

СИСТЕМА ФОРМИРОВАНИЯ ДОКУМЕНТОВ

Руководство по установке

Версия 1.4.0

nexign

Настоящая документация может быть использована только для поддержки работоспособности продуктов, установленных на основании договора с АО «Нэксайн». Документация может быть передана на основании договора, по которому производится (производилась или будет производиться) установка продуктов, или явно выраженного согласия АО «Нэксайн» на использование данной документации. Если данный экземпляр документации попал к вам каким-либо иным образом, пожалуйста, сообщите об этом в АО «Нэксайн» по адресу, приведенному ниже.

Все примеры, приведенные в документации (в том числе примеры отчетов и экранных форм), составлены на основании тестовой базы АО «Нэксайн». Любое совпадение имен, фамилий, названий компаний, банковских реквизитов и другой информации с реальными данными является случайным.

Все встречающиеся в тексте торговые знаки и зарегистрированные торговые знаки являются собственностью их владельцев и использованы исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили в документации ошибки или опечатки или предполагаете их наличие, пожалуйста, сообщите об этом в АО «Нэксайн».

Все имущественные авторские права сохраняются за АО «Нэксайн» в соответствии с действующим законодательством.

© АО «Нэксайн», 1992–2022

АО «Нэксайн»

Россия, 199155, Санкт-Петербург, ул. Уральская, д.4 лит.Б, помещение 22Н

Тел.: + 7 (812) 326-12-99; факс: + 7 (812) 326-12-98.

office@nexign.com; www.nexign.com

Содержание

1. Предварительные условия	4
1.1. Подготовка инфраструктуры	4
1.1.1. JAVA JDK	4
1.1.2. Установка liquibase	4
1.1.3. Установка Tomcat	4
1.1.4. Настройка Tomcat	4
1.1.5. Установка и настройка RabbitMQ для REPORT_ENGINE	5
1.2. Настройка доступа к SFTP-серверу	6
1.3. Конфигурирование внешних запросов	7
2. Установка продукта	8
2.1. Установка DGS	8
2.1.1. Установка схемы данных DGS	8
2.1.2. Установка модулей DGS	8
2.2. Установка REPORT_ENGINE	11
2.2.1. Установка сервера конфигурации (RPT_CONFIG)	11
2.2.2. Установка сервера получения метаданных (RPT_METADATA)	11
2.2.3. Установка модуля «Интерфейс взаимодействия с сервером отчетов» (RPT_INTERFACE)	12
2.2.4. Установка общих библиотек построения отчетов (RPT_COMMON)	12
2.2.5. Установка сервера построения данных (RPT_DATA_BUILD)	12
2.2.6. Установка сервера построения документов на основе FRX-отчетов (RPT_FRNET_BUILD)	14
2.2.7. Установка редактора FRX-отчетов (RPT_FRNET_EDIT)	15
2.2.8. Установка отчетов	15
2.2.9. Параметры установки и настройки REPORT_ENGINE	17
3. Запуск и остановка продукта	23
4. Обновление	24
5. Проверка работоспособности	25

1. Предварительные условия

Продукт «Система формирования документов» (DGS) устанавливается на серверы под управлением РЕД ОС.

Требования к оборудованию и программному обеспечению, а также функциональная архитектура продукта описаны в документе «Система формирования документов. Руководство по эксплуатации» [DGS-DOC_G3].

1.1. Подготовка инфраструктуры

В разделе приведено описание установки и настройки инфраструктуры, необходимой для работы продукта.

1.1.1. JAVA JDK

Для корректной работы продукта необходимо наличие виртуальной машины JAVA. Рекомендуется использовать Java JDK версии 1.8.

После установки Java JDK настройте переменные окружения:

1. Создайте переменную среды `JAVA_HOME` и установите `JAVA_HOME` в корневой каталог с Java (без `\` в конце и без `bin`).
2. В переменную `Path` добавьте путь к `bin` Java.

