



ООО «БФТ»

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

ined@bftcom.com
bftcom.com

Утвержден

БАРМ.00004-56 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.00004-56 32 01-1

Листов 400

© 2024, 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» – 2.56.0. Последние изменения внесены 28.06.2024 г.

СОДЕРЖАНИЕ

1	Общие сведения о программе.....	12
1.1	Назначение программы.....	14
1.2	Системные требования.....	14
1.2.1	Требования к серверам.....	14
1.2.1.1	Общие требования.....	14
1.2.1.2	Сервер приложений системы АЦК.....	15
1.2.1.3	Сервер ЭП	16
1.2.1.4	Сервер базы данных АЦК.....	18
1.2.1.5	Совмещенный сервер приложений и сервер базы данных.....	20
1.2.1.6	Web-сервер	20
1.2.1.7	Сервер приложения для взаимодействия с ЕИС.....	21
1.2.2	Требования к клиентской части.....	21
1.2.2.1	Общие требования.....	21
1.2.2.2	Windows-клиент.....	22
1.2.2.3	Web-клиент	23
1.2.2.4	Требования к программному окружению при использовании криптографических функций.....	23
1.2.2.5	Требования к программному окружению при использовании протокола HTTPS и алгоритмов шифрования ГОСТ.....	24
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.....	25
1.2.3.3	Линии связи между сервером приложений и Windows-клиентом.....	25
1.2.3.4	Линии связи между сервером приложений и WEB-клиентом.....	25
1.2.3.5	Линии связи для сервера ЭП (онлайн-соединение).....	25
1.2.4	Требования к безопасности.....	26
1.2.4.1	Демилитаризованная зона (DMZ).....	26
1.2.5	Требования к офисным приложениям.....	27
1.2.6	Ограничения.....	37
2	Структура программы.....	41
2.1	Сервер приложений.....	43
2.1.1	Компоненты сервера приложений.....	43

2.1.2	Утилиты для работы с данными. XML-скрипты.....	46
2.1.3	Утилиты для работы со структурой БД. SQL-скрипты.....	46
2.1.4	Файл свойств сервера приложений.....	48
2.1.4.1	Свойства базы данных.....	48
2.1.4.2	Свойства лога сервера приложений.....	51
2.1.4.3	Свойства управления заданиями сервера.....	54
2.1.4.4	Управление пользовательскими сессиями.....	55
2.1.4.5	Свойства сетевых параметров сервера.....	56
2.1.4.6	Свойства сетевых параметров СУБД PG.....	58
2.1.4.7	Настройки пула потоков.....	59
2.1.4.8	Свойства репликации.....	60
2.1.4.9	Логирование версий записей.....	60
2.1.4.10	OpenORB	60
2.1.4.11	Управление вложениями в документах.....	60
2.1.4.12	Свойства системы автоматического обновления.....	61
2.1.4.13	Настройка репликации нового сайта.....	64
2.1.4.14	Настройка параметров ЭП.....	64
2.1.4.15	Настройка параметров лицензии.....	65
2.1.4.16	Настройка сервиса бюджетного мониторинга.....	65
2.1.4.17	Поддержка HTTP-соединения для Win-клиента.....	66
2.1.4.18	Настройка кластерного режима работы.....	66
2.1.4.19	Настройка взаимодействия с ЕИС.....	67
2.1.4.20	Настройка взаимодействия с системой Аванпост.....	68
2.1.4.21	Настройка хранилища ключей доверенных сертификатов.....	69
2.1.4.22	Настройка прокси-сервера по протоколу https.....	70
2.1.4.23	Кэширование профилей на сервере АЦК.....	70
2.1.4.24	Настройка сервиса помощи входа по сертификату (СПВС).....	71
2.1.4.25	Настройка сервиса передачи сообщений при формировании отчетов по Приказу 243н.....	71
2.2	Система лицензионной защиты.....	72
2.3	Клиентское приложение.....	73
2.3.1	Описание клиентского приложения.....	73
2.3.2	Параметры командной строки клиента.....	76
3	Основные понятия	78
3.1	Клиентские объекты, процессоры, провайдеры.....	79

3.2	Бюджет	81
3.3	Справочники	81
3.4	Документы. Редактор сценариев обработки документа.....	82
3.5	Репликация. Объекты репликации. Реплицируемые таблицы.....	82
3.6	Сайты. Типы соединений сайтов.....	84
3.7	Сообщения. Очереди сообщений.....	85
3.8	Планировщик. Задания планировщика. Расписание запуска заданий.....	88
3.8.1	Макросы для параметров begin_date и end_date.....	89
3.8.2	AssignExport.....	90
3.8.3	AttachMoverLauncher.....	91
3.8.4	AutoControlProductPlanNotControlEIS.....	92
3.8.5	AutoEBUnloadLauncher.....	93
3.8.6	Backup DB	94
3.8.7	Backuper	96
3.8.8	Bank Update ED807.....	96
3.8.9	CADDownloader.....	97
3.8.10	CassaBUAUExportPlan.....	98
3.8.11	CertDownloader.....	99
3.8.12	Consistency check.....	100
3.8.13	ConsRepLauncher.....	100
3.8.14	ControlEventFactValueProcessor.....	101
3.8.15	ControlExecMessenger.....	101
3.8.16	ControlResultsInEISUploader.....	101
3.8.17	CreateNotifyMsg.....	102
3.8.18	CRLDownloader.....	103
3.8.19	CurdateSynchronizer.....	103
3.8.20	DataObjectChangeLogCleaner.....	104
3.8.21	DBCloneActivate.....	105
3.8.22	DBCloneBackuper.....	105
3.8.23	DBCloneCheck.....	106
3.8.24	DBCloneStatistical.....	107
3.8.25	DebitCardLauncher.....	108

3.8.26	DeleteNotifyMsg.....	108
3.8.27	DeliveryFactDocLauncher.....	108
3.8.28	DocRetentor.....	109
3.8.29	DocumentCleaner.....	110
3.8.30	EisRequestProtocolLoading.....	111
3.8.31	Exchange	112
3.8.32	FhdPlanStructTemplateImport.....	115
3.8.33	FORepLauncher.....	115
3.8.34	GZFactDocImport.....	115
3.8.35	ImportBudgServiceFed2.....	117
3.8.36	ImportDictionariesUI.....	118
3.8.37	ImportDocFromEis.....	126
3.8.38	ImportDocLauncher.....	127
3.8.39	ImportEhdDictionaries.....	128
3.8.40	ImportFinancingIndicatorsEHD	132
3.8.41	ImportFinancingIndicatorsEHDUI.....	134
3.8.42	ImportFedProjects.....	138
3.8.43	ImportRegProjCard.....	156
3.8.44	ImportTaskProjects.....	157
3.8.45	ImportUgpSubjectBroadcast.....	158
3.8.46	ImportUnitProcessor.....	201
3.8.47	KIExport	207
3.8.48	MailCleaner	208
3.8.49	MessageControlEvent.....	208
3.8.50	Messenger	209
3.8.51	MsgCleaner.....	210
3.8.52	MsgRepeater.....	211
3.8.53	NoteToChangePasswTask.....	213
3.8.54	NotifyMsgSender.....	213
3.8.55	NotificationSenderTask.....	213
3.8.56	OkpdImport	214
3.8.57	OkvedImport.....	214
3.8.58	OkvInfoImport.....	215

3.8.59	OosExport	227
3.8.60	PlanAssExport.....	227
3.8.61	PlanInExport.....	228
3.8.62	PlanInReflmport.....	228
3.8.63	PlanSrcExport.....	229
3.8.64	PlanXMLExport.....	230
3.8.65	PIBorExport.....	230
3.8.66	PIStateEventDocExport.....	230
3.8.67	PIStateProgDocExport.....	231
3.8.68	PIStateSubProgDocExport.....	232
3.8.69	ProcessGrantsReducedFin.....	232
3.8.70	ProgDocExport.....	234
3.8.71	PurposefulGrantCleanupTask.....	234
3.8.72	RegionalProjectCardEhdExportProcessor.....	235
3.8.73	RegProjActualityStatusUpdater.....	243
3.8.74	RelmportFailedImportedDocs.....	244
3.8.75	Replication	245
3.8.76	ReportingCompanyRPPProcessor.....	247
3.8.77	ResultAmountCheck.....	247
3.8.78	RplServiceLogCleaner	251
3.8.79	RplGenerationStatusTask.....	251
3.8.80	RtRplRunner.....	252
3.8.81	SchExpDocs	252
3.8.82	Session cleaner.....	253
3.8.83	SQLTracer	254
3.8.84	SSLTrustManager.....	254
3.8.85	Statistics	255
3.8.86	STORAGE_COMPRESSOR.....	255
3.8.87	SyseventCleaner.....	255
3.8.88	SysInfoShower.....	256
3.8.89	TransactMonitorFlusher.....	257
3.8.90	Transport	257
3.8.91	UnDocLauncher.....	260

3.8.92	UniAnalyticalReportDeleteCompleted.....	261
3.8.93	UniAnalyticalReportUpdate.....	261
3.8.94	UniAnalyticalTaskProcessor.....	261
3.8.95	UniContractProcessor.....	261
3.8.96	UpdateRegionalListRegNum.....	263
3.8.97	UploadOKSFromEIS.....	264
3.8.98	Validation	265
3.9	Выполнение XML-заданий.....	265
3.10	Xml для методики создания бюджета 2022 года.....	266
4	База данных	267
4.1	Основные принципы построения БД.....	268
4.2	Основные принципы работы с БД.....	269
4.2.1	SQLExecutor.....	270
4.2.1.1	Назначение SQLExecutor. Параметры запуска. Файл конфигурации	270
4.2.1.2	SQL-скрипт. Назначение, типы, нумерация, структура, синтаксис..	271
4.2.1.3	Утилиты запуска sql-скриптов.....	273
4.2.2	XMLExecutor.....	274
4.2.2.1	Назначение, параметры запуска.....	274
4.2.2.2	XML-скрипт. Структура, назначение.....	276
4.2.2.3	XML-скрипты системы. Описание, зависимости запуска.....	277
4.2.2.4	Утилиты запуска XML-скриптов.....	280
4.2.3	Утилиты для работы с БД.....	281
4.2.3.1	Описание утилиты executor.cmd(sh).....	281
4.2.3.2	Утилита DBUpdate.cmd(sh).....	284
4.2.3.3	Утилита convert.cmd.....	285
4.2.3.4	Утилита comparer.cmd.....	285
4.2.3.5	Утилита update.cmd(sh).....	285
4.2.3.6	Создание новой БД.....	286
4.2.3.7	Резервное копирование/восстановление БД.....	287
5	Установка и настройка системы	288
5.1	Установка и настройка сервера приложений.....	289
5.1.1	Установка и настройка JRE.....	289

5.1.1.1	Настройка Java Cryptography Extension (JCE).....	291
5.1.2	Запуск и остановка сервера приложений. Файл конфигурации.....	292
5.2	Установка и настройка сервера БД.....	293
5.3	Подготовка БД к работе.....	293
5.3.1	Методика приведения БД к нужной версии.....	294
5.4	Установка, настройка и запуск клиентской части.....	294
5.5	Установка сборки отчетов.....	297
5.6	Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) учреждениях.....	297
5.6.1	Порядок предоставления справочной информации.....	301
5.6.2	Заполнение полей справочника «Организации» для интеграции с ЕИС..	302
5.6.3	Обработка приложенных документов при организации обмена с ЕИС....	306
5.6.4	Присоединение файлов к справочникам.....	307
5.6.5	Присоединение файлов к документу.....	309
5.7	Настройка соединения Win-клиента через зашифрованный канал https.....	315
5.8	Установка и настройка Nginx.....	327
5.8.1	Предварительная работа.....	327
5.8.2	Установка Nginx.....	328
5.8.3	Первичные настройки Nginx.....	329
5.8.4	Настройка «липких» сессий.....	335
5.8.4.1	Nginx-sticky-module.....	335
5.8.4.2	Настройка sticky session.....	340
5.8.4.3	Аргументы модуля.....	341
5.8.5	Установка Nginx Vhost Traffic Status.....	341
5.8.5.1	Nginx-module-vts-master.....	341
5.8.5.2	Настройка статистики.....	342
5.8.6	Ускорение работы nginx.....	343
5.8.7	Подключение SSL.....	347
5.8.8	Окончательный вид конфигурационных файлов.....	348
5.9	Сервис помощи входа по сертификату (СПВС).....	354
5.9.1	Методы сервиса.....	355
5.9.2	Описание JSON-типов.....	358

6	Обновление системы.....	360
6.1	Обновление сервера.....	361
6.2	Пакетное обновление БД.....	362
6.3	Обновление клиентской части.....	363
6.4	Обновление системы для работы с веб-клиентом.....	363
6.5	Автоматическое обновление клиентских компонент.....	364
6.5.1	Описание настроек системы для осуществления автоматического обновления системы.....	364
6.5.1.1	Настройки сервера приложений, влияющие на работу модуля обновления.....	364
6.5.1.2	Системные настройки.....	365
6.5.1.3	Настройки клиентской части.....	366
6.5.2	Обновление клиента вручную.....	367
6.5.3	Рекомендации по переходу.....	368
6.6	Обновление отчетов.....	369
6.6.1	Обновление специальных отчетов.....	369
7	Порядок технического обслуживания.....	371
7.1	Регламентно-профилактические мероприятия.....	372
7.1.1	Регламентно-профилактические мероприятия по ППО.....	372
7.1.1.1	Установка эталонной сборки.....	374
7.1.1.2	Параметры распределения JAVA памяти.....	375
7.1.1.3	Параметры настроек в конфигурационных файлах АЦК.....	376
7.1.1.4	Использование утилиты «jvisualvm».....	379
7.1.2	Регламентно-профилактические мероприятия по СПО.....	380
7.1.3	Мониторинг виртуальной инфраструктуры.....	381
7.1.4	Наличие тестового СП.....	381
7.1.5	Мониторинг структуры системы хранения данных.....	382
7.1.5.1	Оценка производительности СХД.....	382
7.1.5.2	Повышенный Service Time СХД.....	382
7.1.5.3	Утилизация CPU СХД.....	383
7.1.6	Проверка работоспособности.....	383
7.1.7	Проверочное восстановление резервной копии.....	385

8	Аванпост FAM.....	386
8.1	Сценарий взаимодействия с Клиентским приложением.....	387
8.2	Сценарий взаимодействия с Web-клиентом.....	389
9	Описание команд ОС Linux.....	391
10	Отслеживание изменений системных объектов.....	395
11	Устранение блокировок по коду цели.....	399



Общи сведения о программе



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. 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.

Минимальное количество аппаратных разъемов (сокетов) ЦПУ для всех серверов: 2 шт.

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	RedOS 7.1 / 7.2 / 7.3; Alt Linux 9.1; Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43; РОСА «Кобальт» 7.3 (3.10); SUSE Linux 11; CentOS 7; Windows Server 2008 R2 (64) / 2012 R2 / 2019 Standart (1809); Windows 7 (64) (только для малых объектов с количеством ПБС до 25).
	JDK (JRE)	Sun Java SE 8 update 201 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 ядра 3 ГГц Intel Xeon 64 bit	4 Гбайт	1 Гбит/с
	Усиленная (с доказательствами подлинности)	4 ядра 3 ГГц Intel Xeon 64 bit		

Общие сведения о программе
Системные требования

Максимальное количество пользователей, одновременно выполняющих проверку ЭП для группы ЭД (более 50)*	Вид используемой ЭП	Рекомендуемая конфигурация		
		CPU	RAM	Ethernet
до 10	Усиленная, Усиленная (XML), Усиленная (со штампом времени)	4 ядра 3 ГГц Intel Xeon 64 bit		
	Усиленная (с доказательствами подлинности)	8 ядер 3 ГГц Intel Xeon 64 bit		
до 20	Усиленная, Усиленная (XML), Усиленная (со штампом времени)	8 ядер 3 ГГц Intel Xeon 64 bit		
	Усиленная (с доказательствами подлинности)	16 ядер 3 ГГц 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	RedOS 7.1 / 7.2 / 7.3; Alt Linux 9.1; Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43; РОСА «Кобальт» 7.3 (3.10); SUSE Linux 11; Windows Server 2008 R2 (64) / 2012 R2; Windows Server 2019 Standart (1809)*; Windows 7* (64) (только для малых объектов с количеством ПБС до 25). Внимание! Некоторое программное обеспечение работает с ограничениями, см. раздел Ограничения ³⁷ .
	JDK (JRE)	Sun Java SE 8 update 201 и выше
	ЭЦП	<ul style="list-style-type: none"> • КриптоПро CSP 4.0, 5.0. Опционально: КриптоПро TSP Client, КриптоПро OCSP Client. <ul style="list-style-type: none"> • ViPNet CSP версии 4.2. • КриптоПро JCP 2.0. Для следующих видов ЭП: Усиленная (64Б), Усиленная (XML), Усиленная (со штампом времени).

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:

Общие сведения о программе
Системные требования

Таблица 5 – Системные требования к конфигурации сервера СУБД

Тип сервера	Тип ПО	Программное окружение
Сервер СУБД	ОС в соответствии с требованиями к СУБД	<p>RedOS 7.1 / 7.2 / 7.3; Alt Linux 9.1; Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43; РОСА «Кобальт» 7.3 (3.10); Oracle Enterprise Linux 6 / 7; RHEL 6, 7; SUSE Linux 11; CentOS 7; Windows Server 2008 R2 (64) / 2012 R2 / 2019 Standart (1809) (</p> <p>Внимание! По результатам нагрузочного тестирования ОС настоятельно не рекомендуется использовать с СУБД PostgreSQL, Postgres Pro при количестве онлайн-пользователей 250 и более);</p> <p>Windows 7* (64) (только для малых объектов с количеством ПБС до 25).</p> <p>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения³⁷.</p>
	СУБД Oracle	<p>Oracle 11g (рекомендуемая 11.2.0.4) Oracle 12c (рекомендуемая 12.1.0.2, 12.2 не поддерживается)</p>
	СУБД Firebird	Firebird-2.5.2
	СУБД Postgres Pro Standart	<p>Внимание! по результатам нагрузочного тестирования СУБД настоятельно не рекомендуется использовать с ОС Windows.</p> <ul style="list-style-type: none"> • Postgres Pro Standart 11.16. • Postgres Pro Standart 14, версия 14.6 и выше (рекомендуемая).
	СУБД Postgres Pro Certified	<p>Внимание! по результатам нагрузочного тестирования СУБД настоятельно не рекомендуется использовать с ОС Windows.</p> <ul style="list-style-type: none"> • Postgres Pro Certified 11.11.1. • Postgres Pro Certified 14, версия 14.6 и выше (рекомендуемая).
	СУБД PostgreSQL	<p>Внимание! по результатам нагрузочного тестирования СУБД настоятельно не рекомендуется использовать с ОС Windows.</p> <ul style="list-style-type: none"> • PostgreSQL 11, версия 11.8 и выше. • PostgreSQL 12, версия 12.7 и выше. • PostgreSQL 13, версия 13.8 и выше (рекомендуемая). • PostgreSQL 14, версия 14.6 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.

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	RedOS 7.1 / 7.2 / 7.3 Alt Linux 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 РОСА «Кобальт» 7.3 (3.10) SUSE Linux 11 CentOS 7 Windows 7 (64) / 2008 (64) / 2008 R2 / 2019 Standart (1809) <i>Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения ³⁷.</i>
	JDK (JRE)	Sun Java SE 8 update 201 и выше
	Веб-сервер	Apache Tomcat 8.5.76 и выше, 9.0.59 и выше

В случае необходимости организации подключений к web-серверу из сети Интернет, необходимо, чтобы web-сервер располагался в безопасной части сети – «демитаризованной зоне»^[26].

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	RedOS 7.2 / 7.3 Alt Linux 9.1 Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43 РОСА «Кобальт» 7.3 (3.10) SUSE Linux 11 клиентские ОС: Windows 7
	JDK (JRE)	Sun Java SE 8 update 201 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 ГГц; Память: 2 Гбайт; Диск: 10 Гбайт (свободно); Монитор и карта: 1024x768; Сеть: 10 Мбит/с; ОС: RedOS 7.1, Alt Linux 9.1, Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43, РОСА «Кобальт» 7.3 (3.10); Windows 7 / 8 / 10. ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ^[27] .	Процессор: Intel Core 2 Duo; Память: 4 Гбайт; Диск: 20 Гбайт (свободно); Монитор и карта: 1280x1024 или 1440x900; Сеть: 100 Мбит/с и выше; ОС: RedOS 7.1 / 7.2 / 7.3, Alt Linux 9.1, Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43, РОСА «Кобальт» 7.3 (3.10); Windows 7 / 8 / 10 ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ^[27] .

Примечание. 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4 Гбит/с.

Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел [Ограничения](#)^[37].

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: Intel P-4 2,8 ГГц; Память: 2 Гбайт; Диск: 10 Гбайт (свободно); Монитор и карта: 1024x768; Сеть: 100 Мбит/с; ОС: RedOS 7.1, Alt Linux 9.1, Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43, РОСА «Кобальт» 7.3; Windows 7 / 8 / 10. ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ²⁷ .	Процессор: Intel Core 2 Duo; Память: 4 Гбайт; Диск: 20 Гбайт (свободно); Монитор и карта: 1280x1024 или 1440x900; Сеть: 100 Мбит/с и выше; ОС: RedOS 7.1 / 7.2 / 7.3, Alt Linux 9.1, Astra Linux (Orel) 2.12, версии: 2.12.29, 2.12.43, РОСА «Кобальт» 7.3; Windows 7 / 8 / 10. ПО: Перечень минимальных требований для ПО приведен в разделе Требования к офисным приложениям ²⁷ .

Примечание. 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4 Гбит/с.

Внимание! Некоторое программное обеспечение работает с ограничениями, см. раздел [Ограничения](#)³⁷.

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:

Внимание! 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4 Гбит/с.

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

Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел [Ограничения](#)³⁷.

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 10 – 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 ЭП согласно требованиям Компании “КРИПТО-ПРО”) Опционально: КриптоПро TSP Client, КриптоПро OSCP Client Браузер: Яндекс.Браузер 19.4.2, MS Internet Explorer 10.0, 11.0 Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения ³⁷ .	ОС: Windows 7 СКЗИ: ViPNet CSP версии 4.2 Браузер: MS Internet Explorer 10.0, 11.0 Внимание! Некоторое программное обеспечение работает с ограничениями, см.раздел Ограничения ³⁷ .
ОС: RedOS, Alt Linux, Astra Linux. СКЗИ: КриптоПро CSP 4.0, 5.0 (версия должна соответствовать установленной на сервере ЭП согласно требованиям Компании “КРИПТО-ПРО”), Браузер: Яндекс.Браузер 19.10.3, Chromium-gost 72.0.3626	

Примечание. 64x разрядная операционная система рекомендуется при оперативной памяти не меньше 4 Гбит/с.

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 серверов приложений АЦК-Финансы к серверам приложений систем АЦК-Госзаказ, АЦК-Планирование, АЦК-Бюджетный учет участвующих в он-лайн обмене документами, подключение сервера приложения АЦК к сервером базы данных должны 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 10 Мбит/с. Реальная скорость передачи данных от любого рабочего места до сервера, в момент пиковых нагрузок, должна быть не ниже 20 Кбит/с.

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:

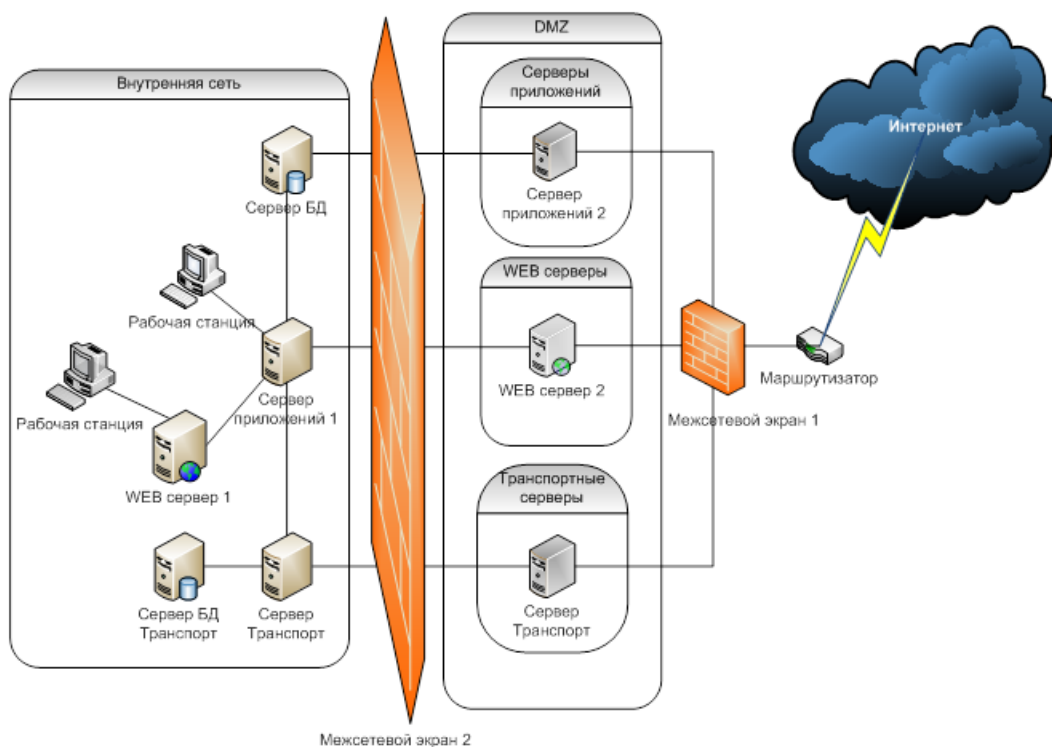


Рисунок 1 – Рекомендуемая схема сети

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.

Общие сведения о программе
Системные требования

Таблица 11 – Системные требования к офисным приложениям рабочего места с установленным на нем Windows-клиентом

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	MS Office	MS Office
Al t L i n u x	9.1	-	-	Li b r e O f f i c e 6 .3 .6 .2	-
R e d O	7.1	-	-	-	-
	7.2	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	-	-

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	L	M
	7.3	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	L i b r e O f f i c e 7 . 2 . 4 . 1	-
А s t r a L i n u x	2.12.29	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	-	-
	2.12.43	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	-	
Р О С А « К о б а л ь т »	7.3 (3.10)	-	-	-	-

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	MS Office
Windows	7	-	-	MS Office 2007 / 2010 / 2013 / 2016

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	P7-Офис	MS Office
	8	-	-	- MS Office 2007 / 2010 / 2013 / 2016

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	MS Office
	10	-	-	- MS Office 2007 / 2010 / 2013 / 2016

Общие сведения о программе
Системные требования

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

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	MS Office
Linux	9.1	-	-	LibreOffice 6.3.6.2
RedOS	7.1	-	-	LibreOffice 5.3.4.1
	7.2	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	-

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	L i b r e O f f i c e	M S O f f i c e
	7.3	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	L i b r e O f f i c e 7 . 2 . 4 . 1	-
А s t r a L i n u x	2.12.29	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	-	-
	2.12.43	МойОфис 2022.01	Р7-Офис Профессиональный (десктопная версия) 7.3.0	-	
Р О С А « К о б а л ь т »	7.3 (3.10)	-	-	-	-

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	MS Office
Windows	7	-	-	MS Office 2007 / 2010 / 2013 / 2016

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	P7-Офис	MS Office
	8	-	-	- MS Office 2007 / 2010 / 2013 / 2016

Общие сведения о программе
Системные требования

ОС / продукт	Версия ОС	МойОфис	Р7-Офис	MS Office
	10	-	-	- MS Office 2007 / 2010 / 2013 / 2016

1.2.6 Ограничения

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

Таблица 13 – Перечень программного обеспечения с ограничениями

Наименование программного обеспечения	Примечание	Ограничения
RedOS 7.1	Операционная система	<ul style="list-style-type: none"> • Прекращена поддержка ОС вендором; • 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>Для win-клиента, web-клиента, сервера ЭП:</p> <ul style="list-style-type: none"> • не работает ЭП вида «усиленная (со штампом времени CAdES-T)».
Astra Linux 2.12.29	Операционная система	<p>Для win-клиента:</p> <ul style="list-style-type: none"> • некорректное отображение неактивных кнопок. <p>Для сервера ЭП:</p> <ul style="list-style-type: none"> • не работает ЭП вида «усиленная с доказательствами подлинности» с включенным параметром расширения подписи на сервере.
Astra Linux 2.12.43	Операционная система	<p>Для windows-клиента:</p> <ul style="list-style-type: none"> • Некорректное отображение неактивных кнопок.
РОСА «Кобальт» 7.3 (3.10)	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 видов «CAdES-BES», «CAdES-X Long Type 1». <p>Для Web-клиента:</p> <ul style="list-style-type: none"> • Есть некритичные проблемы при работе с отчетами.

Общие сведения о программе
Системные требования

Наименование программного обеспечения	Примечание	Ограничения
Alt Linux 9.1	Операционная система	<p>Для windows-клиента:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности». • Не работает ЭП вида «усиленная (со штампом времени CAdES-T)». <p>Для web-клиента:</p> <ul style="list-style-type: none"> • Не работает ЭП для типа подписи «усиленная с доказательствами подлинности». • Не работает ЭП видов «CAdES-BES», «CAdES-X Long Type 1». • Не работает ЭП вида «усиленная (со штампом времени CAdES-T)».
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.
RHEL 6	Операционная система	Прекращен выпуск общедоступных патчей
SUSE Linux 11	Операционная система	Прекращена поддержка ОС вендором
Windows Server 2012 R2 / 2019 Standart (1809)	Операционная система	<p>Для Сервера СУБД:</p> <ul style="list-style-type: none"> • по результатам нагрузочного тестирования ОС настоятельно не рекомендуется использовать с СУБД PostgreSQL, Postgres Pro при количестве онлайн-пользователей 250 и более. <p>Для windows-клиента, web-клиента, сервера ЭП:</p> <ul style="list-style-type: none"> • не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают.
Windows Server 2008 R2	Операционная система	<p>Для Сервера СУБД: по результатам нагрузочного тестирования ОС настоятельно не рекомендуется использовать с СУБД PostgreSQL, Postgres Pro при количестве онлайн-пользователей 250 и более.</p> <p>Для win-клиента, web-клиента, сервера ЭП:</p> <ul style="list-style-type: none"> • не работает ЭП для типа подписи «усиленная с доказательствами подлинности», другие типы подписи работают. <p>Прекращена поддержка ОС вендором.</p>

Общие сведения о программе
Системные требования

Наименование программного обеспечения	Примечание	Ограничения
Windows 7, 8	Операционная система	Прекращена поддержка ОС вендором
КриптоПро 4/5	Криптопровайдер	Для windows-клиента, web-клиента, сервера ЭП: <ul style="list-style-type: none"> • длительное время проверки ЭП формата CAdES-T на стороне криптопровайдера.
МойОфис 2022.01		Есть проблемы при работе с отчетами (критичность ограничения определяется индивидуально): <ul style="list-style-type: none"> ▪ Есть проблемы при работе со специальными отчетами АЦК; ▪ «МойОфис» не поддерживает свойство «переносить строго по разрывам строк и столбцов»; ▪ «МойОфис» не поддерживает скрытие номера первой страницы; ▪ «МойОфис» не поддерживает функциональность «Напечатать активные листы»; ▪ «МойОфис» не поддерживает отображение содержимого открываемого файла с заранее заданным масштабом, отличным от 100%; ▪ «МойОфис» не поддерживает функциональность «Вписать все столбцы на одну страницу»; ▪ «МойОфис» не поддерживает функциональность защиты ячеек с помощью пароля; ▪ «МойОфис Текст» не поддерживает установку произвольной позиции для поля номер страницы и вычисляемое поле количество страниц.



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 (ОМТ) 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 14 – 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 Sentel.

Структура программы
Сервер приложений

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](#)²⁷⁴ 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 15 – Утилиты для работы с БД под управлением ОС Windows

Название утилиты	Описание
<code>./sql/backup.cmd</code>	Создание резервной копии БД.
<code>./sql/comparer.cmd</code>	Сравнение структуры эталонной и целевой БД.
<code>./sql/change_site.cmd</code>	Не используется.
<code>./create_db.cmd</code>	Создание БД начального уровня (создание новой БД, создание объектов БД по SQL-скриптам, заполнение системных справочников, создание бюджета и т.п.).
<code>./create_db_interbase.cmd</code>	Создание новой БД (Firebird).
<code>./sql/DBUpdate.cmd</code>	Обновление БД на версию системы.
<code>./filloutdiff.cmd</code>	Не используется.
<code>./filloutdiff_tfo.cmd</code>	Не используется.
<code>./sql/fix_version.cmd</code>	Установка для записей БД значений VERSION, равных null.
<code>./integrity_check.cmd</code>	Проверка целостности установки системы.
<code>./makediff.cmd</code>	Не используется
<code>./sql/merge.cmd</code>	Объединение данных из нескольких БД в одну.
<code>./sql/perform.cmd</code>	Создание в новой БД всех объектов инфраструктуры по метаданным из SQL-скриптов.
<code>./sql/restore.cmd</code>	Восстановление БД из резервной копии.
<code>./sql/sql.cmd</code>	Выполнение SQL-скриптов для БД.
<code>./sql/update.cmd</code>	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](#)²⁷⁰.

