF-8. На каждый сформированный XML-документ формируется ЭП. ЭП формируется с помощью сертификатов, выданных УЦ Федерального казначейства. Требования к электронной подписи:

- сформирована в соответствии со стандартом CadES-BES (<http://tools.ietf.org/html/rfc5126.html#section-4.3.1>);
- отсоединенная (detached);
- формат – Base64.

ЭП сохраняется как файл с именем, соответствующим имени сообщения, и с расширением isf.

XML-документы архивируются в формате GZIP вместе с ЭП (в соответствии со стандартом DEFLATE <http://tools.ietf.org/html/rfc1951>), информация в данном формате доступна для загрузки в анонимном доступе в режиме «только для чтения». Размер каждого сформированного файла не должен превышать 1 Мб (допустимая погрешность – 10%). В случае если размер файла превышает вышеуказанный размер, выгружаемые сведения разбиваются на два или более XML-документов.

На данном этапе предоставление справочной информации осуществляется в режиме полной выгрузки (в состав полной выгрузки входят все позиции справочников, актуальные на момент выгрузки).

К XML-документу могут быть добавлены электронные копии бумажных документов, при этом допускается передача только содержимого, а не ссылки для скачивания. В качестве формата файлов рекомендуется использовать PDF.



Если к XML-документу не были приложены электронные копии бумажных документов, то допускается возможность приложить их на Официальном сайте ГМУ в личном кабинете.

Предоставление размещенной информации осуществляется посредством файлового обмена, для этого используется FTP-сервер Официального сайта ГМУ.

Предоставление размещенной информации осуществляется накопительным итогом и содержит размещенную информацию с момента предыдущей выгрузки.

Выгрузка размещенной информации осуществляется на Официальный сайт ГМУ осуществляется в автоматическом режиме в соответствии с настройками системы.

5.6.2 Заполнение полей справочника «Организации» для интеграции с ЕИС

Примечание. Действия  **Новый** и  **Удалить** недоступны для пользователей, для которых в функциональной роли назначено специальное право пользования справочником Организации.

Для организации обмена с ЕИС у записей справочника *Организации* обязательно должны быть заполнены следующие поля (согласно форматов Минфина РФ):

- **Код ОГС** – 13-значный код организаций государственного сектора, заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ИНН** – идентификационный номер налогоплательщика, недоступно на редактирование, значение заполняется при загрузке справочника с внешнего ресурса.
- **КПП** – код причины постановки создаваемой организации на учет, недоступно на редактирование, значение заполняется при загрузке справочника с внешнего ресурса
- **Краткое наименование** – краткое наименование организации, используемое в системе, недоступно на редактирование, значение заполняется при загрузке справочника с внешнего ресурса.
- **Название** – название организации, недоступно на редактирование, значение заполняется при загрузке справочника с внешнего ресурса.
- **Код вышестоящего ГРБС (РБС)** – код организации с ролью распорядителя. Заполняется

пользователем. Значение выбирается из справочника *Организации*. Указывается орган государственной власти (орган местного самоуправления), осуществляющий функции и полномочия учредителя в отношении бюджетных и автономных учреждений, орган государственной власти (органа местного самоуправления), осуществляющий бюджетные полномочия главного распорядителя бюджетных средств, в ведении которых находятся казенные учреждения.


Примечание. При выгрузке данных по ГРБС (РБС) должны быть указаны **ИНН, КПП, Наименование, Краткое наименование, Код ОГС.**

- **Телефон/факс** – номер телефона/факса организации, заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно. При заполнении указывается код города и двумя дефисами.
- **Роли** – роль учреждения. Из списка выбирается одно из значений: *Автономное учреждение, Бюджетное учреждение, Контрагент и т. п.*
- **ОГРН** – указывается основной государственный регистрационный номер учреждения (13 символов). Заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ОКФС** – указывается основной государственный регистрационный номер учреждения (2 символа). Заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ОКОПФ** – указывается значение из общероссийского классификатора организационно-правовых форм (2 символа). Заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ОКОГУ** – указывается значение из общероссийского классификатора органов государственной власти и управления (7 символов). Заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно
- **ОКПО** – указывается значение из общероссийского классификатора предприятий и организаций (10 или 8 символов). Заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ОКАТО** – код ОКАТО, соответствующий ОКАТО справочника *Территории*, заполняется пользователем 11 символами в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ОКТМО** – код ОКТМО, соответствующий ОКТМО справочника *Территории*, заполняется пользователем 8 символами в случае если оно не заполнено, редактируется пользователем, если заполнено неверно.
- **ОКВЭД** – код ОКВЭД, соответствующий Общероссийскому классификатору видов экономической деятельности, заполняется пользователем **в точном соответствии с классификатором** в случае

если оно не заполнено, редактируется пользователем, если заполнено неверно.

- **Код административной подчиненности** – код административной подчиненности, выбирается из справочника Код административных поступлений и выбытий. Для корректной выгрузки необходимо очищать данное поле от заполненных значений.
- **Принадлежность к территории** – указывается территориальная принадлежность организации из справочника *Территории*, заполняется пользователем в случае если оно не заполнено, редактируется пользователем, если заполнено неверно. У территории должен быть указан код ОКАТО 11 символов.
- **Публично-правовое образование** – указывается принадлежность организации к публично-правовому образованию из справочника *Публично-правовые образования*. У публично-правового образования должен быть указан код 11 символов.

Остальные поля общей части необязательны для заполнения и на выгрузку данных общей информации по учреждению не влияют.

На закладке **Ответственные лица** по кнопке **Новая**  необходимо добавить хотя бы одну запись с типом должности *-2-Руководитель*, поставить признак **Актуальное**, заполнить поля **Фамилия**, **Имя**, **Отчество**, **Наименование должности**. Это будет руководитель организации, по которой производится выгрузка.

На закладке **Местонахождение организации** для каждого региона необходимо заполнить наименование субъекта РФ, района РФ, города, населенный пункт, улицу, где находится организация. Для каждого заполненного значения указывается код из классификатора адресов России с указанием символов в соответствии с таблицей:

Таблица 26 – Формат ввода полей закладки **Местонахождение организации**

Наименование поля в системе	Формат ввода
Субъект РФ	Текст
Код КЛАДР	11 знаков
Район	Текст
Код КЛАДР	11 знаков
Город	Текст
Код КЛАДР	11 знаков
Населенный пункт	Текст
Код КЛАДР	11 знаков

Улица	Текст
Код КЛАДР	15 знаков
Номер дома	Текст
Номер офиса (квартиры)	Текст
Почтовый индекс	6 знаков

Примечание. Если ввести большее или меньшее число знаков, то информация не выгрузится.

Список необязательных для заполнения полей для интеграции с ЕИС

Заполнение данных на закладке **НПА** является необязательным. Здесь указываются реквизиты правового акта органа, осуществляющего функции и полномочия учредителя, о назначении членов наблюдательного совета.

Примечание. Заполнение доступно только для учреждений, имеющих значение признака Автономное учреждение или Бюджетное учреждение.

Выгрузка производится только по одному документу, указанному на этой закладке (на сайте отсутствует возможность принимать несколько правовых актов учредителя). Если будет указано несколько документов, выгрузка будет производиться по любому из них.

Таблица 27 – Формат ввода полей закладки **НПА**

Наименование поля в системе	Формат ввода
Вид правового акта	Текст
Орган издающий документ	Текст
Дата правового акта	Дата
Номер правового акта	Текст
Наименование правового акта	Текст
Тип документа	Текст
Ссылка для скачивания документа	Текст

Заполнение данных на закладке **Иные ОКВЭД** является необязательным. В поле **Код ОКВЭД** указывается информация об иных видах деятельности учреждения. Для корректной выгрузки код следует указывать в строгом соответствии с «Общероссийским классификатором видов экономической деятельности» с точкой между знаками.

Примечание. Информация об основном виде деятельности указывается в заголовочной части справочника в поле **ОКВЭД**.

Заполнение данных на закладке **Филиалы и представительства** является необязательным. На закладке указывается информация о филиалах и представительствах, если таковые имеются у учреждения. Заполнение поля **Учреждение** создавшее обособленное структурное подразделения необязательно. При его заполнении необходимо учесть, чтобы данное учреждение входило в список учреждений, имеющих филиалы.


Примечание. Данный контроль установлен ЕИС, его несоблюдение может повлечь ошибки при выгрузке.


При заполнении полей блока **Обособленные подразделения** необходимо заполнить поля, указанные в таблице ниже.