1.1.2. Установка liquibase

Для работы liquibase дополнительно скачайте liquibase-библиотеку `slf4j-api-1.7.5.jar` (библиотека журналирования) и поместите в папку `/lib` каталога установки.

1.1.3. Установка Tomcat

Для работы генератора отчетов `REPORT_ENGINE` установите не менее двух экземпляров сервера Tomcat – для сервера приложений и сервера конфигурации.

Tomcat можно настроить для запуска вручную или для работы в виде сервиса (службы).

В случае успешной установки Tomcat должен отвечать на установленном порту, например: `http://localhost:8080`.

1.1.4. Настройка Tomcat

По умолчанию у сервера Tomcat семь конфигурационных файлов (в формате XML), которые находятся в каталоге `%tomcat%\conf`.

Список необходимых действий по настройке серверов Tomcat приведен в [Табл. 1](#).

Табл. 1. Настройка конфигурационных файлов Tomcat

Файл конфигурации	Настройка файла конфигурации
<code>catalina.policy</code>	

Файл конфигурации	Настройка файла конфигурации
catalina.properties	<p>В переменной <code>common.loader</code> укажите путь к проприетарным библиотекам доступа к базе данных:</p> <pre>common.loader= \${catalina.base}/lib,\${catalina.base}/lib/*.jar, \${catalina.home}/lib,\${catalina.home}/lib/*.jar,</pre>
server.xml	<ol style="list-style-type: none"> Установите значения параметров (объем пула JDBC-соединений и пр.) в зависимости от ожидаемой нагрузки. Настройте порты для сервера приложений и сервера конфигурации, на которых будут отвечать Tomcat, например, порт 9191 для сервера конфигурации и 9090 для сервера приложений: <pre><Server port="8005 8006" shutdown="SHUTDOWN"> <Connector port="9191 9090" protocol="org.apache.coyote.http11.Http11NioProt ocol" redirectPort="8441 8440" URIEncoding="UTF- 8"> <Connector port="8009 8010" protocol="AJP/1.3" redirectPort="8441 8440" /></pre> <ol style="list-style-type: none"> Если предполагаются входные параметры для отчетов, содержащие кириллицу, укажите в описании коннектора кодировку 1251: <pre><Connector port="9292" protocol="HTTP/1.1" connectionTimeout="20000" URIEncoding="windows-1251"/></pre>
tomcat-users.xml	Настройте пользователей и их права в соответствии с политикой безопасности
logging.properties	Настройте параметры журналирования Tomcat
web.xml	
context.xml	

1.1.5. Установка и настройка RabbitMQ для REPORT_ENGINE

Взаимодействие модулей компонента `REPORT_ENGINE` между собой обеспечивает сервер сообщений RabbitMQ.

Выполните следующие действия:

1. Установите RabbitMQ Server – см. официальную документацию на <http://www.rabbitmq.com/>.
2. Перейдите в папку \RabbitMQ Server\<...>\sbin.
3. Запустите команды:

```
rabbitmqctl add_user ps-tdp tdp <enter>
rabbitmqctl add_vhost tdp <enter>
rabbitmqctl set_permissions -p tdp ps-tdp ".*" ".*" ".*" <enter>
```

Настройка RabbitMQ

На серверах с RabbitMQ создайте необходимые для работы модулей компонента REPORT_ENGINE сущности:

- для обработки заявок на построение отчета:
 - точку обмена (Exchange) – ps.re.report_request;
 - очередь (Queue) – ps.re.report_request;
 - ключ маршрутизации (Routing_key) – ps.re.report_request_add;
- для запросов на построение отчета по технологии FastReport Mono:
 - точку обмена – ps.re.template_request;
 - очередь – ps.re.template_request.<technology>, например: ps.re.template_request.frx.mono;
 - ключ маршрутизации – ps.re.template_request_add.<technology>, например: ps.re.template_request_add.frx.mono;
- для обработки статистических сообщений:
 - точку обмена – ps.re.report_statistic;
 - очередь – ps.re.report_statistic.interface;
 - ключ маршрутизации – ps.re.report_statistic_add.interface.#;
- для обработки сообщений о готовности отчета:
 - точку обмена – ps.re.report_result;
 - очередь – ps.re.report_result.interface;
 - ключ маршрутизации – ps.re.report_result_add.interface.#;
- для обработки ответов от внешних сервисов о готовности данных:
 - точку обмена – ps.re.external_response;
 - очередь – ps.re.external_response.<host>;
 - ключ маршрутизации – ps.re.external_response_add.<host>.