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
```

- б) 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>azk2.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) Сетевой буфер отправки iiop:

Имя: iiop.sendBufferSize

Описание: Настройка сетевого буфера отправки iiop. Применяется при анализе проблем и рекомендациях технического сопровождения системы. Настройка не влияет на работу Java приложений с сетью в ОС Windows.

3) Сетевой буфер приема iiop:

Имя: iiop.receiveBufferSize

Описание: Настройка сетевого буфера приема iiop. Применяется при анализе проблем и рекомендациях технического сопровождения системы. Настройка не влияет на работу Java приложений с сетью в ОС Windows.

4) Алгоритм Нейгла как клиентское подключение:

Имя: iiop.clientNoDelay

Описание: Отключение алгоритма Нейгла как клиентского подключения.

5) Алгоритм Нейгла как сервер:

Имя: iiop.clientNoDelay

Описание: Отключение алгоритма Нейгла как сервера.

6) Алгоритма Нейгла:

Имя: iiop.noDelay

Описание: Отключение алгоритма Нейгла.

7) 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 подсоединение 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.

8) Использование SSL:

Имя: azk.UseSSL

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

Значение по умолчанию: azk.UseSSL=true

9) Порт для подключения по SSL:

Имя: ssliop.port

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

Значение по умолчанию: ssliop.port=2002

10) 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

11) 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>`

12) 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 Свойства сетевых параметров СУБД PG

1) Сетевой буфер приема при работе с СУБД PG:

Имя: `azk.db.pg.receiveBufferSize`

Описание: Настройка сетевого буфера приема при работе с СУБД PG. Применяется при анализе проблем и рекомендациях технического сопровождения системы. Настройка не влияет на работу Java приложений с сетью в ОС Windows.

2) Сетевой буфер передачи при работе с СУБД PG:

Имя: `azk.db.pg.sendBufferSize`

Описание: астройка сетевого буфера передачи при работе с СУБД PG. Применяется при анализе проблем и рекомендациях технического сопровождения системы. Настройка не влияет на работу Java приложений с сетью в ОС Windows.

3) Имя приложения подключающегося к СУБД PG:

Имя: `azk.db.pg.ApplicationName`

Описание: Задается имя (буквы/цифры) приложения подключающегося к СУБД PG. Значение передается в параметр **ApplicationName** строки подключения к СУБД PG. Имя отображается в информации о сессии к СУБД.

Значение: Если задано значение **azkinstance** – система подставляет в значение параметра значение переменной **INSTANCE** заданной при старте сервера приложений.

4) Поддержка подключения к СУБД PG от СП с помощью механизма KeepAlive.

Имя: azk.db.pg.tcpKeepAlive

Описание: Настройка поддержки подключения к СУБД PG от СП с помощью механизма KeepAlive. Применяется при анализе проблем и рекомендациях технического сопровождения системы.

5) Режим кеширования запросов на стороне сервера и параметризации запросов.

Имя: azk.db.pg.preferQueryMode

Описание: Настройка режима кеширования запросов на стороне сервера и параметризации запросов. Применяется при анализе проблем и рекомендациях технического сопровождения системы.

Значение по умолчанию: azk.db.pg.preferQueryMode =extended. Дополнительно допустимо значение *extendedCacheEverything*.

2.1.4.7 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

2.1.4.8 Свойства репликации

1) Максимальный размер пакета репликации:

Имя: azk.rpl.maxpacketsize

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

Значение по умолчанию: azk.rpl.maxpacketsize = 200000

2.1.4.9 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

2.1.4.10 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).

2.1.4.11 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.bs.sys.azkserver.udod.AzkDocAttachEx

3) Класс, управляющий хранением вложений для справочников:

Имя: azk.refattach.class

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

Значение по умолчанию: azk.refattach.class=com.bs.sys.server.system.RefAttachEx

4) Класс, управляющий хранением вложений в почтовых сообщениях:

Имя: azk.mailattach.class

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

Значение по умолчанию: azk.mailattach.class=com.bs.sys.order.system.GzMailAttachEx

5) Открытие вложений, хранящихся в БФТ е.Архив:

Имя: azk.docarchive_attach.browser_open

Описание: Имя класса в «АЦК-Планирование», выполняющего функции открытие вложений, хранящихся файлов вложений в БФТ е.Архиве.

Значение по умолчанию:

2.1.4.12 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

2.1.4.13 Настройка репликации нового сайта

Не используется.

2.1.4.14 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/

2.1.4.15 Настройка параметров лицензии

1) Путь к файлу лицензии:

Имя: azk.license.name

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

2.1.4.16 Настройка сервиса бюджетного мониторинга

1) Адрес сервиса ЭБ (КБМК)

Имя: fns.wsdlLocation, fns.wsdlLocation.endPoint

Описание: Адрес, по которому отправляется конверт сообщения в СМЭВ 3.0 при выгрузке сведений о лицевых счетах в ЭБ (КБМК).

Пример:

#fns.wsdlLocation=http://127.0.0.1:8088/SMEV?wsdl

#fns.wsdlLocation.endPoint=http://127.0.0.1:8088/SMEV?wsdl

2.1.4.17 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
```

2.1.4.18 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) Параметр времени опроса очереди принудительного обновления журнала поколений:

Имя: `azk.rpllog.service.rplassignqueue.scan`

Описание: регулирует время опроса очереди принудительного обновления журнала поколений (`rpllog`) из таблицы `rplassignqueue`

При использовании более одного сервера приложений, только на одном устанавливается необходимое значение `azk.rpllog.service.rplassignqueue.scan`. Параметр регулирует время опроса очереди принудительного обновления журнала поколений (`rpllog`). Очередь наполняется другими серверами приложений кластера системы по необходимости. Опрос очереди работает только если на данном сервере приложений включен `rpllogservice` (параметр `azk.rpllog.service.disabled=false`).

Значение по умолчанию: сервис разрешен.

Примечание. Процедура обновления журнала поколений (**rplassign**). Процедура предназначена для пересчета журнала поколений и присвоения актуальных номеров поколений изменившихся записей. Процедура проверяет не запущен ли ее экземпляр с другого СП, создает блокировку таблицы **rplassignlock** (если блокировка не удачная, процедура не выполняется), генерирует новый номер поколений записей, формирует перечень обновляемых записей журнала поколений, обновляет по перечню записей без номера поколения, обновляет номера поколений для очистки журнала поколений с учетом номеров поколений в объектах репликации, обновляет текущий номер поколения в таблице **system site**.

3) Параметр ограничения периода запуска обновления журнала поколений:

Имя: azk.rpllog.service.timeout.assign.generation

Описание: параметр регулирует таймаут, в течении которого не возможен запуск обновления журнала поколений

Ограничение запуска обновления журнала поколений через **notifyAssignGeneration** и через очередь принудительного обновления. Задается в мс, не менее 1000мс, значение по умолчанию 10000. Без необходимости значение по умолчанию менять не рекомендуется.

Значение по умолчанию: сервис разрешен.

2.1.4.19 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>

```

2.1.4.20 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
#способы определения
azk.openid.provider.identfieldscope=snils
azk.openid.provider.identfield=snils
azk.openid.scim.setauthusernamefrom=personid

```

Примечание. В случае, если переадресация произошла некорректно, при работе через клиентское приложение, нужно изменить **localhost** на **localopenidazk.bftcom.com**, данная замена осуществляется ручным способом, либо при помощи утилиты **AzkHostEditor**.

2.1.4.21 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
```

2.1.4.22 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
```

2.1.4.23 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.

2.1.4.24 Настройка сервиса помощи входа по сертификату (СПВС)

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

- Адрес сервиса идентификации СБ:

Имя: `azk.integration.security_service.identification.uri`

Описание: Настройка для авторизации пользователя по jwt токenu.

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

2.1.4.25 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 16 – 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	

Структура программы
Клиентское приложение

Файлы клиентского приложения	Описание
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».
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.

Структура программы
Клиентское приложение

Файлы клиентского приложения	Описание
Reports.ocx	Динамическая библиотека, содержащая клиентские объекты для работы с отчетами.
ServIntf.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	
az_lib_rt.bpl, AnalysisPack.bpl, Azk2ClientPack.bpl, Azk2ExpensePack.bpl, clientpack.bpl, ehlib.bpl, kerpacck.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 ОСХ 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. Максимальная длина префикса не должна превышать пятьдесят символов.

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, квитанки, 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.00004-56 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 `<site_number>*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 на статус «*завершен (4)*».

Порядок работы службы обработки сообщений таков:

- 1) сканируется список сообщений на 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, 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 на 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.

Для БФТ.ТОП

Получение и обработка репликационных XML-сообщений через RMI. Взаимодействие выполняется через:

1. Результат приема сообщений

- в случае успешного приема сообщения от АЦК должна отправить в АЦК через RMI сообщение со следующими данными:
 - ID принятого сообщения от АЦК;
 - Код статуса «*получен*» принятого сообщения (**DISPSTATUS** равно 2).
- в случае ошибки при приеме сообщения от АЦК должна отправить в АЦК через RMI сообщение со следующими данными:
 - ID принимаемого сообщения от АЦК;
 - Текст ошибки (**DIAGNOSIS** равно тексту ошибки длиной не более 2000 символов).

2. Результат обработки сообщения

- в случае успешной обработки сообщения от АЦК должна отправить в АЦК через RMI сообщение со следующими данными:
 - ID обработанного сообщения от АЦК;

- Код статуса «*обработан*» обработанного сообщения (**DISPSTATUS = 3**).
- в случае ошибки при обработке принятого сообщения от АЦК должна отправить в АЦК через RMI сообщение со следующими данными:
 - ID обрабатываемого сообщения от АЦК;
 - Текст ошибки (**DIAGNOSIS** равно тексту ошибки длиной не более 2000 символов).

Примечание. При отправке сообщения, у которого, в поле **Получатель** указан сайт с типом соединения 38 = «Обмен с БФТ.ТОП через брокер Kafka», с помощью работы планировщика **Transport** или **Exchange**, в поле **Дата отправки** записывается время отправки сообщения в формате ДД:ММ:ГГГГ ЧЧ:ММ:СС. При получении АЦК через RMI от БФТ.ТОП результата приема сообщения с `msg_dispstatus = 2` (2 - это код статуса «Получен»), то в поле **Дата получения** записывается время в формате ДД:ММ:ГГГГ ЧЧ:ММ:СС.

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/schtask.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.00004-56 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 AttachMoverLauncher

XML для запуска:

```
<DATASTORAGE action="move"/>
```

Параметры: storageName, source, dest, thread_count, budgetID, startDate, endDate, ATTACH_MAX_TOTAL_SIZE, DOCUMENTCLASS_ID, DOCUMENT_ID, ATTACH_ID, DOCCATEGORY_ID, allow_send_to_archive_wo_sign.

Описание: задание планировщика *AuBuLinegroup Cleaner* предназначено для переноса вложений в другое хранилище.

Принимает параметры:

- **storageName** – хранилище данных. Обязательный параметр.
- **source** – устройство хранения (источник). Обязательный параметр.
- **dest** – устройство хранения (приемник). Обязательный параметр.
- **thread_count** – количество потоков. Принимает значение от 1 до n, где n - количество процессоров, доступных виртуальной машине Java. Обязательный параметр.
- **budgetID** – ID бюджетов через запятую. Необязательный параметр.
- **startDate** – дата вложения с. Формат YYYY-MM-DD. Необязательный параметр.
- **endDate** – дата вложения по. Формат YYYY-MM-DD. Необязательный параметр.
- **ATTACH_MAX_TOTAL_SIZE** – максимальный суммарный размер всех перемещаемых вложений в Мегабайтах. Необязательный параметр.
- **DOCUMENTCLASS_ID** – ID классов документов через запятую. Необязательный параметр.
- **DOCUMENT_ID** – ID документов через запятую. Необязательный параметр.

- **ATTACH_ID** – ID вложений через запятую. Необязательный параметр.
- **DOCCATEGORY_ID** – ID категорий вложений через запятую. Необязательный параметр.
- **allow_send_to_archive_wo_sign** – Разрешить передавать вложения в БФТ.е-Архив без ЭП. Допустимые значения: true или false. Необязательный параметр.

3.8.4 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:

```
<GENERICSCHTASK action="run"
class_name="com.bssys.azk.server.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.*

3.8.5 AutoEBUnloadLauncher

Задание планировщика *AutoEBUnloadLauncher* предназначено для выгрузки ЭД «Распоряжение о совершении казначейских платежей», ЭД «Заявка БУ/АУ на выплату средств» в рамках бюджетного мониторинга со статуса «Подготовлен».

XML для запуска:

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

Параметры: class_id, budget_id.

Описание: Автоматическая выгрузка ЭБ (КБМК).

Принимает параметры:

- **budget_id** – ID бюджета, по которому осуществляется выгрузка ЭД. Необязательный параметр. Возможно указание нескольких параметров через запятую.
- **class_id** – ID класса документа, которые выгружаются планировщиком. Необязательный параметр. Возможно указание нескольких параметров через запятую. Работу параметра распространяется только на ЭД «Распоряжение о совершении казначейских платежей», ЭД «Заявка БУ/АУ на выплату средств».

Задание планировщика выполняет автоматическую выгрузку ЭБ (КБМК).

Если в параметре **class_id** планировщика указано значение 309, или параметр **class_id** не указан, или параметр указан, но без конкретных значений (т.е. если указано «*class_id=*»), то отбираются ЭД «Распоряжение о совершении казначейских платежей по ЛС УКС» на статусе «Подготовлен». При этом если указан параметр планировщика **budget_id** с конкретным значением (значениями), отбираются ЭД только из бюджетов с соответствующими ID.

Если в параметре **class_id** планировщика указано значение 195, или параметр **class_id** не указан, или параметр указан, но без конкретных значений (т.е. если указано «*class_id=*»), то отбираются ЭД «Заявка БУ/АУ на выплату средств» на статусе «Подготовлен» и «Импортирован», у которых поле **Статус ЭБ (КБМК)** пусто или «Ошибка экспорта», и значение в поле **Счет УФК** группы полей **Плательщик** начинается на «03215», «03225», «03235». При этом, если указан параметр планировщика **budget_id** с конкретным значением (значениями), отбираются ЭД только из бюджетов с соответствующими ID.

Над отобранными ЭД выполняется автоматически действие, аналогичное ручному действию по кнопке **Выгрузить** на панели инструментов списковой формы ЭД «Распоряжение о совершении казначейских платежей», ЭД «Заявка БУ/АУ на выплату средств» при выбранном режиме выгрузки «Запрос на проведение операций».

3.8.6 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_2.56.0_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.

3.8.7 Backuper

XML для запуска:

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

Параметры: backuppath, backupname, -checkrestore.

Описание: Задание планировщика *Backuper* 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
```

3.8.8 Bank Update ED807

XML для запуска:

```
<GENERICSCHTASK action="run" class_name="com.bssys.shared.BankUpdateED807Launcher"/>
```

Параметры: dir, archiveDir, InfoTypeCode.

Описание: Автоматическое обновление справочника *Банки* из ED807.

Процедура используется для автоматического обновления справочника на основании электронного сообщения, предоставляемого по требованиям УФЭБС. При выполнении процедуры осуществляется контроль активности карточки банка при наличии

ограничений по организации или счету. При соответствии статуса карточки банка хотя бы одному значению из системного параметра **Статусы банка (через запятую) для определения неактивности банка** (пункт меню **Сервис→Системные параметры→Организации**, закладка **Настройки**), в справочнике *Банки* снимается признак **Активный**.

Примечание. При формировании признака активности и статуса карточки банка учитывается дата начала ограничений и дата закрытия счета. Карточки для организаций со счетами, не соответствующими корреспондентскому счету, не формируются.

Принимает следующие параметры:

- **dir** – путь к каталогу загрузки, в котором определяется файл для обновления справочника. Обязательный параметр.
- **archiveDir** – путь к архивному каталогу для сохранения обработанных файлов. Если параметр не указан, файлы обработки удаляются из каталога загрузки.
- **InfoTypeCode** – режим обработки. Указывается один из двух режимов:
 - *FIRR* – полное обновление справочника.
 - *SIRR* – изменение справочника.

Если параметр не указан, режим обработки определяется из файла, который содержит обновление для справочника.

Пример:

```
dir=E:\Bank UpdateED807;archiveDir=E:\Bank UpdateED807Archive;InfoTypeCode=FIRR;
```

3.8.9 CADownloader

XML для запуска:

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

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

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

3.8.10 CassaBUAUExportPlan

XML для запуска:

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

Параметры: finYear, budget, budget_id, dest, dir, user, password, wsdlLoc, site, queue, end_date, id_manager, id_org, id_grantinvestment_code, kif.

Описание: планировщик *CassaBUAUExportPlan* предназначен для экспорта субсидий БУ/АУ.

Принимает параметры:

- **finYear** – год бюджета, для которого делается выгрузка. Обязательный параметр.
- **budget** – наименование бюджета, для которого делается выгрузка. Обязательный параметр, если не заполнен **budget_id**.
- **budget_id** – id бюджета, для которого делается выгрузка (допустимо указывать несколько id через запятую). Обязательный параметр, если не указаны параметры **finYear** и **budget**.
- **dest** – назначение выгрузки файла. Возможные значения: 1 – передать файл веб-сервису, 2 – сохранить файл в файловой системе, 3 – передать сообщением. Обязательный параметр.
- **dir** – адрес директории выгрузки в файловой системе. Обязательный параметр, если **dest** – 2.
- **user** – пользователь веб-сервиса. Обязательный параметр, если **dest** – 1.
- **password** – пароль для входа на веб-сервис. Обязательный параметр, если **dest** – 1.
- **wsdlLoc** – адрес веб-сервиса. Обязательный параметр, если **dest** – 1.
- **site** – номер сайта на который необходимо отправить сообщение. Обязательный параметр, если значение параметра **dest** – 3.
- **queue** – наименование очереди сообщений, по которой будет происходить отправка. Обязательный параметр, если значение параметра **dest** – 3.
- **end_date** – дата окончания выгружаемых данных. Если параметр не указан –

используется текущая дата.

- **id_manager** – id учредителя, по которому делается выгрузку (допустимо указывать несколько id через запятую).
- **id_org** – id учреждения, по которому делается выгрузку (допустимо указывать несколько id через запятую).
- **id_grantinvestment_code** – id кода субсидии, по которому делается выгрузку (допустимо указывать несколько id через запятую).
- **kif** – КВФО, по которому делается выгрузку (допустимо указывать несколько КВФО через запятую).

Пример использования веб-сервиса:

```
finYear=год бюджета; budget=наименование_бюджета; wsdlLoc=адрес_веб_сервиса;
user=пользователь_веб_сервиса; password=пароль_пользователя_веб_сервиса; dest=1
```

Пример использования выгрузки в файл:

```
finYear=год бюджета; budget=наименование_бюджета;
dir=директория_выгрузки_в_файловой_системе; dest=2
```

3.8.11 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.

3.8.12 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.

3.8.13 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.

3.8.14 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.

3.8.15 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.

3.8.16 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:

```
<GENERICSTASK 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.

3.8.17 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.

3.8.18 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*. При выполнении планировщика и включении признака **Отозван** сертификату, отключается признак **Актуальное** у соответствующей записи справочника *Ответственные лица*.

Примечание. При выполнении запроса на отключение признака **Отозван**, осуществляется проверка несоответствия текущего значения и сохраняемого. Таким образом, признак **Отозван** у сертификатов, у которых он и так уже отключен, изменен не будет.

Принимает параметр:

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

3.8.19 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».

3.8.20 DataObjectChangeLogCleaner

XML для запуска:

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

Параметры: PERIOD, OBJECTS, pg_work_mem.

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

Принимает параметры:

- **PERIOD** – период очистки журнал, по умолчанию 8.
- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

- **OBJECTS** – список объектов для которых производить очистку.

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

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

3.8.21 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.
```

3.8.22 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
```

3.8.23 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 зеркало (клон) переходит в состояние
ошибка.
```

3.8.24 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.

3.8.25 DebitCardLauncher

XML для запуска:

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

Параметры: нет.

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

Рекомендуемый период запуска: не используется.

3.8.26 DeleteNotifyMsg

XML для запуска:

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

Параметры: нет;

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

Рекомендуемый период запуска: не используется.

3.8.27 DeliveryFactDocLauncher

XML для запуска:

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

Параметры: ProcessClasses, budget_id, imported_only.

Описание: Задание планировщика предназначено для автоматического перевода ЭД «Факт поставки» со статуса «*подготовлен*» на статус «*принят*».

Принимает параметры, которые может указывать администратор системы:

- **ProcessClasses** – перечисляются классы документов. Возможно указание нескольких параметров через запятую. Обязательный параметр.
- **budget_id** – наименование бюджета, для которого требуется выгрузить данные. Возможно указание нескольких параметров через запятую. Необязательный параметр.
- **imported_only** – при необходимости ограничения планировщика по ЭД «Факт поставки», которые были импортированы из внешних систем. Может принимать значения *1* и *0*. Необязательный параметр.

При задании планировщика выполняется:

- в базе всех классов осуществляется поиск, указанных в параметре **ProcessClasses** на статусе «*подготовлен*».
- Если у планировщика заполнен параметр **budget_id**, список найденных ЭД «Факт поставки» на статусе «*подготовлен*» ограничивается теми, у которых идентификатор бюджета совпадает со значениями из параметра.
- Если у планировщика задан параметр **imported_only** равный значению *1*, ограничивается список ЭД «Факт поставки» на статусе «*подготовлен*», у которых идентификатор бюджета совпадает со значениями из параметра **budget_id**, только с заполненным полем **foreign_key**.
- Над всеми документами из получившегося списка выполняется действие **process** (**Принять**).

Примечание. Любые ошибки при выполнении обработки документов по действию **process** (**Принять**) должны в штатном порядке фиксироваться в «Журнале ошибок документа».

3.8.28 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.

3.8.29 DocumentCleaner

XML для запуска:

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

Параметры: path, delete, nosave, pg_work_mem.

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.
- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_tet=384
```

- **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=?]
```

3.8.30 EisRequestProtocolLoading

XML для запуска:

```
<GENERICSCHTASK action="run" class_name="com.bftcom.shared.EisRequestProtocolLoading"/>
```

Параметры: budget_id, class_id, finYear.

Описание: Запрос статуса проверки платежных данных в ЕИС.

Принимает параметры:

- **budget_id** – ID бюджета, по которому осуществляется загрузка результата контроля ЭД в ЕИС. Необязательный параметр, с возможностью указать несколько ID бюджетов (через запятую).
- **class_id** – ID класса документа, по которым необходимо загружать результат контроля планировщиком. Необязательный параметр, с возможностью указания нескольких значений (через запятую). Распространяется только на ЭД «Заявка на оплату расходов», «Заявка БУ/АУ на получение наличных денег», «Заявка БУ/АУ на выплату средств» (остальные значения игнорируются).
- **finYear** – год бюджета, в котором осуществляется задание планировщика. Обязательный параметр, если не заполнен **budget_id**.

При запуске планировщика:

- Если в параметре **class_id** планировщика указаны конкретные значения,

отбирается ЭД из параметра **class_id**, у которых значение в поле **Счет** группы полей **Плательщик** имеет тип счета равный *Лицевой счет в ФО*, в поле **Вид реестра** значение 2 – *Реестр контрактов (открытый)* и в поле **Внешний статус** со значением *Ожидает загрузки на контроль в ЕИС*. При этом если указан параметр планировщика **budget_id** с конкретным значением (значениями), отбирается ЭД только из бюджетов с соответствующими ID.

- Если в параметре **class_id** планировщика не указан, или параметр указан, но без конкретных значений, отбираются ЭД «Заявка на оплату расходов», «Заявка БУ/АУ на получение наличных денег», «Заявка БУ/АУ на выплату средств», у которых значение в поле **Счет** группы полей **Плательщик** имеет тип счета равный *Лицевой счет в ФО*, в поле **Вид реестра** значение 2 – *Реестр контрактов (открытый)* и в поле **Внешний статус** со значением *Ожидает загрузки на контроль в ЕИС*. При этом если указан параметр планировщика **budget_id** с конкретным значением (значениями), отбираются ЭД только из бюджетов с соответствующими ID.
- Над отобранными ЭД выполняется группировка ЭД по **EIS_REF_ID**, который передается в **refld**.
- Опрашивается сервис ЕИС, по адресу, указанному в бюджетном параметре **URL получения результатов загрузки документов в ЕИС по протоколу HTTPS (Сервис→Бюджетные параметры→Закупки)** на получение «Протокола загрузки информации HTTPS-интеграции» или на получение ответа на запрос проведения проверки платежных данных *controlPaymentDetailsResponse* по **refld**. При этом опрос проводится через десять минут, после получения первоначального протокола с значением в элементе *"result" = "processing"* и не менее, чем через тридцать минут повторный опрос сервиса ЕИС.

3.8.31 Exchange

XML для запуска:

```
<COMMONPARAMSTASK action="run">
```

```
<TASK>
```



```

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

```

Параметры: site_id, error.email, error.email.subject, pg_work_mem.

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.

- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

- **error. email** – указывается адрес или адреса электронной почты для отправки информации.
- **error. email. subject** – указывается текст сообщения. Если параметр не заполнен, заполняется тестом: «Службное сообщение АЦК».

В случае прерывания работы задания планировщика формируется электронное письмо со следующими реквизитами:

- с темой из параметра **error.email.subject**;
- в теле письма цитируется ошибка;
- в качестве адресов получателей подставлять значение параметра **error.email** (для отправки требуется разделять их точкой с запятой).
- **serverMode** – параметр определяет протокол соединения между серверами. Работа параметра осуществляется для типа соединения – **Прямое соединение серверов (MSGSITE.SMSGLINKTYPE_ID = 1)**. Если параметр **serverMode** в задании:
 - не указан, подключение к другому СП идёт по *iiop*;
 - указан со значением *iiop* (**serverMode = iiop**), подключение к другому серверу приложений идёт по протоколу *iiop*;
 - указан со значением *http* или *https* (**serverMode = http**), подключение к другому СП идёт по протоколу *http* или *https*.

Пример использования:

error.email=admin1@mail.ru, admin2@mail.ru; error.email.subject=Ошибка планировщика Exchange в АЦК-Финансы 2022 для сайта АЦК-Планирование 2022

***Примечание.** Независимый от работы репликатора механизм предотвращения параллельного запуска обновления журнала поколений. При запуске процедуры обновления журнала поколений, на основе считанных объектов репликации для каждого объекта вычисляется какие данные необходимо послать на сайт-получатель, в журнале поколений находятся записи с номером поколения больше или равным «Отправленная версия» и/или с номером поколения меньше или равным текущему номеру поколения. Полученные данные проходят постобработку с помощью процессора репликации, если он указан. Сформированные данные помещаются в сообщения.*

Сформированное сообщение отправляется с адреса, указанного в системных параметрах электронной почты (пункт меню **Сервис→Системные параметры**, группа настроек **Система**, закладка **Настройки электронной почты**)

Примечание. В случае прерывания обработки репликационного сообщения если работа планировщика не прерывается, но ошибка фиксируется в поле *Диагноз* обрабатываемого сообщения, формируется электронное сообщение.

3.8.32 FhdPlanStructTemplateImport

XML для запуска:

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

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

3.8.33 FORepLauncher

XML для запуска:

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

Параметры: pg_work_mem.

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

pg_work_mem – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

3.8.34 GZFactDocImport

XML для запуска:

```
<GENERICSCHTASK action="run" class_name="com.bftcom.shared.GZFactDocImport" />
```

Параметры: budget_id, class_id, finYear.

Описание: Выгрузка фактов оплаты в АЦК-Госзаказ.

Принимает параметры:

- **budget_id** – ID бюджета, по которому осуществляется выгрузку ЭД. Необязательный параметр, с возможностью указать несколько ID бюджетов (через запятую).
- **class_id** – ID класса документа, по которым необходимо выгружать планировщиком. Необязательный параметр, с возможностью указания нескольких значений (через запятую). Распространяется только на ЭД:
 - «Заявка на оплату расходов»;
 - «Уведомление о возврате средств в бюджет»;
 - «Справка по расходам»;
 - «Справка по специальным средствам»;
 - «Распоряжение о перечислении на карты «Мир»»;
 - «Распоряжение БУ/АУ о перечислении на карты «Мир»»;
 - «Распоряжение о совершении казначейских платежей по ЛС УКС»;
 - «Распоряжения на зачисление средств на л/с»;
 - «Заявка БУ/АУ на получение наличных денег»;
 - «Заявка БУ/АУ на выплату средств»;
 - «Справка-уведомление об уточнении операций БУ/АУ»;
 - «Справка по операциям БУ/АУ».
- **finYear** – год бюджета, в котором осуществляется задание планировщика. Обязательный параметр, если не заполнен **budget_id**.

При запуске планировщика:

- Если в параметре **class_id** планировщика указан конкретные значения, отбирается ЭД из параметра **class_id** на статусе «*обработка завершена*» или «*обработан*» и в поле **Регистрация в системе закупок** значение *Не пройдена*. При этом если указан параметр планировщика **budget_id** с конкретным значением (значениями), отбирается ЭД только из бюджетов с соответствующими ID.
- Если в параметре **class_id** планировщика не указан, или параметр указан, но без

конкретных значений, отбираются ЭД «Заявка на оплату расходов», «Уведомление о возврате средств в бюджет», «Справка по расходам», «Справка по специальным средствам», «Распоряжение о перечислении на карты «Мир», «Распоряжение БУ/АУ о перечислении на карты «Мир», «Распоряжение о совершении казначейских платежей по ЛС УКС», «Распоряжения на зачисление средств на л/с», «Заявка БУ/АУ на получение наличных денег», «Заявка БУ/АУ на выплату средств», «Справка-уведомление об уточнении операций БУ/АУ», «Справка по операциям БУ/АУ» на статусе «*обработка завершена*» или «*обработан*» и в поле **Регистрация в системе закупок** значение *Не пройдена*. При этом если указан параметр планировщика **budget_id** с конкретным значением (значениями), отбираются ЭД только из бюджетов с соответствующими ID.

- Над отображенными ЭД выполняется автоматически действие, аналогичное ручному действию **Выгрузить факт оплаты**.

3.8.35 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_rpp.log, где:
 - dd_mm_yyyy - день, месяц, год;
 - rpp – 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.

3.8.36 ImportDictionariesUI

XML для запуска:

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

Описание: Загрузка и импорт справочников из внешних систем.

Параметры: BUDGET_ID=<id бюджета>, file_path=<адрес папки>.

Рекомендуемый период запуска: 1 раз в сутки.

Процесс выполняет загрузку информации по общим справочникам муниципальных образований посредством файлового обмена в виде файлов формата *xml*. Источником данных являются сторонние финансовые системы. Данные загружаются в справочник *Финансовые показатели (ЕХД)*.

Обмен производится без использования прямого подключения или обращения к БД сторонних систем.

Принимает параметры:

- **BUDGET_ID=<id бюджета>** – указывается ID бюджета, который будет присваиваться записям при обработке задания планировщика, а так же определяет финансовый год файлов, которые необходимы для обработки, по ФГ указанного бюджета.
- **file_path=<адрес папки>** – путь к источнику файлов для загрузки.

Атрибутный состав:

- **Классификатор ведомственных статей расходов (КВСР)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Таблица 17 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–
VERSION	VERSION	–
Бюджет	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий КВСР	PARENT_DESCRIPTION	Не заполняется.
Наименование вышестоящего КВСР	–	–
КВСР	CODE	cdep
Программные расходы	PROGRAM_EXPENSES	Заполняется значением false .
Утвержден на уровне субъекта (муниципального образования)	APPROVED_ENTITY_LEVEL	Заполняется значением false .
Национальный проект	IS_NAT_PROJECT	Заполняется значением false .
Строки		

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Атрибут АЦК-П	Атрибут XML
Краткое наименование	CAPTION	dep_n
Полное наименование	DESCRIPTION	dep_n
Не утвержден	UNAPPROVED	Заполняется значением false .
Начало действия	BEGIN_DATE	Заполняется датой начала текущего финансового года, который определяется из заголовка файла.
Окончание действия	END_DATE	Не заполняется.
Последнее изменение	CHANGE_DATE	–

- **Функциональный классификатор расходов (КФСР)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Таблица 18 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–
VERSION	VERSION	–
Бюджет	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий КФСР	PARENT_DESCRIPTION	Определяется верхнеуровневая запись по идентичному коду до точки («.») внутри бюджета. Например, если текущий код = 0303 (03.03), то есть запись с кодом 0300 (03.00), которая для текущей записи будет верхнеуровневой.
Наименование вышестоящего КФСР	–	–
КФСР	CODE	cdiv
Утвержден на уровне субъекта (муниципального образования)	APPROVED_ENTITY_LEVEL	Заполняется значением false .
Строки		
Краткое наименование	CAPTION	div_n
Полное наименование	DESCRIPTION	div_n
Не утвержден	UNAPPROVED	Заполняется значением false .
Начало действия	BEGIN_DATE	Заполняется датой начала текущего финансового года, который определяется из заголовка файла.
Окончание действия	END_DATE	Не заполняется.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Атрибут АЦК-П	Атрибут XML
Последнее изменение	CHANGE_DATE	–

- **Классификатор целевых статей расходов (КЦСР)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Таблица 19 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–
VERSION	VERSION	–
Бюджет	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий КЦСР	PARENT_DESCRIPTION	Не заполняется.
Наименование вышестоящего КЦСР	–	–
КЦСР	CODE	ctgt
Программные расходы	PROGRAM_EXPENSES	Заполняется значением false .
Утвержден на уровне субъекта (муниципального образования)	APPROVED_ENTITY_LEVEL	Заполняется значением false .
Национальный проект	IS_NAT_PROJECT	Заполняется значением false .
Строки		
Краткое наименование	CAPTION	tgt_n
Полное наименование	DESCRIPTION	tgt_n
Не утвержден	UNAPPROVED	Заполняется значением false .
Начало действия	BEGIN_DATE	Заполняется датой начала текущего финансового года, который определяется из заголовка файла.
Окончание действия	END_DATE	Не заполняется.
Последнее изменение	CHANGE_DATE	–

- **Классификатор видов расходов (КВР)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Основные понятия
 Планировщик. Задания планировщика. Расписание запуска заданий

Таблица 20 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–
VERSION	VERSION	–
Бюджет	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий KBP	PARENT_DESCRIPTION	Определяется верхнеуровневая запись по идентичному коду до точки («.») внутри бюджета. Например, если текущий код = 313, то есть запись с кодом 310, которая для текущей записи будет верхнеуровневой, в свою очередь для записи с 3 символами 312 вышестоящей будет запись 312, определяется по первым двум символам.
Наименование вышестоящего KBP	–	–
KBP	CODE	code
Утвержден на уровне субъекта (муниципального образования)	APPROVED_ENTITY_LEVEL	Заполняется значением false .
Строки		
Краткое наименование	CAPTION	exp_n
Полное наименование	DESCRIPTION	exp_n
Не утвержден	UNAPPROVED	Заполняется значением false .
Начало действия	BEGIN_DATE	Заполняется датой начала текущего финансового года, который определяется из заголовка файла.
Окончание действия	END_DATE	Не заполняется.
Последнее изменение	CHANGE_DATE	–

• **Дополнительный код расхода (Доп. КР)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Атрибут АЦК-П	Атрибут XML
VERSION	VERSION	–
Бюджет	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий Доп.КР	PARENT_DESCRIPTION	Определяется автоматически по текущему коду.
Наименование вышестоящего Доп.КР	–	–
Доп.КР	CODE	clcl Если входящий код меньше, чем в установленной маске для текущего кода, то добавляются нули в начале. Например, если маска 8 символов, а на входе код 900, то получаем 00000900.
Утвержден на уровне субъекта (муниципального образования)	APPROVED_ENTITY_LEVEL	Заполняется значением false .
Строки		
Краткое наименование	CAPTION	lcl_n
Полное наименование	DESCRIPTION	lcl_n
Не утвержден	UNAPPROVED	Заполняется значением false .
Начало действия	BEGIN_DATE	Заполняется датой начала текущего финансового года, который определяется из заголовка файла.
Окончание действия	END_DATE	Не заполняется.
Последнее изменение	CHANGE_DATE	–

- **Целевые назначения (Код цели)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Таблица 21 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–
VERSION	VERSION	–
Наименование бюджета	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий код	PARENT_DESCRIPTION	Не заполняется.
Наименование вышестоящего Доп. КР	–	–

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Атрибут АЦК-П	Атрибут XML
Код цели	CODE	cadd_bk
Наименование	DESCRIPTION	add_bk_n
Источник средств	PURPOSEFULGRANT_FSOURCE_CAPTION	Заполняется значением <i>Муниципальные средства.</i>
Форма предоставления	PAY_FORM	Заполняется значением <i>Иная.</i>
Порядок представления	ORDER_FORM	Заполняется значением <i>Не указан.</i>
КВД	KD_CODE	Заполняется значением <i>Не указан.</i>

• **Классификатор операций сектора государственного управления и аналитических кодов (КОСГУ)**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты справочника, которые указаны в таблице ниже.