Таблица 28 – Формат ввода полей закладки **Филиалы и представительства**

Наименование поля в системе	Формат ввода
Тип	Выбор одного из трех типов: - Филиал - Представительство - Обособленное структурное подразделение
Обособленное подразделение	Заполнить поле Обособленное подразделение

5.6.3 Обработка приложенных документов при организации обмена с ЕИС

К организации можно присоединить файл. Для этого организацию необходимо открыть на редактирование кнопкой .

В форме организации нажимается кнопка  и далее необходимо следовать инструкции в разделе **Присоединение файлов к справочникам**¹⁸⁰.

Особенности присоединения файлов к организации:

- Приложение файлов для выгрузки является **необязательным**.
- Для вложенного файла обязательно следует указать категорию вложения:


- Учредительные документы.
- Свидетельства о государственной регистрации.
- Решения учредителя о назначении руководителя.
- Положения о филиалах, представительствах.
- Правовой акт о назначении членов наблюдательного совета.
- Прочие документы.
- Решение учредителя о создании государственного (муниципального) учреждения.

Примечание. По умолчанию, для вложенного файла указания общей информации указывается категория вложения – «Прочие документы».

Файлы можно присоединять к электронным документам. Для присоединения файла к документу необходимо следовать инструкции в разделе [Присоединение файлов к документу](#)¹⁸².

5.6.4 Присоединение файлов к справочникам

К записям справочников можно прикрепить файл произвольного формата, содержащий любые дополнительные сведения.

Перед присоединением файла необходимо открыть запись на редактирование. В форме редактирования нажимается кнопка . Открывается окно списка присоединенных файлов:

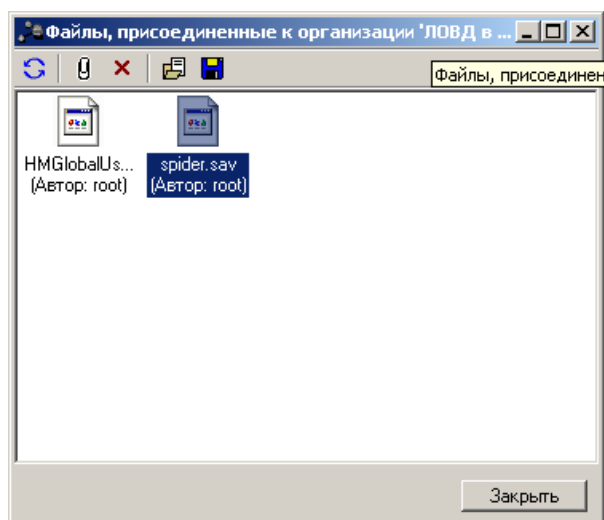



Рисунок 4 – Список присоединенных файлов к справочникам

В верхней части списка располагается панель инструментов. На ней находятся функциональные кнопки, с помощью которых можно выполнить действия: обновить список файлов, присоединить файл к документу, удалить присоединенный файл, открыть присоединенный файл и сохранить как.

Чтобы присоединить файл, необходимо нажать кнопку . На экране появится [окно](#)¹⁸³, в котором выбирается присоединенный файл.¹⁸²

В диалоговом окне выбора файла устанавливается курсор на нужный файл. В поле **Имя файла** вводится наименование файла или дважды нажимается левая кнопка мыши. Выбранный файл отобразится в списке присоединенных файлов.

Для указания категории вложения файлу в контекстном меню выбирается действие **Изменить категорию вложения**:

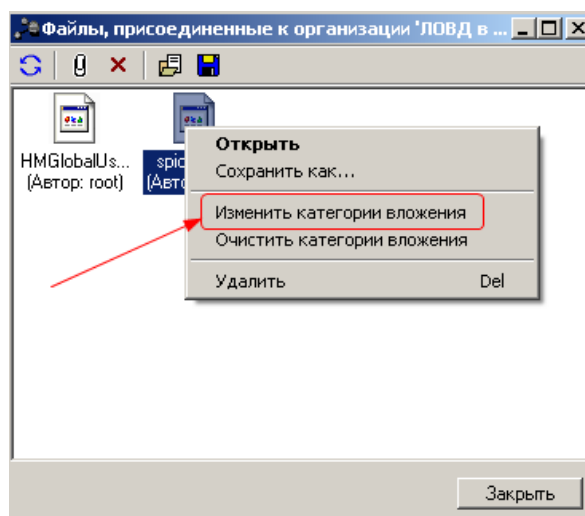



Рисунок 5 – Список присоединенных файлов, изменение категории вложения

Открывается справочник *Категории вложений*, где осуществляется множественный выбор значений¹⁸⁵.

5.6.5 Присоединение файлов к документу

К электронному документу можно прикрепить файл произвольного формата, содержащий любые дополнительные сведения.

Перед присоединением файла необходимо открыть документ на редактирование. В форме редактирования нажимается кнопка . Открывается окно списка присоединенных файлов:

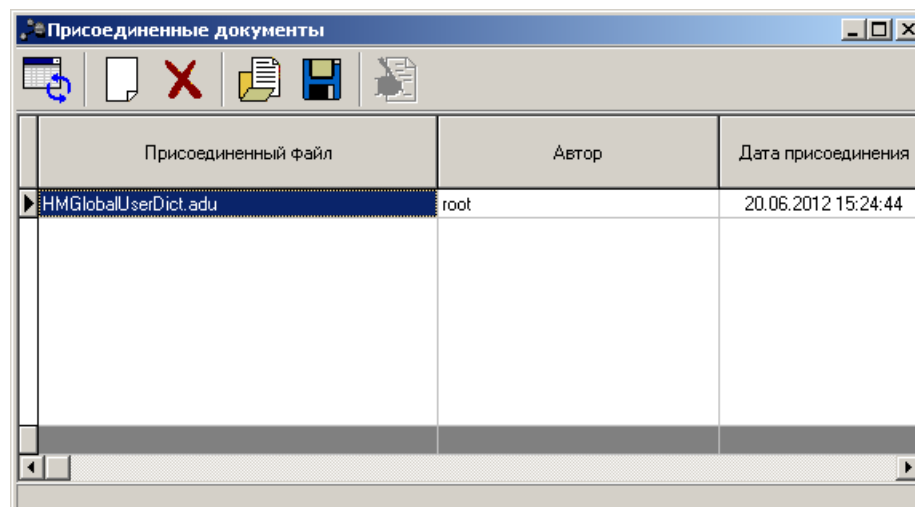


Рисунок 6 – Список присоединенных файлов

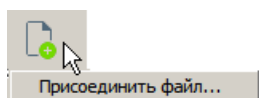
В верхней части списка располагается панель инструментов. На ней находятся функциональные кнопки, с помощью которых можно выполнить действия: обновить список файлов, присоединить файл к документу, удалить присоединенный файл, открыть присоединенный файл и сохранить как.

Список присоединенных файлов состоит из следующих колонок:

- **Присоединенный файл** – наименование присоединенного файла.
- **Автор** – имя автора присоединенного документа.
- **Дата присоединения** – дата/время присоединения документа.
- **Дата приема УДОД** – дата/время приема документа устройством УДОД (удаленная доставка отгрузочных документов).
- **Номер документа-владельца** – номер документа, к которому присоединен файл.
- **Дата документа-владельца** – дата документа, к которому присоединен файл.
- **Группа полей** – группа полей, к которой привязано вложение.
- **Количество ЭП** – количество ЭП, которыми подписано вложение.
- **Категории вложений** – наименование категории вложений.
- **Размер** – размер вложения.