1.2. Настройка доступа к SFTP-серверу

В компоненте REPORT_ENGINE реализована поддержка аутентификации для доступа к SFTP-серверу.

Для аутентификации используются параметры User, sshKey и sshKeyPassphrase.

Приватный ключ sshKey и парольная фраза sshKeyPassphrase, с помощью которой шифруется приватный ключ, хранятся на машинах с функционирующим сервисом, которые взаимодействуют с SFTP-сервером.

Парольная фраза sshKeyPassphrase шифруется с помощью ключа secretKey, который

хранится в настройках продукта как параметр `key`.

Перед взаимодействием с SFTP `REPORT_ENGINE` расшифровывает `sshKeyPassphrase` на локальной машине с помощью `secretKey` (заданного в параметре `key`), и выполняет аутентификацию с помощью приватного ключа `sshKey` и расшифрованного `sshKeyPassphrase` к этому ключу.

Описание параметров `REPORT_ENGINE` см. в разделе [«Параметры установки и настройки REPORT_ENGINE»](#).

1.3. Конфигурирование внешних запросов

Настройте запросы на получение данных от сторонних продуктов для формирования отчетов – пропишите в конфигурационном файле `REPORT_ENGINE` по пути

`ps.config.apps.tdp.report-engine.reportResources.externalRequests` параметры этих продуктов:

- `services` – описание сервисов:
 - `reports` – имя тега является наименованием сервиса, описывающего HTTP-запросы (может быть несколько):
 - `storageCluster` – ссылка на имя кластера ресурсов (имена регистрозависимы);
 - `requestSync` – описание функции синхронного запроса:
 - `method` – метод; например: `POST`;
 - `path` – путь, например: `/openapi/v1/reports/documents`; может содержать макроподстановки (переменные в фигурных скобках), например: `dictionaries/common/branches/{branchId}` – в этом случае при вызове вместо `{branchId}` будет подставлен идентификатор ветки, полученный с использованием класса `ExternalServiceCollector`;
 - `cacheable` – признак кеширования запроса; возможные значения: `true/false`;
 - `requestAsync` – описание функции асинхронного запроса:
 - `method` – метод; например: `POST`;
 - `path` – путь, например: `/openapi/v1/reports/orders`; может содержать макроподстановки аналогично параметру `requestSync.path`;
 - `orderId` – название поля, в котором находится идентификатор заявки, например: `reportOrderId`;
 - `additionalParams` – список дополнительных параметров, которые необходимо дописывать при вызове, кроме `corralationId`; например: `resource` – ссылка на внешний ресурс;
 - `cacheable` – признак кеширования запроса; возможные значения: `true/false`;
 - `requestCheck` – описание функции проверки статуса:
 - `method` – метод; например: `GET`;
 - `path` – путь; например: `/openapi/v1/reports/orders/{orderId}`; может содержать макроподстановки аналогично параметру `requestSync.path`;
 - `statusId` – название поля, в котором находится идентификатор статуса заявки; например: `reportOrderStatusCode`;
 - `statusOk` – значение идентификатора, которое соответствует успешному формированию заявки;
 - `statusFail` – значение идентификатора, которое соответствует неуспешному формированию заявки;
 - `cacheable` – признак кеширования запроса; возможные значения: `true/false`.

2. Установка продукта

Установка продукта выполняется в два этапа:

1. Установка модулей обработчика заказов на формирование отчетов – см. раздел [«Установка DGS»](#).
2. Установка модулей генератора отчетов REPORT_ENGINE – см. раздел [«Установка REPORT_ENGINE»](#).