Таблица 22 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Атрибут XML
ID	ID	–
VERSION	VERSION	–
Бюджет	BUDGET_ID	Определяется параметром BUDGET_ID .
Вышестоящий КОСГУ	PARENT_DESCRIPTION	Определяется автоматически по текущему коду.
Наименование вышестоящего Вышестоящего КОСГУ	–	–
КОСГУ	CODE	citem
Строки		
Краткое наименование	CAPTION	item_n
Полное наименование	DESCRIPTION	item_n
Не утвержден	UNAPPROVED	Заполняется значением false .
Начало действия	BEGIN_DATE	Заполняется датой начала текущего финансового года, который определяется из заголовка файла.
Окончание действия	END_DATE	Не заполняется.
Последнее изменение	CHANGE_DATE	–

• **Справочник Организации**

Обновление проводится по коду записи + бюджет. Заполняются только те атрибуты

Основные понятия
 Планировщик. Задания планировщика. Расписание запуска заданий

справочника, которые указаны в таблице ниже.

Таблица 23 – Описание атрибутивного состава запроса

Наименование поля	Атрибут АЦК-П	Значение для загрузки
ИНН	TAXCODE	corr_inn
КПП	KPP	corr_cpp
Краткое наименование	CAPTION	corr_sn
Полное наименование	DESCRIPTION	corr_n
ОКАТО	OKATO	Определяется значение через поле corr_cate .
Домен	DOMEN_ID	По умолчанию значение <i>верхнеуровневый</i> .
Территория	TERRITORY_ID	corr_cate Поиск ссылки производится по полю ОКТМО в справочнике <i>Территории</i> .
ППО	PPO_ID	Поиск ссылки производится в справочнике <i>Публично-правовые образования</i> по связанной территории в поле TERRITORY_ID .
–	ORGROLES (не заполняется)	Заполняется всеми ролями организации.

Правила загрузки:

- Данные загружаются в справочники: *Классификатор ведомственных статей расходов, Функциональный классификатор расходов, Классификатор целевых статей расходов, Классификатор видов расходов, Дополнительный код расхода, Организации.*
- Формат наименования файлов: <наименование выгружаемой таблицы>_<финансовый год>_<дата выгрузки в формате ДД. ММ. ГГГГ>_<время выгрузки (часы)>_<время выгрузки (минуты)>.

Например:

ReportSC_2022_27.10.2022_14_37,

где:

ReportSC – наименование выгружаемой таблицы,

2022 – финансовый год,

27.10.2022 – дата выгрузки,

14 – время выгрузки (часы),

13 – время выгрузки (минуты).

- В момент загрузки сопоставляется значения финансового года в заголовке и финансового года, указанного в параметре **BUDGET_ID**, если они не совпадают, то файл не обрабатывается.
- Если значение суммы не заполнено, то поле заполняется значением 0 (ноль), если поле КБК не заполнено, то поле заполняется значением справочника = Не указано.

3.8.37 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.

3.8.38 ImportDocLauncher

XML для запуска:

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

Параметры: OnlyWhenControlsOn, ProcessClasses, WorkDate, pg_work_mem.

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.
- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

- **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 на 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 на 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 на 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.
```

3.8.39 ImportEhdDictionaries

XML для запуска:

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

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 24 – 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
lM	int4	-	да
name	varchar	2000	да
parent_code	varchar	100	нет
subjectcode	varchar	100	да
parent_id	int8	-	нет

Таблица 25 – 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

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля в API-интерфейсе	Формат	Длина	Обязательность
Полное наименование	varchar	2000	нет
ИНН	varchar	255	нет
КПП	varchar	255	нет
ППО	varchar	255	нет
ОГРН	varchar	2000	нет
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

3.8.40 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 данные внешних систем*.

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

Рекомендуемый период запуска: 1 раз в сутки.

Процесс выполняет загрузку справочника *Финансовые данные внешних систем* 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.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Атрибутный состав:

После ответа от сервиса формируется набор с данными:

Таблица 26 – 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 .
–	–	–	–	Код цели	Заполняется значением <i>Не указано</i> .
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	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля в API-интерфейсе	Формат	Длина	Описание (от источника данных)	Поле таблицы справочника «Финансовые показатели (ЕХД)»	Логика заполнения
budgetExpendituresAddCode	varchar	100	Код из справочника <i>Дополнительный код расхода</i> .	Доп.КР	
amountAssignCurrent	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 данные внешних систем*:

- Проверяются строки на уникальность по полям: **На дату, КВСР, Код цели, 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.

3.8.41 ImportFinancingIndicatorsEHDI

XML для запуска:

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

Описание: Загрузка и импорт для справочника *Финансовые данные внешних систем* данных РЧБ из внешних систем.

Параметры: BUDGET_ID=<id бюджета>, file_path=<адрес папки>.

Рекомендуемый период запуска: 1 раз в сутки.

Процесс выполняет загрузку расходной части бюджета посредством файлового обмена в виде файлов формата *xml*. Данные загружаются в справочник *Финансовые данные внешних систем*.

Источником данных являются сторонние финансовые системы. Обмен производится без использования прямого подключения или обращения к БД сторонних систем.

Принимает параметры:

- **BUDGET_ID=<id бюджета>** – указывается ID бюджета, который будет присваиваться записям при обработке задания планировщика, а также определяет финансовый год файлов, которые необходимы для обработки, по ФГ указанного бюджета.
- **file_path=<адрес папки>** – путь к источнику файлов для загрузки.

Атрибутный состав:

После ответа от сервиса формируется набор с данными:

Таблица 27 – Описание атрибутивного состава запроса

Наименование поля	Атрибут XML	Наименование поля в API-интерфейсе	Формат	Длина	Значение для загрузки
Бюджет	Определяется настройкой планировщика.	–	–	–	Определяется параметром BUDGET_ID . Обязательное поле.
На дату	Определяется датой в заголовке.	ondate	timestamp	–	Определяется датой в заголовке наименования файла: <i>ReportSC_2022_27.10.2022_14_37</i> Обязательное поле.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Атрибут XML	Наименование поля в API-интерфейсе	Формат	Длина	Значение для загрузки
Организация	<i>dep_n</i>	budgetRecipientAccount	varchar	100	Поиск ссылки по полю Полное наименование в справочнике <i>Организации</i> из тега <i>dep_n</i> . Если найдено несколько организаций, берется первая попавшаяся запись без учета ID . Обязательное поле.
Код цели	<i>cadd_bk</i>	–	–	–	Поиск ссылки в справочнике <i>Целевые назначения</i> по полям Код + Бюджет .
КВСП	<i>cdep</i>	chiefManagerBudgetFunds	varchar	100	Поиск ссылки в справочнике <i>Классификатор ведомственных статей расходов</i> по полям Код + Бюджет .
КФСР	<i>cdiv</i>	budgetExpendituresSectorSubSector	varchar	100	Поиск ссылки в справочнике <i>Функциональный классификатор расходов</i> по полям Код + Бюджет .
КЦСР	<i>ctgt</i>	targetArticleClassifierCode	varchar	100	Поиск ссылки в справочнике <i>Классификатор целевых статей расходов</i> по полям Код + Бюджет .
КВР	<i>sexp</i>	budgetExpendituresClass	varchar	100	Поиск ссылки в справочнике <i>Код вида расхода</i> по полям Код + Бюджет .
Доп.ФК.	Отсутствует.	budgetExpendituresAddFuncCode	varchar	100	Заполняется значением <i>Не указано</i> .
Доп.ЭК.	Отсутствует.	budgetExpendituresAddEcoCode	varchar	100	Заполняется значением <i>Не указано</i> .
Доп. КР	<i>clcl</i>	budgetExpendituresAddCode	varchar	100	Поиск ссылки в справочнике <i>Дополнительный код расхода</i> по полям Код + Бюджет . При обработке входящим кодам слева добавляются нули, если значение кода меньше маски для текущего КБК. К примеру, входящий код <i>900</i> для маски из восьми символов (XXXXXXXX) обрабатывается как <i>00000900</i> .
Ассигнования (текущий финансовый год)	<i>s_sbr_y0</i>	amountAssignCur	numeric	–	Заполняется согласно источнику.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Атрибут XML	Наименование поля в API-интерфейсе	Формат	Длина	Значение для загрузки
Ассигнования (первый год планового периода)	<i>s_sbr_y1</i>	amountAssignFirst	numeric	–	Заполняется согласно источнику.
Ассигнования (второй год планового периода)	<i>s_sbr_y2</i>	amountAssignSecond	numeric	–	Заполняется согласно источнику.
Лимиты (текущий финансовый год)	<i>s_slbo_y0</i>	amountLimit	numeric	–	Заполняется согласно источнику.
Финансирование	<i>s_f_rr</i>	amountFund	numeric	–	Заполняется согласно источнику.
Кассовое исполнение	<i>s_f_rr</i>	amountCashExec	numeric	–	Заполняется значению колонки <i>Финансирование</i> .

Правила загрузки:

- Формат наименования файлов: <наименование выгружаемой таблицы>_<финансовый год>_<дата выгрузки в формате ДД. ММ. ГГГГ>_<время выгрузки (часы)>_<время выгрузки (минуты)>.

Например:*ReportSC_2022_27.10.2022_14_37,**где:**ReportSC – наименование выгружаемой таблицы,**2022 – финансовый год,**27.10.2022 – дата выгрузки,**14 – время выгрузки (часы),**13 – время выгрузки (минуты).*

- В момент загрузки сопоставляется значения финансового года в заголовке и финансового года, указанного в параметре **BUDGET_ID**, если они не совпадают, то файл не обрабатывается.
- Если значение суммы не заполнено, то поле заполняется значением 0, если в поле КБК пусто, то поле заполняется значением справочника = <Не указано>.
- Если файлов для обработки несколько, то обрабатываются все файлы.

- Если при следующей обработке задания планировщика файл за ту же дату, что и был ранее обработан, то полностью обновляются текущие строки за эту дату по данным нового файла.
- Если одна из ссылок в пакете не найдена, то весь пакет не обрабатывается и файл переходит в папку **Error** с указанием в логге наименования файла, и поля с ошибкой.

3.8.42 ImportFedProjects

XML для запуска:

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

Описание: Задание планировщика *ImportFedProjects* предназначено для загрузки и обновления справочников из API-интерфейса с открытых данных единого портала бюджетной системы. Загрузка выполняется путем запуска задания планировщика с заданными параметрами. Перечень справочников: *Результаты реализации проектов, Цели и задачи региональных проектов, Показатели результативности региональных проектов.*

Задание имеет следующие функциональные возможности:

1. Актуализация справочника с частотой, устанавливаемой администратором системы.
2. Возможность загрузки по API полной информации по справочникам, а также частично по установленным параметрам и фильтрам в строке запроса данных.
3. Задание позволяет устанавливать в параметрах возможность фильтрации по адресу загрузки (URL), т.е. URL=<http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data>.

Источником данных результатов реализации проектов являются сервисы на сайте Электронного бюджета: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data>. (Проверить актуальный адрес сервисов можно на сайте Электронного бюджета в разделе [путь] Данные и сервисы / Открытые данные / Информация об основных параметрах реализации региональных проектов, обеспечивающих достижение целей, показателей и результатов федеральных проектов, мероприятия которых относятся к законодательно установленным полномочиям субъекта Российской Федерации, а также к вопросам местного

значения муниципальных образований, расположенных на территории указанного субъекта Российской Федерации ([7710168360-FEDERALPROJECT](#))

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/7710168360-FEDERALPROJECT/data?>

[sortField=code&sortDir=desc](http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?sortField=code&sortDir=desc)

Для API предусмотрены следующие параметры запросов:

- **sortField** – параметр отвечает за сортировку данных по определенному атрибуту. При помощи параметра **sortDir** указывается порядок сортировки.
- **sortDir** – параметр указывает порядок сортировки данных (desc/asc):
 - для выполнения сортировки информации по коду федерального проекта в порядке убывания, в API-интерфейс устанавливаются значения параметров **sortField** и **sortDir=desc**: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?sortField=code&sortDir=desc>.
 - для выполнения сортировки информации по коду федерального проекта в порядке возрастания, в API-интерфейс устанавливаются значения параметров **sortField** и **sortDir=asc**: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?sortField=code&sortDir=asc>.
- **filter** – параметр позволяет осуществить фильтрацию информации по атрибутам.
- **pageSize** – параметр задает число отображаемых элементов на одной странице. Минимум на одной странице может быть отображен 1 элемент, максимум - 1000. По умолчанию выводится 10 элементов. Например, для отображения 100 элементов на одной странице в API-интерфейс устанавливается значение параметра **pageSize=100**: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?pageSize=100>.
- **pageNum** – параметр задает номер отображаемой страницы. Например, для отображения 2-й страницы, в API-интерфейс устанавливается значение параметра **pageNum=2**: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?pageNum=2>. При совместном использовании параметров **pageNum** и **offset** приоритетным будет параметр **offset**, так как позволяет более точно задать позицию.

- **offset** – параметр задает номер элемента, с которого нужно начать вывод данных. Например, для отображения данных с 11-го элемента в API-интерфейс устанавливается значение параметра `offset=10`: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?offset=10>. При совместном использовании параметров **pageNum** и **offset** приоритетным будет параметр **offset**, так как позволяет более точно задать позицию.
- **blocks** – параметр позволяет вывести в ответ только запрошенные наборы атрибутов (блоки). Например, для отображения блоков `participants` и `results` в API-интерфейс устанавливается: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?blocks=participants,results>.

Атрибутный состав записи для загрузки справочника *Результаты региональных проектов*

Основная информация включает набор атрибутов: `prcode` - код национального проекта, `code` - код федерального проекта, по которым возможна фильтрация при загрузке данных посредством планировщика.

Описание атрибутов блока PARTICIPANTS

Блок PARTICIPANTS с набором атрибутов: `metaid`, `recorded`, `startdate`, `enddate`, `fiо`, `headpost`, `immsupervisor`, `percemploy`, `roleinproj`, `recordupdate`; подблок `results` с набором атрибутов: `recorded`, `code`, `name`.

Для вывода блока в API-интерфейсе необходимо указать: <http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?blocks=participants>

Таблица 28 – Описание атрибутов блока PARTICIPANTS

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание
<code>metaid</code>	NUMBER	1	20	Да	Идентификатор всех версий записи реестра Участники федерального проекта
<code>recordid</code>	NUMBER	1	20	Да	Идентификатор записи реестра Участники федерального проекта
<code>startdate</code>	DATE	-	-	Да	Дата начала действия записи
<code>enddate</code>	DATE	-	-	Нет	Дата окончания действия записи

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание
fio	STRING	1	2000	Да	Фамилия, инициалы
headpost	STRING	0	2000	Нет	Должность
immsupervisor	STRING	0	2000	Нет	Непосредственный руководитель
percemploy	NUMBER	1	17,2	Да	Занятость в проекте (процентов)
roleinproj	STRING	1	2000	Да	Роль в федеральном проекте
recordupdate	DATE	-	-	Да	Дата обновления записи
results	Подблок «Результаты»				
recordid	NUMBER	1	20	Да	Идентификатор результата федерального проекта
code	STRING	1	255	Да	Код результата федерального проекта
name	STRING	1	2000	Да	Наименование результата федерального проекта

Описание атрибутов блока TASKS

Блок Tasks с набором атрибутов: metaid, recorded, startdate, enddate, code, name, realizationdate; подблок finsupports с набором атрибутов: metaid, recorded, startdate, enddate, fo2019, fo2020, fo2021, fo2022, fo2023, fo2024, finsource.

Для вывода блока в API-интерфейсе необходимо указать:

<http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data?blocks=tasks>

Таблица 29 – Описание атрибутов блока TASKS

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание
metaid	NUMBER	1	20	Да	Идентификатор всех версий задач федерального проекта
recordid	NUMBER	1	20	Да	Идентификатор задачи федерального проекта
startdate	DATE	-	-	Да	Дата начала действия записи
enddate	DATE	-	-	Нет	Дата окончания действия записи
code	STRING	1	10	Да	Код задачи
name	STRING	1	2000	Да	Наименование задачи

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание
recordupdate	DATE	-	-	Да	Дата обновления записи
nprecordid	NUMBER	1	20	Да	Идентификатор задачи национального проекта

Таблица 30 – Описание атрибутов подблока RESULTS блока TASKS

results	Подблок «Результат федерального проекта»				
metaid	NUMBER	1	20	Да	Идентификатор всех версий результата федерального проекта
recordid	NUMBER	1	20	Да	Идентификатор результата федерального проекта
startdate	DATE	-	-	Да	Дата начала действия записи
enddate	DATE	-	-	Нет	Дата окончания действия записи
code	STRING	1	255	Да	Код результата федерального проекта
name	STRING	1	2000	Да	Наименование результата федерального проекта
respexec	STRING	1	2000	Да	Ответственный исполнитель
numbercharact	STRING	1	2000	Да	Характеристика результата федерального проекта
typeres	STRING	1	2000	Да	Тип результата
costwaycode	STRING	0	5	Нет	Код направления расходов
directionexpenses	STRING	0	2000	Нет	Наименование направления расходов
direxpcoderesf	STRING	0	5	Нет	Код направления расходов_Резервный фонд
direxpnameresf	STRING	0	2000	Нет	Наименование направления расходов_Резервный фонд
isincludeinnp	NUMBER	0	1	Нет	Признак результата ФП Включен в НП Может принимать значения: 1 («Да»), 0 («Нет»), значение не указано.
recordupdate	DATE	-	-	Да	Дата обновления записи

Правила загрузки

Задание позволяет устанавливать в параметрах возможность фильтрации по:

- адресу загрузки (URL), т.е. URL=<http://budget.gov.ru/epbs/registry/7710168360-FEDERALPROJECT/data>;
- коду федерального проекта (code заголовочной части), т.е. пример: code=P1 - в этом случае загрузка должна происходить только по федеральному проекту "P1";
- коду национального проекта (prcode), т.е. пример prcode=P - в этом случае загрузка должна происходить только по национальному проекту "P".

Данные загружаются в справочник *Результаты реализации проектов* по следующему алгоритму:

Значение поля **recordid** найдено в поле **recordid_f** с признаком **Федеральный**

- Да, проверить условие:

проверить условие:

соответствие остальных полей **Наименование, Тип результата, Код результата,**

Дата окончания, Ответственный исполнитель, Характеристика:

да, не обновлять запись справочника *Результаты реализации проектов*

нет, обновить запись в справочнике *Результаты реализации проектов*

- Нет, добавить запись в справочник *Результаты реализации проектов* (в случае отсутствия соответствующих записей в справочниках *Типы результатов* и *Участники проектов* осуществляется добавление записей и в эти справочники).

Соответствие полей и правила загрузки данных

Данные загружаются из блоков PARTICIPANTS и подблока RESULTS блока TASKS.

Таблица 31 – Описание атрибутов блока TASKS

Наименование поля записи справочника «Результаты реализации проектов»/«Участники проектов»	Наименование атрибута	Правила загрузки
Справочник <i>Участники проектов</i> (заполняется в случае отсутствия соответствующей записи в справочнике для поля Ответственный исполнитель справочника <i>Результаты реализации проектов</i>)		
Блок PARTICIPANTS		

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля записи справочника «Результаты реализации проектов»/«Участники проектов»	Наименование атрибута	Правила загрузки
Роль участника проекта	roleinproj	Заполняется значением поля Участник проекта справочника <i>Роли участников проекта</i>
Фамилия, имя, отчество	fio	
Должность	headpost	Заполняется текстовым значением (без ссылки на справочник <i>Должности</i>)
Справочник <i>Результаты реализации проектов</i>		
Подблок RESULTS блока TASKS		
recordid_f	recordid	
признак «Федеральный» (CheckBox)	-	Признак Федеральный проставляется автоматически при добавлении записи в справочник посредством выполнения задания планировщика
Наименование	name	
Тип результата	types	Сопоставить со значением в поле Наименование справочника <i>Типы результата</i> . Если не совпадает - добавить запись в справочник <i>Типы результата</i>
признак «Загружен из ЭБ»	-	Выставляется активным.
признак «Актуальность»	-	Выставляется активным.
Единица измерения	okeicode	Заполняется ссылкой на запись справочника <i>Единицы измерения</i> (с активным признаком Актуальность) по соответствию ОКЕИ со значением поля okeicode блока RESULTS. Если соответствие не найдено или найдено неоднозначно, то заполняется записью справочника <i>Единицы измерения</i> с наименованием «НЕ УКАЗАНО» (id=0).
ОКЕИ	okeicode	Заполняется значением ОКЕИ найденной единицы измерения справочника <i>Единицы измерения</i> .

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля записи справочника «Результаты реализации проектов»/«Участники проектов»	Наименование атрибута	Правила загрузки
Подведение итогов	cbaccumulationtype	Заполняется: - значением <i>Накопительный итог</i> , если cbaccumulationtype блока RESULTS равно <i>Накопительный итог</i> ; - значением <i>Значение на последний год</i> , если cbaccumulationtype блока RESULTS равно <i>Значение на последний год</i> . Если cbaccumulationtype не заполнено, то поле заполняется значением <i>Значение на последний год</i> .
Вид	typevaluemrk	Заполняется: - значением <i>Возрастающий</i> , если typevaluemrk блока RESULTS равно <i>Возрастающий</i> ; - значением <i>Убывающий</i> , если typevaluemrk блока RESULTS равно <i>Убывающий</i> . Если поле typevaluemrk не заполнено, то поле заполняется значением <i>Возрастающий</i> .
Код результата	code	
Дата окончания	enddate	
Ответственный исполнитель	respexec	Сопоставить значение поля respexec подблока RESULTS, лока TASKS со значением поля fo блока PARTICIPANTS и определить значение поля headpost блока PARTICIPANTS. При сопоставлении полей respexec и headpost с полями Фамилия, имя, отчество и Должность справочника <i>Участники проектов</i> найти соответствующую запись и в поле Ответственный исполнитель создать на нее ссылку. Если совпадений в справочнике <i>Участники проектов</i> не найдено: добавить соответствующую запись в справочник <i>Участники проектов</i> с заполнением полей Роль участника проекта, Фамилия, имя, отчество и Должность

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля записи справочника «Результаты реализации проектов»/«Участники проектов»	Наименование атрибута	Правила загрузки
Характеристика	numbercharact	
Скрытое системное поле для хранения ID задачи национального проекта в результате	nprecordid	Храним с системном поле. Необходимо для автоматического подтягивания задачи национального проекта для строк федерального результата при загрузке региональных проектов.

Атрибутный состав записи справочника *Показатели результативности региональных проектов* и *Цели и задачи региональных проектов*

Основной блок

Таблица 32 – Описание атрибутов основного блока

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"
metaid	NUMBER	1	20	Да	Идентификатор всех версий федерального проекта	-
recordid	NUMBER	1	20	Да	Идентификатор федерального проекта	-
startdate	DATE	-	-	Да	Дата начала действия записи	-
enddate	DATE	-	-	Нет	Дата окончания действия записи	-
code	STRING	1	255	Да	Код федерального проекта	-
fullname	STRING	1	2000	Да	Полное наименование федерального проекта	-
shortname	STRING	1	2000	Да	Сокращенное наименование федерального проекта	-

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"
curator	STRING	0	2000	Нет	Куратор проекта	-
person	STRING	0	2000	Нет	Руководитель проекта	-
kvsr	STRING	0	2000	Нет	Администратор проекта	-
seniorofficial	STRING	0	2000	Нет	Старшее должностное лицо	-
startdateproj	DATE	-	-	Да	Дата начала федерального проекта	-
enddateproj	DATE	-	-	Да	Дата окончания федерального проекта	-
recordupdate	DATE	-	-	Да	Дата обновления записи	-
idnp	NUMBER	0	20	Нет	Идентификатор национального проекта	-
nrcode	STRING	0	512	Нет	Код национального проекта	-
nprname	STRING	0	2000	Нет	Наименование национального проекта	-

Описание атрибутов блока PURPOSES (цели, показатели и показатели по субъектам)

Из данного основного блока формируем запись в справочнике *Цели и задачи региональных проектов*. Всем записям должны быть присвоены поля:

- **Тип** = Цель федерального проекта/Задача федерального 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 загрузке записи в справочник *Цели и задачи региональных проектов* выполняем

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

также вставку в скрытое поле значение поля **code** основного блока, в котором будет храниться отнесение к коду федерального проекта. Поле заполнять только при загрузке текущим заданием и включается в уникальность. Необходимость вызвана дублями целей/задач, которые включены в разные федеральные проекты.

Таблица 33 – Описание атрибутов блока PURPOSES

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник «Показатели региональных проектов»	Соответствие с данными системы
metaid	NUMBER	1	20	Да	Идентификатор всех версий цели федерального проекта	-	
recordid	NUMBER	1	20	Да	Идентификатор цели федерального проекта	Да	Храним в скрытом системном поле recordid_fed. Новое скрытое системное поле в справочнике <i>Цели и задачи регионального проекта</i> .
startdate	DATE	-	-	Да	Дата начала действия записи	-	
enddate	DATE	-	-	Нет	Дата окончания действия записи	Да	Дата достижения
name	STRING	1	2000	Да	Наименование цели федерального проекта	Да	Наименование
recordupdate	DATE	-	-	Да	Дата обновления записи	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник «Показатели региональных проектов»	Соответствие с данными системы
recordid	NUMBER	1	20	Да	Идентификатор показателя национального проекта	-	
mrkname	STRING	1	2000	Да	Наименование показателя национального проекта	-	

Описание атрибутов подблока MARKS «Показатели федерального проекта»

Загрузку выполняем по данному блоку с учетом подблока SUBJECTMARKS.

Для загружаемой записи должен быть заполнен блок с показателями и их значениями.

При загрузке цели в блоке показателей необходимо заполнять значение поля **Вид**. Для заполнения проводим анализ значений подблока SUBJECTMARKS.

Сравниваем значение атрибутов value... с минимальным годом и с максимальным годом.

- Если значение с максимальным годом больше чем с минимальным годом, то вид показателя = Прогрессирующий;
- Если значение с максимальным годом меньше чем с минимальным годом, то вид показателя = Регрессирующий;
- Если значение с максимальным годом = значению с минимальным годом, то вид показателя = Стабильный.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Таблица 34 – Описание атрибутов подблока MARKS

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
metaid	NUMBER	1	20	Да	Идентификатор всех версий показателя федерального проекта	-	
recordid	NUMBER	1	20	Да	Идентификатор показателя федерального проекта	Да	Храним в скрытом системном поле recordid_fed . Новое скрытое системное поле в справочнике <i>Показатели результативности регионального проекта</i> .
startdate	DATE	-	-	Да	Дата начала действия записи		
enddate	DATE	-	-	Нет	Дата окончания действия записи		
mrkname	STRING	1	2000	Да	Наименование показателя	Да	Наименование показателя
typemrk	STRING	1	2000	Да	Тип показателя	Да	Если значение - Дополнительный, то устанавливаем активность признака Дополнительный показатель , иначе не устанавливаем. Всем загружаемым записям устанавливаем признак Федеральный проект .

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
okeirecordid	NUMBER	1	20	Да	Идентификатор актуальной версии записи справочника Единицы измерения	-	
okeicode	STRING	1	2000	Да	Код единицы измерения	-	
okeiname	STRING	1	2000	Да	Наименование единицы измерения	-	
basicvalueind	NUMBER	1	19,4	Да	Базовое значение показателя федерального проекта	-	
setdate	DATE	-	-	Да	Дата расчета базового значения показателя федерального проекта	-	
value2018	NUMBER	1	19,4	Да	Значение показателя на 2018 год	-	
value2019	NUMBER	1	19,4	Да	Значение показателя на 2019 год	-	
value2020	NUMBER	1	19,4	Да	Значение показателя на 2020 год	-	
value2021	NUMBER	1	19,4	Да	Значение показателя на 2021 год	-	
value2022	NUMBER	1	19,4	Да	Значение показателя на 2022 год	-	
value2023	NUMBER	1	19,4	Да	Значение показателя на 2023 год	-	
value2024	NUMBER	1	19,4	Да	Значение показателя на 2024 год	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
recordupdate	DATE	-	-	Да	Дата обновления записи	-	

Описание атрибутов подблока SUBJECTMARKS - Подблок «Показатели федерального проекта по субъектам РФ» подблока «Показатели федерального проекта»

Берутся только те значения, у которых значение поля **subjectcode** равно значению в поле **Код субъекта РФ** в системных параметрах в ветке «Региональные проекты».

Для загружаемой записи должен быть заполнен блок с показателями и их значениями.

Таблица 35 – Описание атрибутов подблока SUBJECTMARKS

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
subjectcode	STRING	1	2	Да	Код субъекта		Равно значению в поле Код субъекта РФ в системных параметрах в ветке «Региональные проекты»
subjectname	STRING	1	2000	Да	Наименование субъекта	-	
basicvalueid	NUMBER	1	19,4	Да	Базовое значение показателя по субъектам федерального проекта	Да	Поле Значение в строке раздела «Значения показателя» с типом значения <i>Базовое</i> .

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
setdate	DATE	-	-	Да	Дата расчета базового значения показателя по субъектам федерального проекта	Да	Поле На дату в строке раздела «Значения показателя» с типом значения <i>Базовое</i> . Загружается в одной строке для базового показателя
value2018	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2018 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения <i>Плановое</i> , где Период = 2018. Определяется по названию атрибута value
value2019	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2019 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения = <i>Плановое</i> , где Период = 2019. Определяется по названию атрибута value
value2020	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2020 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения = <i>Плановое</i> , где Период = 2020. Определяется по названию атрибута value

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
value2021	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2021 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения = <i>Плановое</i> , где Период = 2021. Определяется по названию атрибута value
value2022	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2022 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения = <i>Плановое</i> , где Период = 2022. Определяется по названию атрибута value
value2023	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2023 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения = <i>Плановое</i> , где Период = 2023. Определяется по названию атрибута value
value2024	NUMBER	1	19,4	Да	Значение показателя по субъектам на 2024 год	Да	Поле Значение в строке раздела «Значения показателя» с типом значения = <i>Плановое</i> , где Период = 2024. Определяется по названию атрибута value
okeirecordid	NUMBER	1	20	Да	Идентификатор записи справочника Единицы измерения	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля	Формат	Минимальная длина	Максимальная длина	Обязательность	Примечание	Загрузка в справочник "Показатели региональных проектов"	Соответствие с данными системы
okeicode	STRING	1	2000	Да	Код единицы измерения	Да	Осуществляется поиск в справочнике <i>Единицы измерения</i> в группе «Не указано» с аналогичным значением в поле Код ОКЕИ . Если найдено, то подставляется, если не найдено, ошибку в лог. Если найдено несколько, то на любой их них.
okeiname	STRING	1	2000	Да	Наименование единицы измерения	-	
recordupdate	DATE	-	-	Да	Дата обновления записи	-	

Правила загрузки

1. Берутся данные API и осуществляется поиск в системе записи с аналогичным значением в поле **recordid_fed** записи справочника *Цели и задачи региональных проектов*. Если запись найдена, то сравниваются значения основных атрибутов записи, если отличаются, то осуществляется обновление, если нет, то пропускается и далее осуществляется переход к показателям. Если запись отсутствует, то вставляется в справочник.
2. По аналогии с пунктом 1 осуществляется поиск, обновление и вставка записи в справочнике *Показатели результативности региональных проектов*, формируя при этом связь с целью/задачей в форме редактирования записи справочника *Цели и задачи региональных проектов*.

3.8.43 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, BUDGET_ID=<id бюджета.

Описание: Задание планировщика *ImportRegProjCard* 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_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.
- **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.

*Примечание. Если "name" блока "roiv" = пусто, то текущее ** значение поля **Региональный орган исполнительной власти** не обновляется (не очищается). Пример входящих данных:*