Чтобы присоединить файл к документу, необходимо выбрать *Присоединить файл...*

в раскрывающемся списке



. На экране появится окно, в котором выбирается присоединяемый файл:

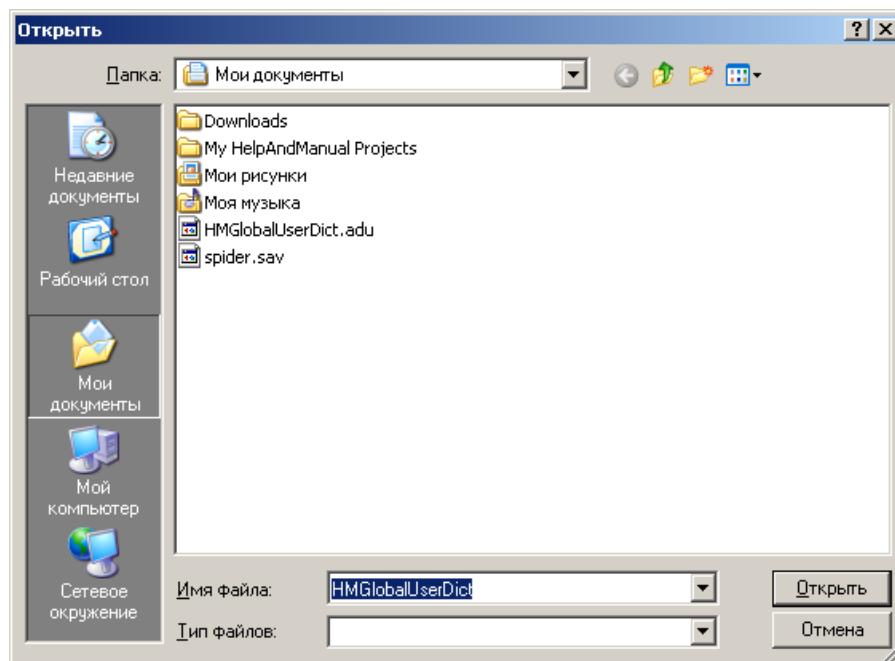


Рисунок 7 – Окно выбора присоединяемого файла

Примечание. При присоединении файла осуществляется контроль доступных категории файлов для назначения прикрепляемому файлу. Если у пользователя в правах в правах пользователя для выбранного статуса документа отсутствуют доступные категории файлов для назначения прикрепляемому файлу, на экран выводится соответствующее сообщение об ошибке, файл не прикрепляется.

В диалоговом окне выбора файла устанавливается курсор на нужный файл. В поле **Имя файла** вводится наименование файла или дважды нажимается левая кнопка мыши. Выбранный файл отобразится в списке присоединенных файлов.

К документу можно одновременно присоединить сразу несколько файлов или все файлы, содержащиеся в папке. Для этого необходимо, удерживая клавиши **<Ctrl>** или **<Shift>**, выделить файлы, которые требуется прикрепить к документу. Чтобы выделить все файлы в папке, используется сочетание клавиш **<Ctrl+A>**. После выделения файлов нажимается кнопка **Открыть**, вложения привязываются к группам полей и выбранные файлы отображаются в списке присоединенных файлов.

Если файл успешно привязан к группе полей или выбран параметр **Не привязывать** в окне привязки вложений к группам полей, осуществляется контроль полного доступа к категории для выбранного статуса. Если для выбранного статуса у пользователя полный

доступ только к одной категории, эта категория выбирается и файл прикрепляется. В противоположном случае проверяется значение системного параметра **Выбор категории вложения** (пункт меню **Сервис**→**Системные параметры**→**Присоединенные файлы**), если в параметре выбрано значение *Назначить категорию по умолчанию*, осуществляется контроль полного доступа к категории для выбранного статуса. Если контроль пройден, для выбранного файла устанавливается категория *Категория по умолчанию*, файл прикрепляется. Если контроль не пройден, на экран выводится окно выбора категории вложений.

При добавлении новых вложений к документу файлам по умолчанию присваивается категория вложения *Категория по умолчанию*. Для изменения категории вложения файла в контекстном меню выбирается действие **Изменить категорию вложения**:

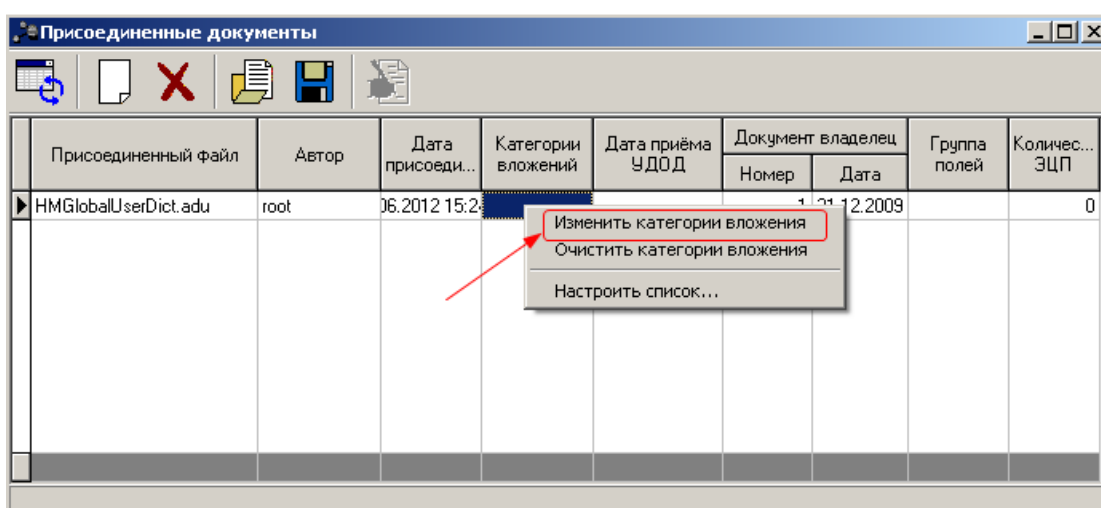


Рисунок 8 – Список присоединенных файлов, изменение категории вложения

Открывается справочник *Категории вложений*, где осуществляется множественный выбор значений:

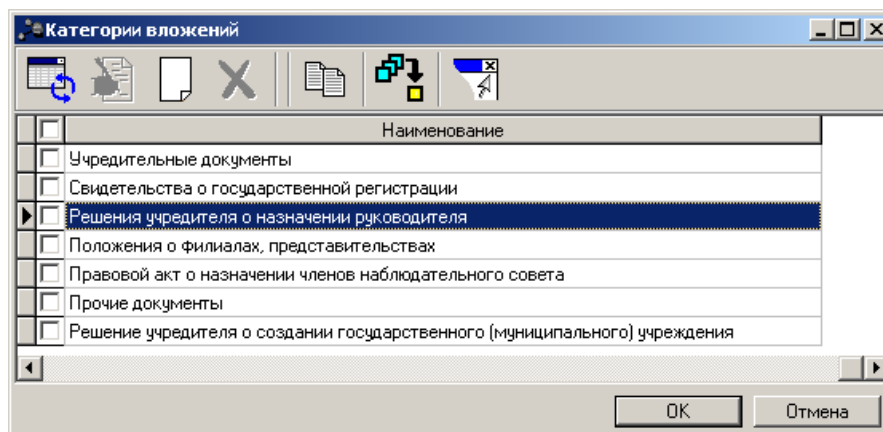


Рисунок 9 – Выбор категории вложений

Для закрытия справочника *Категории вложений* нажимается кнопка **ОК** или **Отмена**.

Выход из списка файлов, присоединенных к документу, осуществляется нажатием кнопки **Закреть**.


***Примечание.** Количество и размер присоединяемых файлов настраивается через пункт меню **Сервис**→**Системные параметры**, группа настроек **Присоединенные документы** (подробнее см. в документации «[БАРМ.00004-55 32 01-3 Система «АЦК-Планирование»](#). АРМ ФО. Блок администрирования. Подсистема администрирования. Настройка и сервис системы. Руководство администратора»).*

Действия пользователей по добавлению и удалению присоединенных файлов регистрируются в *Журнале действий пользователя*. *Журнал действий пользователя* содержит перечень всех произведенных над документом операций с указанием даты и имени пользователя, изменившего документ. *Журнал действий пользователя* вызывается из формы редактирования документа с помощью комбинации клавиш **<Ctrl+F11>**:

Пользователь	ФИО пользователя	Код статуса	Статус	Код предыдущего статуса	Предыдущий статус	Операция	Дата	Адрес клиента
staoe	Стахов Ольга Евгеньевна	0	Отложен	0	Отложен	Вставка	29.12.2015 10:36:59	192.168.33.25
staoe	Стахов Ольга Евгеньевна	0	Отложен	0	Отложен	Вставка	29.12.2015 10:36:59	192.168.33.25
staoe	Стахов Ольга Евгеньевна	0	Отложен	0	Отложен	Изменение	29.12.2015 10:38:41	192.168.33.25
staoe	Стахов Ольга Евгеньевна	0	Отложен	0	Отложен	Изменение	29.12.2015 10:39:48	192.168.33.25
staoe	Стахов Ольга Евгеньевна	0	Отложен	0	Отложен	Изменение	11.01.2016 11:52:48	192.168.33.27
staoe	Стахов Ольга Евгеньевна	17	Обработка	17	Обработка в бухгалтер	Перед проведением	11.01.2016 12:35:13	192.168.33.27
staoe	Стахов Ольга Евгеньевна	17	Обработка	17	Обработка в бухгалтер	Проводка прошла	11.01.2016 12:35:13	192.168.33.27
staoe	Стахов Ольга Евгеньевна	17	Обработка	17	Обработка в бухгалтер	Перед проведением	11.01.2016 12:35:13	192.168.33.27