2.1. Установка DGS

В разделе приведено описание установки схемы данных DGS и модулей обработчика заказов на формирование отчетов (модулей DGS).

2.1.1. Установка схемы данных DGS

Для установки схемы данных DGS требуется установленный Liquibase 3.5.3 или выше.

Порядок установки:

1. Перейдите в каталог to_user_dgs/scr.
2. Выдайте права пользователю базы данных, от имени которого будет устанавливаться схема:
 - заполните параметры в файле grants.yaml:
 - dgs_user – имя существующего пользователя базы данных;
 - data_tablespace – имя табличного пространства для данных;
 - index_tablespace – имя табличного пространства для индексов;
 - запустите команду:

```
./liquibase --driver=org.postgresql.Driver  
--changeLogFile=db.changelog.yml --url=<url> --username=<username>  
--password=<password> update
```

где:

- url – строка подключения к базе данных;
 - username, password – имя и пароль пользователя, имеющего привилегию SYSDBA.
3. Заполните параметры установки в файле install_params.yaml.
 4. Выполните команду:

```
./liquibase --defaultsFile=install_params.properties update
```

2.1.2. Установка модулей DGS

Для установки модулей DGS используются Helm-чарты, которые описывают манифесты ресурсов для Kubernetes.

Зависимости Helm-чартов указаны в файле Chart.yaml.

Параметры Helm-чартов сохранены в файле values.yaml.

Для установки:

1. Настройте конфигурационные параметры модулей `DGS_API`, `DGS_PROCESSOR`, `DGS_COMBINER` и `DGS_DBADMIN` в файлах `helm/dgs/<module_name>/values.yaml`.
Описание параметров приведено в [Табл. 2](#) - [Табл. 7](#).
2. Выполните установку модулей `DGS` с помощью Helm-чартов.

Табл. 2. Общие параметры

Параметр	Признак/ значение по умолчанию	Описание
<code>replica.Count</code>	1	Количество реплик пода
<code>nodeSelector</code>	{ }	Ограничение выбора узла
<code>affinity</code>	{ }	Ограничение использования модулей на указанных узлах
<code>tolerations</code>	[]	Описание базового объекта, состоящего из трех обязательных параметров: <code>key</code> , <code>value</code> , <code>effect</code>
<code>serviceAccount.name</code>	default	Профиль для аутентификации и авторизации выполнения API-запросов
<code>extraLabels</code>	{ }	Дополнительные параметры для манифеста k8s
<code>priorityClassName</code>		Приоритетность пода

Табл. 3. Параметры образа

Параметр	Признак/ значение по умолчанию	Описание
<code>image.repository</code>	docker-dev- local.nexign.com	Путь размещения образа
<code>image.pullPolicy</code>	Always	Политика загрузки образа
<code>imagePullSecrets</code>	default	Политика доступа к данным секретного реестра

Табл. 4. Параметры сервиса

Параметр	Признак/ значение по умолчанию	Описание
<code>service.type</code>	NodePort	Тип сервиса Kubernetes
<code>service.ports.http.port</code>	8081	Порт для обращения к сервису внутри кластера
<code>service.ports.http.protocol</code>	TCP	Протокол для передачи данных
<code>service.ports.http.targetPort</code>	8080	Внутренний порт приложения

Параметр	Признак/ значение по умолчанию	Описание
service.ports.http.nodePort	30009	Наружный порт для подключения к сервису извне кластера
service.clusterIP		IP-адрес сервиса в кластере

Табл. 5. Параметры используемых ресурсов

Параметр	Признак/ значение по умолчанию	Описание
resources.limits.cpu	800m	Максимальная нагрузка на процессор
resources.limits.memory	2000Mi	Максимальное ограничение используемой приложением памяти
resources.requests.cpu	400m	Резервируемая нагрузка на процессор
resources.requests.memory	1000Mi	Резервируемое ограничение используемой приложением памяти