```
"roiv": {
```

```
  "recordid": "",
```

```
  "name": ""
```

3.8.44 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_nac.
- 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).

3.8.45 ImportUgpSubjectBroadcast

XML для запуска:

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

Описание: Получение данных о показателях и результатах (мероприятиях) паспортов ГП

По планировщику пополняются внутренние справочники Результаты (мероприятия) и Показатели (индикаторы) результативности, а также уточняются ранее загруженные значения, пополняются значения внешних справочников Типы результатов и типовые наименования контрольных точек РФ, Субъекты РФ.

Пополнение значений справочника *Результаты (мероприятия)*

В полученном xml используются только записи с тегом SubjectCode группы Subject равным системному параметру **Код субъекта** и тегом SubjectId, равному **ID субъекта в ЭБ**, расположенному в **Системные параметры→Планирование**.

Из полученного xml обновляются записи справочника *Результаты (мероприятия)* при выполнении одного из следующих условий:

- значения поля **Код федерального уровня** и значения в теге FederalUniqueResultCode подподгруппы GeneralInformation подгруппы Result группы Results и значение поля **Источник загрузки** равно *Данные ГП РФ*;
- значения поля **Наименование** и значения в теге ResultsName подподгруппы GeneralInformation подгруппы Result группы Results и значение поля **Источник загрузки** равно *Данные ГП РФ*.

Если значение поля **Код федерального уровня** и значение в теге FederalUniqueResultCode подгруппы GeneralInformation группы Result не совпадают, создается новую запись в справочнике *Результаты (мероприятия)*; в поле **Код федерального уровня** записывается значение из тега FederalUniqueResultCode.

Внутри тега SendingResults/Results/Result/GeneralInformation вложены теги с значениями полей записи справочника:

Наименование поля в АЦК	Формат АЦК	Наименование атрибута в ЭБ	Формат ЭБ	Правила заполнения

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Код федерального уровня	Текст, 10 символов	FederalUniqueResultCode	Текст, 10 символов <i>Пример: X412670000</i>	Загружаемое, недоступное для редактирования. Если такое значение существует, то остальные поля перезаписываются новыми значениями, если нет, то создается новая запись справочника.
Код	Текст, 10 символов	UniqueResultCode	Текст, 10 символов <i>Пример: X412670000</i>	«Уникальный код результата» Загружаемое, недоступное для редактирования.
Этап	перечисление:: 10 - Исполнение, 20 - Планирование ФЗО ЭБ	Stage	(перечисление: 10 - Исполнение, 20 - Планирование ФЗО ЭБ	«Этап» Загружаемое, скрытое.
Наименование результата	Текст, 2000 символов	ResultsName	Текст, 2000 символов <i>Пример: Созданы и функционируют Центры опережающей профессиональной подготовки</i>	Загружаемое, доступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Мета ID результата	Текст, 36 символов	VersionMetald	Текст, 36 символов Пример: 9934b332-fea5-4b80-872d-12cde48ec28d	Скрытое. Загружаемое, недоступное для редактирования. При загрузке из ЭБ через СМЭВ перезаписывать поле Версия только в случае когда пришедший номер версии больше чем имеется в справочнике (при совпадении поля Мета ID и тега xml VersionMetald).
Версия результата	Число, целое	VersionNumber	Число, целое Пример: 1	Скрытое. Загружаемое, недоступное для редактирования.
Признак Финансовый	Булево true=1; false=2	MonetaryType	Число целое 1 значение Пример: 1	Загружаемое, недоступное для редактирования. Загруженный финансовый тип имеет варианты: 1 - финансовый; 2 - нефинансовый

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Единица измерения	Ссылка на справочник <i>Единицы измерения</i>	ОКЕИ ОКЕИCode	Строка 100 символов <i>Пример: 643</i>	Загружаемое, доступное для редактирования. По полученному коду по совпадению поля Код ОКЕИ определяется ссылка на справочник <i>Единицы измерения</i> .
Год базового значения	Число, целое, 4 знака	BasicValue Year	Число целое 4 знака <i>Пример: 2020</i>	Загружаемое, недоступное для редактирования.
Базовое значение	Число с запятой	BasicValue Value	Число с запятой <i>Пример: 0.0</i>	Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

<p>Тип результата</p>	<p>Ссылка на справочник <i>Типы результатов</i></p>	<p>ResultType ResultTypeId ResultTypeName</p>	<p>Строка <i>Пример: Создание (реорганизация) организации (структурного подразделения)</i></p>	<p>Загружаемое. Сначала определяется по совпадению тега "ResultTypeId" и поля "ИД типа результата из ЭБ" справочника "Типы результатов", а за тем (если не найдено) по полученному наименованию результата по полному совпадению текста с полем ResultTypeName определяется ссылка на справочник "Типы результатов"</p> <ul style="list-style-type: none"> • Если найдено – заполняется поле Типы результата значением найденной строки справочника; • если не найдено :
------------------------------	---	---	--	---

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

				<ul style="list-style-type: none"> • заполняется поле Тип результата ссылкой на строку справочника с наименованием <i>Иные мероприятия (результаты)</i> • заполняется поле Описание типа результата значением из поля ResultTypeName
Дата начала	Дата	ImplementationPeriod DateStart	Дата <i>Пример: 2022-01-01T00:00:00</i>	Загружаемое, недоступное для редактирования.
Дата окончания	Дата	ImplementationPeriod DateEnd	Дата <i>Пример: 2022-01-01T00:00:00</i>	Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Признак Актуальность	Булево false= true: true= false	AchievementStatus	Булево <i>Пример: false</i>	Загружаемо е, доступное для редактирова ния. Значение меняется на обратное, если загружено false то признак актуальност и true и наоборот.
Вид результ ата	Значени е из списка Убываю щий=1; Возраст ающий= 2; Не указано =3	Dynamic	Число целое 1 знак <i>Пример: 2</i>	Загружаемо е, доступное для редактирова ния. Загруженная динамика имеет виды: 1 - Убывающая; 2 - Возрастающ ая; 3 - Отсутствует
Подвед ение итогов	Значени е из списка Накопит ельный итог=1; Значени е на последн ий год=2; Не указано =3	TypeCalculation	Число целое 1 знак <i>Пример: 2</i>	Загружаемо е, доступное для редактирова ния. Загруженны й тип расчета имеет варианты: 1 - Накопитель ный итог; 2 - Дискретно; 3 - Отсутствует

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Источник загрузки	Значение из списка Данные ГП РФ			Доступное для редактирования, для загруженных записей всегда <i>Данные ГП РФ.</i>
Уровень результата	Федеральный			Для загруженных сведений одно состояние <i>Федеральный.</i>
Принадлежность результата	Не указано			Для загруженных сведений одно состояние <i>Не указано.</i>
Закладка <u>Федеральные КБК</u>				
Код для строки таблиц	Текст, 200 символов	FederalКБК	Текст, 26 символов	Загружаемое, недоступное для редактирования. Автонумерация строки. создается столько строк, сколько имеется кодов КБК в тегах, относящихся к результату.
Закладка <u>Значения по годам</u> с таблицей				

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Год для строки таблицы	Число целое 4 знака	ResultValues Years ValuesYear Year	Число целое 4 знака <i>Пример: 2024</i>	Загружаемое, недоступное для редактирования. Сортировка по возрастанию лет. Автонумерация строки. создается столько строк, сколько имеется перечислений лет в тегах, относящихся к результату.
Значение для строки таблицы	Число с десятичной запятой	ResultValues Years ValuesYear Value	Число с десятичной запятой <i>Пример: 2.0</i>	Загружаемое, недоступное для редактирования. В созданную выше строку с годом добавляем значение.
Мета ИД значения результата	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример: 9934b332-fea5-4b80-872d-12cde48ec28d</i>	Скрытое. Загружаемое, недоступное для редактирования.
Версия значения результата	Число, целое	VersionNumber	Число, целое <i>Пример: 1</i>	Скрытое. Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Дата окончания значения	Дата	ResultValues Years ValuesYear DateEnd	Дата Пример: 2024-12-31T00:00:00	Загружаемое, недоступное для редактирования. В созданную выше строку с годом добавляем дату окончания для значения.
Закладка <u>Связь со структурным элементом ГП РФ</u>				
Код программы	Число, целое	ResultsMarking Links FederalStructuralElement StateProgramCode	Текст 2 знака Пример: 01	Загружаемое, недоступное для редактирования.
Наименование программы	Текстовое, 2000 символов	ResultsMarking Links FederalStructuralElement StateProgramName	Текст, 2000 символов Пример: Развитие здравоохранения	Загружаемое, недоступное для редактирования.
Мета ID программы	Текст, 36 символов	VersionMetald	Текст, 36 символов Пример: 9934b332-fea5-4b80-872d-12cde48ec28d	Скрытое. Загружаемое, недоступное для редактирования.
Версия программы	Число, целое	VersionNumber	Число, целое Пример: 1	Скрытое. Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Тип элемента	Значение из списка: Федеральный проект =10; Комплекс процессных мероприятий = 30	ResultsMarking Links FederalStructuralElement FederalStructuralElementType	Число целое <i>Пример: 10</i>	Загружаемое, недоступное для редактирования. Загруженный федеральный структурный элемент имеет виды: 10 - Федеральный проект 30 - Комплекс процессных мероприятий
Наименование элемента	Текст, 2000 символов	ResultsMarking Links FederalStructuralElement FederalStructuralElementName	Текст, 2000 символов <i>Пример: Профессионалитет</i>	Загружаемое, недоступное для редактирования.
Мета ИД элемента	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример: 9934b332-fea5-4b80-872d-12cde48ec28d</i>	Скрытое. Загружаемое, недоступное для редактирования.
Версия элемента	Число, целое	VersionNumber	Число, целое <i>Пример: 1</i>	Скрытое. Загружаемое, недоступное для редактирования.
Закладка Контрольные точки узел SendingDocuments\SendingResults\Results\Result\ResultValues Years\Values Year\Checkpoints\Checkpoint				
Наименование	Текст, 2000 символов	CheckpointName	Текст, 2000 символов <i>Пример: Развитие здравоохранения</i>	Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

<p>Типовое наименование контрольной точки</p>	<p>Ссылка на поле "Наименование" справочника "Типы контрольных точек"</p>	<p>Выбор пользователя</p>		<p>Заполняется после загрузки по выбору пользователя из справочника <i>Типы контрольных точек</i> список доступных значений предфильтрован по соответствию поля Тип результата текущей строки справочника <i>Мероприятия</i> (<i>результаты</i>).</p>
<p>Срок выполнения</p>	<p>Дата</p>	<p>DateEnd</p>	<p>Дата <i>Пример: 2024-12-31T00:00:00</i></p>	<p>Загружаемое, недоступное для редактирования. Если дата пустая, подставляется значение поля DateEnd вышестоящего значения результата по году.</p>

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

<p>Вид контрольной точки</p>	<p>выбор из списка если 0, то выбирается "Специальная"; если 2, то выбирается "Рекомендуемая"; если 3, то выбирается "Дополнительная";</p>	<p>AdditionalControlPointType</p>	<p>Целое число <i>Пример:</i> 0 - <i>Специальная</i>; 2 - <i>Рекомендуемая</i>; 3 - <i>Дополнительная</i></p>	<p>Загружаемое, недоступное для редактирования.</p>
<p>Значение</p>	<p>Десятичное число</p>	<p>Value</p>	<p>Десятичное число</p>	<p>Загружаемое, недоступное для редактирования.</p>
<p>Код ОКЕИ</p>	<p>Ссылка на поле "Код ОКЕИ" справочника "Единицы измерения"</p>	<p>ОКЕИ ОКЕИCode</p>	<p>Строка 100 символов</p>	<p>Загружаемое, недоступное для редактирования.</p>

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Единица измерения	Ссылка на поле "Описание" справочника "Единицы измерения" по соответствию предшествующего поля "Код ОКЕИ"	Определяется по значению предшествующего поля		недоступное для редактирования.
Не актуально	булево	Actuality	булево	Загружаемое, недоступное для редактирования.
Идентификатор типовой контрольной точки	long	AdditionalControlPointId	long	Загружаемое, недоступное для редактирования.
Мета ИД контрольной точки	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример:</i> 9934b332-fea5-4b80-872d-12cde48ec28d	Скрытое. Загружаемое, недоступное для редактирования.
Версия контрольной точки	Число, целое	VersionNumber	Число, целое <i>Пример:</i> 1	Скрытое. Загружаемое, недоступное для редактирования.
Закладка Объекты узел SendingDocuments\SendingResults\Results\Result\Result\Values Years\Values Year\Objects\Object				
Год	целочисленное 4 знака	ResultValues Years\Values Year	целочисленное 4 знака	Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Тип объекта	выбор из списка если 1, то выбирается "ОКС"; если 2, то выбирается "Иные";	ObjectType	Целое число Пример: 1 - ОКС; 2 - Иные	«Тип объекта» Загружаемое, недоступное для редактирования.
Форма собственности	выбор из списка если 1, то выбирается "Федеральная"; если 2, то выбирается "Региональная"; ; если 3, то выбирается "Муниципальная";	OwnershipForm	Целое число Пример: 1 - Федеральная, 2 - Региональная; 3 - Муниципальная	«Форма собственности» Загружаемое, недоступное для редактирования
Код ОКС	целочисленное текст преобразуем в число	ObjectCodeOKS	Текст, 2000 символов	«Уникальный код объекта «Строительство» Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование	Текст, 2000 символов	ObjectName	Текст, 2000 символов <i>Пример: Развитие здравоохранения</i>	«Наименование объекта» Загружаемое, недоступное для редактирования.
Значение (Мощность)	Десятичное число	Value	Десятичное число	«Значение» Загружаемое, недоступное для редактирования.
Единица измерения (Мощность)	Ссылка на поле "Описание" справочника "Единицы измерения" где поле "Код ОКЕИ"= ОКЕИCode	ОКЕИ ОКЕИCode	строка, 100 символов	«Код ОКЕИ» Загружаемое, недоступное для редактирования.
Характеристика	Текстовое 8000 символов	Description	Тестовое 8000 символов	«Характеристика объекта» Загружаемое, недоступное для редактирования.
ОКТМО	Текстовое 11 символов	ОКТМО ОКТМОCode	Текстовое 11 символов	«Код ОКТМО» Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Адрес объекта в соответствии с ФИАС	Текст, 2000 символов	ObjectAddress	Текст, 2000 символов	«Адрес объекта в соответствии с ФИАС» Загружаемое, недоступное для редактирования.
Информационная система	Текст, 2000 символов	DataSource	Текст, 2000 символов	«Информационная система (источник данных)» Загружаемое, редактируемое.
Мета ID объекта	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример: 9934b332-fea5-4b80-872d-12cde48ec28d</i>	«Сквозной идентификатор объекта» Скрытое. Загружаемое, недоступное для редактирования.
Версия объекта	Число, целое	VersionNumber	Число, целое <i>Пример: 1</i>	«Номер версии» Скрытое. Загружаемое, недоступное для редактирования.
Закладка Объекты , закладка Контрольные точки узел SendingDocuments\SendingResults\Results\Result\Result\Values Years\Values Year\Objects\Object\Checkpoints\Checkpoint				

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование	Текст, 2000 символов	CheckpointName	Текст, 2000 символов <i>Пример: Развитие здравоохранения</i>	«Наименование контрольной точки» Загружаемое, недоступное для редактирования.
Срок выполнения	Дата	DateEnd	Дата <i>Пример: 2024-12-31T00:00:00</i>	«Дата окончания» Загружаемое, недоступное для редактирования. Если дата пустая, то подставляем значение поля DateEnd вышестоящего значения результата по году

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Вид контрольной точки	выбор из списка если 0, то выбирается "Специальная"; если 1, то выбирается "Обеспечивающая"; если 2, то выбирается "Рекомендуемая"; если 3, то выбирается "Дополнительная";	AdditionalControlPointType	Целое число Пример: 0 - Специальная; 1 - Обеспечивающая; 2 - Рекомендуемая; 3 - Дополнительная	«Тип контрольной точки» Загружаемое, недоступное для редактирования.
Значение	Десятичное число	Value	Десятичное число	«Значение» Загружаемое, недоступное для редактирования.
Код ОКЕИ	Ссылка на поле "Код ОКЕИ" справочника "Единицы измерения"	ОКЕИ ОКЕИCode	Строка 100 символов	«Код ОКЕИ» Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Единица измерения	Ссылка на поле "Описание" справочника "Единицы измерения" по соответствию предшествующего поля "Код ОКЕИ"	Определяется по значению предшествующего поля		«Наименование ОКЕИ» недоступно для редактирования.
Не актуально	булево	Actuality	булево	«Признак неактуальности КТ» Загружаемое, недоступно для редактирования.
Идентификатор типовой контрольной точки	long	AdditionalControlPointId	long	«Идентификатор типовой контрольной точки из справочника «Типовые КТ» Загружаемое, недоступно для редактирования.
Мета ИД контрольной точки	Текст, 36 символов	VersionMetald	Текст, 36 символов Пример: 9934b332-fea5-4b80-872d-12cde48ec28d	«Сквозной идентификатор контрольной точки» Скрытое. Загружаемое, недоступно для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

<p>Версия контрольной точки</p>	<p>Число, целое</p>	<p>VersionNumber</p>	<p>Число, целое <i>Пример: 1</i></p>	<p>«Номер версии» Скрытое. Загружаемое, недоступное для редактирования.</p>
<p>Закладка <u>Объекты</u> закладка <u>Объем финансового обеспечения</u> узел SendingDocuments\SendingResults\Results\Result\ResultValues Years\Values Year\Objects\Object\ObjectFinancialSources\SourceFinancing</p>				

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Колонка с источником финансирования	Определение колонки для заполнения, при этом строка определяется годом для "31" заполняем "Всего"; для "32" заполняем "Федеральные средства"; для "33" заполняем "Бюджет ФОМС"; для "34" не передаем; для "35" заполняем "Внебюджетные источники"	SourceFinancingId	long <i>Пример: 31</i>	«Идентификатор из справочника «Источники финансирования» Загружаемое, недоступное для редактирования.
не заполняем		SourceFinancingName	Тестовое 2000 символов <i>Пример:</i> <i>1. Всего</i> <i>2. Бюджет субъекта РФ</i> <i>3. Из них: консолидированный бюджет субъекта рф</i> <i>4. Из них в том числе за счёт мбт из фб бюджетам рф</i>	«Наименование источника финансирования» Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

			<p>5. <i>Бюджет территориального государственного внебюджетного фонда (бюджет тфомс)</i></p> <p>6. <i>Внебюджетные источники</i></p>	
Год	Целочисленное 4 знака	FinancialSecurityYears Year Year	Целочисленное	«Финансовое обеспечение по году» Загружаемое, недоступное для редактирования.
Значение	Десятичное число	Value	Десятичное число	«Значение финансового обеспечения» Загружаемое, недоступное для редактирования.
<p>Закладка Объекты закладка Характеристики объекта</p> <p>узел SendingDocuments\SendingResults\Results\Result\ResultValues Years\Values Year\Objects\Object\ObjectCharacteristics\ObjectCharacteristic</p>				
Мета ИД характеристики объекта	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример: 9934b332-fea5-4b80-872d-12cde48ec28d</i>	«Сквозной идентификатор характеристики объекта» Скрытое. Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Версия характеристики объекта	Число, целое	VersionNumber	Число, целое <i>Пример: 1</i>	«Номер версии» Скрытое. Загружаемое, недоступное для редактирования.
Наименование характеристики	Текст, 2000 символов	CharacteristicName	Текст, 2000 символов <i>Пример: Развитие здравоохранения"</i>	«Наименование характеристики» Загружаемое, недоступное для редактирования.
Год	Целочисленное 4 знака	Year	Целочисленное	«Год» Загружаемое, недоступное для редактирования.
Значение	Десятичное число	Value	Десятичное число	«Значение» Загружаемое, недоступное для редактирования.
Код ОКЕИ	Ссылка на поле "Код ОКЕИ" справочника "Единицы измерения"	ОКЕИ ОКЕИCode	Строка 100 символов	«Код ОКЕИ» Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Единица измерения	Ссылка на поле "Описание" справочника "Единицы измерения" по соответствию предшествующего поля "Код ОКЕИ"	Определяется по значению предшествующего поля		«Наименование ОКЕИ» недоступное для редактирования.
--------------------------	---	---	--	---

Пополнение значений справочника Показатели (индикаторы) результативности

В полученном xml используются только записи с тегом SubjectCode группы Subject равным системному параметру **Код субъекта** и тегом SubjectId, равному **ID субъекта в ЭБ**, расположенному в **Системные параметры**→**Планирование**.

Из полученного xml обновляются записи справочника *Показатели (индикаторы) результативности* при одновременном совпадении одной из комбинаций:

- значение поля **Источник загрузки** равно *Данные ГП РФ*, и значение поля **Мета ID показателя** и значения в теге VersionMetald подподгруппы GeneralInformation подгруппы Indicator группы Indicators;
- значение поля **Источник загрузки** равно *Данные ГП РФ* и значение поля **Наименование показателя** совпадает с значением тега IndicatorName.

Если условия не выполняются, то создается новая запись в справочнике *Показатели (индикаторы) результативности*, в поле **Мета ID показателя** записывается значение из тега VersionMetald.

Внутри тега GeneralInformation вложены теги с значениями полей записи справочника:

Основные понятия
 Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля в АЦК	Формат АЦК	Наименование атрибута в ЭБ	Формат ЭБ	Правила заполнения
Этап	перечисление:: 10 - Исполнение, 20 - Планирование ФЗ о ЭБ	Stage	(перечисление: 10 - Исполнение, 20 - Планирование ФЗ о ЭБ	«Этап» Загружаемое, скрытое.
Наименование показателя	Текст, 2000 символов	IndicatorName	Текст, 2000 символов <i>Пример: Доступность дошкольного образования для детей в возрасте от 1,5 до 3 лет</i>	Загружаемое, доступное для редактирования

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

М е т а И Д п о к а з а т е л я	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример:</i> adcaecdc-413c-8a6d-b772-b9b86e6a33d5	Скрытое. Загружаемое, недоступное для редактирования. При загрузке из ЭБ через СМЭВ перезаписывается поле Версия только в случае когда пришедший номер версии больше чем имеется в справочнике (при совпадении поля Мета ID и тега xml VersionMetald).
В е р с и я п о к а з а т е л я	Число, целое	VersionNumber	Число, целое <i>Пример:</i> 20	Скрытое. Загружаемое, недоступное для редактирования.
У р о в е н ь п о к а з а т е л я	Текстовое 200 знаков «ГП РФ»=1; «ФП вне НП»=2; «КПМ»=3	IndicatorLevel	Число целое 1 знак <i>Пример:</i> 1	Загружаемое, доступное для редактирования. Загруженный уровень показателя имеет виды: 1 - «ГП РФ», 2 - «ФП вне НП», 3 - «КПМ»

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Динамика	Значение из списка Возрастающий=1; Убывающий=2; Не указано=3	Dynamic	Число целое 1 знак <i>Пример: 1</i>	Загружаемое, доступное для редактирования. Загруженная динамика имеет виды: 1 - Возрастающий; 2 - Убывающий; 3 - Отсутствует
Тип расчета	Значение из списка Накопительный=1; Дискретный=2; Не указано=3	TypeOfCalculation	Число целое 1 знак <i>Пример: 2</i>	Загружаемое, доступное для редактирования. Загруженный тип расчета имеет виды: 1 - Накопительный итог; 2 - Дискретно; 3 - Отсутствует
Варианты планирования значения	Значение из списка <ul style="list-style-type: none"> • Годовое = 1 • Годовое помесечной детализацией=3 • Полугодовое • Квартальное =2 • Месячное • На определенную дату=4 	Periodicity	Число целое 1 знак <i>Пример: 2</i>	Загружаемое, доступное для редактирования. Загруженная периодичность имеет виды: 1 - Годовая; 2 - Квартальная; 3 - Месячная; 4 - Отсутствует

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Е Д и н и ц а и з м е р е н и я	Ссылка на справочник "Единицы измерения"	ОКЕИ ОКЕICode	Строка 100 символов <i>Пример: 643</i>	Загружаемое, доступное для редактирования. По полученному коду по совпадению поля Код ОКЕИ определяется ссылка на справочник <i>Единицы измерения</i> .
Закладка <i>Значения по годам</i>				
Г о д б а з о в о г о з н а ч е н и я	Число целое 4 знака	BasicValue Year	Число целое 4 знака <i>Пример: 2020</i>	Загружаемое, недоступное для редактирования.
Б а з о в о е з н а ч е н и е	Число с запятой	BasicValue Value	Число с запятой <i>Пример: 0,0</i>	Загружаемое, недоступное для редактирования.
Закладка <i>Значения по годам</i> Таблица заполняется из значений группы PlanIndicator.				

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Г О Д Д Л Я С Т Р О К И Т А Б Л И Ц Ы	Число целое 4 знака	PlanIndicator ValuesYear Year	Число целое 4 знака <i>Пример: 2024</i>	Загружаемое, недоступное для редактирования. Сортировка по возрастанию лет. Автонумерация строки. Создается столько строк, сколько имеется перечислений лет в тегах, относящихся к результату.
З н а ч е н и е Д Л Я С Т Р О К И Т А Б Л И Ц Ы	Число с десятичной запятой	PlanIndicator ValuesYear Value	Число с десятичной запятой <i>Пример: 100.8</i>	Загружаемое, недоступное для редактирования. В созданную выше строку с годом добавляем значение.
Закладка <i>Связь со структурным элементом ГП РФ</i>				

Основные понятия
 Планировщик. Задания планировщика. Расписание запуска заданий

К О Д п р о г р а м м ь	Число, целое	IndicatorsMarking Links StateProgramCode	Текст 2 знака <i>Пример: 02</i>	Загружаемое, недоступное для редактирования.
Н а и м е н о в а н и е п р о г р а м м ь	Текст, 2000 символов	IndicatorsMarking Links StateProgramName	Текст, 2000 символов <i>Пример: Развитие образования</i>	Загружаемое, недоступное для редактирования.
М е т а И Д п р о г р а м м ь	Текст, 36 символов	VersionMetalId	Текст, 36 символов <i>Пример: ec3706f3-4b19-8391-4c5a-e39c82fafac9</i>	Скрытое. Загружаемое, недоступное для редактирования.

Основные понятия
 Планировщик. Задания планировщика. Расписание запуска заданий

Версия программы	Число, целое	VersionNumber	Число, целое <i>Пример: 45</i>	Скрытое. Загружаемое, недоступное для редактирования.
Тип элемента	Значение из списка: Федеральный проект =10; Комплекс процессных мероприятий = 30	IndicatorsMarking Links FederalStructuralElement FederalStructuralElementType	Число целое <i>Пример: 10</i>	Загружаемое, недоступное для редактирования. Загруженные федеральные типы структурных элементов имеют варианты: 10 - Федеральный проект 30 - Комплекс процессных мероприятий
Наименование элемента	Текст, 2000 символов	IndicatorsMarking Links FederalStructuralElement FederalStructuralElementName	Текст, 2000 символов <i>Пример: Профессионалитет</i>	Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

М е т а И д э л е м е н т а	Текст, 36 символов	VersionMetald	Текст, 36 символов <i>Пример:</i> 9934b332-fea5-4b80-872d-12cde48ec28d	Скрытое. Загружаемое, недоступное для редактирования.
В е р с и я э л е м е н т а	Число, целое	VersionNumber	Число, целое <i>Пример: 1</i>	Скрытое. Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование родител'sкого показателя ГПРФ	Текст, 2000 символов	FederalIndicator IndicatorName	Текст, 2000 символов <i>Пример: Удвоение ВВП</i>	Загружаемое, недоступное для редактирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Метаидентификаторского показателя	Текст, 36 символов	FederalIndicator VersionMetald	Текст, 36 символов Пример: 9934b332-fea5-4b80-872d-12cde48ec28d	Скрытое. Загружаемое, недоступное для редактирования.
-----------------------------------	--------------------	-----------------------------------	---	---

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Версия родителяского показателя	Число, целое	FederalIndicator VersionNumber	Число, целое <i>Пример: 1</i>	Скрытое. Загружаемое, недоступное для редактирования.
---------------------------------	--------------	-----------------------------------	----------------------------------	---

Загрузка справочников *Типы результатов*, *Типы контрольных точек* и *Соответствие Типов субсидий*, *Типов результатов* и *Типов контрольных точек*

В первую очередь заполняется справочник *Типы результатов* значениями из CheckPointTypesRF, которые загружаются из ЭБ по планировщику, при этом:

- если загружается тип результата, в котором значение тега ResultTypeName имеется в поле **CAPTION** строки справочника *Типы результатов*, то в связанной таблице производится поиск по совпадению поля **ИД типа результата из ЭБ** и тега ResultTypeId;
- если совпадение не найдено, заполняются поля связанной таблицы;
- если совпадение найдено, то обновляются сведения в остальных полях строки связанной таблицы;
- если значение тега ResultTypeName не встречается в поле **CAPTION** справочника *Типы результатов*, формируется новая строка по правилу:

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля в АЦК	Формат АЦК	Наименование атрибута в ЭБ	Формат ЭБ	Правила заполнения
CAPTION	Текст, 2000 символов	SpecialControlPoints ResultsType ResultTypeName	Текст, 2000 символов <i>Пример:</i> <i>Строительство (реконструкция, техническое перевооружение, приобретение) объекта недвижимого имущества</i>	Отображается, недоступное для редактирования.
IS_ACTUAL	булево			Отображается, недоступное для редактирования устанавливается значение <i>true</i> .
Связанная таблица				
ИД результата ЭБ	Нумератор ИД	SpecialControlPoints ResultsType ResultTypeCode	Нумератор ИД <i>Пример: 75</i>	Не отображается, недоступное для редактирования
Код результата	Текст, 2000 символов	SpecialControlPoints ResultsType ResultTypeCode	Текст, 2000 символов <i>Пример: Т1</i>	Не отображается, недоступное для редактирования
Принадлежность	Значение из списка: 1- Проектная часть =1; 3- Процессная часть =2;	SpecialControlPoints ResultsType StructuralElement	целочисленное <i>Пример: 1</i>	Не отображается, доступное для редактирования Загруженные структурные элементы имеют варианты: 1 - Ведомственный/Федеральный/Региональный проект; 2 - Комплекс процессных мероприятий

Во вторую очередь заполняется справочник *Типы контрольных точек* значениями из CheckPointTypesRF, которые загружаются из ЭБ по планировщику, при этом:

- если загружается тип контрольной точки, в котором значение тега CheckpointName имеется в поле **CAPTION** строки справочника *Типы контрольных точек*, то обновляются сведения в остальных полях строки справочника;
- если значение тега CheckpointName не встречается в поле **CAPTION** справочника

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Типы контрольных точек, формируется новая строка по правилу:

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование поля в АЦК	Формат АЦК	Наименование атрибута в ЭБ	Формат ЭБ	Правила заполнения
CAPTION	Текст, 2000 символов	LinkedControlPoints ControlPoint CheckpointName	Текст, 2000 символов <i>Пример: Земельный участок предоставлен заказчику</i>	Отображается, недоступное для редактирования.
IS_ACTUAL	булево			Отображается, недоступное для редактирования устанавливается значение <i>true</i> .
ИД контрольной точки из ЭБ	Нумератор ИД	LinkedControlPoints ControlPoint CheckpointId	Нумератор ИД <i>Пример: 162</i>	Не отображается, недоступное для редактирования.
KIND	Значение из списка: 1 - Специальная =0; 2 - Обеспечивающая =1; 3 - Дополнительная = 3; 4 - Рекомендуемая =2;	LinkedControlPoints ControlPoint AdditionalControlPointType	целочисленное <i>Пример: 0</i>	Отображается, недоступное для редактирования загруженные виды контрольных точек имеют варианты: 0 - Специальная 1 - Обеспечивающая 2 - Рекомендуемая 3 - Дополнительная
DESCRIPTION	Текст, 2000 символов			Отображается, недоступное для редактирования устанавливается значение <i>Загружено из ЭБ</i> .
TYPE_RESULT S_ID	Нумератор ИД	SpecialControlPoints Results Type ResultTypeCode	Текст, 2000 символов <i>Пример: Строительство (реконструкция, техническое перевооружение, приобретение) объекта недвижимого имущества</i>	Отображается, недоступное для редактирования. Определяется ссылка на справочник <i>Типов результатов</i> при полном совпадении в нем поля CAPTION с тегом ResultTypeName, в который вложен блок LinkedControlPoints/ControlPoint с обрабатываемой контрольной точкой с наименованием в теге CheckpointName.

В третью очередь заполняется справочник **Соответствие Типов субсидий, Типов результатов и Типов контрольных точек** значениями из CheckPointTypesRF, которые загружаются из ЭБ по планировщику, при этом:

- если загружается тип контрольной точки, в котором значение тега CheckpointName имеется в поле **TYPE_CONTROL_POINTS_CAPTION** строки справочника *Соответствие Типов субсидий, Типов результатов и Типов контрольных точек*, производится поиск по вышестоящему тегу ResultTypeName и поля TYPE_RESULTS_CAPTION:
- если совпадение не найдено, то формируется новая строка справочника;
- если совпадение найдено, то обновляются сведения в остальных полях строки;
- если значение тега CheckpointName не встречается в поле **TYPE_CONTROL_POINTS_CAPTION** справочника *Соответствие Типов субсидий, Типов результатов и Типов контрольных точек* то формируется новая строка по правилу:

Наименование поля в АЦК	Формат АЦК	Наименование атрибута в ЭБ	Формат ЭБ	Правила заполнения
TYPE_CONTROL_POINTS_ID	Нумератор ИД	LinkedControlPoints ControlPoint CheckpointName	Текст, 2000 символов <i>Пример: Земельный участок предоставлен заказчику</i>	Отображается, недоступное для редактирования. Определяется ссылка на строку справочника <i>Типы контрольных точек</i> по совпадению в нем поля CAPTION с значением тега CheckpointName. Поле TYPE_CONTROL_POINTS_CAPTION пересчитывается по значению в поле TYPE_CONTROL_POINTS_ID .

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

TYPE_RESULTS_ID	Нумератор ИД	SpecialControlPoints ResultsType ResultTypeCode	Текст, 2000 символов Пример: <i>Строительство (реконструкция, техническое первооружение, приобретение) объекта недвижимого имущества</i>	Отображается, недоступное для редактирования. Определяется ссылка на справочник <i>Типов результатов</i> при полном совпадении в нем поля CAPTION с тегом ResultTypeName, в который вложен блок LinkedControlPoints/C ontrolPoint с обрабатываемой контрольной точкой с наименованием в теге CheckpointName. Поле TYPE_RESULTS_CAPT ION пересчитывается по значению в поле TYPE_RESULTS_ID .
AFFILIATION	Значение из списка: 1- Проектная часть =1; 3- Процессная часть =2;	SpecialControlPoints ResultsType StructuralElement	Целочисленное Пример: 1	Отображается, недоступное для редактирования. Для загружаемого из тега CheckpointName во вложенном блоке LinkedControlPoints/C ontrolPoint по значению тега StructuralElement для ResultTypeName выбирается значение в поле AFFILIATION по условию: <ul style="list-style-type: none"> ▪ если "StructuralElement"= 1 (1 - Ведомственный/Фе деральный/Региона льный проект), то поле "AFFILIATION"=1; ▪ если "StructuralElement"= 2 (2 - Комплекс процессных мероприятий), то поле "AFFILIATION"=3;

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

IS_ACTUAL	булево			Отображается, недоступное для редактирования. Устанавливается значение <i>true</i> .
TYPE_SUBSIDY_ID	целочисленное			Отображается, недоступное для редактирования. Устанавливается значение <i>0</i> . Поле TYPE_SUBSIDY_CAPTION пересчитывается по значению в поле TYPE_SUBSIDY_ID .

Загрузка справочника *Субъекты РФ*

В системе создается скрытый объект, который заполняется на основании полученного по планировщику из ЭБ xml (справочник *SubjectsRF*).

Пользователь заполняет системный параметр **Код субъекта РФ** по пути **Системные параметры**→**Планирование**. По заполненному полю **Код субъекта РФ** из скрытого объекта по соответствию поля **Код субъекта РФ** предзаполняются поля **ID субъекта в ЭБ** и **Наименование субъекта** с возможностью редактирования. Если по введенному **Коду субъекта РФ** в скрытом объекте отсутствуют соответствующие записи, то поля с параметрами **ID субъекта в ЭБ** и **Наименование субъекта** пусты.

Наименование поля в АЦК	Формат АЦК	Наименование атрибута в ЭБ	Формат ЭБ	Правила заполнения
ID субъекта в ЭБ	long	SubjectId	long <i>Пример: 52622816</i>	Не отображается, Загружаемое, недоступное для редактирования
Код субъекта РФ	Число, 2 знака	SubjectCode	текстовое <i>Пример: 86</i>	Не отображается, Загружаемое, недоступное для редактирования
Наименование субъекта	Текст, 2000 знаков	SubjectName	текстовое <i>Пример: Ханты-Мансийский автономный округ - Югра</i>	Не отображается, Загружаемое, недоступное для редактирования

3.8.46 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:

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Таблица 36 – 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 запись, 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 запись, у которой значение в поле Раздел равно «1».</p>

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

№	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.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Атрибутный состав:

После ответа от сервиса формируется набор с данными:

Таблица 37 – 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** записи в списке дополнительных наименований значением 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 38 – 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	ОКЕИ	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

3.8.47 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
```

3.8.48 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.