Рисунок 10 – Журнал действий пользователей

При добавлении или удалении присоединенного файла в *Журнал действий* пользователей вносится соответствующая запись с указанием параметров файла и действия. В колонке *Присоединенный файл* указывается наименование файла, над которым было произведено действие. В колонке *Примечание* отображаются комментарии пользователей относительно произведенных действий над документом. Текст комментария вводится пользователем вручную в поле **Текст примечания** закладки **Примечание** формы

просмотра записи журнала, которая открывается на редактирование нажатием кнопки  на панели инструментов:

Дерево | XML | Примечание

Текст примечания:

Сохранить | Закрыть

Рисунок 11 – Форма просмотра записи журнала действий пользователей

Для сохранения введенного комментария нажимается кнопка **Сохранить**.

Примечание. Доступ к операциям с вложениями в электронные документы настраивается в функциональной роли пользователя для всех статусов обработки документа отдельно по каждой категории вложения из справочника Категории вложений. Настройка доступа к операциям с вложениями рассмотрена в «БАРМ.00004-55 32 01-4 Система «АЦК-Планирование». АРМ ФО. Блок администрирования. Подсистема администрирования. Управление доступом. Руководство администратора».

5.7 Настройка соединения Win-клиента через шифрованный канал https

Для соединения Win-клиента через шифрованный канал https необходимо настроить работу сервера приложений и delphi-клиента:

1. С помощью OpenSSL сгенерировать сертификат сервера приложений. В данном примере используется сборка openssl из проекта *finroot\dominant\client\Crypto\openssl*. При генерации необходимо задать атрибут **common name** сертификата. Для сертификата CA можно указать, например, *BFT TEST CA*, для серверного – *azkserver*, для клиентского имя пользователя. Перед генерацией сертификатов в файле *v3-server.ext* необходимо прописать реальное имя сервера приложений, к которому будет подключаться клиент, иначе https-соединение не будет устанавливаться (сертификат сервера не пройдет валидацию).

Генерация сертификата корневого CA, сертификата сервера приложений и сертификата клиента с использованием openssl

(в файле *v3-server.ext* вместо *azkserver* прописать реальное имя сервера, а также прописать правильный IP. 1)

```
set OPENSSL_CONF=openssl.cfg
openssl genrsa -out ca.key 4096
openssl req -new -sha256 -x509 -days 1095 -key ca.key -out ca.crt
openssl genrsa -out serverkey.pem 4096
openssl req -new -key serverkey.pem -sha256 -out server.csr -extensions server
openssl x509 -req -in server.csr -days 365 -sha256 -CAcreateserial -CA ca.crt -CAkey ca.key -out
```

```
server.crt -extfile v3-server.ext
```

```
openssl pkcs12 -export -in server.crt -inkey serverkey.pem -name "azk server" -out server.pfx
```

2. Установить сертификат *ca.crt* в доверенные корневые центры сертификации. Файл *server.pfx* (контейнер в формате PKCS12) скопировать в папку сервера приложений.

3. Если требуется клиентская аутентификация сформировать сертификат клиента:

```
openssl genrsa -out clientkey.pem 4096
```

```
openssl req -new -key clientkey.pem -sha256 -out client.csr -extensions usr_cert
```

```
openssl x509 -req -in client.csr -days 365 -sha256 -CAcreateserial -CA ca.crt -CAkey ca.key -out client.crt
```

```
openssl pkcs12 -export -in client.crt -inkey clientkey.pem -name "maincontroller" -out client.pfx
```

Формирование контейнера *pkcs12* необязательно. Допустимо использовать как файл *client.pfx*, так и файлы *client.crt* и *clientkey.pem*.

4. Если файл настроек Tomcat *embedded_tomcat_server.xml* не используется, создать его необходимо из шаблона *embedded_tomcat_server.xml.template*, иначе отредактировать файл.

Примечание. Описание файла настроек Tomcat приведено в разделе **Конфигурация встроенного Tomcat**.

Раскомментировать блок **Connector port="8443"**, прописать в нем атрибуты:

```
keystoreFile="../../server.pfx"
```

```
keystorePass="пароль к контейнеру server.pfx"
```

```
keystoreType="PKCS12"
```

```
keyAlias="azk server"
```

В файле свойств сервера приложений раскомментировать или добавить строку:

```
azk.tomcat.embedded.config=embedded_tomcat_server.xml
```

5. Если требуется клиентская аутентификация, необходимо создать контейнер с доверенными сертификатами, например с именем *mytruststore.jks* (в него добавляется сертификат клиента, сформированный ранее):

```
keytool -import -alias client -keyalg RSA -keystore mytruststore.jks -storetype JKS -file client.crt
```

Прописать в *embedded_tomcat_server.xml* в блоке настроек для **Connector port="8443"**

```
clientAuth="true"  
truststoreFile="..../mytruststore.jks"  
truststorePass="пароль к контейнеру mytruststore.jks"  
truststoreType="JKS"
```

6. Запустить сервер приложений, в логе появится строка:

```
ServletContainer,INFO,Http11Protocol,Starting ProtocolHandler ["http-bio-8443"]
```

7. Для подключения Win-клиента настроить файл *Azk2CInt.ini* блок **HTTP**.

8. Вместо *azkserver* указать реальное имя сервера:

```
URL=https://azkserver:8443/exec
```

Если используется библиотека *HttpLib=INDY*, необходимо прописать параметры:

- **CACertificate** – путь к сертификату CA *ca.crt*.
- **ClientCertificate** – путь к *client.crt*.
- **ClientPrivateKey** – путь к *clientkey.pem*.
- **ClientPrivateKeyPass** – пароль.

В **ClientCertificate** можно указать путь к файлу *client.pfx*, тогда **ClientPrivateKey** необязателен для заполнения.

Если пароль не указан, будет выводиться диалог для ввода пароля к приватному ключу.

Если кроме корневого центра сертификации в цепочке сертификатов есть промежуточные центры сертификации, все файлы центров сертификации объединяются в один файл, а в **CACertificate** указывается путь к объединенному файлу сертификатов.

Если используется библиотека *HttpLib=WINHTTP*, необходимо установить клиентский сертификат *client.pfx* в личные сертификаты и прописать параметр **ClientCertificate**. Вместо *root* указать *common name*, заданный при генерации сертификата:

```
ClientCertificate=CURRENT_USER\My\root
```



Обновление системы

Под процессом обновления системы «АЦК-Планирование» следует понимать комплекс мероприятий по обновлению файлов серверной и клиентской части, приведению БД в требуемое состояние, обновление отчетов.

Перед отправкой обновления на объект составляется пакет измененных файлов, создаются соответствующие списки файлов. Вместе с пакетом изменений поставляется инструкция по его применению, действуя по которой можно безошибочно провести обновление стенда.

6.1 Обновление сервера

Обновление серверной части со сборки на сборку производится путем копирования всех измененных файлов поверх существующих файлов: сервера приложений, библиотек, драйверов, SQL- и XML-скриптов.

Внимание! Сервер приложений при копировании файлов должен быть остановлен.

Итак, обновление серверной части выполняется следующим образом:

1. Остановить клиентскую и серверную части приложения.
2. Скопировать обновленные файлы в соответствующие каталоги «АЦК-Планирование».
3. Запустить `./sql/update.cmd(sh)`.

На экране в процессе проводимого обновления появляются обобщенные сообщения об операциях, осуществляемых в текущий момент. Например:

```
Update DB
=====
- drop triggers
- rollback old scripts
-perform new scripts
=====
Update FAILED! see .\update.log
```

Подробная расшифровка операций проводимого обновления записываются в файл, задаваемый в `update.cmd` переменной `LOG_FILE`. Например:

```
set LOG_FILE=.\\update.log
```


Если обновление прошло успешно, то в конечной строке на экране выдается сообщение *Update SUCCEEDED*. Если же при обновлении возникла ошибка, то сообщения закончатся строкой *Update FAILED! see .update.log*. О причине неудачного обновления можно узнать из log-файла.