Табл. 6. Параметры подключения к ZooKeeper

Параметр	Признак/ значение по умолчанию	Описание
zookeeper.host	zookeeper	Хост подключения к ZooKeeper
zookeeper.port	2181	Порт подключения к ZooKeeper
zookeeper.root	ps/config/apps	Корневой путь по умолчанию

Табл. 7. Параметры модуля

Параметр	Признак/ значение по умолчанию	Описание
<module_name>.image.path	ps/tdp/dgs_1/ <module_name>	Путь до Helm-чарта в хранилище
<module_name>.replicaCount	1	Количество реплик пода
<module_name>.minAvailable	1	Минимальное количество реплик пода для функционирования приложения
application.server.port	8080	Внутренний порт приложения
application.server.servlet.context_path		Путь к контексту приложения
<module_name>.readinessProbe.httpGet.path	/actuator/health	URI для проверки готовности приложения

Параметр	Признак/ значение по умолчанию	Описание
readinessProbe.initialDelaySeconds	70	Задержка перед первой проверкой контейнера на жизнеспособность, в секундах
readinessProbe.timeoutSeconds	1	Тайм-аут ожидания ответа от сервиса, в секундах
readinessProbe.periodSeconds	15	Период между проверками контейнера на жизнеспособность, в секундах
readinessProbe.failureThreshold	5	Количество попыток опроса модуля (ping)
<module_name>.livenessProbe		Набор параметров для проверки контейнера на жизнеспособность

2.2. Установка REPORT_ENGINE

1. Последовательно установите все функциональные модули REPORT_ENGINE.
2. Установите FRX-отчеты (см. подраздел «[Установка отчетов](#)»).

Подробное описание установки и настройки модулей REPORT_ENGINE приведено в соответствующих подразделах.

2.2.1. Установка сервера конфигурации (RPT_CONFIG)

Модуль RPT_CONFIG содержит файлы конфигурации, которые используют остальные модули REPORT_ENGINE.

RPT_CONFIG установите на сервер с развернутым сервером приложений Tomcat:

1. Скопируйте war-файл дистрибутива из каталога rpt_config/app в каталог %tomcat%\webapps.
2. Создайте дерево настроек на сервере конфигурации ZooKeeper в ветке ps/config/apps/tdp/report_engine/<host_name>; пример дерева см. в файле samples/zookeeper.yml.
3. Задайте значения конфигурационных параметров RPT_CONFIG в файле conf/TdpApplication/TdpRptConfig.properties.

Описание параметров см. в разделе «Параметры установки и настройки REPORT_ENGINE», секция [rpt_config](#).

2.2.2. Установка сервера получения метаданных (RPT_METADATA)

Модуль RPT_METADATA установите на сервер с Tomcat:

1. Скопируйте war-файл из каталога rpt_metadata/app дистрибутива в каталог webapps сервера Tomcat.
2. Настройте параметры конфигурации в файле conf/TdpApplication/RptMetadata.xml.

Описание параметров см. в разделе «Параметры установки и настройки REPORT_ENGINE», секция [metadata](#).

3. Создайте корневой каталог для хранения отчетов с именем, заданным в параметре

reportPath файла конфигурации RptMetadata.xml.

2.2.3. Установка модуля «Интерфейс взаимодействия с сервером отчетов» (RPT_INTERFACE)

Модуль RPT_INTERFACE установите на сервер с развернутым Tomcat:

1. Скопируйте war-файл из каталога rpt_interface/app дистрибутива в каталог \webapps установленного сервера Tomcat.

Описание параметров см. в разделе «Параметры установки и настройки REPORT_ENGINE», секция [rpt_interface](#).