3.8.49 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».

3.8.50 Messenger

XML для запуска:

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

Параметры: ProcessClasses, WorkDate, error.email, error.email.subject.

Описание: Задание планировщика *Messenger* предназначено для обработки очередей 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.

- **error. email** – указывается адрес или адреса электронной почты для отправки информации.
- **error. email. subject** – указывается текст сообщения. Если параметр не заполнен, заполняется тестом: «*Службное сообщение АЦК*».

В случае прерывания работы задания планировщика формируется электронное письмо со следующими реквизитами:

- с темой из параметра **error.email.subject**;

- в теле письма цитируется ошибка;
- в качестве адресов получателей подставлять значение параметра **error.email** (для отправки требуется разделять их точкой с запятой).

Пример использования:

```
error.email=admin1@mail.ru, admin2@mail.ru; error.email.subject=Ошибка планировщика  
Messenger в АЦК-Финансы 2022 для сайта АЦК-Планирование 2022
```

Сформированное сообщение отправляется с адреса, указанного в системных параметрах электронной почты (пункт меню **Сервис→Системные параметры**, группа настроек **Система**, закладка **Настройки электронной почты**)

Примечание. В случае прерывания обработки репликационного сообщения если работа планировщика не прерывается, но ошибка фиксируется в поле *Диагноз обрабатываемого сообщения*, формируется электронное сообщение.

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.

3.8.51 MsgCleaner

XML для запуска:

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

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

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.
- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

- **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_yyyy-mm-dd hh-mm-ss.xml*, где **yyyy-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.

3.8.52 MsgRepeater

XML для запуска:

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

Параметры: CHECK_FAILED, ProcessClasses, WorkDate, error.email, error.email.subject, pg_work_mem.

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. Предусмотренные параметры нельзя изменить.

- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

pg_work_mem=384

- **error. email** – указывается адрес или адреса электронной почты для отправки информации.
- **error. email. subject** – указывается текст сообщения. Если параметр не заполнен, заполняется тестом: «Служебное сообщение АЦК».

В случае прерывания работы задания планировщика формируется электронное письмо со следующими реквизитами:

- с темой из параметра **error.email.subject**;
- в теле письма цитируется ошибка;
- в качестве адресов получателей подставлять значение параметра **error.email** (для отправки требуется разделять их точкой с запятой).

Пример использования:

error.email=admin1@mail.ru, admin2@mail.ru; error.email.subject=Ошибка планировщика MsgRepeater в АЦК-Финансы 2022 для сайта АЦК-Планирование 2022

Сформированное сообщение отправляется с адреса, указанного в системных параметрах электронной почты (пункт меню **Сервис**→**Системные параметры**, группа настроек **Система**, закладка **Настройки электронной почты**)

Примечание. В случае прерывания обработки репликационного сообщения если работа планировщика не прерывается, но ошибка фиксируется в поле Диагноз обрабатываемого сообщения, формируется электронное сообщение.

Процедура обеспечивает исполнение сообщений на сервере приложений на статусе

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

3.8.53 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.

3.8.54 NotifyMsgSender

XML для запуска:

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

Параметры: нет;

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

3.8.55 NotificationSenderTask

XML для запуска:

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

Параметры: нет.

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

Рекомендуемый период запуска: по усмотрению.

3.8.56 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*.

3.8.57 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.

3.8.58 OkvInfoImport

XML для запуска:

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

Параметры: SITE_ID, file_path.

Описание: Загрузка данных в справочники ОКВ из папки.

Рекомендуемый период запуска: раз в час.

Процесс, предназначенный для обновления информации в справочниках *Объекты капитальных вложений*, *Информация об объектах капитальных вложений*, *Направление инвестирования* и *Группы направлений инвестирования*.

Принимает следующие параметры:

- **SITE_ID** – номер сайта БФТ.Капвложения;
- **file_path** – каталог, в который выгружает XML система БФТ.Капвложения/каталог импорта в АЦК-Планирование;
- **out_file_path** – название параметра в планировщике для лога загрузки.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

При приеме информационного сообщения, загружаемого через задание планировщика *OkvInfolmport* производится анализ атрибутивного состава входящей XML. Если во входящей XML присутствует атрибут с именем BUDGET_INVEST_TARGET (в составе DOCAZKOCIPAS), то загрузка выполняется по новому формату, иначе – по старому формату.

При обработке файла по новому формату предварительно проводится проверка: если в составе атрибутов DOCAZKOCIPAS присутствует тег AZK-ID и он не пустой, по значению кода AZK_ID производится поиск в таблице OKS (Объекты капитального строительства и вложений) существующую строку. Если строка найдена, обновляются все поля; если строка по AZK-ID не найдена, проводится обычная загрузка нового объекта.

Таблица 39 – Описание атрибутов из БФТ.Капвложения (QDP)

Но мер п/п	Наименование поля ЭД «Информация об ОКВ»	Наименование реквизита xml	Правило заполнения
1	Вышестоящий код	HIUIN	Значение из xml-файла.
2	Код ОКС	UIN	Указывается ссылка на запись справочника <i>Объекты капитальных вложений</i> . Поиск нужной записи в справочнике <i>Объекты капитальных вложений</i> осуществляется по соответствию значения из xml-файла, значению в поле УИН БФТ.Капвложения . Если значение не найдено, то оно должно быть добавлено в справочник.
3	УИН БФТ.Капвложения	-	Заполняется на основании поля УИН БФТ.Капвложения справочника <i>Объекты капитального строительства и вложений</i> по соответствующему Код ОКС .
	Расшифровка записи справочника:		
4	Направление инвестирования	OCIINVESTTYPECAPTION	Значение из xml-файла, при загрузке осуществляется подбор значений из справочника <i>Направления инвестирования</i> , если значение, указанное в реквизите OCIINVESTTYPECAPTION не найдено, то оно должно добавиться в справочник.
5	ГРБС, осуществляющий финансирование объекта	ORG_ID	Значение из xml-файла.
6	Информация о включении в АИП	INAIP	Значение из xml-файла.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Но ме р п/п	Наименование поля ЭД «Информация об ОКВ»	Наименование реквизита xml	Правило заполнения
7	Наименование объекта	CAPTION	Значение из xml-файла. Значение загружается в справочники <ul style="list-style-type: none"> Информация об ОКВ Объекты капитальных вложений (повторная загрузка может обновить значение в справочнике по уникальному полю Код)
8	На дату	DATECHANGE	Значение из xml-файла.
9	Внешний ID (ID паспорта ОКВ внешний)	FOREIGN_ID	Значение из xml-файла.

Таблица 40 – Описание атрибутов из БФТ.Капвложения (ICE)

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника Объекты капитальных вложений	справочник а Направлени я инвестиров ания	справочник а Информац ия об ОКВ	
EXTRWT	-	X	X	X	
WTFORMAT	-	X	X	X	
DOCAZKOCIPAS	Информация об одном объекте	X	X	X	
AGREEMENT_UIN	Уникальный реестровый номер соглашения	Уникальный реестровый номер соглашения	-	-	
AREA_CODE	-	-	-	-	
BUILDING_CODE	-	-	-	-	
CITY_CODE	-	-	-	-	
HOUSE_CODE	-	-	-	-	
LOCALITY_CODE	-	-	-	-	
STREET_CODE	-	-	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
AREA_NAME	Район	Местонахождение (адрес при наличии)	-	-	
CITY_NAME	Город	Формируется как комбинация совокупности полей в формате:	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
LOCALITY_NAME	Населенный пункт	<p><если заполнено значение атрибута AREA_NAME, то заполняется им>, <если заполнено значение атрибута CITY_NAME, то заполняется значением «г.» и значением атрибута CITY_NAME>, <если заполнено значение атрибута LOCALITY_NAME, то заполняется значением «ул.» и значением атрибута LOCALITY_NAME>, <если заполнено значение атрибута HOUSE_NUMBER, то заполняется значением «д.» и значением атрибута HOUSE_NUMBER>, <если заполнено значение атрибута BUILDING_NUMBER, то заполняется значением «стр./кор.» и значением атрибута BUILDING_NUMBER>, <если заполнено значение атрибута BUILDING_LETTER, то заполняется значением «лит.» и значением атрибута BUILDING_LETTER>, <если заполнено значение атрибута FLAT_NUMBER, то заполняется значением «кв./оф.» и значением атрибута</p>	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
STREET_NAME	Улица	FLAT_NUMBER>,<если и заполнено значение атрибута POSTAL_CODE, то заполняется значением "Индекс:" и значением атрибута POSTAL_CODE>. Если итоговое значение больше допустимого структурой БД, то обрезается итоговое значение до максимально допустимого.	-	-	
HOUSE_NUMBER	Номер дома		-	-	
BUILDING_NUMBER	Номер строения/корпуса		-	-	
BUILDING_LETTER	Литера		-	-	
FLAT_NUMBER	Номер квартиры/офиса		-	-	
POSTAL_CODE	Индекс		-	-	
AZK_ID	ИД объекта в системе АЦК-Планирование	-	-	-	Используется на начальной стадии алгоритма загрузки данных в справочник <i>Объекты капитальных вложений</i> , <i>Информация об объектах капитальных вложений</i> и <i>Направления инвестирования</i> и <i>Группы направлений инвестирования</i> .
BUDGET_INVEST_TARGET	Цель предоставления бюджетной инвестиции	Цель предоставления бюджетной инвестиции	-	-	Отличительный признак определения варианта форматов загрузки
CAPTION	Наименование объекта	Наименование объекта	-	Наименование объекта (в строке записи справочника)	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
COMMISSIONING_P LAN_DATE	Срок ввода в эксплуатацию	Срок ввода в эксплуатацию	-	-	
CUSTOMER_ID	Застройщик (Заказчик)	Заказчик	-	-	
DATECHANGE	На дату	-	-	На дату (в строке записи справочника)	
DEADLINE_MONTH_ COUNT	Нормативный срок выполнения работ (по ПОС), мес.	-			
FINANCINGFB	Тип софинансирования	Поле Тип софинансирования	-	-	
FOREIGN_ID	Внешний ID (ID паспорта ОКВ внешний)	-	-	Скрытое поле EXTERNAL_ID	
HIUIN	Код вышестоящего объекта	Вышестоящий код	-	Вышестоящий код	
INAIP	Информация о включении в АИП	-	-	Признак включения в программу (в строке записи справочника).	
INVESTMENTUK	Предоставление взноса в УК	Предоставление взноса в УК	-	-	
OIV_ID	Отраслевой орган исполнительной власти	-	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника Объекты капитальных вложений	справочник а Направлени я инвестирова ния	справочник а Информац ия об ОКВ	
OKSCODE_1	Код ОКС→ Вид бюджета (1 разряд)	Поле Вид бюджета combobox со значениями: пусто, S, M.	-	-	
OKSCODE_1_20	Код ОКС в МОКВ	Поле Код ОКС Отбрасывается первый символ (S или M): <ul style="list-style-type: none"> • 18 (с софинансировани ем из ФБ) • 19 (остальные) 	-	Поле Код ОКС	
ORG_ID	ID организации, владельца объекта	Наименование застройщика (определение по ID) и ГРБС, осуществляющий финансирование объекта (определение по ID)	-	ГРБС, осуществля ющий финансиро вание объекта (определен ие по ID)	
TERRITORY_ID	Территория	Территория			
TOTAL_INVESTMENT	Сумма полей Сметная стоимость табличной части <i>Информация о направления х инвестирова ния</i>	Общий объем инвестиций. Если значение атрибута = null, то подставляется его = 0.	-	-	
TOTAL_INVESTMENT_OKEI	Единица измерения стоимости строительств а	Единица измерения стоимости строительства (Значение ОКЕИ -383 - рубли)	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
UIN	Код объекта	УИН БФТ.Капвложения	-	УИН БФТ.Капвло жения	
OWNERSHIPTYPE_CODE	Код формы собственности	Форма собственности Если значение: - 14, то подставляется "муниципальная собственность", - 13, то подставляется "собственность субъекта Российской Федерации", - кроме 12 и 13, то подставляется "собственность организации".	-	-	
OWNERSHIPTYPE_NAME	Форма собственности	-	-	-	
CAPACITY_AMOUNT	Количественное значение мощности	Поле Мощность объекта капитального строительства строки таблицы <i>Направление инвестирования</i> . Вставляется во добавляемые строки таблицы <i>Направление инвестирования</i> .	-	-	
CAPACITY_CAPTION	Наименование мощности	-	-	-	
CAPACITY_UNIT	Единицы измерения мощности	-	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
CAPACITY_UNIT_OKEI	ОКЕИ единицы измерения мощности	-	-	-	
CAPACITY_UNIT_ID	ID единицы измерения мощности в АЦК-Планировании	Поле Единица измерения мощности строки таблицы <i>Направление инвестирования</i> . Добавлять по наличию в справочнике <i>Единицы измерения</i> . Отображать значение поля Наименование . Вставляется во добавляемые строки таблицы <i>Направление инвестирования</i> .	-	-	
DEVELOPERS	Перечень подрядчиков	X	X	X	
DEVELOPER	Информация об одном подрядчике	X	X	X	
CAPTION	Наименование	-	-	-	
INN	ИНН	-	-	-	
ADDRESS	Адрес	-	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
OCICOOINATES	Информация о перечне координат	Координаты Формируется как комбинация	X	X	
OCICOOINATE	Информация о координате одного объекта	совокупности полей в формате: Широта: <Значение атрибута LATITUDE> - Долгота: <Значение атрибута	X	X	
LATITUDE	Широта	LONGITUDE> - Позиция:<Значение атрибута POSITION>	-	-	
LONGITUDE	Долгота	Если в составе несколько строк, то указывается из через точку с запятой «;» Если итоговое значение больше допустимого структурой базы данных, то итоговое значение обрезается до максимально допустимого.	-	-	
POSITION	Позиция (порядковый номер)				
INVESTTYPES	Перечень этапов и направлений инвестирования	X	X	X	
INVESTTYPE	Информация об одном этапе/направлении инвестирования	X	X	X	
ECD_AMOUNT	Сметная стоимость в ценах года утверждения ПСД, руб.	-	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника <i>Объекты капитальных вложений</i>	справочник а <i>Направлени я инвестиров ания</i>	справочник а <i>Информац ия об ОКВ</i>	
ECD_UNIT_OKEI	Единица измерения стоимости строительства	-	-	-	
START_PLAN_DATE	Планируемые сроки начала работ	-	-	-	
END_PLAN_DATE	Планируемые сроки окончания работ	-	-	-	
INVESTSTAGE	Этап реализации направления инвестирования	-	-	-	
OCIINVESTTYPECAPTION	Направление инвестирования	Поле Направление инвестирования строки таблицы <i>Направление инвестирования</i> .	Направлен ие инвестирования. Если при загрузке в справочнике присутствует запись с аналогичным наименованием, то пропускается при загрузке в справочник.	Направлен ие инвестирования	
CODE PL_FAIP_OBJ	Объект ФАИП	Объект ФАИП заполняется ссылкой на справочник <i>Объекты ФАИП</i>	-	-	
CAPTION PL_FAIP_OBJ		-	-	-	
CUSTOMER_ID PL_FAIP_OBJ		-	-	-	

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

Наименование атрибута/элемента входящего объекта	Описание	Поле системы АЦК-Планирование и правила заполнения			Дополнительно
		справочника Объекты капитальных вложений	справочник а Направлени я инвестиров ания	справочник а Информац ия об ОКВ	
DATETO	Дата начала действия	Дата начала действия	-	-	
DATEFINISH	Дата окончания действия	Дата окончания действия	-	-	

3.8.59 OosExport

XML для запуска:

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

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

3.8.60 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
```

3.8.61 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
```

3.8.62 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.

3.8.63 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
```

3.8.64 PlanXMLExport

XML для запуска:

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

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

3.8.65 PIBorExport

XML для запуска:

```
<PLBOREXPORT 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
```

3.8.66 PIStateEventDocExport

XML для запуска:

```
<PLSTATEEVENTDOCEXPORT 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
```

3.8.67 PISStateProgDocExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PIStateProgDocExport* 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 – *PIStateProgDocExport.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
```

3.8.68 PISubProgDocExport

XML для запуска:

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

Параметры: budget_id, directory.

Описание: Задание планировщика *PISubProgDocExport* предназначено для экспорта ЭД «Направление (подпрограмма)».

Рекомендуемый период запуска: по усмотрению.

Процесс, предназначенный для выгрузки строк ЭД «Направление (подпрограмма)» для портала «Открытый бюджет» в формате xml-файлов – *PISubProgDocExport.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
```

3.8.69 ProcessGrantsReducedFin

XML для запуска:

```
<PLANASSDOC_PROCESSOR action="reduced_fin_grants"/>
```

Описание: Перевод соглашения на статус «*уменьшено фин. обеспечение*»

При активации задания планировщика:

1. осуществляется проверка наличия записей со ссылками на расходные ЭД в отдельной таблице, которые подпадают по условия отбора бюджетных строк;
2. предусмотрен запрет на активацию второго экземпляра планировщика, если один запущен, то второй запуститься не должен;
3. осуществляется группировка бюджетных строк по набору полей: **Бланк расходов+Бюджетополучатель+РО+КВСР+КФСР+КЦСР+КВР+КОСГУ+Доп.ФК+Доп.ЭК+Доп.КР+КВФО+Код цели** из бюджетных строк ЭД «Бюджетная заявка на изменение ассигнований», ссылки на которые присутствуют в отдельной таблице;

4. осуществляется проверка наличия по ЭД «Соглашение по МБТ» на статусах *«утвержден»*, в котором присутствуют строки с идентичным набором полей: **БР+БП+КБК+РО+КЦ** на закладке **Сумма обязательств в разрезе бюджетных строк**. Если таких соглашений нет, то такая бюджетная строка пропускается для дальнейших проверок;

5. для каждой бюджетной строки определяется общая сумма ассигнований по 1-му, 2-му и 3-му годам планирования по данным ЭД «Бюджетная заявка»/ЭД «Справка об ассигнованиях» на статусе *«утвержденный бюджет»* (за исключением документов по исходной версии планирования), ЭД «Бюджетная заявка на изменение ассигнований»/ЭД «Справка об изменении бюджетных назначений» на статусе *«утвержденный бюджет»*, на статусах *«обработка завершена»* и *«направлен на утверждение»* (с активным признаком **Без внесения изменений в закон о бюджете** по версиям, отличным от исходной);

6. для каждой бюджетной строки определяется объем принятых обязательств по ЭД «Соглашение по МБТ» на статусах *«утвержден»*, *«выполнена актуализация»*, *«уменьшено фин. обеспечение»*, *«недостижение целевых показателей»*, *«внесение изменений»* (при наличии родительского соглашения с идентичным финансовым годом предоставляющего бюджета), в которых отбираются строки с идентичным набором полей: **БР+БП+КБК+РО+КЦ** закладки **Сумма обязательств в разрезе бюджетных строк** со значениями полей **Сумма (1-й год планирования)**, **Сумма (2-й год планирования)**, **Сумма (3-й год планирования)**;

7. Осуществляется проверка для каждого из годов на непревышение суммы принятых обязательств по соглашениям над суммой ассигнований;

8. Если проверка для бюджетной строки не выполняется, то осуществляется проверка наличия ЭД «Соглашение по МБТ» с данной бюджетной строкой на статусе *«утвержден»*. Если такое соглашение/соглашения найдены, то:

8.1. Если подключен функционал в части работы с расходными документами, бюджетные строки по которым включены в электронные соглашения (planassdoc_agreement_amendment_enabled.xml), то в систему «АЦК-Финансы» отправляется сообщение о смене статуса связанного ЭД «Сведения о соглашении» с указанием doc_id соглашения:

```
<DOCUMENT
```

```
action = "новое название" class="273" DISPSTATUS_ID=55 DOCUMENT_ID=" "
```

```
/DOCUMENT>
```

При получении положительного ответа от системы «АЦК-Финансы», осуществляется

проверка, если статус ЭД «Соглашение по МБТ» *«утвержден»*, то статус ЭД «Соглашение по МБТ» меняется на *«уменьшено фин. обеспечение»*.

При получении отрицательного ответа от системы «АЦК-Финансы» по одному соглашению статус соглашения не меняется, но продолжается работа с другими соглашениями;

8.2. Если не подключен функционал в части работы с расходными документами, бюджетные строки по которым включены в электронные соглашения (planassdoc_agreement_amendment_enabled.xml), осуществляется перевод соглашения на статус *«уменьшено фин. обеспечение»* без отправки сообщения в систему «АЦК-Финансы».

3.8.70 ProgDocExport

XML для запуска:

```
<PROGDOCEXPOR 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
```

3.8.71 PurposefulGrantCleanupTask

XML для запуска:

```
<GENERICSCHTASK action="run"
```

```
class_name="com.bssys.azk server.purposefulgrant.PurposefulGrantCleanupTask"/>
```

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

3.8.72 RegionalProjectCardEhdExportProcessor

XML для запуска:

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

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 системы «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

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

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.

Атрибутный состав:

Таблица 41 – Описание атрибутивного состава отправляемого файла

Название атрибута/элемента	Описание элемента	Описание сбора
REGIONAL_PROJECTS_<ИМЯ>		
REGIONAL_PROJECT	Информация по одному региональному проекту.	В данном блоке отражается информация по одному региональному проекту ЭД «Карточка регионального проекта» (отбираются ЭД «Карточка регионального проекта» на статусе « <i>утвержден</i> » и « <i>изменение</i> »).
MASTER_ID	Сайт-источник (целочисленный формат, 15 символов).	–
CLIENT_ID	Сайт-приемник (целочисленный формат, 15 символов).	–
RPL_OBJ_NAME	Наименование объекта репликации (текстовый формат, 100 символов).	Начинается с 0, каждый последующий запуск увеличивает значение на единицу.
PREV_VERSION	Предыдущая версия формат (целочисленный формат, 15 символов)	–
VERSION	Версия формат (целочисленный формат, 15 символов).	Начинается с 1, каждый последующий запуск увеличивает значение на единицу относительно последней выгрузки.
REGIONAL_PROJECT		
ID	Идентификатор выгрузки.	–
ON_DATE	Дата выгрузки.	Заполняется значением системной даты сервера приложений при запуске задания.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

METAID	Значение скрытого поля, загруженного с API ЕПБС.	Заполняется значением системного поля METAID , которое хранит в себе значение, загруженное в ЭД с API ЕПБС (Единого портала бюджетной системы Российской Федерации).
NAME	Наименование регионального проекта.	Значение поля Наименование регионального проекта (NAME) ЭД «Карточка регионального проекта».
PROJ_CODE	Код регионального проекта.	Значение поля Код регионального проекта (PROJ_CODE) ЭД «Карточка регионального проекта».
FED_PROJECT_CAPTION	Наименование федерального проекта.	Значение поля Наименование федерального проекта (DD_FED_PROJECT_CAPTION) ЭД «Карточка регионального проекта».
FED_PROJECT_CODE	Код федерального проекта.	Значение поля Код федерального проекта (DD_FED_PROJECT_CODE) ЭД «Карточка регионального проекта».
REG_PROJ_CARD_RESULT_LINES	Результаты и мероприятия региональных проектов.	Информация о перечне строк результатов и мероприятий. Из полученных ЭД отбираются строки с закладок Результаты, контрольные точки и мероприятия с типами «результат» и «мероприятие», у которых в таблицах на закладках КБК или Направление расходов присутствуют данные.
REG_PROJ_CARD_RESULT_LINES		
REG_PROJ_CARD_RESULT_LINE	Строка результата (мероприятия).	Информация по одной строке результата или мероприятия согласно описанию. Из полученных ЭД отбираются строки с закладок Результаты, контрольные точки и мероприятия с типами «результат» и «мероприятие», у которых в таблицах на закладках КБК или Направление расходов присутствуют данные.
REG_PROJ_CARD_RESULT_LINE		
ID	Идентификатор строки ЭД «Карточка регионального проекта».	Значение системного поля ID строки результата/мероприятия.
METAID	Значение скрытого поля, загруженного с API ЕПБС.	Заполняется значением системного поля METAID , которое хранит в себе значение, загруженное в ЭД с API ЕПБС (Единого портала бюджетной системы Российской Федерации).

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

CAPTION	Наименование строки результата/мероприятия.	Значение поля Наименование события (CAPTION) строки результата/мероприятия ЭД «Карточка регионального проекта».
CODE	Код строки результата/мероприятия.	Значение поля Код события (CODE) строки результата/мероприятия ЭД «Карточка регионального проекта».
EVENT_TYPE	Тип события.	Значение поля Наименование типа события (DD_EVENT_TYPE) строки результата/мероприятия ЭД «Карточка регионального проекта». Возможны следующие значения: <ul style="list-style-type: none"> • <i>Результат</i> • <i>Мероприятие</i>
START_DATE	Дата начала.	Значение поля Дата начала (START_DATE) строки результата/мероприятия ЭД «Карточка регионального проекта».
END_DATE	Дата окончания.	Значение поля Дата окончания (END_DATE) строки результата/мероприятия ЭД «Карточка регионального проекта».
ORG_CAPTION	Наименование организации.	Значение поля Организация (DD_ORG_CAPTION) строки результата/мероприятия ЭД «Карточка регионального проекта».
IS_EXECUTION	Признак Выполнено .	Значение поля Выполнено (IS_EXECUTION) строки результата/мероприятия ЭД «Карточка регионального проекта». <ul style="list-style-type: none"> • 0 - признак не активен • 1 - признак активен
SUM_CARD_RESULTS	Итоговые суммы по результату/мероприятию.	Раздел содержит итоговые, агрегированные данные FINANCIALSOURCES по всем годам и суммам без учета источников средств.
SUM_CARD_RESULTS		
SUM_CARD_RESULT	Информация по строке с суммами за год.	Информация по строке одного финансового года.
SUM_CARD_RESULT		

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

FINYEAR	Финансовый год суммы.	<p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>АЦК-Финансы</i>, то значение определяется результатом значение поля Год строки агрегированной строки блока FINANCIALSOURCES.</p> <p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>Внешние системы</i>, то значение вычисляется из поля Бюджет строки справочника <i>Финансовые данные внешних систем</i>. Проставляется значение поля Финансовый год для указанного бюджета.</p>
CBR_SUM	Сумма сводной бюджетной росписи.	Результат суммы сводной бюджетной росписи по всем источникам в пределах одного финансового года блока FINANCIALSOURCES.
LBO_SUM	Сумма лимитов бюджетных обязательств.	<p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>АЦК-Финансы</i>, то значение определяется результатом суммы лимитов бюджетных обязательств по всем источникам в пределах одного финансового года блока FINANCIALSOURCES.</p> <p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>Внешние системы</i>, то заполняются значения из справочника <i>Финансовые данные внешних систем</i>, из колонки <i>Лимиты</i>, суммой строк, у которых значение На дату меньше даты формирования выгрузки или равно ей.</p> <p>Если активна настройка Настройка соответствия источника финансовых показателей для бюджетов, то данные собираются по указанным в настройке бюджетам.</p>

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

BUDGETCOMMITMENTS_SUM	Сума поставленных на учет БО.	<p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>АЦК-Финансы</i>, то значение определяется результатом суммы BUDGETCOMMITMENTS по всем источникам в пределах одного финансового года блока FINANCIALSOURCES.</p> <p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>Внешние системы</i>, то заполняются значения из справочника <i>Финансовые данные внешних систем</i>, из колонки <i>Финансирование</i>, суммой строк, у которых значение На дату меньше даты формирования выгрузки или равно ей.</p> <p>Если активна настройка Настройка соответствия источника финансовых показателей для бюджетов, то данные собираются по указанным в настройке бюджетам.</p>
CASHEXECUTION_SUM	Сумма кассового исполнения.	<p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>АЦК-Финансы</i>, то значение определяется результатом суммы CASHEXECUTION по всем источникам в пределах одного финансового года блока FINANCIALSOURCES.</p> <p>Если в настройке Источник финансовых показателей для фактических показателей выбрано значение <i>Внешние системы</i>, то заполняются значения из справочника <i>Финансовые данные внешних систем</i>, из колонки <i>Кассовое исполнение</i>, суммой строк, у которых значение На дату меньше даты формирования выгрузки или равно ей.</p> <p>Если активна настройка Настройка соответствия источника финансовых показателей для бюджетов, то данные собираются по указанным в настройке бюджетам.</p>
FINANCIALSOURCES	Финансовое обеспечение результата/мероприятия по источникам.	Раздел содержит детализированные суммы по всем годам и источникам финансирования.

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

FINANCIALSOURCES		
FINANCIALSOURCE	Информация по строке по результату/мероприятия по источникам.	Раздел собирает информацию по одной строке источника и года.
FINANCIALSOURCE		
REGPROJFINVALUE	Код источника средств.	Значение поля Код строки справочника <i>Финансовые источники средств</i> .
FINYEAR	Финансовый год.	Значение поля Год (FINYEAR) по строке источника и результату/мероприятия блока REG_PROJ_CARD_FPROV_AMTS ЭД «Карточка регионального проекта».
CBR	Сумма сводной бюджетной росписи.	Значение суммы по полю Бюджет (FPROV_BUDGET_AMOUNT) по одному финансовому году блока (выполняется запрос актуальной суммы как по действию Актуализировать суммы списка результатов).
LBO	Сумма лимитов бюджетных обязательств.	Значение суммы по полю ЛБО (LIMIT_BO_AMOUNT) по одному финансовому году.
BUDGETCOMMITMENTS	Сума поставленных на учет БО.	Значение суммы по полю Ученные БО (ACCOUNTED_BO_AMOUNT) по одному финансовому году.
CASHEXECUTION	Сумма кассового исполнения.	Значение суммы по полю Кассовое исполнение (CASH_EXEC_AMOUNT) по одному финансовому году.
PURCHASES	Перечень закупок.	Информация о перечне всех строк закупок. Каждое выполнение запрашивает суммы из ЕХД в объеме атрибутивного состава текущей выгрузки. Запрос аналогичен запросам по таким же полям блока Сведения о закупках в региональных проектах .
PURCHASES		
PURCHASE	Информация по строке закупок.	Информация по одной строке закупки.
PURCHASE		
FINYEAR	Финансовый год	Поле Финансовый год блока Сведения о закупках в региональных проектах .
DEALDATE	Плановая дата заключения контракта (договора).	Поле Плановая дата заключения контракта (договора) справочника блока Сведения о закупках в региональных проектах .

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

IKZ	Идентификационный код закупки.	Поле ИКЗ блока Сведения о закупках в региональных проектах .
IKZ_26	Идентификационный код закупки (1-26 разряд).	Поле ИКЗ системное блока Сведения о закупках в региональных проектах .
PLANPOSITION	Номер позиции плана-графика.	Поле Номер позиции плана-графика блока Сведения о закупках в региональных проектах .
CUSTOMERNAME	Название организации-заказчика.	Поле Заказчик Наименование блока Сведения о закупках в региональных проектах .
CUSTOMERTAXCODE	ИНН заказчика.	Поле Заказчик ИНН блока Сведения о закупках в региональных проектах .
NMCK	Лимит НМЦК (Начальной минимальной цены контракта).	Поле Лимит (НМЦК) блока Сведения о закупках в региональных проектах .
CONTRACTS		
CONTRACT	Информация по строке контракта.	Информация по одной строке контракта
CONTRACT		
IKZ	Идентификационный код закупки.	Поле ИКЗ блока Сведения о закупках в региональных проектах .
IMIT_AMOUT	Лимит НМЦК.	Поле Лимит НМЦК блока Сведения о закупках в региональных проектах .
CONTRACT_EXEC_FACT_DATE_FROM	Фактический срок исполнения контракта (договора) с.	Поле Фактический срок исполнения контракта (договора) с блока Сведения о закупках в региональных проектах .
CONTRACT_EXEC_FACT_DATE_TO	Фактический срок исполнения контракта (договора) по.	Поле Фактический срок исполнения контракта (договора) по блока Сведения о закупках в региональных проектах .
CONTRACT_EXEPIRE_DATE_FROM	Срок действия с контракта (договора).	Поле Срок действия с контракта (договора) блока Сведения о закупках в региональных проектах .
CONTRACT_EXEPIRE_DATE_TO	Срок действия по контракта (договора).	Поле Срок действия по контракта (договора) блока Сведения о закупках в региональных проектах .
SAVINGS_PERCENTAGE	Процент экономии.	Поле Процент экономии блока Сведения о закупках в региональных проектах .
DOC_NUMBER	Номер контракта.	Поле Номер контракта блока Сведения о закупках в региональных проектах .

Основные понятия

Планировщик. Задания планировщика. Расписание запуска заданий

CON_NUMBER	Реестровый номер контракта.	Поле Реестровый номер контракта блока Сведения о закупках в региональных проектах .
SUBJECT	Наименование объекта закупки (предмет контракта).	Поле Предмет контракта блока Сведения о закупках в региональных проектах .
DISPSTATUS_ID	Статус (стадия).	Поле Статус (стадия) блока Сведения о закупках в региональных проектах .
CONTRACT_SIGN_FACT_DATE	Фактическая дата заключения контракта.	Поле Фактическая дата заключения контракта (договора) блока Сведения о закупках в региональных проектах .
CONTRACT_AMOUNT	Сумма контракта (договора).	Поле Сумма контракта (договора) блока Сведения о закупках в региональных проектах . Сумма контракта (договора) справочника <i>Сведения о закупках в региональных проектах</i> .
PAYMENT_AMOUNT	Оплата контракта (договора).	Поле Оплата контракта (договора) блока Сведения о закупках в региональных проектах .
DELIVERY_AMOUNT	Сумма поставки.	Поле Сумма поставки справочника блока Сведения о закупках в региональных проектах .
CLAIM_WORKS	Наличие претензионных работ.	Поле Наличие претензионных работ блока Сведения о закупках в региональных проектах .
EIS_LINK	Гиперссылка на ЕИС (Единую информационную систему в сфере закупок).	Поле Гиперссылка на ЕИС блока Сведения о закупках в региональных проектах .

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)**.

3.8.73 RegProjActualityStatusUpdater

XML для запуска:

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

Описание: Обновление статуса актуальности данных ЭД КП

Мониторинг актуальности, осуществляемый данным заданием, необходим для проверки и оповещения пользователя о наличии более актуального планового ЭД «Карточка проекта» для текущего исполнения. Процесс производит запрос по всем ЭД «Карточка проекта», связанных с ЭД «Исполнение события» и ЭД «Исполнение проекта» на статусах, отличных от «*архив*» и «*удален*», и проставляет документам исполнения значение признака **Актуальность** по следующему алгоритму:

1. Отбираются все ЭД «Исполнения проекта» на статусах, отличных от «*утвержден*», «*удален*», «*архив*»;
2. Фиксируется значение поля **Дата утверждения** и ID связанного с отображенными документами ЭД «Карточка проекта»;
3. Фиксируется значение поля **На дату** текущего документа исполнения проекта;
4. Производится поиск ЭД «Карточка проекта» на статусе «*утвержден*» и значением поля **Дата утверждения** в диапазоне от даты из пункта 2 до даты из пункта 3, и ID больше, чем в пункте 2;
5. Если такие документы найдены, то во всех соответствующих ЭД «Исполнения проекта» и связанных с ним ЭД «Исполнения события» проставляется активный признак **Актуальность**, в противном случае признак проставляется неактивным.

3.8.74 RelimportFailedImportedDocs

XML для запуска:

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

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**.

3.8.75 Replication

XML для запуска:

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

Параметры: SITE, error.email, error.email.subject, pg_work_mem.

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
```

- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

- **error. email** – указывается адрес или адреса электронной почты для отправки информации.
- **error. email. subject** – указывается текст сообщения. Если параметр не заполнен, заполняется тестом: «Служебное сообщение АЦК».

В случае прерывания работы задания планировщика формируется электронное письмо со следующими реквизитами:

- с темой из параметра **error.email.subject**;
- в теле письма цитируется ошибка;
- в качестве адресов получателей подставлять значение параметра **error.email** (для отправки требуется разделять их точкой с запятой).

Пример использования:

```
error.email=admin1@mail.ru, admin2@mail.ru; error.email.subject=Ошибка планировщика  
Replication в АЦК-Финансы 2022 для сайта АЦК-Планирование 2022
```

Примечание. Независимый от работы репликатора механизм предотвращения параллельного запуска обновления журнала поколений. При запуске процедуры обновления журнала поколений, на основе считанных объектов репликации для каждого объекта вычисляется какие данные необходимо послать на сайт-получатель, в журнале поколений находятся записи с номером поколения больше или равным «Отправленная версия» и/или с номером поколения меньше или равным текущему номеру поколения. Полученные данные проходят постобработку с помощью процессора репликации, если он указан. Сформированные данные помещаются в сообщения.

Сформированное сообщение отправляется с адреса, указанного в системных параметрах электронной почты (пункт меню **Сервис**→**Системные параметры**, группа настроек **Система**, закладка **Настройки электронной почты**)

3.8.76 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.

3.8.77 ResultAmountCheck

XML для запуска:

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

Описание: Задание планировщика **ResultAmountCheck** выполняет запрос обновления сумм по плановым документам ЭД «Карточка проекта» и ЭД «Структурный элемент», в зависимости от параметров настроек.

Задание имеет следующие функциональные возможности:

1. применяется к ЭД «Карточка проекта» на всех статусах, кроме «архив», «удален», «изменение»;
2. применяется к ЭД «Структурный элемент» на всех статусах, кроме *архив*, «удален», «закрито», «отказан»;
3. обновляется значение параметра **Требуется актуализация сумм** в ЭД «Структурный элемент» и ЭД «Карточка проекта».

Параметры

Для формирования запроса необходимо ввести параметры запроса.

- Параметр: **BUDGET_ID** = "<ID бюджета>" - определяет, для какого бюджета должен обрабатывать планировщик.
- Параметр: **regionalProjectCard** - логическое, false или true. Если true, то проводится проверка актуальности сумм по ЭД «Карточка проекта», в противном случае не проверяется. Значение по умолчанию =true. Пример заполнения: "**regionalProjectCard=true**".
- Параметр: **plStateEventDoc** логическое, false или true. Если true, то проводится проверка актуальности сумм по ЭД «Структурный элемент», в противном случае не проверяется. Значение по умолчанию = true. Пример заполнения: "**plStateEventDoc=true**".

Дополнительно:

- Если оба параметра **plStateEventDoc** и **regionalProjectCard** = false, или оба параметра отсутствуют, то задание планировщика завершается сообщением: *«Задание не может быть выполнено. Не определены значения параметров **plStateEventDoc** и **regionalProjectCard**»*.
- **BUDGET_ID** = пусто, или параметр не установлен, то задание планировщика не запускается. Выводится сообщение: *«Задание не может быть выполнено. Не определено значение параметра **BUDGET_ID**»*.

Алгоритм отработки задания

ЭД «Карточка проекта»

1. Проверяется параметр **regionalProjectCard**, если = true, то осуществляется переход к п.2., в противном случае ничего не делается.

2. Отбираются карточки проекта со статусами, кроме «архив», «удален», «изменение».

3. Для каждого ЭД «Карточка проекта», для событий с типом = *Результат/Мероприятие* производится запрос строк КБК (только по бюджету, указанному в **BUDGET_ID**) по полям с суммами из ЭД «Бюджетная заявка» или «Справка об ассигнованиях» + «Заявка на изменения». ЭД «Карточка проекта», закладка **КБК**, для инструмента **Заполнить автоматически по выбранным КЦСР**.

4. Сформированные данные помещаются во временную таблицу и сравниваются с текущими значениями сумм в поле ЭД «Карточка проекта», закладка **Результаты контрольные точки мероприятия** → закладка **Финансовое обеспечение** → закладка **КБК**.

5. Построчно сравниваются строки КБК (только по бюджету, указанному в параметре **BUDGET_ID**). Строки считаются отличными, если:

- количество строк КБК различается как в меньшую, так и в большую сторону;
- если при одинаковом значении КБК суммы в сопоставляемых строках различаются.

Если набор строк КБК, полученных на закладке **КБК** текущего результата отличается от полученных в запросе п.4, то полю **Требуется актуализация сумм** присваивается значение = true, в противном случае - false.

6. При переходе на форму редактирования ЭД «Карточка проекта», где поле **Требуется актуализация сумм** = true, система должна возвращать информационное сообщение вида:

«В <Тип события> <Номер> <Наименование события> строки <номера строк, через запятую> не соответствуют данным расходных документов».

Примечание. Такое сообщение выводится, если суммы в существующих строках различаются.

«В <Тип события> <Номер> <Наименование события> требуется актуализация строк КБК».

Примечание. Такое сообщение выводится, только если количество строк различается».

Для каждого события сообщение выводится с новой строки, в конце выводится кнопка **ОК**, закрывающая информационное сообщение.

ЭД «Структурный элемент»

1. Проверяется параметр **plStateEventDoc**, если = true, то осуществляется переход к п.2. в противном случае ничего не делается.

2. Отбираются карточки проекта со статусами, кроме «*архив*», «*удален*», «*закрыто*», «*отказан*».

3. Для каждого ЭД «Структурный элемент», для записей на закладке **Мероприятия (результаты)** производится запрос строк КБК (только по бюджету, указанному в **BUDGET_ID**) по полям с суммами из ЭД «Бюджетная заявка»/ЭД «Бюджетная заявка на изменение» или ЭД «Справка об ассигнованиях»/ЭД «Справка об изменении бюджетных назначений». ЭД «Структурный элемент», закладка **Финансовое обеспечение**, закладка **Сумма ассигнований (плановый период)** для строки Мероприятия.

4. Сформированные данные помещаются во временную таблицу и сравниваются с текущими значениями сумм в поле ЭД «Структурный элемент», закладка **Мероприятия (результаты)**→закладка **Финансовое обеспечение**→закладка **Сумма ассигнований (плановый период)**.

5. Построчно сравниваются сформированные строки (только по бюджету, указанному в параметре **BUDGET_ID**). Строки считаются отличными, если:

- количество строк различается как в меньшую, так и в большую сторону;
- если у строк совпадают по параметрам уникальности, суммы в сопоставляемых строках различаются.

Если набор строк, полученных на закладке **Сумма ассигнований (плановый период)** текущего мероприятия, отличается от полученных в запросе п.4, то полю **Требуется актуализация сумм** присваивается значение = true, в противном случае - false.

6. При переходе на форму редактирования ЭД «Карточка проекта», где поле **Требуется актуализация сумм** = true, система должна возвращать информационное сообщение вида:

«В Мероприятии <Наименование мероприятия> строки <номера строк, через запятую> не соответствуют данным расходных документов».

Примечание. Такое сообщение выводится, если суммы в существующих строках различаются.

«В Мероприятии <Номер> <Наименование мероприятия> требуется актуализация плановых сумм».

Примечание. Такое сообщение выводится, только если количество строк различается).

Для каждого события сообщение выводится с новой строки, в конце выводится кнопка ОК, закрывающая информационное сообщение.

3.8.78 RplServiceLogCleaner

XML для запуска:

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

Параметры: daysAgo.

Описание: Очистка журнала работы сервиса **RplLogService**.

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

daysAgo – предназначен для указания количества последних дней хранимых в журнале работы сервиса **RplLogService** данных. Может принимать значения типа «*Целое положительное число*». Значение по умолчанию *61*. Параметр доступен для указания в поле **Параметры** в окне редактирования записи АРМ «Расписание планировщика». Предназначен для указания в поле **Параметры** окна редактирования строки АРМ «Расписание планировщика».

Пример использования:

```
daysAgo=72
```

При запуске задания **RplServiceLogCleaner** удаляются строки из таблицы **RPLSERVICEWORKLOG**, у которых значение в поле **START_TIME** меньше разницы текущая дата и **daysAgo**.

3.8.79 RplGenerationStatusTask

XML для запуска:

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

Параметры: pg_work_mem.

Описание: Статистика значений в RplLog.

- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во

время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_tem=384
```

3.8.80 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*.

3.8.81 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.00004-56 32 01-3 Система «АЦК-Планирование»](#). АРМ ФО. Блок администрирования. Подсистема администрирования. Настройка и сервис системы. Руководство администратора».*

3.8.82 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.

3.8.83 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.

3.8.84 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.

3.8.85 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.

3.8.86 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.

3.8.87 SyseventCleaner

XML для запуска:

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

Параметры: MSGTYPE, pg_work_mem.

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.

pg_work_mem – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

```
pg_work_mem=384
```

3.8.88 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.

3.8.89 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"/>
```

3.8.90 Transport

XML для запуска:

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

Параметры: site_id, period_from, period_end, error.email, error.email.subject, pg_work_mem.

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`.

- **pg_work_mem** – задает значение **work_mem** при работе с СУБД PostgreSQL во время выполнения операций задания, по умолчанию указано значение «384».

Пример использования:

`pg_work_mem=384`

- **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

- **error. email** – указывается адрес или адреса электронной почты для отправки информации.
- **error. email. subject** – указывается текст сообщения. Если параметр не заполнен, заполняется тестом: «*Службное сообщение АЦК*».

В случае прерывания работы задания планировщика формируется электронное письмо со следующими реквизитами:

- с темой из параметра **error.email.subject**;
 - в теле письма цитируется ошибка;
 - в качестве адресов получателей подставлять значение параметра **error.email** (для отправки требуется разделять их точкой с запятой).
- **serverMode** – параметр определяет протокол соединения между серверами. Работа параметра осуществляется для типа соединения – **Прямое соединение серверов (MSGSITE.SMSGLINKTYPE_ID = 1)**. Если параметр **serverMode** в задании:
 - не указан, подключение к другому СП идёт по *iiop*;
 - указан со значением *iiop* (**serverMode = iiop**), подключение к другому серверу приложений идёт по протоколу *iiop*;
 - указан со значением *http* или *https* (**serverMode = http**), подключение к другому СП идёт по протоколу *http* или *https*.

Пример использования:

error.email=admin1@mail.ru, admin2@mail.ru; error.email.subject=Ошибка планировщика Transport в АЦК-Финансы 2022 для сайта АЦК-Планирование 2022

Сформированное сообщение отправляется с адреса, указанного в системных параметрах электронной почты (пункт меню **Сервис**→**Системные параметры**, группа настроек **Система**, закладка **Настройки электронной почты**)

Примечание. В случае прерывания обработки репликационного сообщения если работа планировщика не прерывается, но ошибка фиксируется в поле Диагноз обрабатываемого сообщения, формируется электронное сообщение.

3.8.91 UnDocLauncher

XML для запуска:

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

Параметры: profile_name, budget_id.

Описание: Выгрузка отчетов «Информация о мониторинге достижения результатов предоставления субсидии».

Задание автоматически создает ЭД «Универсальный документ» для массовой выгрузки и сохранения в системе отчетов «Информация о мониторинге достижения результатов предоставления субсидии».

Выгружаются файлы отчетных форм по профилю диалогового окна отчета, указанному в параметрах задания; создается ЭД «Универсальный документ» на статусе «подготовлен» с прикрепленным выгруженным файлом отчетной формы значениями в полях:

- **Тип документа:** *Отчет*;
- **Получатель (Организация):** значения финоргана для бюджета отчета, указанного в профиле отчета;
- **Тема:** значение поля **Заголовок отчета** профиля отчета, а также, через пробел, поля **Субсидия** в кавычках;
- **Дата формирования отчета:** дата выгрузки файла отчета;
- **Период формирования отчета:** значение поля **по:** отчетного периода.

Задание доступно при подключении подсистемы ведения соглашений и мониторинга результатов предоставления субсидий юридическим лицам, индивидуальным предпринимателям, физическим лицам – производителям товаров, работ, услуг.

Принимает параметры:

- profile_name – профиль отчета в диалоговом окне.
- budget_id – идентификационный номер бюджета профиля.

Пример указания параметров в расписании планировщика:

```
PROFILE_NAME=Профил1;BUDGET_ID=200000003484
```

3.8.92 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.

3.8.93 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.

3.8.94 UniAnalyticalTaskProcessor

XML для запуска:

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

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

3.8.95 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](#)⁸⁹.

- **status** – статус документа. Может принимать значения: 6, 10, 28.
- **with_pay_docs** – параметр позволяет выгружать 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*.

3.8.96 UpdateRegionalListRegNum

XML для запуска:

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

Параметры: URL, log_path, name_subject.

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

Принимает параметры:

- **URL** – указывается адрес, с которого осуществляется загрузка информации, значение по умолчанию <https://budget.gov.ru/epbs/registry/7710568760-REGIONLISTS/data>. Обязательный параметр;
- **log_path** – указывается 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. Обязательный параметр;

- **name_subject** – указывается значение для фильтрации данных. Обязательный параметр.

3.8.97 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.

3.8.98 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

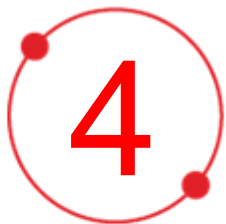
При необходимости выполнить какое-либо xml-задание, в том числе и размещенное в файле, (например, в каталоге */XML/...*) можно воспользоваться специальным сервисом меню **Сервис**→**XML-задания**. В данном диалоговом окне доступно вручную набрать xml-задание или открыть файл с таким заданием и выполнить 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 42 – 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 43 – 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.

База данных
Основные принципы работы с БД

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.

База данных
Основные принципы работы с БД

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](#)²⁸⁶.

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 **\XmlTemplate**. 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.00004-56 32 01-8](#)»



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 вкладке **Дополнительно** нажать

кнопку **Переменные среды**). В форме переменных сред присвоить путь к каталогу инсталляции JRE переменной окружения JAVA_HOME:

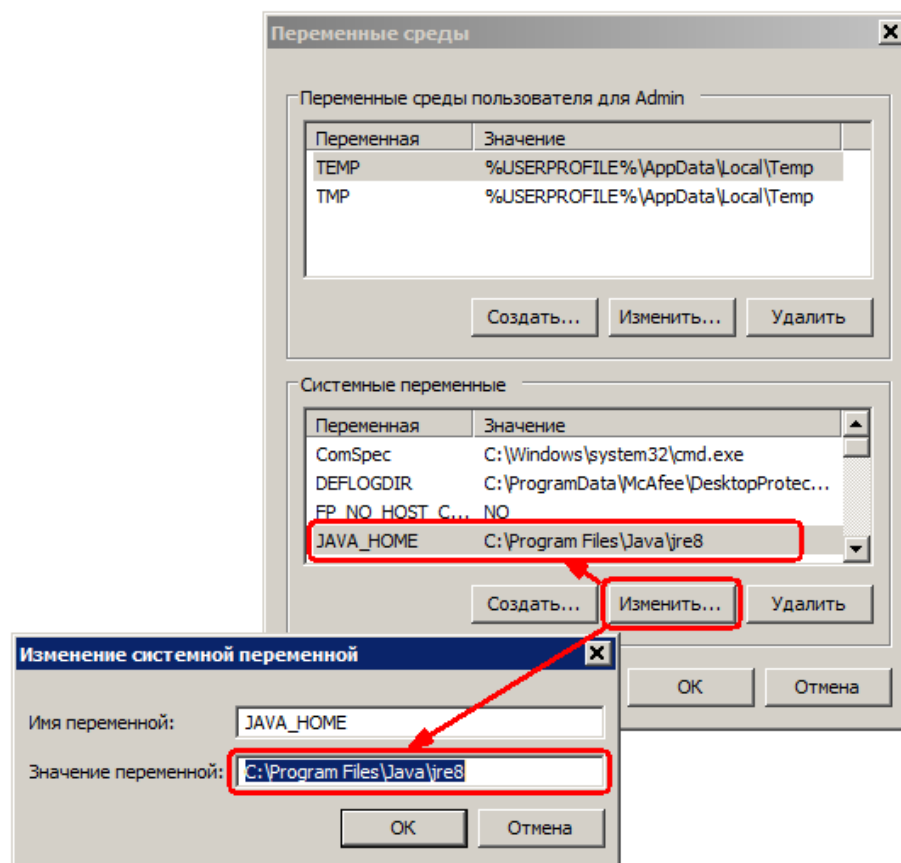


Рисунок 2 – Редактирование переменных сред

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](#)²⁸⁹.
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](#)⁴³ «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**, **xmirtl180.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.

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) ур

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**:

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 производится 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.

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) учр

- **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 ОКATO 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 44 – 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

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) учр

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 45 – 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 46 – 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**³⁰⁷.

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](#)³⁰⁹.

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:

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) уч

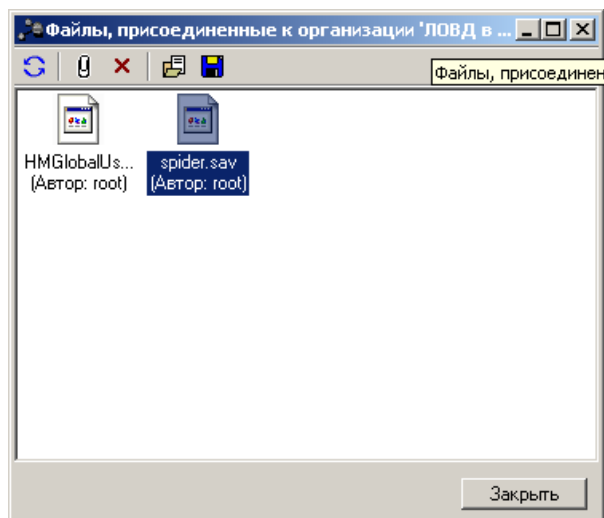



Рисунок 4 – Список присоединенных файлов к справочникам

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](#)³¹⁰, в котором выбирается присоединенный файл.³⁰⁹

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**:

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) уч

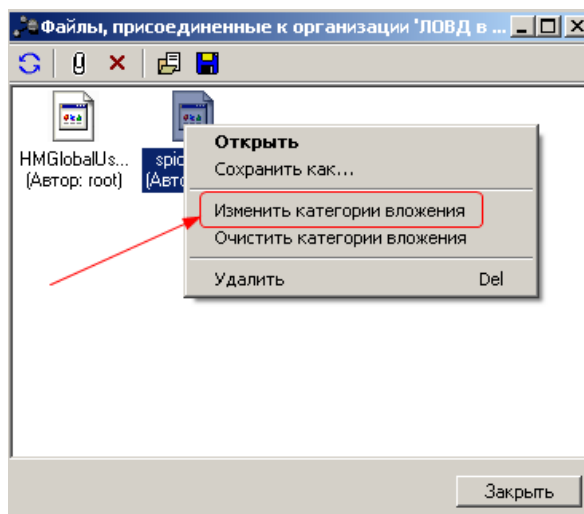



Рисунок 5 – Список присоединенных файлов, изменение категории вложения

Открывается справочник *Категории вложений*, где осуществляется множественный выбор значений³¹².

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:

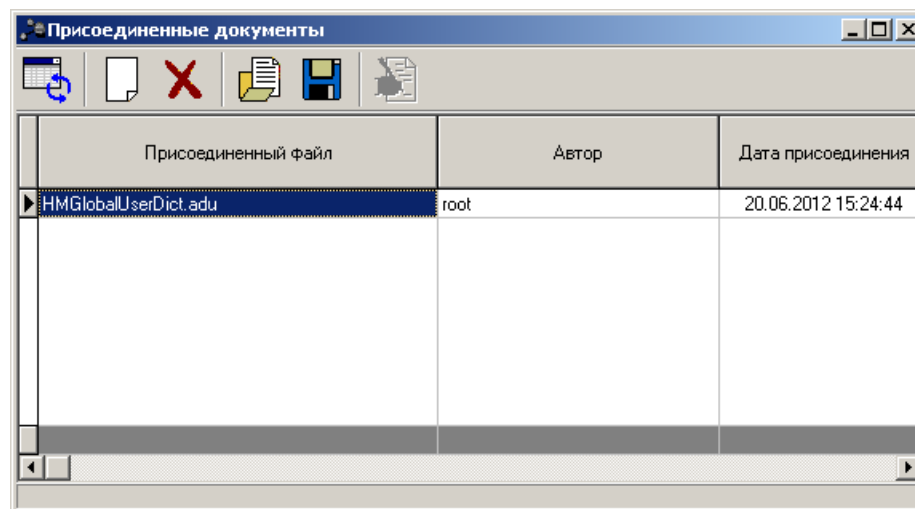


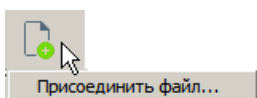
Рисунок 6 – Список присоединенных файлов

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:

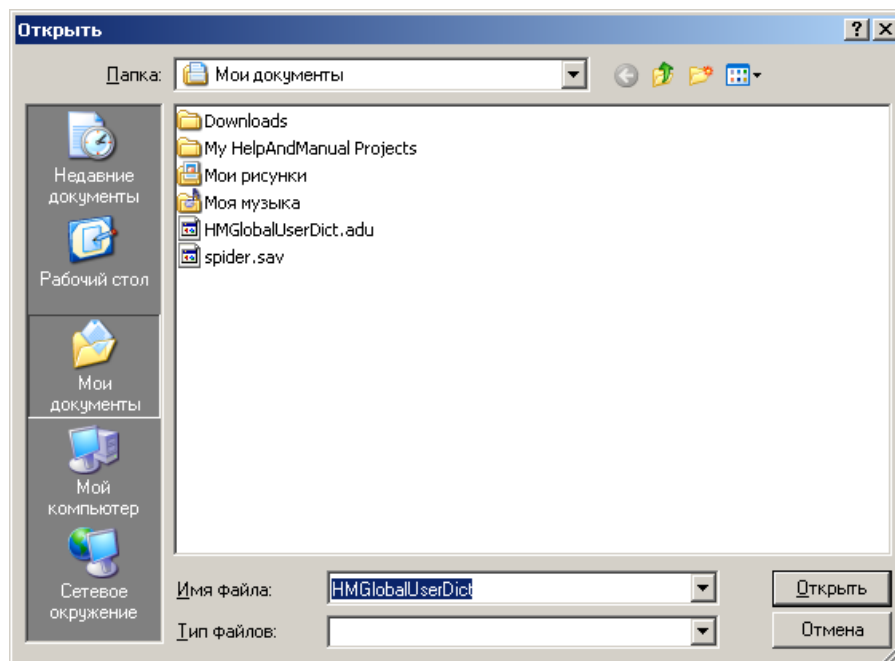


Рисунок 7 – Окно выбора присоединяемого файла

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

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) у

доступ только к одной категории, эта категория выбирается и файл прикрепляется. В противоположном случае проверяется значение системного параметра **Выбор категорий вложения** (пункт меню **Сервис**→**Системные параметры**→**Присоединенные файлы**→**Выбор категорий вложения по классам документов**), 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**:

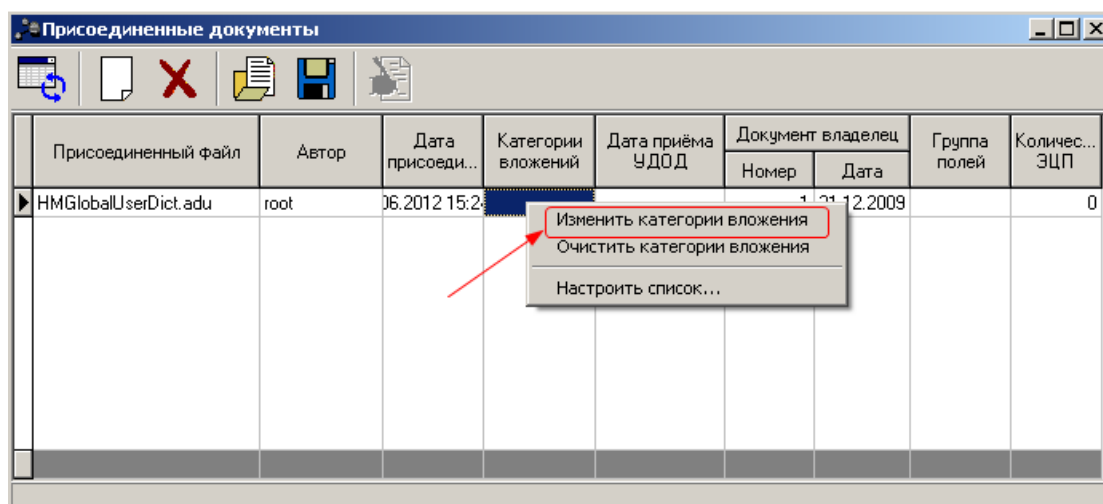


Рисунок 8 – Список присоединенных файлов, изменение категории вложения

Открывается справочник *Категории вложений*, где осуществляется множественный выбор значений:

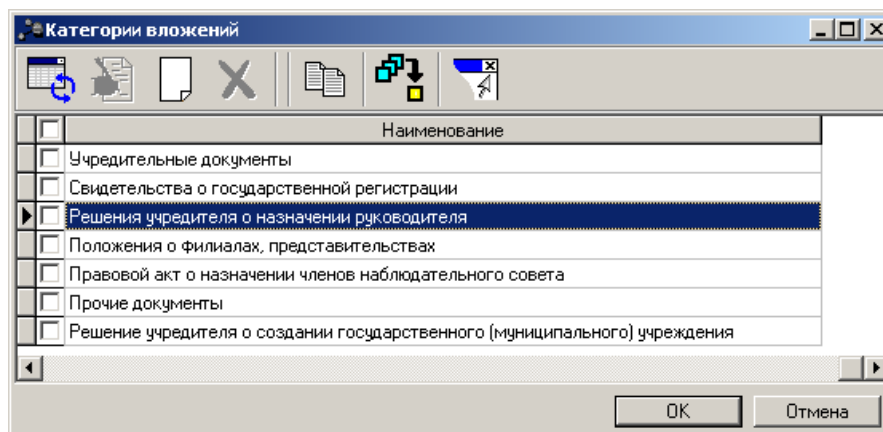


Рисунок 9 – Выбор категории вложений

Для закрытия справочника *Категории вложений* нажимается кнопка **ОК** или **Отмена**.

Выход из списка файлов, присоединенных к документу, осуществляется нажатием кнопки **Заккрыть**.

***Примечание.** Количество и размер присоединяемых файлов настраивается через пункт меню **Сервис**→**Системные параметры**, группа настроек **Присоединенные документы** (подробнее см. в документации «[БАРМ.00004-56 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>**:

Установка и настройка системы

Настройка выгрузки данных на Официальный сайт для размещения информации о государственных (муниципальных) ур

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:

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.00004-56 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

Клиентское приложение АЦК может подключаться к серверу приложений АЦК по TLS соединению (https):

- Поддерживается работа клиентского приложения с сервером приложений по TLS с использованием RSA ключей;
- Поддерживается режим односторонней проверки подключения к серверу;
- Поддерживается режим двусторонней проверки подключения;
- Сервер проверяет возможность доступа клиентского приложения на основе клиентского сертификата;
- Поддерживается работа клиентского приложения по TLS с внешним веб сервером, осуществляющим балансировку подключений по нескольким серверам приложений.

Примечание. Ключи и сертификаты необходимые для организации защищенного подключения могут быть сформированы как самостоятельно так и могут быть получены от иных (внешних) удостоверяющих центров.

➤ Прямое подключение клиентского приложения АЦК к серверу приложений АЦК:

1) Настраивается встроенный в сервер приложений АЦК tomcat:

- настраивается возможность подключения клиентских приложений к серверу приложений АЦК через http/https (файл свойств и конфигурационный файл встроенного tomcat);
- настраивается работа с TLS портом;
- для односторонней проверки сервера настраивается только работа с ключом и сертификатом

сервера;

- для двусторонней проверки сервера и клиентского приложения;
 - настраивается работа с ключом и сертификатом сервера;
 - создается файл-хранилище для сертификатов клиентских приложений, куда помещаются все сертификаты клиентских приложений имеющих право доступа к серверу приложений.