4. Запустить сервер приложений **StartServer.bat**.
5. Выполнить новые XML-скрипты в рамках **@update.lst**.

Пример:

```
C:\2.55.0.1\XML\xml.cmd @update.lst
```

Выполнение скрипта должно закончиться без ошибок. Об успешном выполнении скрипта говорит ключевое слово «successful», в отличие от «error» в случае неудачи.

6. Перезапустить сервер приложений.

Примечание. Если в процессе обновления возникают нетипичные для обновления производимые действия, то об этом дополнительно говорится в прилагаемом к обновлению файлу *!readme.txt*.

Отдельно обновлять серверную часть, без обновления клиентской части категорически запрещается. Так как несогласованная деятельность клиентского и серверного приложения из разных версий сборок может привести к неправильной работе системы «АЦК-Планирование» и непредсказуемым последствиям.

6.2 Пакетное обновление БД

Чтобы привести БД в состояние, пригодное для использования в новой сборке, необходимо исполнить на ней те SQL-скрипты, которые были созданы с момента последнего обновления БД.

Новые скрипты исполняются последовательно, по порядку возрастания номеров.

Перед обновлением внимательно ознакомьтесь с содержанием описания проведения обновления. Возможно, необходимо исполнить сервисные скрипты или откатить некоторые из уже примененных скриптов.

Последние версии обновлений БД поставляются со списками файлов, в той последовательности, в которой они должны быть применены в БД.

Внимание! *Делайте резервное копирование БД перед проведением обновления. Подробнее о резервном копировании/восстановлении БД см. в документе «[БАРМ.00004-55 32 01-8 <% 32_01-8%>](#)».*

6.3 Обновление клиентской части

Обновление клиентской части со сборки на сборку производится путем копирования измененных файлов поверх существующих. При этом все клиентские приложения должны быть закрыты, иначе при копировании возникнет ошибка доступа к исполняемому файлу.

После копирования клиентская часть готова к работе. После первого запуска клиентского приложения на клиентском компьютере происходит автоматическая регистрация библиотек с клиентскими объектами. Ход выполнения процесса можно контролировать по панели прогресса.

Отдельно обновлять клиентскую часть, без обновления серверной части категорически запрещается. Так как несогласованная деятельность клиентского и серверного приложения из разных сборок может привести к неправильной работе системы «АЦК-Планирование» и непредсказуемым последствиям в конечном итоге.

6.4 Обновление системы для работы с веб-клиентом

Для работы веб-клиента помимо стандартных действий по установке/обновлению системы «АЦК-Планирование» необходимо выполнить действия:

- во время выполнения xml-скриптов выполнить скрипты, содержащиеся в файле [XML\@module.lst](#);
- выполнить xml-скрипты XML\Roles\pbs.xml и XML\Roles\pbsFullAccess.xml, содержащие описание функциональных ролей для ПБС.

Примечание. *Подробнее о выполнении xml-скриптов см. в разделе [XMLExecuter](#)¹⁴⁷.*

6.5 Автоматическое обновление клиентских компонент

Автоматическое обновление предназначено для обновления файлов клиентских приложений, расположенных на удаленной машине, в случае их изменения в эталонном дистрибутиве. В системе для этого предусмотрено два возможных пути обновления – через сервер приложений и через общую сетевую папку, к которой удаленный клиент может обращаться напрямую, минуя сервер.

Настройка модуля автоматического обновления осуществляется в следующем порядке:

1. Настройка сервера приложений. Осуществляется администратором системы, является обязательной.
2. Системные настройки программы. Осуществляется администратором системы или пользователем, который обладает правами доступа к данному пункту меню программы. Системные настройки являются едиными для всех пользователей системы.
3. Параметры пользователя системы. Настройка осуществляется администратором системы или пользователем, который обладает правами доступа к данному пункту меню программы. Данная настройка может быть как общей, так и индивидуальной для каждой клиентской машины, пользователя ОС Windows, префикса запуска. По умолчанию клиентское приложение настроено на работу согласно системным настройкам программы.

Подробное описание настроек системы и дополнительных возможностей модуля обновления приведены ниже.

6.5.1 Описание настроек системы для осуществления автоматического обновления системы

6.5.1.1 Настройки сервера приложений, влияющие на работу модуля обновления

Для возможности обновления клиента через сервер приложения в файле **Azk2Server.properties** необходимо заполнить параметр **azk.clientupdate.path**, находящийся в блоке «UPDATER клиентской части».

Параметр указывает путь от сервера приложений, к местоположению файлов, входящих в состав клиента. Для корректной работы путь необходимо прописать в формате Unix, т.е. обратные слэши должны быть удвоены. Важно, что это путь именно от сервера приложения (файлы не обязательно должны находиться на папке, отданной в свободный доступ) к каталогу, в котором будут храниться файлы клиентского приложения. Это может быть любой каталог сервера. Сервер должен иметь доступ на чтение к этому каталогу.

Примеры:

```
azk.clientupdate.path=\\server\azk2\Client\
```

```
azk.clientupdate.path=d:\\temp\client
```

6.5.1.2 Системные настройки

В сервисных параметрах (**Сервис**→**Системные параметры**) имеется ветвь **Обновление клиентской части**. На закладке **Настройки обновления** определяется способ обновления по умолчанию, действующий для всех пользователей системы:

- **Отключено**

При активизации данного параметра автоматическое обновление производиться не будет. Это значение является значением по умолчанию для данной настройки.

- **Через общую папку**

Параметр должен содержать путь от клиентских машин к файлам обновления, которые могут находиться как на основном сервере, так и на другом сетевом ресурсе. Поэтому указанный путь должен быть доступен с пользовательских машин, и у пользователей должен быть доступ к нему на чтение. Открывать данную папку на запись крайне не рекомендуется!

- **Через сервер приложения**

При активизации данного параметра при обновлении клиент будет обращаться к серверу приложений, обновление будет происходить через сервер, с использованием настройки указанной в разделе [Настройки сервера приложений, влияющие на работу модуля обновления](#)¹⁹⁵.

- **Через расширенный список**

При включении параметра обновление выполняется через одну из общих папок, указанных в поле **Список папок**. При указании в поле списка папок используется разделитель «;». В случае отсутствия доступного каталога обновление выполняется через сервер приложений.

- **Запретить игнорировать разницу в 1 ч.**

Для предотвращения сбоев обновления при переходе на летнее/зимнее время реализована функция игнорирования разницы в 1ч. и 2 сек. для обновления клиентского приложения. Параметр устанавливается для отмены игнорирования разницы.

На закладке **Настройки обновления** также для всех пользователей системы могут быть определены **Допустимые версии клиента для работы с текущим сервером**.

По умолчанию к серверу разрешено подключаться только клиенту версии, совпадающей с версией сервера. Если необходимо произвести подключение клиента версии, отличной от версии сервера, то в параметре **Допустимые версии клиента для работы с текущим сервером** через запятую указывается список версий, для которых подключение будет разрешено.

В случае несовпадения версии клиента и версии сервера, а также непопадания в список разрешенных версий в настройке, в системе проверяется наличие специального права «Позволять входить в систему с клиентской версией, отличной от допустимых сервером». Если специальное право имеется, то клиентское приложение формирует предупреждение о несовпадении версий. В случае отсутствия права подключения клиент не запускается.

В целях снижения нагрузки на сервер приложений и задействования других каналов сети Интернет для возможности переадресации обновлений на другие TCP соединения, на закладке **Сервер обновлений** указываются параметры подключения к выделенному серверу обновления. Формат настроек подключения соответствует формату *Azk2CInt.ini*.

На закладке **Описание ресурсов** в поле **Ресурсы**, указываются файлы, которые следует обновлять. Данный текст представляет собой XML, описывающий ресурсы обновления и файлы, входящие в состав ресурсов. По умолчанию автоматически проставляются данные из файла **sysparam.xml**. При изменении текста XML допустимо использование масок файлов (как *.OCX), но для файлов с расширением EXE маски использовать не рекомендуется. Маски также могут использоваться в именах каталогов.