2. Включите дополнительное журналирование – задайте настройки журналирования, специфичные для RPT_INTERFACE, в файле RptInterface_logback.xml (размещение этого файла определяет администратор платформы, на которой запускается модуль, по умолчанию – каталог conf сервера Tomcat):

- в файле setenv.sh/setenv.bat добавьте в переменную CLASSPATH путь к каталогу с файлом RptInterface_logback.xml, например:
 - для setenv.sh:

```
CLASSPATH="$CLASSPATH;%CATALINA_HOME/conf/RptInterface"
```

- для setenv.bat:

```
set "CLASSPATH=%CLASSPATH%;%CATALINA_HOME%/conf/RptInterface"
```

- в файле \${catalina_home}/conf/logback-real.xml после тега <configuration> добавьте строку:

```
<include file="${catalina.base}/conf/RptInterface_logback.xml"/>
```

3. Скопируйте файл RptInterface_logback.xml из каталога rpt_interface/conf дистрибутива в указанный на первом шаге каталог.

2.2.4. Установка общих библиотек построения отчетов (RPT_COMMON)

Скопируйте все библиотеки из каталога дистрибутива rpt_common/lib в каталог \lib сервера Tomcat.

2.2.5. Установка сервера построения данных (RPT_DATA_BUILD)

Модуль RPT_DATA_BUILD установите на сервер с развернутым Tomcat:

1. Скопируйте war-файл из каталога rpt_data_build/app дистрибутива в каталог \webapps сервера Tomcat.

Описание параметров см. в разделе «Параметры установки и настройки REPORT_ENGINE», секция [rpt_data_build](#).

2. Если отчеты используют подключения к базе данных или внешние сервисы для сбора

данных, то на сервере конфигурации ZooKeeper в ветке `/ps/config/apps/tdp/report-engine/reportResources` задайте значения параметров:

- `dataSources` – описание соединений с базой данных:
 - `defaultDataSource` – наименование соединения с базой данных по умолчанию;
 - `bis` – JDBC-соединение с базой данных; имя тега является наименованием соединения (их может быть несколько); параметры соединения:
 - `type` – тип соединения;
 - `url` – URL для подключения к базе данных;
 - `user` – имя пользователя базы данных;
 - `password` – пароль пользователя базы данных; зашифрованное значение, если задано значение параметра `secretKey`;
 - `driverClass` – имя класса драйвера;
 - `min` – минимальный размер пула соединений (количество подключений, создаваемых при старте приложения); во время малой нагрузки размер пула будет находиться около этого значения;
 - `max` – максимальный размер пула соединений; если генераторы отчетов активно используют подключение к базе данных, то это значение должно быть приближено к значению конфигурационного параметра `rpt_data_build.builderCount`;
 - `maxAge` – время жизни соединения в миллисекундах; при превышении данного значения соединение будет пересоздано при следующем обращении; значение по умолчанию: 600000 (10 минут);
- `externalRequests` – конфигурация ресурсов для работы с HTTP:
 - `url` – адрес сервера единой точки входа (`SEP_SRV`);
 - `applicationCode` – пользователь доверенного канала.

3. Если необходимо, включите дополнительное журналирование.

Настройка журналирования

Модуль `RPT_DATA_BUILD` поддерживает журналирование на уровне сервера приложений Tomcat. Описание общих настроек журналирования см. в документации на продукт «ТОМКАТ» (ТОМКАТ).

Также `RPT_DATA_BUILD` обеспечивает дополнительное журналирование информации в основной `log`-файл.

Для включения дополнительного журналирования специфичные для `RPT_DATA_BUILD` настройки задайте в файле `RptDataBuild_logback.xml` (размещение файла определяет администратор платформы, на которой запускается модуль, по умолчанию – каталог `conf` сервера Tomcat):

1. В файле `setenv.sh/setenv.bat` добавьте в переменную `CLASSPATH` путь к каталогу с файлом `RptDataBuild_logback.xml`, например:

- для `setenv.sh`:

```
CLASSPATH="$CLASSPATH;$CATALINA_HOME/conf/RptDataBuild"
```

- для `setenv.bat`:

```
set "CLASSPATH=%CLASSPATH%;%CATALINA_HOME%/conf/RptDataBuild"
```

- В файле `${catalina_home}/conf/logback-real.xml` после тега `<configuration>` добавьте строку:

```
<include file="${catalina.base}/conf/RptDataBuild_logback.xml"/>
```

- Скопируйте файл `RptDataBuild_logback.xml` из каталога `rpt_data_build/conf` дистрибутива в указанный на первом шаге каталог.