2) Настраивается подключение клиентского приложения АЦК:

- настраивается возможность подключения клиентского приложений к серверу приложений АЦК через http/https;
- для односторонней проверки подключения к серверу указывается сертификат **УЦ** выдавшего сертификат сервера;
- для двусторонней проверки сервера и клиентского приложения;
 - указывается сертификат **УЦ** выдавшего сертификат сервера;
 - указывается сертификат, ключ и пароль на ключ клиентского приложения.

➤ Подключение клиентского приложения АЦК через сторонний веб сервер, осуществляющий балансировку подключений к серверам приложений АЦК:

1) Настраивается сторонний веб-сервер:

- настраивается подключение к серверам приложений АЦК по протоколу http в режиме балансировки;
- для односторонней проверки сервера настраивается только работа с ключом и сертификатом сервера;
- для двусторонней проверки сервера и клиентского приложения настраивается работа с ключом и сертификатом сервера согласно документации на используемый веб сервер;

2) Настраивается встроенный в сервер приложений АЦК tomcat. Настраивается возможность подключения клиентских приложений к серверу приложений АЦК через http/https (файл свойств и конфигурационный файл встроенного **tomcat**);

3) Настраивается подключение клиентского приложения АЦК:

- настраивается возможность подключения клиентского приложений к серверу приложений АЦК через http/https;
- для односторонней проверки подключения к серверу указывается сертификат **УЦ** выдавшего сертификат сервера;
- для двусторонней проверки сервера и клиентского приложения;
 - указывается сертификат **УЦ** выдавшего сертификат сервера;

- указывается сертификат, ключ и пароль на ключ клиентского приложения.

➤ Требования к ключам и сертификатам:

- Для односторонней работы должен быть получен ключ и сертификат сервера приложений;
- Поддерживаются ключи с криптографическим алгоритмом **RSA**;
- Ключи должны быть выданы через удостоверяющий центр. Ключи не должны быть самоподписанными;
- Ключи сервера должны содержать атрибуты:
 - **Использование ключа (keyUsage)** с параметрами Шифрование данных, Шифрование ключей, Цифровая подпись;
 - **Улучшенный ключ (extendedKeyUsage)** с параметрами Проверка подлинности сервера (1.3.6.1.5.5.7.3.1), Проверка подлинности клиента (1.3.6.1.5.5.7.3.2);
- Ключи клиентского приложения должны содержать атрибуты:
 - **Использование ключа (keyUsage)** с параметрами Шифрование данных, Шифрование ключей, Цифровая подпись;
 - **Улучшенный ключ (extendedKeyUsage)** с параметром Проверка подлинности клиента (1.3.6.1.5.5.7.3.2);
- Сертификаты ключей сервера должны содержать дополнительный атрибут **Дополнительное имя субъекта (subjectAltName OID 2.5.29.17)**. В дополнительном имени субъекта должны быть указаны корректные **DNS/IP** соответствующие адресам серверов приложений АЦК, на которые осуществляется подключение клиентских приложений;
- Сертификаты и ключи для серверной и клиентских частей удобнее использовать в формате **PKCS12 (*.pfx)**;
- Сертификаты **УЦ**, сервера и клиентских частей в формате **X509** и кодировке **base64** могут потребоваться для настроек подключений через сторонний веб-сервер.
- Сертификаты и ключи сервера/клиента могут быть получены у любого **УЦ**, но должны соответствовать обозначенным в данном разделе требованиям.

Настройки сервера приложений:

➤ Работа с клиентским приложением по http/https:

1) Настраивается файл свойств сервера приложений АЦК, указанный в запускаящем файле:

azk.tomcat.embedded.config - указывается наименование и расположение файла конфигурации встроенного tomcat;

Пример. azk.tomcat.embedded.config=tomcat_server.xml

`azk.servlet.exec.mapping` - указывается наименование точки подключения клиентского приложения для работы с сервером приложений АЦК по http/http;

Пример. azk.servlet.exec.mapping = /exec

2) Настраивается конфигурационный файл встроенного tomcat:

Настраивается порт подключений по http/https:

- `port="10183"` – номер порта на котором будут приниматься подключения;
- `protocol="HTTP/1.1"` – протокол;
- `maxPostSize="52428800"` – максимальный размер *POST* запроса;
- `tcpNoDelay="true"` – включение опции сетевого сокета *TCP_NO_DELAY*;
- `maxCookieCount="5000"` – количество *cookie*, рассчитывается исходя из количества подключений и количества *cookie* создаваемых приложением для подключения;
- `acceptCount="500"` – количество подключений в очереди если достигнут лимит подключений.

➤ Настройка односторонней проверки на стороне сервера приложений АЦК:

Для работы встроенного **tomcat** по **TLS** с односторонней проверкой необходимы дополнительные настройки порта для подключений в конфигурационном файле встроенного **tomcat**.

Пример. tomcat_server.xml

- `SSLEnabled="true"` – включение **TLS**;
- `scheme="https"` – ответ на запрос схемы использования https;
- `secure="true"` – ответ на запрос безопасности соединения;
- `clientAuth="false"` – отключение проверки сертификата подключаемого клиента (двусторонняя проверка);
- `sslProtocol="TLS"` – тип **SSL** протокола **TLS**;
- `keystoreFile="..server.pfx"` – расположение и наименование файла содержащего сертификат и ключ серверной части (в формате *PKCS12*). Путь указывается относительно временной папки создаваемой сервером приложений при работе с подключениями через встроенный **tomcat**. Генерируемое наименование папки **tomcat.<номер порта http/https>**;
- `keystorePass="123"` – пароль на контейнер с сертификатом и ключом серверной части;

Пример. server.pfx

- `keystoreType="PKCS12"` – тип контейнера с сертификатом и ключом в формате *PKCS12*;

- **keyAlias="azkserver"** – условное именованное обозначение (алиас) серверного ключа;
- **algorithm="SunX509"** – алгоритм работы с контейнером содержащим сертификат серверной части.

➤ Настройка двусторонней проверки на стороне сервера приложений АЦК:

Для работы встроенного **tomcat** по **TLS** с двусторонней проверкой необходимы дополнительные настройки порта для подключений в конфигурационном файле встроенного **tomcat**.

Пример. tomcat_server.xml

- **SSLEnabled="true"** – включение **TLS**;
- **scheme="https"** – ответ на запрос схемы использования https;
- **secure="true"** – ответ на запрос безопасности соединения;
- **clientAuth="false"** – отключение проверки сертификата подключаемого клиента (двусторонняя проверка);
- **sslProtocol="TLS"** – тип **SSL** протокола **TLS**;
- **keystoreFile="..server.pfx"** – расположение и наименование файла содержащего сертификат и ключ серверной части (в формате *PKCS12*). Путь указывается относительно временной папки создаваемой сервером приложений при работе с подключениями через встроенный **tomcat**. Генерируемое наименование папки **tomcat.<номер порта http/https>**;
- **keystorePass="123"** – пароль на контейнер с сертификатом и ключом серверной части;

пример, server.pfx

- **keystoreType="PKCS12"** – тип контейнера с сертификатом и ключом в формате *PKCS12*;
- **keyAlias="azkserver"** – условное именованное обозначение (алиас) серверного ключа;
- **algorithm="SunX509"** – алгоритм работы с контейнером содержащим сертификат серверной части;
- **truststoreFile="..mytruststore.jks"** – расположение файла содержащего хранилище сертификатов клиентов имеющих право подключиться к серверу;
- **truststorePass="123"** – пароль к файлу хранилищу сертификатов клиентов;
- **truststoreType="JKS"** – тип хранилища, например, *JavaKeyStore*;
- **truststoreAlgorithm="SunX509"** – алгоритм работы с сертификатами в хранилище.

➤ Создание файла хранилища сертификатов клиентов в формате JavaKeyStore (JKS):

В случае использования хранилища сертификатов в формате *JKS*, рассматриваемого в примере настройки, необходимо воспользоваться утилитой **Java - keytool**.

```
keytool -import -alias client -keyalg RSA -keystore mytruststore.jks -storetype JKS -file client.crt
```

- **import** – импорт сертификата в хранилище;
- **alias client** – алиас хранилища;
- **keyalg RSA** – сертификаты для ключей **RSA**;
- **keystore mytruststore.jk** – наименование файла хранилища сертификатов. создается в той папке где была запущена утилита **keytool**;
- **storetype JKS** – тип хранилища **JKS**;
- **file client.crt** – импорт сертификата из файла **client.crt**.

В хранилище необходимо добавить все сертификаты всех клиентов которые должны будут подключаться к серверу приложений АЦК.

➤ Примеры настроек:

1) Файл свойств сервера приложений (*.properties):

```
# -----  
# ! Параметры встроенного томката !  
# -----  
# Рабочая папка. По дефолту берется из java.io.tmpdir  
azk.tomcat.embedded.basedir = ./tmp  
# Конфигурационный файл с описанием коннекторов встроенного томката (опционально)  
# из настройки берется только коннектор  
azk.tomcat.embedded.config=tomcat_server.xml  
# Брать из конфигурационного файла в том числе и настройки Engine (по умолчанию false)  
#azk.tomcat.embedded.engine.enable=false  
  
# -----  
# ! Параметры сервлета, принимающего команды от делфи-клиента !  
# -----  
# Мappings  
azk.servlet.exec.mapping = /exec
```

2) Файл параметров встроенного tomcat (фрагмент *tomcat_server.xml*):

```
<Connector port="9183" protocol="HTTP/1.1"  
    connectionTimeout="20000"  
    maxPostSize="52428800"
```



```

        tcpNoDelay="true"
        maxCookieCount="5000"
        acceptCount="500"
        redirectPort="10183"
    />

    <Connector port="10183" protocol="HTTP/1.1"
        connectionTimeout="20000"
        maxPostSize="52428800"
        tcpNoDelay="true"
        maxCookieCount="5000"
        acceptCount="500"
        SSLEnabled="true"
        scheme="https"
        secure="true"
        clientAuth="true"
        sslProtocol="TLS"
        keystoreFile=" ../server.pfx"
        keystorePass=
123"
        keystoreType="PKCS12"
        keyAlias="azk server"
        algorithm="SunX509"
        truststoreFile=" ../mytruststore.jks"
        truststorePass="123"
        truststoreType="JKS"
        truststoreAlgorithm="SunX509"
    />

```

Настройки клиентского приложения АЦК:

➤ Настройки подключения к серверу приложений по протоколам http/https:

Настройки необходимые для работы клиентского приложения АЦК проводятся в файле настроек подключений - по умолчанию Azk2CInt.ini:

- Блок **SERVER** - параметр ServerMode=HTTP - работа клиентского приложения с сервером приложений АЦК по протоколам http/https;

Установка и настройка системы

Настройка соединения Win-клиента через шифрованный канал https

- Блок **HTTP** - параметр **URL**=<протокол http или https>://<адрес сервера приложений dns или IP>:<порт подключения>/<точка подключения>;

Пример. URL=https://192.168.1.111:443/ehes - подключение к серверу приложений по протоколу https (для примера) на порт 443 на точку подключения /ehes

- Блок **HTTP** - параметр **HttpLib**=<библиотека для работы по http/https> - указывается тип библиотеки для работы с http/https либо **INDY** либо **WINHTTP**;
 - **INDY** - компоненты delphi - по умолчанию;
 - **WINHTTP** - работа через API Windows - **IWinHttpRequest**.
- Блок **HTTP** - параметр **ProxyEnable**=true/false - включение/отключение работы через http/https прокси. По умолчанию **false**;
- Блок **HTTP** - параметр **ProxyServer**=<адрес прокси сервера> - работа через прокси сервер http/https;
- Блок **HTTP** - параметр **ProxyPort**=<порт прокси сервера>;
- Блок **HTTP** - параметр **ProxyAuthentication**=<тип авторизации через прокси сервер>;
 - **none/false** - без авторизации;
 - **manual/true** - имя пользователя и пароль от прокси сервера будут взяты из параметров **ProxyUser** и **ProxyPassword**;
 - **auto** - аутентификационные данные будут взяты из системы.
- Блок **HTTP** - параметр **ProxyUser**=<пользователь для аутентификации на прокси сервере>;
- Блок **HTTP** - параметр **ProxyPassword**=<пароль для прокси сервера>.

➤ Настройки подключения к серверу приложений по TLS с односторонней проверкой:

- Блок **HTTP** - параметр **CACertificate**=<путь и имя файла в формате x509, кодировке base64, содержащего сертификат УЦ которым был выдан сертификат сервера>.

Пример. ACertificate=ca.crt - файл расположен в папке где находится исполняемый файл клиентского приложения и имеет наименование ca.crt

➤ Настройки подключения к серверу приложений по TLS с двусторонней проверкой:

- Блок **HTTP** - параметр **CACertificate**=<путь и имя файла в формате x509, кодировке base64, содержащего сертификат УЦ которым был выдан сертификат сервера>;

Пример. ACertificate=ca.crt - файл расположен в папке где находится исполняемый файл клиентского приложения и имеет наименование ca.crt

- Блок **HTTP** - параметр **ClientCertificate**=<путь и имя файла, содержащего сертификат клиента>. Формат файла может быть как **PKCS12 (pfx)**, так и **X509/base64 (pem,cer,crt)**;

Пример. ClientCertificate=client.pfx - файл формата PKCS12 расположен в папке где находится исполняемый файл клиентского приложения и имеет наименование client.pfx

- Блок **HTTP** - параметр **ClientPrivateKey=<путь и имя файла, содержащего закрытый ключ клиента>**. Формат файла может быть как *PKCS12 (pfx)*, так и *X509/base64 (pem,cer,crt)*;

Пример. ClientPrivateKey=client.pfx - файл формата PKCS12 расположен в папке где находится исполняемый файл клиентского приложения и имеет наименование client.pfx

- Блок **HTTP** - параметр **ClientPrivateKeyPass=<пароль к закрытому ключу клиента>**. Если пароль будет не указан, потребуется ввод пароля от ключа при подключении клиентского приложения к серверу приложений АЦК.

Пример. ClientPrivateKeyPass=123 - пароль к ключу клиента 123

➤ Настройки подключения к серверу приложений по TLS если используется тип библиотеки WINHTTP:

- Сертификат УЦ, сертификат и ключ клиентского приложения (для двусторонней проверки) - должны быть импортированы в соответствующие хранилища ОС;
- Блок **HTTP** - параметр **ClientCertificate=<путь и CN сертификата клиентского приложения в хранилище ОС>**.

Пример. ClientCertificate=CURRENT_USER\My\client

➤ Настройки подключения по TLS при использовании стороннего веб сервера:

1) На стороне сервера приложений АЦК настраивается возможность подключения по http;

2) На стороне стороннего веб сервера настраивается средствами этого веб сервера:

- возможность подключения по https;
- возможность двусторонней проверки - если требуется;
- в случае настройки двусторонней проверки - настраивается вывод ошибки 403 в случае если клиентской часть предъявлен некорректный сертификат;
- балансировка на http порты серверов приложений АЦК.

3) На стороне клиентского приложения АЦК настраивается:

- возможность подключения по https на адрес и порт веб сервера, с указанием сертификатов(цепочки сертификатов) УЦ;
- возможность двусторонней проверки - если требуется.

Пример настроек для выдачи сертификатов и ключей с помощью открытого**ПО openssl:**

➤ Необходимо создать файл конфигурации openssl для генерации ключа и сертификата УЦ, ключа и сертификата сервера, ключа и сертификата клиента.

В файле должно быть настроено:

- **KEY_SIZE** – Длина ключа;
- **KEY_COUNTRY** – обозначение страны;
- **KEY_PROVINCE** – обозначение региона;
- **KEY_CITY** – обозначение города;
- **KEY_ORG** – обозначение организации;
- **KEY_ORGUNIT** – обозначения подразделения;
- **KEY_EMAIL** – обозначение адреса электронной почты;
- **HOME** – папка расположения данных;
- **KEY_DIR** – папка для расположения ключей;
- **RANDFILE** – файл генератора случайных чисел;
- **default_md** – функция хеша;
- блоки политик – **polycy** – Возможность ввода атрибутов сертификата при его генерации;
- блоки шаблонов генерации ключей/сертификатов – **usr_cert, server, v3_ca** – Использование и расширенное использование ключа для ключей УЦ, сервера и клиента;
- блоки дополнительной информации для ключей и сертификатов – **alt_names, alt_server_name** – дополнительные имена субъектов сертификатов необходимые для сверки имени сервера с адресом подключения.

➤ Необходимо настроить дополнительные файлы для дополнительных атрибутов сертификатов:

1) **v3_ca.ext**, для примера, содержит дополнительные имена и адреса сервера УЦ:

```
subjectAltName = @alt_names
```

```
[alt_names]
```

```
DNS.1           = oel8x64.local
```

```
DNS.2           = oel8x64
```

```
DNS.3           = 127.0.0.1
```

```
DNS.4      = ::1
IP.1       = 172.0.0.1
IP.2       = ::1
IP.3       = 192.168.1.111
```

2) **v3_server.ext**, для примера, содержит дополнительные имена и адреса сервера приложений:

```
subjectAltName = @alt_server_names
```

```
[alt_server_names]
```

```
DNS.1      = hdesktop.local
DNS.2      = hdesktop
DNS.3      = 127.0.0.1
DNS.4      = ::1
DNS.5      = localhost
IP.1       = 172.0.0.1
IP.2       = ::1
IP.3       = 192.168.1.4
```

➤ Пример создания ключей и сертификатов для сервера (openssl под Linux):

1) Создаем переменную **OPENSSL_CONF** для использования нашего файла конфигурации **openssl**;

```
export OPENSSL_CONF=openssl.cfg
```

2) Генерируем ключ **RSA УЦ**:

- с именем файла **ca.key**;
- длиной ключа 4069;

```
openssl genrsa -out ca.key 4096
```

3) Генерируем сертификат **УЦ**:

- с именем файла **ca.crt**;
- на 1095 дней;
- в формате **x509**;
- атрибуты берутся из конфигурационного файла **openssl** и вводятся при генерации сертификата;

```
openssl req -new -sha256 -x509 -days 1095 -key ca.key -out ca.crt
```

4) Генерируем ключ сервера АЦК (или стороннего веб сервера) и запрос на сертификат:

- с именем файла **serverkey.pem**;
- используя блок **server** и связанные с ним блоки из конфигурационного файла **openssl**;
- запрос на сертификат с именем файла **server.csr**;

```
openssl genrsa -out serverkey.pem 4096
```

```
openssl req -new -key serverkey.pem -sha256 -out server.csr -extensions server
```

5) Генерируем сертификат ключа сервера АЦК (или стороннего веб сервера):

- по формату **x509**;
- с именем файла **server.crt**;
- подписывая ключом ранее созданного **УЦ**;
- используя файл доп. свойств сертификата **v3-server.ext** (доп. имена субъекта);

```
openssl x509 -req -in server.csr -days 365 -sha256 -CAcreateserial -CA ca.crt -CAkey ca.key -out server.crt -extfile v3-server.ext
```

6) Создаем файл по формату PKCS12 для сервера приложений АЦК (или стороннего веб сервера):

- используем сертификат **server.crt**;
- используем ключ **serverkey.pem**;
- даем алиас **azkserver**;
- с именем файла **server.pfx**;

```
openssl pkcs12 -export -in server.crt -inkey serverkey.pem -name "azkserver" -out server.pfx
```

7) Генерируем ключ клиента АЦК и запрос на сертификат:

- с именем файла **clientkey.pem**;
- используя блок настроек **usr_cert** и связанные с ним блоки из конфигурационного файла **openssl**;
- запрос на сертификат с именем файла **client.csr**;

```
openssl genrsa -out clientkey.pem 4096
```

```
openssl req -new -key clientkey.pem -sha256 -out client.csr -extensions usr_cert
```

8) Генерируем сертификат ключа клиента АЦК:

- по формату **x509**;
- с именем файла **client.crt**;

- подписывая ключом ранее созданного УЦ;

```
openssl x509 -req -in client.csr -days 365 -sha256 -CAcreateserial -CA ca.crt -CAkey ca.key -out client.crt
```

9) Создаем файл по формату **PKCS12** для клиентского приложения АЦК:

- используем сертификат **client.crt**;
- используем ключ **clientkey.pem**;
- даем алиас **client**;
- с именем файла **client.pfx**;

```
openssl pkcs12 -export -in client.crt -inkey clientkey.pem -name "client" -out client.pfx
```

5.8 Установка и настройка Nginx

5.8.1 Предварительная работа

Для успешного запуска **Nginx** на ОС Linux предварительно необходимо установить пакеты **OPENSSL** (если планируется использование **SSL**), **Zlib**, **PRCE**.

Для того, чтобы установить пакет **OPENSSL** необходимо выполнить следующие действия:

- скачать дистрибутив с <https://www.openssl.org/source/openssl-0.9.8.tar.gz>;
- распаковать архив: **tar xvzf zlib-1.2.11.tar.gz**.

Для того, чтобы установить **Zlib** необходимо выполнить следующие действия:

- скачать дистрибутив с <http://www.zlib.net/zlib-1.2.11.tar.gz>;
- распаковать архив: **tar xvzf zlib-1.2.11.tar.gz**.

Для того, чтобы установить **PRCE** необходимо выполнить следующие действия:

- скачать дистрибутив с <https://sourceforge.net/projects/pcre/files/pcre/8.45/> библиотеку **pcre-8.45.tar.gz**;
- распаковать архив **tar xvzf pcre-8.45.tar.gz**.

5.8.2 Установка Nginx

Для того, чтобы установить nginx выполняются следующие действия:

- Скачать дистрибутив с <http://nginx.org/download/nginx-1.13.6.tar.gz>;
- Распаковать архив: `tar xvzf nginx-1.13.6.tar.gz`;
- Добавить пользователя и группу nginx, под которой будет работать сервер (если не существует ни того, ни другого).

```
groupadd nginx
```

```
useradd -m -s /dev/null -g nginx -d /usr/local/nginx nginx
```

- Сконфигурировать исходный код:

```
#!/configure --prefix=/usr/local/nginx/ --conf-path=/etc/nginx/nginx.conf --pid-  
path=/var/run/nginx.pid --user=nginx --with-http_ssl_module --with-http_gzip_static_module --  
with-http_stub_status_module --without-http_ssi_module --without-http_userid_module --  
without-http_autoindex_module --without-http_geo_module --without-http_referer_module --  
without-http_memcached_module --with-pcre=/tmp/pcre-8.45/ --with-openssl=/tmp/openssl-1.1.1g  
--with-zlib=/tmp/zlib-1.2.11/
```

Внимание! Выделенная часть кода должна ссылаться на исходники (Source).

```
Configuration summary
+ using PCRE library: /tmp/pcre-8.38/
+ using openssl library: /tmp/openssl-0.9.8
+ using zlib library: /tmp/zlib-1.2.11/

nginx path prefix: "/usr/local/nginx/"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx modules path: "/usr/local/nginx/modules"
nginx configuration prefix: "/etc/nginx"
nginx configuration file: "/etc/nginx/nginx.conf"
nginx pid file: "/var/run/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"
```

Рисунок 12 – Исходный код

При возникновении ошибки, необходимо добавить ключ – fPIC:

```
--with-pcre=/tmp/ontest/pcre-8.38/ --with-pcre-opt=-fPIC --with-openssl=/tmp/ontest/openssl-1.1.1a/ --  
with-openssl-opt=-fPIC --with-zlib=/tmp/ontest/zlib-1.2.11/ --with-zlib-opt=-fPIC --add-dynamic-  
module=/tmp/ontest/nginx-module-vts-master/
```


Без fPIC динамическая библиотека будет грузиться в память каждого процесса, который ее использует. С fPIC - все процессы будут разделять одну копию библиотеки.

```
objs/nginx_modules.o \
-Wl,-z,relro -Wl,-z,now -pie -ldl -lpthread -lpthread -lcrypt /etc/pcrcr-8.45//.libs/libpcrcr.a /etc/openssl-1.1.1g//.openssl/lib/libssl.a /et
c/openssl-1.1.1g//.openssl/lib/libcrypto.a -ldl -lpthread /etc/zlib-1.2.11//libz.a \
-Wl,-E
/bin/ld: /etc/pcrcr-8.45//.libs/libpcrcr.a(libpcrcr_la-pcrcr_compile.o): перемещение R_X86_64_32S для «_pcrcr_0P_lengths» не может использоваться
при создании общего объекта; перекомпилируйте с -fPIC
/etc/pcrcr-8.45//.libs/libpcrcr.a: could not read symbols: Некорректное значение
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
make[1]: *** [objs/nginx] Ошибка 1
make[1]: Выход из каталога `/etc/nginx'
make: *** [build] Ошибка 2
[root@webgz nginx]# S
```

Рисунок 13 – Ошибка

- Сибрать: **#make**;
- Выполнить: **#make install**.

5.8.3 Первичные настройки Nginx

Для корректной работы нужно задать следующие первичные настройки:

- Указать количество рабочих процессов. Количество рабочих процессов, зависит от ваших задач и ресурсов оборудования:

```
worker_processes 1
```

- Уточнить, куда писать логи ошибок в соответствии с уровнями уведомлений.

```
error_log logs/error.log;
error_log logs/error.log notice;
error_log logs/error.log info;
```

- Настроить секции **events**. Здесь можно задать максимальное количество соединений, которые одновременно будет обрабатывать один процесс-worker, и метод, который будет использоваться для получения асинхронных уведомлений о событиях в ОС. Конечно же, можно выбрать только те методы, которые доступны на вашей ОС, и были включены при компиляции. Эти параметры могут оказать значительное влияние на производительность сервера. Их надо подбирать индивидуально, в зависимости от ОС и комплектующих. Общие правила подбора параметров:

Модули работы с событиями:

- **select** и **poll** обычно медленнее и довольно сильно нагружают процессор, зато доступны

практически везде, и работают практически всегда;

— **kqueue** и **epoll** — более эффективны, но доступны только во FreeBSD и Linux 2.6, соответственно;

— **rtsig** — довольно эффективный метод, и поддерживается даже очень старыми линуксами, но может вызывать проблемы при большом числе подключений;

— **/dev/poll** — работает в несколько более экзотических системах, типа соляриса, и в нём довольно эффективен.

- Параметр **worker_connections** отвечает за общее максимальное количество обслуживаемых клиентов будет равно **worker_processes * worker_connections**;
- Системный вызов **sendfile**. Позволяет отправить данные в сеть, минуя этап их копирования в адресное пространство приложения. Во многих случаях это существенно повышает производительность сервера, поэтому параметр **sendfile** лучше всегда включать.

```
sendfile on
```

- Параметр **keepalive_timeout** отвечает за максимальное время поддержания keepalive-соединения, в случае, если пользователь по нему ничего не запрашивает.

```
keepalive_timeout 65
```

- Настройки проксирования **nginx**. Чаще всего, nginx используется именно как сервер-прокси, соответственно они имеют довольно большое значение. В частности, размер буфера для проксируемых запросов имеет смысл устанавливать не менее, чем ожидаемый размер ответа от сервера-бэкенда. При медленных (или, наоборот, очень быстрых) бэкендах, имеет смысл изменить таймауты ожидания ответа от бэкенда. Чем больше эти таймауты, тем дольше будут ждать ответа ваши пользователи, при замедленной работе бэкенда.

```
proxy_buffers 8 64k;
```

```
proxy_intercept_errors on;
```

```
proxy_connect_timeout 1s;
```

```
proxy_read_timeout 3s;
```

```
proxy_send_timeout 3s
```

- В nginx начиная с версии 0.7.44 появилась возможность кешировать отдаваемые

страницы, что может увеличить работы сайта в тысячу раз. Как это правильно сделать: сперва создадим папку, где nginx будет хранить данные кеша.

```
proxy_cache_path temp/cashe levels=2 keys_zone=default:5m inactive=10m max_size=50m
```

Здесь:

Создаем кеш зону **default** (память под ключи в 5Мб) с настройками:

inactive: хранить кеш 10 минут

max_size: максимальный размер кеш данные 50Мб

```
#user nobody;
worker_processes 1;

error_log logs/error.log;
error_log logs/error.log notice;
error_log logs/error.log info;

pid logs/nginx.pid;

events {
    worker_connections 1024;
    use select;
}

http {
    include mime.types;
    default_type application/octet-stream;

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    #                '$status $body_bytes_sent "$http_referer" '
    #                '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log logs/access.log main;

    sendfile on;
    #tcp_nopush on;

    #keepalive_timeout 0;
    keepalive_timeout 65;

    proxy_cache_path temp/cashe levels=2 keys_zone=default:5m inactive=10m max_size=50m;

    proxy_buffers 8 64k;
    proxy_intercept_errors on;
    proxy_connect_timeout 1s;
    proxy_read_timeout 3s;
    proxy_send_timeout 3s;
```

Рисунок 14 – Кэш зона

- У **Nginx** есть свои виртуальные хосты. Поэтому для нашего сайта нужно указать виртуальный хост **localhost**:

```
listen 81;
server_name localhost
```

Кодировка по умолчанию:

```
charset koi8-r
```

Описать **location**, которые будут вести на бэкенд, к **apache**, запущенному на порту

8080:

```
location / {
proxy_pass http://127.0.0.1:8080;
proxy_redirect off;
log_not_found off;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header Host $http_host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_connect_timeout 120;
proxy_send_timeout 120;
proxy_read_timeout 180;
proxy_cache_valid 200 301 302 304 5m;
proxy_hide_header "Set-Cookie";
proxy_ignore_headers "Cache-Control" "Expires";
proxy_cache_use_stale error timeout invalid_header http_500 http_502 http_503 http_504;
proxy_cache default
```

Здесь:

```
proxy_set_header Host $host; — передаем имя хоста.
proxy_set_header X-Real-IP $remote_addr; — передаем IP адрес запрашивающего в заголовке X-Real-IP (иначе в логах Apache будут адреса Nginx сервера, а не реальных клиентов).
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; — Заголовок X-Forwarded-For содержит список прокси серверов, по которым прошел запрос до настоящего момента. Передаем ему значение переменной $proxy_add_x_forwarded_for. Она содержит в себе полученный заголовок X-Forwarder-For плюс добавляет свой сервер в этот список.
proxy_connect_timeout 120; — время таймаута для подключения к проксируемому серверу (120 секунд).
proxy_send_timeout 120; — время таймаута для отправки данных на проксируемый сервер (120 секунд).
```

proxy_read_timeout 180; — время таймаута для чтения ответа от проксируемого сервера (180 секунд).

proxy_cache_valid 200 301 302 304 5m; - Кешировать указанные коды ответов 5 минут

proxy_hide_header "Set-Cookie";- Защита от раздачи одинаковой куки в кешированном ответе

proxy_ignore_headers "Cache-Control" "Expires"; - Игнорировать параметры кеша заданные бекэндом

proxy_cache_use_stale error timeout invalid_header http_500 http_502 http_503 http_504;-

Указывает в каких случаях клиенту можно отдать несвежий ответ из кеша

proxy_cache default;- активировать зону кеширования default

Чтобы отключить кеширование для конкретного **location**, то нужно указать внутри него **proxy_cache off**.

```

server {
    listen      81;
    server_name localhost;

    charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    html;
        index  index.html index.htm;
    }

    #error_page 404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root    html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    location ~ /\.php$ {

        proxy_pass      http://127.0.0.1:8080;
        proxy_redirect  off;
        log_not_found   off;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_connect_timeout 120;
        proxy_send_timeout 120;
        proxy_read_timeout 180;
        proxy_cache_valid 200 301 302 304 5m;
        proxy_hide_header "Set-Cookie";
        proxy_ignore_headers "Cache-Control" "Expires";
        proxy_cache_use_stale error timeout invalid_header http_500 http_502 http_503 http_504;
        proxy_cache default;
    }
}

```

Рисунок 15 – Виртуальные хосты

- Проверить правильность синтаксиса в **nginx -t**. Должно быть – **test is successful**;
- **nginx** работает как стандартное консольное приложение (не сервис) и управляется при помощи следующих команд:

nginx -s stop – быстрое завершение;

nginx -s quit – плавное завершение;

nginx -s reload – изменение конфигурации, запуск новых рабочих процессов с новой конфигурацией, плавное завершение старых рабочих процессов;

nginx -s reopen – переоткрытие лог-файлов;

- Проверить работоспособность **nginx**:

<http://localhost:81/>

Первичная настройка проведена:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Рисунок 16 – Проверка работоспособности

5.8.4 Настройка «липких» сессий

В **Nginx** существует метод **sticky** использующий cookie для балансировки, правда только в коммерческой версии. Есть и бесплатный путь — использование внешних модулей.

5.8.4.1 Nginx-sticky-module

Указанный модуль создает **cookie** — чем делает каждый браузер уникальным — и далее использует его для переадресации запросов на один и тот же сервер. При отсутствии **cookie**, например, при первом запросе, сервер выбирается случайным образом. Проект был разветвлен, оригинальная версия больше не поддерживается, поддерживаемое ответвление называется **nginx-sticky-module-ng** и находится здесь <https://bitbucket.org/nginx-goodies/nginx-sticky-module-ng/src/master/>. Обратите внимание на подзаголовок **DEPRECATED** и ниже на ссылку на поддерживаемую версию — перед инсталляцией модуля потребуется внести в исходный код некоторые изменения. Дело в том, что начиная с версии **nginx 1.5.8** поменялся API для nginx метода **ngx_sock_ntop()**, поэтому в файле **ngx_http_sticky_misc** с архива **nginx-sticky-module** строку **digest->len = ngx_sock_ntop(in,**

digest->data, len, 1) стоит заменить на **digest->len = ngx_sock_ntop(in, sizeof(struct sockaddr_in), digest->data, len, 1)**.