6.5.1.3 Настройки клиентской части

В сервисных параметрах пользователя (**Сервис**→**Параметры пользователя**) имеется группа настроек **Обновление клиентской части**. Настройки группы могут быть как

общие, так и индивидуальные для каждой клиентской машины, пользователя ОС Windows, префикса запуска. Здесь задаются параметры:

- **При появлении новых версий файлов**

Данный параметр определяет режим работы модуля обновления. Допустимы следующие варианты:

- **Всегда обновлять** (рекомендуется) (при активизации данного параметра клиентское приложение будет периодически проверять появление новых версий файлов, и в случае обнаружения автоматически производить обновление);
- **Спрашивать** (при активизации данного параметра клиентское приложение будет периодически проверять появление новых версий файлов, и в случае их обнаружения, система автоматически будет формировать запрос о необходимости обновления, и далее действовать по выбору клиента);
- **Не проверять появление новых версий** (при активизации данного параметра система не будет проверять наличие новых версий файлов).

- **Способ обновления**

Данный параметр определяет, как будет проходить обновление клиентского приложения в случае появления новых версий файлов:

- **Согласно настройке сервера** (при активизации данного параметра клиентское приложение будет производить обновление согласно настройкам, указанным в [Системных параметрах](#)¹⁹⁶);
- **Через общую папку** (при активизации данного параметра клиентское приложение будет обращаться напрямую к удаленному ресурсу, где должны содержаться файлы для обновления);
- **Через сервер приложения** (при активизации данного параметра клиентское приложение будет обращаться к [серверу приложений](#)¹⁹⁵, и обновляться через настройки);
- **Через расширенный список** (при включении параметра обновление выполняется через одну из общих папок, указанных в поле **Список папок**. При указании в поле списка папок используется разделитель «;»).

6.5.2 Обновление клиента вручную

В случае необходимости произвести обновление клиента вручную, нужно запустить файл **updater.exe**. В результате откроется специальное окно обновления, в котором необходимо настроить следующие параметры:

- **Тип обновления**

- **Общая папка** – обновление будет осуществляться копированием файлов из каталога, указанного в строке **Полный путь**. При активном сервере приложения данный параметр будет взят из системной настройки Через общую папку (пункт меню **Сервис**→**Системные параметры**, раздел **Обновление клиентской части**, закладка **Настройки обновления**);
 - **Удаленное соединение** – обновление будет осуществляться, используя удаленный сервер приложения (при этом используется путь, указанный в **azk.clientupdate.path**);
 - **Расширенный список** – обновление будет осуществляться через одну из общих папок, указанных в поле **Список папок**.
- **Список ресурсов**

Список ресурсов, которые должна обновлять программа клиентского обновления. В данный момент доступны компоненты *Конфигурация подключения*, *Клиентское приложение* и *Отчеты*.

6.5.3 Рекомендации по переходу

После обновления на нужную версию подсистема обновления отключается по умолчанию:

- **Настройка типа обновления клиента** (пункт меню **Сервис**→**Параметры пользователя**, раздел **Обновление клиентской части**): *Согласно настройке сервера*,
- **Способ обновления** (пункт меню **Сервис**→**Системные параметры**, раздел **Обновление клиентской части**, закладка **Настройки обновления**): *Отключено*.

Во избежание излишней нагрузки на сервер приложений, рекомендуется использовать режим обновления – **Через общую папку**. Общая папка должна иметь доступ только на чтение!

Для активизации рекомендуемого режима необходимо установить в **группе настроек Способ обновления**¹⁹⁶ параметр **Через общую папку** и указать путь к общей папке, к которой имеют доступ на чтение все клиентские машины. После этого можно приступить к установке дистрибутива клиентской части на машины пользователей.

В системе также поддерживается одновременная работа части пользователей с локальных рабочих станций, а другой части с общей папки. Таким образом, установка дистрибутива клиента «АЦК-Планирование» на рабочие станции всех пользователей не является необходимым условием для начала работы с удаленным обновлением.

Режим обновления **Через сервер приложения** рекомендуется устанавливать при инсталляции удаленных клиентов, которые не имеют доступа к ресурсам локальной сети ФО.

При выборе режима обновления **Через расширенный список** должно выполняться следующее требование. При наличии нескольких локальных сетей, взаимодействующих с сервером приложений, наименование каждой общей папки должно быть **уникальным в рамках одной локальной сети**. Для обеспечения уникальности в качестве наименования папки можно использовать **GUID**. При наличии нескольких локальных сетей, взаимодействующих с сервером приложений, желательно, чтобы были одинаковыми:

- названия компьютеров, на которых находятся общие папки;
- названия общих папок.

6.6 Обновление отчетов

Обновление основных отчетов производится одновременно с выполнением процедуры обновления сервера, при выполнении скрипта **@update.lst** (путем вызова **\module\bft.reports@reports.lst**). Для отдельного пролития основных отчетов производится вызов на исполнение XML-скрипта: **@azk2_rep.lst**.

Например:

```
C:\2.55.0.1\XML\module\bft.reports\xml.cmd @azk2_rep.lst
```

6.6.1 Обновление специальных отчетов

Для обновления специальных отчетов (индивидуальных отчетов для каждого объекта) нужно выполнить следующее:

1. Предварительно обновить версию основного отчетного билда.
2. Выполнить XML-скрипты в рамках **@azk2rep_[ObjectName].lst** из каталога объекта (располагаются в папках **\XML\module\bft.reports\[ObjectName]**, где **[ObjectName]** – имя соответствующего объекта).

Например:

`C:\2.55.0.1\XML\module\bft.reports\xml.cmd`

`C:\2.55.0.1\XML\Rostov\@azk2rep_Rostov.lst`

Примечание. Если новые отчеты не доступны в главном меню «АЦК-Планирование», то администратор может добавить новый пункт в меню для вызова отчета. Создание меню и добавление пунктов меню рассмотрено в документации «[БАРМ.00004-55 32 01-3 Система «АЦК-Планирование»](#). АРМ ФО. Блок администрирования. Подсистема администрирования. Настройка и сервис системы. Руководство администратора».



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.

7.1 Регламентно-профилактические мероприятия

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

7.1.1 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 29 – 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"> • <i>StartServer.bat/StartServer.sh</i> (в зависимости от ОС); • <i>Azk2Server.properties, Server.properties</i>; • <i>Azk2logback.xml</i>; • <i>lconflserver.xml</i> (для веб-сервера). 	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 XMX, XMS, сборщик мусора UseG1GC, интервал его запуска MaxGCPauseMillis.</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 webapps, work, temp и копирование <i>azk.war</i> в каталог webapps 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, Out of memory, sql</i> -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.

7.1.1.1 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. 1-9²⁰⁵](#)).
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.

7.1.1.2 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.

7.1.1.3 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.

7.1.1.4 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.

7.1.2 Регламентно-профилактические мероприятия по СПО

Регламентно-профилактические мероприятия по СПО включают в себя следующие действия, выполняемые с определенной периодичностью:

Таблица 30 – 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"> • <i>server.xml</i>, • <i>nginx.conf/httpd.conf/httpd2.conf/apache.conf</i>. 	До и после выполнения процедуры обновления сборки системы/изменения количества серверов приложений.
2.	Резервирование конфигурационных файлов (для nginx – nginx.conf , для apache – httpd.conf/httpd2.conf/apache.conf , для tomcat – server.xml) на физическом отдельном диске.	Периодически и перед выполнением процедуры обновления/реконфигурации сборки/серверов/гипервизоров/СХД.
3.	Мониторинг загрузки компонентов JVM.	Постоянно.
4.	Мониторинг балансировщиков (Nginx: connections accepted per sec, connections reading, connections waiting, connections writing, connects active, Number of processes nginx, requests per sec).	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 sysctl.conf, limits.conf, security.java .	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.

№	Мероприятия	Периодичность выполнения
10.	Мониторинг системных журналов, содержащих информацию об успешности авторизации для выявления угроз безопасности, хакерских атак методом перебора и др..	Постоянно.

7.1.3 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.

7.1.4 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.

7.1.5 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.

7.1.5.1 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.

7.1.5.2 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.

7.1.5.3 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.

7.1.6 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.

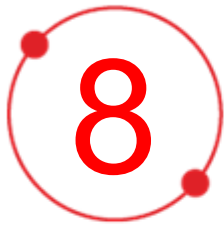
После настройки и изменения конфигураций аппаратного обеспечения проверяется работоспособность аппаратного обеспечения системы (подсистемы) и в случае их неработоспособности, восстановлена существовавшая ранее работоспособная конфигурация. Проверка работоспособности выполняется по аналогии с ежедневной проверкой работоспособности.

7.1.7 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.

8.1 Сценарий взаимодействия с Клиентским приложением

1. При запуске Клиентского приложения в АРМ запускается Web-браузер, происходит переадресация на страницу логина в FAM, где вводится логин и пароль/сертификат.
2. В случае успешного ввода FAM возвращает Web-браузер на клиентском АРМ ссылку для запуска клиентского приложения, ссылка содержит код авторизации.
3. Запускается клиентское приложение, создается сессия пользователя, код авторизации передается на сервер приложений, сервер приложений получает от FAM Access Token (одноразовый код доступа) и Refresh Token (код обновления

доступа) по коду авторизации, после получения Access Token (одноразовый код доступа) пользователю предоставляется доступ к работе в клиентском приложении.

4. В случае, если Access Token истекает, сервер приложений запрашивает в FAM новый Access Token (одноразовый код доступа) по Refresh Token (код обновления доступа).
5. В случае, если Refresh Token (код обновления доступа) истек, пользовательская сессия завершается (LogOut), и пользователь принудительно выходит из системы в клиентском приложении, оно прекращает работу на клиентском АРМ с соответствующим сообщением.

При запуске Клиентского приложения, в случае, если пользователь уже вошел под своим логином и работает в Web-клиенте или другом клиентском приложении:


1. Запускается браузер, пользователь перенаправляется в FAM.
2. FAM по информации в Cookie браузера определяет, что пользователь уже вошел под своим логином, поэтому форма ввода логина и пароля/сертификата пользователю не отображается, в браузер возвращается ссылка для запуска клиентского приложения.
3. Далее запускается клиентское приложение, создается сессия пользователя, код авторизации передается на сервер приложений, сервер приложений получает от FAM Access Token (одноразовый код доступа) и Refresh Token (код обновления доступа) по коду авторизации, после получения Access Token (одноразовый код доступа) пользователю предоставляется доступ к работе в клиентском приложении.
4. В случае, если Access Token (одноразовый код доступа) истекает, сервер приложений запрашивает в FAM новый Access Token (одноразовый код доступа) по Refresh Token (код обновления доступа).
5. В случае, если Refresh Token (код обновления доступа) истек, пользовательская сессия завершается (LogOut), и пользователь принудительно выходит из системы в клиентском приложении, оно прекращает работу на клиентском АРМ с соответствующим сообщением.

8.2 Сценарий взаимодействия с Web-клиентом

1. При запуске Web-клиента в АРМ, в системе создается пользовательская сессия, пользователь переадресуется на страницу FAM, где вводит логин и пароль/сертификат.
2. В случае успешного ввода логина и пароля FAM переадресует пользователя обратно в Web-приложение, в Redirect URI передается код авторизации.
3. Код авторизации передается на сервер приложений, сервер приложений получает от FAM Access Token (одноразовый код доступа) и Refresh Token (по коду авторизации, после получения Access Token (одноразовый код доступа) пользователю предоставляется доступ к работе в Web-приложении).
4. В случае, если Access Token (одноразовый код доступа) истекает, сервер приложений запрашивает в FAM новый Access Token (одноразовый код доступа) по Refresh Token.
5. В случае, если Refresh Token истек, пользовательская сессия завершается (LogOut), пользователь принудительно выходит из системы.

При запуске Web-приложения, в случае, если пользователь уже работает в Клиентском приложении:

1. Пользователь переадресуется в FAM.
2. FAM по информации в Cookie браузера определяет, что пользователь уже вошел под своим логином, поэтому форма ввода логину и пароля/сертификата пользователю не отображается, пользователь переадресуется в Web-клиент с кодом авторизации.
3. Далее код авторизации передается на сервер приложений, сервер приложений получает от FAM Access Token (одноразовый код доступа) и Refresh Token (по коду авторизации, после получения Access Token (одноразовый код доступа) пользователю предоставляется доступ к работе в Web-приложении).
4. В случае, если Access Token (одноразовый код доступа) истекает, сервер приложений запрашивает в FAM новый Access Token (одноразовый код доступа) по Refresh Token.
5. В случае, если Refresh Token истек, пользовательская сессия завершается (LogOut), пользователь принудительно выходит из системы.



9

Описание команд ОС Linux

В этом разделе описываются некоторые команды ОС Linux, которые могут использоваться при установке СУБД, а также при настройке системы «АЦК-Планирование».

Команды навигации по файловой системе:

- *ls* <маска файлов> – выдать список файлов в текущем каталоге соответствующих маске (если указана), ключи:
 - l – подробный формат.
 - a – вывод скрытых файлов.
- *cd* <путь> – смена текущего каталога.
- *pwd* – вывод текущего положения в файловой системе.

Работа с файлами:

- *cp* <что_копировать> <куда_копировать> – копировать файл(ы).
- *mv* <что_перемещать> <куда_перемещать> – переместить или переименовать файл.
- *ln -s* <на_что_сделать_ссылку> <имя_ссылки> – создать символическую ссылку.
- *rm* <файлы> – удалить файлы.
- *rm -r* <файлы/каталоги> – рекурсивное удаление каталогов.
- *mkdir* <каталог> – создать новый каталог.
- *rmdir* <каталог> – удалить пустой каталог.
- *less* <файл> – просмотр содержимого файла.
- *mount* <что_монтировать> <куда_монтировать> – монтирование устройств в файловую систему.

Сеть:

- *ifconfig* <интерфейс> <down/up> – старт/остановка сетевого интерфейса.
- *ping* <адрес> – проверка доступности сетевого адреса.
- *traceroute* <адрес> – проверка сетевого маршрута.

Установка/удаление пакетов:

- *rpm* – установка/удаление системных пакетов, ключи:

-i <имя файла> – инсталляция.

-e <имя пакета> – удаление.

-q <имя пакета> – запрос (содержимого и т.п.).

Прочее:

- *crontab* <имя файла> – активация нового файла расписания заданий.
- *chkconfig* – установка/удаление сервиса в системе, ключи:
 - add <имя файла> – установка сервиса.
 - del <имя файла> – удаление сервиса.
 - list – показ списка сервисов.
- *man* <команда> – вывод справки по команде.
- *top* – интерактивный список текущих процессов, отсортированных по использованию центрального процессора.
- *uname -a* – вывести информацию о версии операционной системы.
- *free* – вывести информацию по использованию памяти.
- *df -h* – вывести информацию о свободном и используемом месте на дисках.
- *echo* <строка> – вывод строки/содержимого переменной на экран.
- *grep* – фильтрация выводимой информации (используется в сочетании с другими командами).

Переменные системы:

- *export* <переменная> = <значение> – установка значения переменной окружения.
- *set* – просмотр значений всех переменных окружения.

Управление правами и пользователями:

- *useradd* <имя пользователя> – добавление пользователя в систему.
- *userdel* <имя пользователя> – удаление пользователя из системы.
- *groupadd* <имя группы> – добавление группы в систему.
- *groupdel* <имя группы> – удаление группы из системы.

- *passwd* <имя пользователя> – установка пароля пользователя.
- *chgrp* <имя файла/каталога> – смена группы-владельца файла/каталога, ключи:
 - R – рекурсивная смена группы-владельца.
- *chown* <имя файла/каталога> – смена владельца файла/каталога, ключи:
 - R – рекурсивная смена владельца.
- *chmod* <опции режима> <имя файла/каталога> – смена режима доступа к файлу/каталогу, ключи:
 - R – рекурсивная смена режима.
- *su* - <имя пользователя> – смена текущего пользователя.

Архивы:

- *tar -x* <имя файла> – распаковка tar-архива.
- *gzip -dr* <имя файла> – распаковка gz-архива.
- *cpio -idmv* <<имя файла> – распаковка cpio-архива.

Управление процессами:

- *ps* – просмотр списка процессов, ключи:
 - A – список всех процессов.
 - f – подробная информации.
- *kill* <идентификатор процесса> – запрос на остановку заданного процесса, ключи:
 - 9 – принудительное завершение процесса.



10

Отслеживание изменений системных объектов

При совместном администрировании разными администраторами имеется возможность отслеживания изменений системных объектов с целью получения полной информации об изменениях.

Для включения механизма отслеживания необходимо выполнить XML-скрипт **historymodify.xml**.

XML-скрипт должен иметь следующую структуру:

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

В разделе **CDATA** указывается перечень отслеживаемых параметров.

где,

SYSPARAM – системные и отчетные параметры.