Для того, чтобы узнать, с какими параметрами был установлен **nginx** (в том числе, из репозитория), нужно набрать команду:

```
nginx -V
```

Чтобы установить этот модуль, нужно скомпилировать **Nginx** с этим модулем:

```
./configure --prefix=/usr/local/nginx/ --conf-path=/etc/nginx/nginx.conf --pid-path=/var/run/nginx.pid --
user=nginx --with-http_ssl_module --with-http_gzip_static_module --with-http_stub_status_module --
without-http_ssi_module --without-http_userid_module --without-http_autoindex_module --without-
http_geo_module --without-http_referer_module --without-http_memcached_module --with-
pcre=/tmp/pcre-8.38/ --with-openssl=/tmp/openssl-1.1.1g --with-zlib=/tmp/zlib-1.2.11/ --add-
module=/tmp/nginx-sticky-module
make
make install clean
```

Загрузить

init.d

script

c

<https://www.nginx.com/resources/wiki/start/topics/examples/redhatnginxinit> (только для RHEL, Fedora, CentOS. Tested on CentOS 5.).

```
#!/bin/sh
#
# nginx - this script starts and stops the nginx daemon
#
# chkconfig: - 85 15
# description: NGINX is an HTTP(S) server, HTTP(S) reverse \
#             proxy and IMAP/POP3 proxy server
# processname: nginx
# config:     /etc/nginx/nginx.conf
# config:     /etc/sysconfig/nginx
# pidfile:    /var/run/nginx.pid
# Source function library.
. /etc/rc.d/init.d/functions
# Source networking configuration.
```



```

/etc/sysconfig/network

# Check that networking is up.
[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/sbin/nginx"
prog=$(basename $nginx)

NGINX_CONF_FILE="/etc/nginx/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make_dirs() {
    # make required directories
    user=`$nginx -V 2>&1 | grep "configure arguments:.*--user=" | sed 's/[^\ ]*--user=\([^\ ]*\).*/\1/g' -`
    if [ -n "$user" ]; then
        if [ -z "`grep $user /etc/passwd`" ]; then
            useradd -M -s /bin/nologin $user
        fi
        options=`$nginx -V 2>&1 | grep 'configure arguments:'`
        for opt in $options; do
            if [ `echo $opt | grep '.*-temp-path'` ]; then
                value=`echo $opt | cut -d "=" -f 2`
                if [ ! -d "$value" ]; then
                    # echo "creating" $value
                    mkdir -p $value && chown -R $user $value
                fi
            fi
        done
    fi
}

start() {

```

```
[ -x $nginx ] || exit 5
[ -f $NGINX_CONF_FILE ] || exit 6
make_dirs
echo -n "Starting $prog: "
daemon $nginx -c $NGINX_CONF_FILE
retval=$?
echo
[ $retval -eq 0 ] && touch $lockfile
return $retval
}

stop() {
echo -n "Stopping $prog: "
killproc $prog -QUIT
retval=$?
echo
[ $retval -eq 0 ] && rm -f $lockfile
return $retval
}

restart() {
configtest || return $?
stop
sleep 1
start
}

reload() {
configtest || return $?
echo -n "Reloading $prog: "
killproc $prog -HUP
retval=$?
echo
}
```

```
force_reload() {
    restart
}

configtest() {
    $nginx -t -c $NGINX_CONF_FILE
}

rh_status() {
    status $prog
}

rh_status_q() {
    rh_status >/dev/null 2>&1
}

case "$1" in
    start)
        rh_status_q && exit 0
        $1
        ;;
    stop)
        rh_status_q || exit 0
        $1
        ;;
    restart|configtest)
        $1
        ;;
    reload)
        rh_status_q || exit 7
        $1
        ;;
    force-reload)
        force_reload
        ;;
```

```
status)
    rh_status
    ;;
condrestart|try-restart)
    rh_status_q || exit 0
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"
    exit 2
esac
```

Сохранить в `/etc/init.d/nginx` и дать ему права запуска:

```
chmod a+x /etc/init.d/nginx
```

Использовать команды сервиса и настроить автоматический запуск после перезагрузки:

```
#запуск nginx
service nginx start
#автоматический запуск nginx после перезагрузки
chkconfig nginx on
```

5.8.4.2 Настройка sticky session

Настройка **sticky session** выглядит не сложнее, чем для метода **ip_hash**:

```
upstream backend {
    sticky;
    server backend1.example.com;
    server backend2.example.com;
}
```

По умолчанию будет создаваться **cookie** с именем **route** и временем жизни (expires) 1 час.

5.8.4.3 Аргументы модуля

Метод может принимать несколько аргументов:

```
sticky [name=route] [domain=.foo.bar] [path=/] [expires=1h] [hash=index|md5|sha1] [no_fallback]  
[secure] [httponly];
```

name – имя файлов cookie, используемых для отслеживания постоянного **upstream servers**, по умолчанию *route*.

domain – домен, в котором файл cookie будет действителен, по умолчанию: ничего.

path – путь, в котором файл cookie будет действителен, по умолчанию */*.

expires – срок действия файла cookie, по умолчанию: ничего. Это файл cookie сеанса. Ограничение: должно быть длительностью более одной секунды.

hash – хэш-механизм для кодирования вышестоящего сервера. Он не может быть использован с *hmac*. По умолчанию *md5*.

index – он не хэшируется, вместо него используется индекс в памяти, он быстрее, а накладные расходы короче. Сопоставление со списком вышестоящих серверов непоследовательно. Таким образом, при перезагрузке, если **upstreams servers** изменился, значения Индекса не гарантированно соответствуют тому же серверу, что и раньше. Используйте его с осторожностью и только в случае необходимости.

no_fallback – когда этот флаг установлен, **nginx** вернет 502 (ошибка плохого шлюза или прокси-сервера), если запрос приходит с файлом cookie и соответствующий сервер недоступен.

secure – обеспечивают безопасную cookie; передается только через HTTPS.

httponly – позволяет cookie-файлы не просачиваться через **js**.

5.8.5 Установка Nginx Vhost Traffic Status

5.8.5.1 Nginx-module-vts-master

- Скачать с <https://github.com/vozl/nginx-module-vts> ;
- Распаковать архив **unzip nginx-module-vts-master.zip**.

Для того, чтобы узнать, с какими параметрами был установлен **nginx** (в том числе, из репозитория), нужно набрать команду:

```
nginx -V
```

Чтобы установить этот модуль, нужно выполнить последовательность действий:

```
./configure --prefix=/usr/local/nginx/ --conf-path=/etc/nginx/nginx.conf --pid-path=/var/run/nginx.pid --  
user=nginx --with-http_ssl_module --with-http_gzip_static_module --with-http_stub_status_module --  
without-http_ssi_module --without-http_userid_module --without-http_autoindex_module --without-  
http_geo_module --without-http_referer_module --without-http_memcached_module --with-  
pcre=/tmp/pcre-8.38/ --with-openssl=/tmp/openssl-1.1.1g --with-zlib=/tmp/zlib-1.2.11/ --add-  
module=/tmp/nginx-sticky-module --add-dynamic-module=/tmp/nginx-module-vts-master/  
make  
make install clean
```

5.8.5.2 Настройка статистики

Внимание! Модуль статистики называется **ngx_http_vhost_traffic_status_module.so**, и размещается в **/usr/local/nginx/modules/**.

В основном конфигурационном файле **nginx.conf** необходимо добавить строку:

```
load_module /usr/local/nginx/modules//ngx_http_vhost_traffic_status_module.so;
```

В секцию **http** добавить строку:

```
vhost_traffic_status_zone;
```

В секцию **server** добавить:

```
location /status  
{  
    vhost_traffic_status on;  
    vhost_traffic_status_display;  
    vhost_traffic_status_display_format html;  
}
```

Обязательно проверяем синтаксис введенных изменений:

```
Nginx -t
```

```
[root@OEL6u4 sbin]# ./nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@OEL6u4 sbin]# █
```

Рисунок 17 – Проверка синтаксиса введенных изменений

Окончательный вид конфигурационных файлов:

Nginx Vhost Traffic Status

Server main

Host	Version	Uptime	Connections				Requests				Shared memory			
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	usedSize	usedNode
oel6u4	1.19.2	1m 25s	5	0	1	4	6	6	93	1	ngx_http_vhost_traffic_status	1024.0 KiB	6.9 KiB	2

Server zones

Zone	Requests			Responses					Traffic				Cache									
	Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarce	Total
localhost	92	1	3ms	0	90	1	1	0	92574.8 KiB	32.6 KiB	6.2 KiB	359 B	0	0	0	0	0	0	0	0	0	0
*	92	1	1ms	0	90	1	1	0	92574.8 KiB	32.6 KiB	6.2 KiB	359 B	0	0	0	0	0	0	0	0	0	0

Upstreams

gz_webcluster

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Traffic			
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s
172.26.24.8:8080	up	3ms	1	3	300	11	0	3ms	0	9	1	1	0	1150.7 KiB	4.1 KiB	0 B	0 B
172.26.24.8:8081	up	0ms	1	3	300	0	0	0ms	0	0	0	0	0	0 B	0 B	0 B	0 B

php_fpm

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Traffic						
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s		
unix:/var/run/php-fpm.sock	up	0ms	1	1	10	0	0	0ms	0	0	0	0	0	0	0	0	0 B	0 B	0 B	0 B

update interval: sec

[JSON](#) | [GITHUB](#)

Рисунок 18 – Окно статистики

5.8.6 Ускорение работы nginx

- **worker_priority**

worker_priority по умолчанию 0. Задаёт приоритет рабочих процессов от -20 до 20 (отрицательное число означает более высокий приоритет). Ему стоит отдать чуть больший приоритет (-2). Это нужно для того, чтобы при сильной нагрузке на скриптовую часть сайта или **DDoS** атаке, **nginx** продолжал обрабатывать запросы и отдавать статику.

- **worker_processes**

Если **Nginx** выполняет работу нагружающую процессор (например, **SSL** или **gziping**), то оптимально установить эту директиву в значение, равное количеству ядер процессора. Выигрыш при большем значении вы получите только в случае обработки очень большого количества статики:

```
# This number should be, at maximum, the number of CPU cores on your system.  
worker_processes 24;
```

Также, директива **worker_processes**, умноженная на **worker_connections** из секции **event**, даст максимально возможное количество клиентов.

- **worker_rlimit_nofile**

Данная директива указывает, сколько файловых дескрипторов будет использовать **Nginx**. На каждое соединение надо выделять по два дескриптора, даже для статических файлов (**картинки/JS/CSS**): один для соединения с клиентом, а второй — для открытия статического файла. Таким образом, значение **worker_rlimit_nofile** должно быть равным удвоенному значению **Max Clients**.

В системе это значение можно установить из командной строки **ulimit -n 200000** или используя **/etc/security/limits.conf**.

```
# Number of file descriptors used for Nginx.  
worker_rlimit_nofile 200000;
```

- **Логирование**

Логи доступа можно отключить полностью:

```
# Disable access log altogether.  
access_log off;
```

- **Обработка подключений**

Для обработки подключений **Nginx** поддерживает ряд методов. Наиболее эффективным для **Linux** является метод **epoll**:

```
# The effective method, used on Linux 2.6+, optimized to serve many clients with each thread.  
use epoll;
```

- **multi_accept**

Для того, чтобы **Nginx** пытался принять максимальное количество подключений, необходимо включить директиву **multi_accept**. Однако при слишком маленьком значении **worker_connections**, их лимит может быть очень быстро исчерпан:

```
# Accept as many connections as possible, after nginx gets notification about a new connection.  
multi_accept on;
```

• Кеширование

Можно обойтись без кеширования информации о:

- дескрипторах недавно открытых файлов: их размера и даты модификации;
- существовании директорий;
- ошибках при поиске файлов: отсутствие самого файла, отсутствие прав на чтение и т.д.

Лучше не копировать значения директив кеширования, а поиграть с ними, подобрав оптимальные для вашего окружения:

```
# Caches information about open FDs, frequently accessed files.  
open_file_cache max=200000 inactive=20s;  
open_file_cache_valid 30s;  
open_file_cache_min_uses 2;  
open_file_cache_errors on;
```

• keep-alive подключения

Для **keep-alive** подключений можно выключить буферизацию. Это будет полезно при частом запросе маленьких объёмов данных в режиме реального времени, без получения немедленного ответа, когда важна своевременная доставка данных:

```
# Don't buffer data-sends (disable Nagle algorithm).  
tcp_nodelay on;
```

Стоит обратить внимание на ещё две директивы для **keep-alive** подключений:

```
# Timeout for keep-alive connections. Server will close connections after this time.  
keepalive_timeout 30;  
# Number of requests a client can make over the keep-alive connection.  
keepalive_requests 1000;
```

• Дополнительная память

Чтобы высвободить дополнительную память, выделенную под сокеты, включите директиву **reset_timedout_connection**. Она разрешит серверу закрывать подключение тех клиентов, которые перестали отвечать;

```
# Allow the server to close the connection after a client stops responding.  
reset_timedout_connection on;
```

• Тайм-ауты

Можно существенно уменьшить тайм-ауты для директив **client_body_timeout** и **send_timeout**. Первая — ограничивает время на чтение тела запроса от клиента. Вторая — время ответа клиенту. Таким образом, если клиент не начнёт читать данные в указанный промежуток времени, то **Nginx** закроет подключение.

```
# Send the client a "request timed out" if the body is not loaded by this time.  
client_body_timeout 10;  
# If the client stops reading data, free up the stale client connection after this much time.  
send_timeout 2;
```

• Оптимизация работы с файлами

Директива **Sendfile** активирует копирование данных между файловыми дескрипторами средствами ядра, что намного эффективнее связки **read() + write()**, которая требует обмена данными с пользовательским пространством.

```
# Sendfile copies data between one FD and other from within the kernel.  
sendfile on;
```

После включения **sendfile**, можно заставить **Nginx** отправлять заголовки HTTP-ответов одним пакетом, а не отдельным частями.

```
# Causes nginx to attempt to send its HTTP response head in one packet, instead of using partial frames.  
tcp_nopush on;
```

aio включает использование асинхронного обращения к файлам, что избавит от очередей запросов.

```
Aio on;
```

open_file_cache по умолчанию выключена. Задаёт настройку для кэширования информации о файлах, с которыми работает **nginx**. По умолчанию, выключено.

- **Сжатие данных**

Сжатие данных служит для уменьшения размера пересылаемого трафика. Отключить сжатие для этих браузеров можно директивой **gzip_disable**, указав в качестве значения специальную маску «*msie6*», которая соответствует регулярному выражению «*MSIE [4-6].*», но работает быстрее

```
# Compression.
gzip on;
gzip_min_length 10240;
gzip_proxied expired no-cache no-store private auth;
gzip_types text/plain text/css text/xml text/javascript application/x-javascript application/xml;
gzip_disable "msie6";
```

- **Кеширование на стороне клиента**

NGINX умеет сам отдавать статические данные, не заставляя сервер **backend** этим заниматься. В настройке каждого сайта (секция **server**):

```
server {
    ...

    location ~*\.\.+\.(jpg|jpeg|gif|png|css|zip|tgz|gz|rar|bz2|doc|docx|xls|xlsx|exe|pdf|ppt|tar|wav|bmp|rtf|js|svg|woff2|ttf)$ {
        expires modified +1w;
    }
    ...
}
```

Примечание. Не стоит копировать приведенные настройки один в один. Можно применять их по одной, каждый раз запуская какую-нибудь утилиту для нагрузочного тестирования. Это весьма важно для понимания, какие настройки реально ускоряют ваш веб-сервер.

5.8.7 Подключение SSL

Добавить с секцию **server** строки:

```
server {
```

```
listen 81 ssl ;
server_name localhost;
access_log off;

ssl_certificate /HTTPServer/localhost/ssl/domain.pem;
ssl_certificate_key /HTTPServer/localhost/ssl/domain.pem;
ssl_session_cache shared:SSL:1m;
ssl_session_timeout 5m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers GOST2012-GOST8912-GOST8912:GOST2001-GOST89-GOST89:HIGH;
ssl_prefer_server_ciphers on;
```

ssl_certificate – путь до сертификата;

ssl_certificate_key – путь до ключа сертификата;

ssl_protocols – поддерживаемый протоколы;

ssl_ciphers – тип используемого шифра;

ssl_prefer_server_ciphers on – указывает, чтобы при использовании протоколов **SSLv3** и **TLS** серверные шифры были менее приоритетны, чем клиентские.

- **Самостоятельная генерация ключа и сертификата**

Самостоятельно это можно выполнить в два этапа (пример для ознакомления) :

```
Создание сертификата openssl req -newkey rsa:4096 -nodes -out key.csr -keyout key.pem
```

```
Подписание сертификата openssl x509 -req -days 365 -in key.csr -signkey key.pem -out cert.pem
```

Примечание. При использовании самоподписанного сертификата клиент, подключающийся к серверу, получит сообщение об ошибке. На производственный сервер следует устанавливать только сертификаты, подписанные признанным удостоверяющим центром.

5.8.8 Окончательный вид конфигурационных файлов

- **Nginx.conf**

```
# подключим библиотеку статистики
load_module /usr/local/nginx/modules/nginx_http_vhost_traffic_status_module.so;
```

```
user nginx;
worker_processes auto;
worker_priority -2;

error_log /var/log/nginx/error.log crit;
pid /var/run/nginx.pid;

events {
    worker_connections 4000;
    use epoll;
    multi_accept on;
}

http {
    include mime.types;
    default_type application/octet-stream;
    vhost_traffic_status_zone;

    upstream gz_webcluster {
        sticky;
        server 172.26.24.8:8080 max_fails=3 fail_timeout=300;
        server 172.26.24.8:8081 max_fails=3 fail_timeout=300;
    }

    upstream php_fpm {
        server unix:/var/run/php-fpm.sock;
    }

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log off;

    # OPTIMIZED
```

```
open_file_cache max=200000 inactive=20s;
open_file_cache_valid 30s;
open_file_cache_min_uses 2;
open_file_cache_errors on;

sendfile on;
server_tokens off;
tcp_nodelay on;
tcp_nopush on;
Aio on;

keepalive_timeout 30;
keepalive_requests 1000;

client_header_timeout 15;
client_body_timeout 10;
client_body_buffer_size 128k;

send_timeout 10;
reset_timedout_connection on;

gzip on;
gzip_min_length 10240;
gzip_proxied expired no-cache no-store private auth;
gzip_types
application/atom+xml
application/javascript
application/json
application/ld+json
application/manifest+json
application/rss+xml
application/vnd.geo+json
application/vnd.ms-fontobject
```

```
application/x-font-ttf
application/x-web-app-manifest+json
application/x-javascript
application/xhtml+xml
application/xml
application/pdf
application/msword
application/vnd.ms-excel
font/opentype
image/bmp
image/svg+xml
image/x-icon
text/cache-manifest
text/css
text/xml
text/javascript
text/plain
text/vcard
text/vnd.rim.location.xloc
text/vtt
text/x-component
text/x-cross-domain-policy;

gzip_disable "msie6";;

proxy_cache_path /var/cache/nginx levels=2:2 keys_zone=default:1000m;

include /etc/nginx/conf.d/*.conf;
}
```

- **Gzweb.conf**

```
server {
    listen 81 ssl;
    server_name localhost;
    access_log off;
```

```
ssl_certificate /HTTPServer/localhost/ssl/domain.pem;
ssl_certificate_key /HTTPServer/localhost/ssl/domain.pem;
ssl_session_cache shared:SSL:1m;
ssl_session_timeout 5m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers GOST2012-GOST8912-GOST8912:GOST2001-GOST89-GOST89:HIGH;
ssl_prefer_server_ciphers on;

location ~*\.\.+\.(jpg|jpeg|gif|png|css|zip|tgz|gz|rar|bz2|doc|docx|xls|xlsx|exe|pdf|ppt|tar|wav|bmp|rtf|js|
svg|woff2|ttf)$ {
    expires modified +1w;
}

client_header_buffer_size 128k;
large_client_header_buffers 8 16k;

location /status {
    vhost_traffic_status on;
    vhost_traffic_status_display;
    vhost_traffic_status_display_format html;
}

location / {
    proxy_pass http://gz\_webcluster;
    proxy_redirect off;
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    client_max_body_size 50m;
    client_body_buffer_size 128k;
    proxy_connect_timeout 20m;
    proxy_send_timeout 20m;
```



```
proxy_read_timeout 20m;
proxy_buffer_size 4k;
proxy_buffers 4 32k;
proxy_busy_buffers_size 64k;
proxy_temp_file_write_size 64k;
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root html;
}
}
```

• Предотвращение DDoS атак

DDoS атаки один из самых легкорезализуемых и потому часто применяемых видов атак. Защитить свой веб-сервер от атак такого вида можно поменяв следующие настройки.

- Ограничение запросов пользователей

Для ограничения числа запросов от пользователей можно использовать директивы **limit_req_zone** и **limit_req**. Следующий код нужно добавить в раздел **location**, который вложен в раздел **server**:

```
limit_req_zone $binary_remote_addr zone=one:10m rate=30r/m;

server {
    location /admin.html {
        limit_req zone=one;
    }
}
```

- Ограничение на число подключений

Директивы **limit_conn** и **limit_conn_zone** можно использовать для ограничения числа подключения к конкретным зонам. Например, код указанный ниже ограничивает число подключений десятью.

Код должен располагаться в разделе **location** раздела **server**:

```
server {  
  
    location /products/ {  
        limit_conn addr 10;  
    }  
}
```

- Разрыв медленных соединений

Директива как **client_body_timeout** задаёт таймаут при чтении тела запроса клиента. Таймаут устанавливается не на всю передачу тела запроса, а только между двумя последовательными операциями чтения. Если по истечении этого времени клиент ничего не передаст, обработка запроса прекращается с ошибкой. А **client_header_timeout** задаёт таймаут при чтении заголовка запроса клиента. Если по истечении этого времени клиент не передаст полностью заголовок, обработка запроса прекращается с ошибкой.

В раздел **server** нужно добавить следующее:

```
server {  
    client_body_timeout 5s;  
    client_header_timeout 5s;  
}
```

- Отключение вывода списка директорий

Чтобы отключить вывод директорий на странице можно использовать директиву **auto_index**. Она располагается в разделе **location**, а значение должно быть установлено в **off**.

```
location / {  
    auto_index off;  
}
```

5.9 Сервис помощи входа по сертификату (СПВС)

В целях усиления мер по защите данных от несанкционированного доступа (использование средств криптографической защиты информации) и повышения удобства работы с системой (отсутствие необходимости ввода логина и пароля пользователя при

входе в систему) требуется поддержка механизма аутентификации по сертификату при использовании СБ в качестве средства защиты информации и управления доступом субъектов доступа к объектам доступа.

Сервис помощи входа по сертификату – вспомогательный сервис для взаимодействия с сервисами СБ, СЭП и прикладной информационной системой, выполняет функции по работе с хранилищем сертификатов пользователя, генерации и проверке токенов, осуществляет вызовы методов формирования и проверки ЭП, а также аутентификации в СБ.

5.9.1 Методы сервиса

Проверка сертификата и генерация токена

POST /LOGIN

Consumes

- application/json

Produces

- application/json

Таблица 47 – Параметры CertLoginRequest

Type	Name	Schema	Description
Query	certificate	String	Сертификат в формате DER (*.cer), закодированный в Base64

Таблица 48 – Response

HTTP Code	Schema	Description
200	CertLoginResponse	токен сертификата
200	BasicRestResponse	успешно
400	BasicRestResponse	информация об ошибке

Проверка подписи токена

POST /VERIFYTOKEN

Consumes

- application/json

Produces

- application/json

Таблица 49 – Параметры VerifyTokenRequest

Type	Name	Schema	Description
Query	token	String	Токен сертификата, полученный от СПВС
Query	signature	String	Данные подписи в формате Base64
Query	auth	String	Временный код аутентификации, полученный от СБ
Query	redirectUri	String	Адрес (URL) обратного редиректа, полученный от СБ
Query	signatureFormat	enum	Формат подписи. Поддерживается только cades_bes

Таблица 50 – Response

HTTP Code	Schema	Description
302	FOUND	редирект на url, отправленный в запросе в параметре redirectUri
400	BasicRestResponse	в случае ошибки

Авторизация пользователя по JWT-токену

POST /login

Consumes

- */*

Produces

- application/json

Установка и настройка системы
Сервис помощи входа по сертификату (СПВС)

Таблица 51 – Параметры

Type	Name	Schema	Description
Header	Authorization	String	Строка авторизации, формат: "Bearer ey9eyJzd..."

Таблица 52 – Response

HTTP Code	Schema	Description
200	AzkCommonResponse	ответ АЦК об успешной авторизации или ошибке

5.9.2 Описание JSON-типов

Таблица 53 – BasicRestResponse

Name	Type	Description
success	boolean	успешно/ошибка
error (optional)	String	информация об ошибке, если success - false

Таблица 54 – CertLoginRequest

Name	Type	Description
certificate	String	Сертификат в формате DER (*.cer), закодированный в Base64

Таблица 55 – CertLoginResponse

Name	Type	Description
token	String	токен сертификата для подписания на клиенте

Таблица 56 – VerifyTokenRequest

Name	Type	Description
token	String	Токен сертификата, полученный от СПВС
signature	String	Данные подписи в формате Base64
auth	String	Временный код аутентификации, полученный от СБ
redirectUri	String	Адрес (URL) обратного редиректа, полученный от СБ
signatureFormat	enum	Формат подписи. Поддерживается только cades_bes

Таблица 57 – AzkCommonResponse

Name	Type	Description
success	boolean	успешно/ошибка
jsessionid (optional)	String	ID сессии, если success - true

Установка и настройка системы
Сервис помощи входа по сертификату (СПВС)

Name	Type	Description
errors (optional)	AzkErrors	информация об ошибке, если success - false

Таблица 58 – AzkErrors

Name	Type	Description
reason	String	причина ошибки
info (optional)	String	дополнительная информация об ошибке



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:\2.56.0\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 содержимым 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.00004-56 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](#)²⁷⁴.*

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](#)³⁶⁴.

- **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](#)³⁶⁵);
- **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](#)³⁶⁴, 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**³⁶⁵ 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:\2.56.0\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:

Обновление системы

Обновление отчетов

C:\2.56.0\XML\module\bft.reports\xml.cmd

C:\2.56.0\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.00004-56 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):

Порядок технического обслуживания
 Регламентно-профилактические мероприятия

Таблица 59 – 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>lconfserver.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 XXMX, XXMS, 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 azk.war в каталог webapps с последующим запуском tomcat.	После проведения процедуры обновления сборки системы.
8. Мониторинг виртуальной Java машины (далее JVM) для выявления случаев утечки памяти (например, с помощью утилиты jvisualvm).	Периодически, при возникновении проблем с производительностью. Постоянный мониторинг опасен тем, что дополнительно нагружает систему, замедляя работу.
9. Мониторинг логов сервера приложений на наличие ошибок типа: <i>Java heap space</i> , <i>Out of memory</i> , sql-ошибок и пр. Мониторинг логов веб-сервера на наличие ошибок.	Постоянно.
10. Мониторинг и контроль выполнения заданий планировщиков.	Постоянно.
11. Мониторинг и контроль за «системными событиями».	Постоянно.

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

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. 1-9³⁷⁴](#)).
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.pooledexecutor.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 60 – 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/vhttpd2.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/vhttpd2.conf/apache.conf , для tomcat – server.xml) на физическом отдельном диске.	Периодически и перед выполнением процедуры обновления/реконфигурации сборки/серверов/гипервизоров/СХД.
3.	Мониторинг загруженности компонентов JVM.	Постоянно.
4.	Мониторинг балансировщиков (Nginx: 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.

№	Мероприятия	Периодичность выполнения
8.	Мониторинг и контроль за доступностью ресурсов (дисковые хранилища, CPU, RAM).	Постоянный.
9.	Профилактическое обслуживание ИТ-системы (анализ журналов, диагностика, тонкая настройка).	Периодически, 1 раз в месяц.
10.	Мониторинг системных журналов, содержащих информацию об успешности авторизации для выявления угроз безопасности, хакерских атак методом перебора и др..	Постоянно.

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. **Мбайт/s** – количество переданных мегабайт в секунду.

Параметры **IO/s** и **Мбайт/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.

*Примечание. В случае, если переадресация произошла некорректно, нужно изменить **localhost** на **localopenidazk.bftcom.com**, данная замена осуществляется ручным способом, либо при помощи утилиты **AzkHostEditor**. Замена позволяет при аутентификации в клиентском приложении через Аванпост запускать локальный веб-сервис с адресом **localopenidazk.bftcom.com**.*

2. В случае успешного ввода FAM возвращает в 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.



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**.

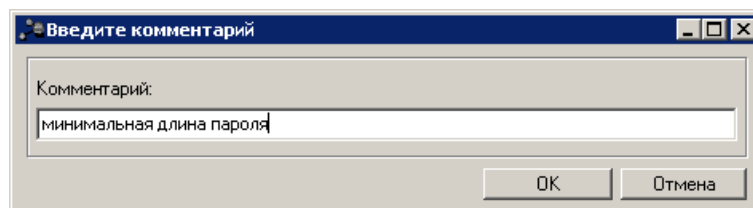


Рисунок 19 – Окно ввода комментария

4. Нажать сочетание клавиш **Ctrl+F6**. Откроется окно просмотра истории изменений.

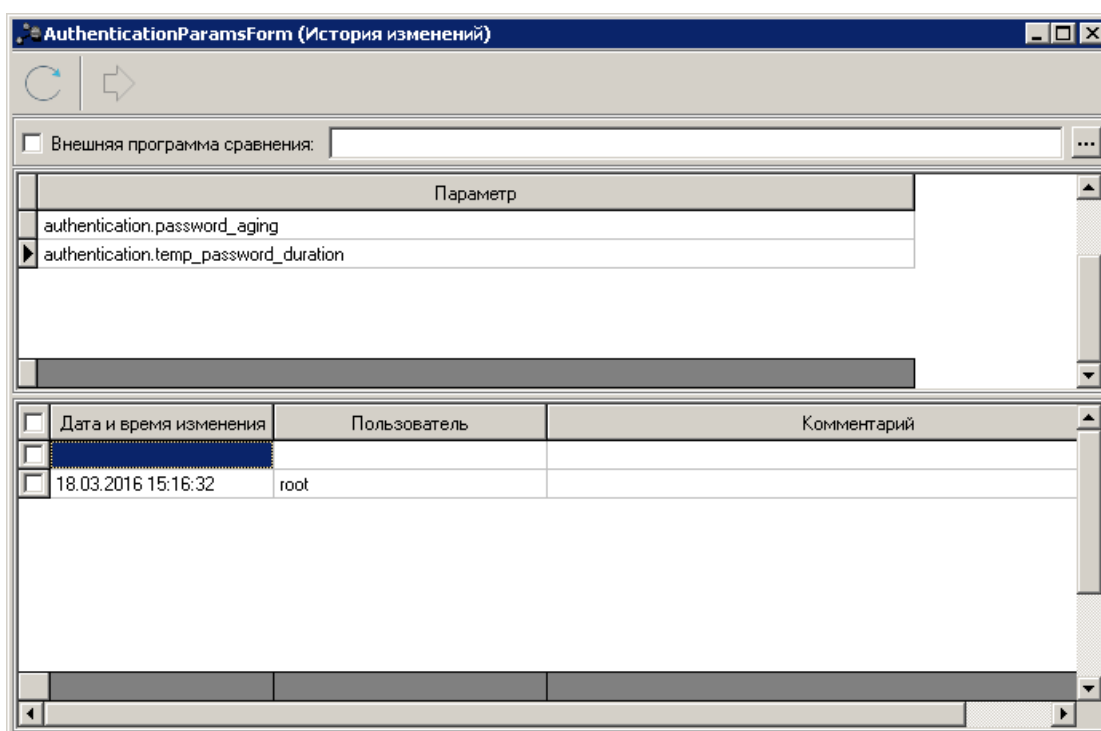



Рисунок 20 – Окно просмотра истории изменений системных параметров

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:

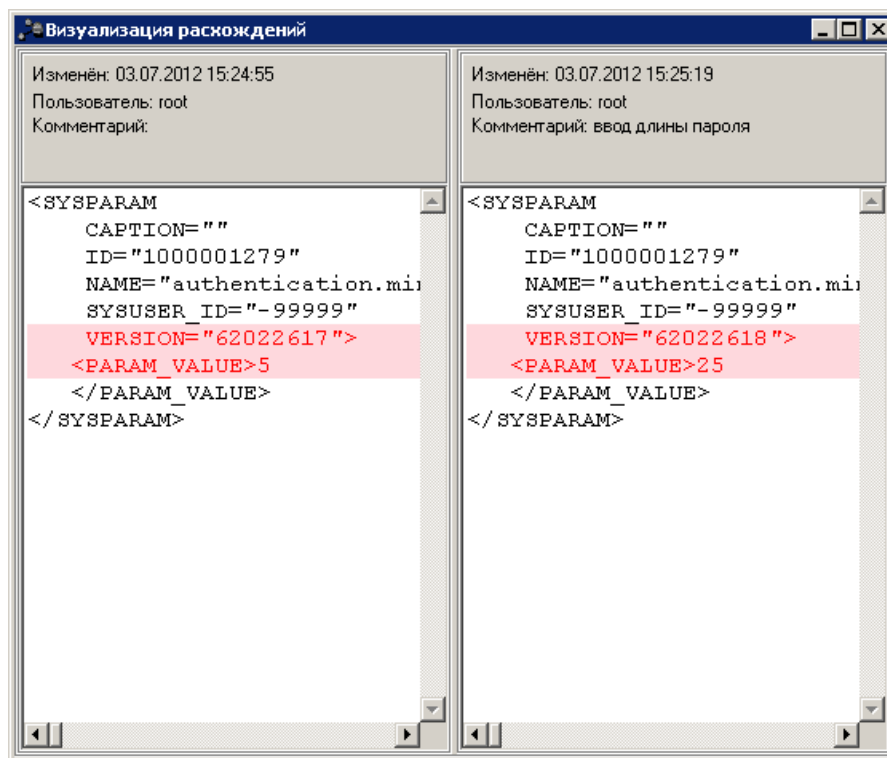



Рисунок 21 – Окно визуализации расхождений

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