Для параметра можно указать режим отслеживания изменений:

- **SYSPARAM = 0** или **SYSPARAM** – не запрашивать комментарий при сохранении измененного объекта;
- **SYSPARAM = 1** – запрашивать комментарий, но он не является обязательным;
- **SYSPARAM = 2** – запрашивать комментарий, выдавать ошибку и не сохранять изменения без введенного комментария.

Для вызова окна просмотра истории изменений системных параметров необходимо:

1. Открыть окно настройки параметров через пункт меню **Сервис**→**Системные параметры** или **Отчеты**→**Отчетные параметры**.
2. Внести изменения в настройки и нажать кнопку **Применить**.
3. При необходимости указать причину изменения настроек в окне ввода комментария. Наличие окна определяется настройкой режима отслеживания в файле **historymodify.xml**.

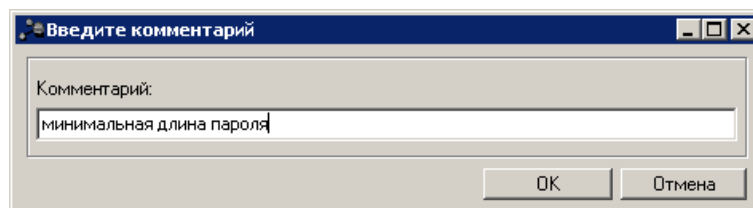


Рисунок 12 – Окно ввода комментария

4. Нажать сочетание клавиш **Ctrl+F6**. Откроется окно просмотра истории изменений.

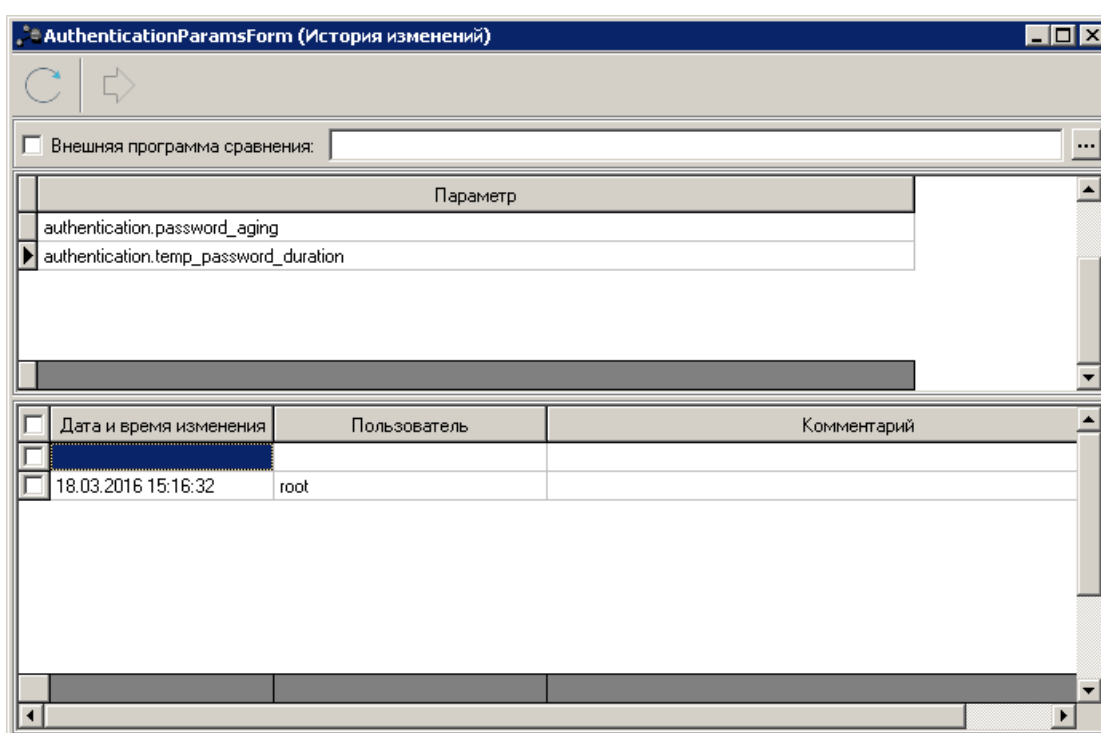


Рисунок 13 – Окно просмотра истории изменений системных параметров

На панели инструментов располагаются функциональные клавиши, с помощью которых можно обновить окно и вызвать окно сравнения измененных параметров.

Для сравнения параметров:

1. В списке *Параметр* выбирается нужный параметр. В таблице отобразится список записей с историей изменения этого параметра с указанием даты изменения, пользователя и его комментария.
2. В таблице отмечается пара записей. При этом становится доступна кнопка панели инструментов.

3. Нажимается кнопка  или сочетание клавиш **Ctrl+D**. Откроется окно:

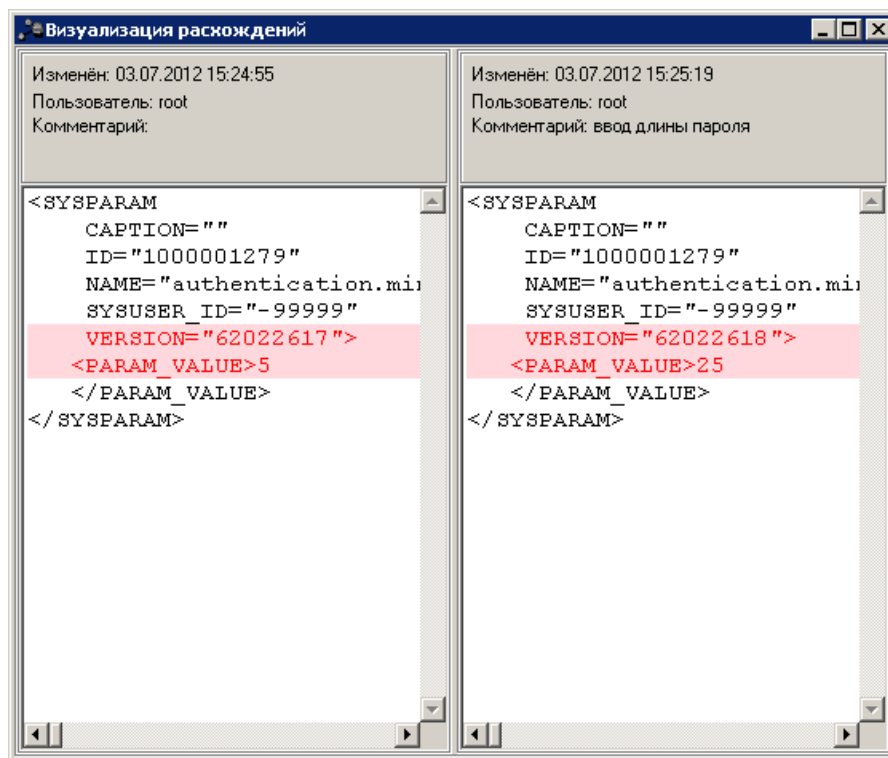



Рисунок 14 – Окно визуализации расхождений

В окне красным цветом отмечены отличающиеся строки.

Для истории изменения параметра с помощью другой программы сравнения необходимо:

1. Выбирать в таблице пару записей.
2. На панели инструментов включить галочку **Внешняя программа сравнения** и с помощью кнопки  указать путь к файлу программы.



11

Устранение блокировок по коду цели

При возникновении ситуации с блокировкой резервирующих проводок по коду цели по техническими причинам (потеря связи с БД, остановка сервера приложений) предусмотрена ручная возможность их разблокировки. Для этого предусмотрен сервис, который запускается заданием планировщика `PurposefulGrantCleanupTask`.

Примечание. Как правило, сервис запускается автоматически при следующем обращении к проводочному движку, если ошибка возникла на этом сервере приложений, а также при первом обращении к проводочному движку – после старта сервера приложений.

Предусмотрена возможность отключения сервиса на некоторых серверах приложений, для этого нужно задать свойство сервера `azk.purposefulgrant.service.disabled=true`.

Запуск задания `PurposefulGrantCleanupTask` следует осуществлять на одном сервере приложений, причем сервис на этом сервере не должен быть отключен.

НАШИ КОНТАКТЫ

Звоните:

(495) 784-70-00

Пишите:

bft@bftcom.com

Будьте с нами online:

www.bftcom.com

Приезжайте:

129085, г. Москва,
ул. Годовикова, д. 9, стр. 17

Дружите с нами в социальных сетях:



vk.com/bftcom



t.me/ExpertBFT_bot