2.2.6. Установка сервера построения документов на основе FRX-отчетов (RPT_FRNET_BUILD)

- Подготовьте сервер для установки `RPT_FRNET_BUILD`:

- установите утилиты:
 - `mono`;
 - `Xfce`;
- обеспечьте наличие шрифтов:
 - `arial`;
 - `arialbd`;
- установите переменную окружения `FONTDIR`, для чего выполните команду:

```
export FONTDIR=/usr/share/fonts  
echo -e '\nexport FONTDIR=/usr/share/fonts' >> ~/.bashrc
```

- Распакуйте архив с дистрибутивом модуля `RPT_FRNET_BUILD`.
- Запустите команду:

```
DISPLAY=:0 mono-service RptFrNetBuildService.exe
```

где 0 – номер дисплея сервера.

- Создайте файл с настройками `tdpcfg.properties` в каталоге установки `RPT_FRNET_BUILD` (см. [«Формат файла tdpcfg.properties»](#)).
- Настройте конфигурационные параметры `RPT_FRNET_BUILD` на сервере конфигурации ZooKeeper.

Описание параметров см. в разделе «Параметры установки и настройки `REPORT_ENGINE`», секция [rpt_frnet_build](#).

Формат файла `tdpcfg.properties`

```
remoteConfig.baseUrl=http\://{rpt_config_server}\:{rpt_config_  
server_port}/rptconfig  
secretKey=VGhpcyBpcyBkZW9uIGtleQ==  
zookeeper.servers={zk_server1:zk_server_port},{zk_server2:zk_  
server2_port}
```

где:

- {rpt_config_server} – сервер конфигурации RPT_CONFIG;
- {rpt_config_server_port} – порт RPT_CONFIG;
- zookeeper.servers – список через запятую серверов ZooKeeper в формате server:port.

2.2.7. Установка редактора FRX-отчетов (RPT_FRNET_EDIT)

Для установки редактора FRX-отчетов (RPT_FRNET_EDIT) запустите из дистрибутива файл rpt_frnet_edit/app/setup.exe.

2.2.8. Установка отчетов

1. Скопируйте файлы FRX-отчетов (см. «[Файлы FRX-отчетов](#)») в каталог, указанный в конфигурационном параметре reportPath модуля RPT_METADATA.
2. Запустите автоматическое обновление описаний отчетов:
установите в значение 1 параметр конфигурации reloadReports, который хранится на сервере ZooKeeper в ветке /ps/config/apps/tdp/report_engine/<host_name>/automation.



Примечание. Через 3 минуты параметр автоматически выключится (перейдет в значение по умолчанию: 0), после чего отчеты будут формироваться согласно последним установленным версиям их описания.

Файлы FRX-отчетов:

- *.jar – генератор FRX-отчета (обязательный);
- report*.xml – описатель отчета (обязательный);
- *.frx – шаблон FRX-отчета (обязательный при использовании технологии FRX; опциональный при использовании технологии NATIVE);
- *.xsd – схема FRX-отчета (опциональный).

Описатель отчета (файл report*.xml; описывает структуру данных FRX-отчета) должен содержать одну из структур, приведённых в примерах.

Пример файла report.xml:

```
<?xml version="1.0" encoding="utf-8"?>
  <report type="frx" format-version="1">
    <id>sample</id>
    <version>001.00</version>

    <description>Report Sample based on PostgreSQL
Package</description>
    <description lang="ru">Пример отчета на основе пакета
PostgreSQL</description>

    <generatorName>universal.postgresql.package</generatorName>
    <schema>sample.xsd</schema>

    <readyForMono>true</readyForMono>
    <param name="packageName">sample_report_frx</param>

    <defaultLanguage>ru</defaultLanguage>

    <template lang="ru">t_ru.frx</template>
    <template lang="en">t_en.frx</template>

  </report>
```

где:

- секция `report` с атрибутами:
 - `type` – технология отчета; возможные значения: `frx.mono`;
 - `format-version` – версия формата;
- `version` – версия отчета;
- `generatorName` – имя генератора данных (в случае использования универсального генератора укажите `universal.postgresql.package`);
- `schema` – файл со схемой данных для отчетов;
- `readyForMono` – флаг совместимости отчета с версией FrNetBuilder для Mono; возможные значения: `true` или `false`;
- `param` – параметр с атрибутом `name`, в котором в случае использования универсального генератора укажите имя пакета, реализующего отчет;
- `defaultLanguage` – язык отчета по умолчанию;
- `template` – наименование файла шаблона отчета (например, `template_ru.frx`) с атрибутом `lang`, в котором укажите язык отчета.

Пример индивидуальной настройки отчетов (файл `settings.xml`)

```
<?xml version="1.0" encoding="utf-8"?>
<report type="frx" format-version="1">
  <id>r1</id>

  <frx>
    <export>
      <html type="object">
        <pictures>>true</pictures>
        <wysiwyg>>true</wysiwyg>
      </html>
      <pdf type="object">
        <title>title</title>
        <author>author</author>
        <hideToolbar>>false</hideToolbar>
        <hideMenubar>>false</hideMenubar>
      </pdf>
    </export>
  </frx>
</report>
```

2.2.9. Параметры установки и настройки REPORT_ENGINE

Параметры подключения к серверу RabbitMQ:

- rabbitMQ_RE_userName – имя пользователя;
- rabbitMQ_RE_password – зашифрованный пароль пользователя;
- rabbitMQ_RE_vhost – виртуальный хост;
- rabbitMQ_RE_host – хост;
- rabbitMQ_RE_port – порт.

Параметры подключения к серверу NGINX для обеспечения функциональности обновления без простоя:

- nginx:
 - nginx_host – хост NGINX;
 - nginx_port – порт NGINX;
 - user – имя пользователя NGINX;
 - password – пароль пользователя.

Параметры конфигурации REPORT_ENGINE:

- ps:report_engine:
 - logLevel – уровень журналирования (TRACE, DEBUG, INFO, WARN, ERROR, FATAL); значение по умолчанию: ERROR;
 - secretKey – секретный ключ; если параметр не задан, то шифрование не применяется и пароли зашифровывать не нужно; если параметр задан, то укажите зашифрованные значения для всех паролей (для каждого компонента, где необходимо);

- S3 – имя объектного хранилища и настройки доступа к нему:
 - `endpoint` – адрес сервера;
 - `mode` – режим интеграции с хранилищем; возможные значения:
 - 0 – хранилище отключено;
 - 1 – только на чтение;
 - 2 – только на запись;
 - 3 – на чтение и запись;
 - `bucketName` – имя `bucket`'а в хранилище отчетов;
 - `region` – регион расположения хранилища.

Параметры подключения к базе данных:

- `db`:
 - `username` – имя пользователя базы данных;
 - `password` – пароль;
 - `url` – URL для подключения к базе данных;
 - `data_tablespace` – табличное пространство данных;
 - `index_tablespace` – табличное пространство индексов;
 - `roles_prefix` – префикс имён ролей для доступа к объектам схемы продукта (значение по умолчанию `RE_ROLE`);
 - `sys` – параметры подключения от имени пользователя с привилегиями `sysdba`:
 - `username` – имя пользователя;
 - `password` – пароль;
 - `url` – URL для подключения;
 - `data` – параметры для создания технического пользователя для приложений:
 - `rpteng_user` – пользователь, которому будет выдан набор ролей `ps.report_engine.db.roles_prefix_*`;
 - `rpteng_pswd` – пароль пользователя.

Параметры модуля RPT_CONFIG

- `rpt_config`:
 - `rpt_config_host` – сервер для установки;
 - `rpt_config_port` – порт сервера;
 - `rptQueue` – параметры конфигурации для сервера очередей:
 - `server` – список серверов в формате `server:port` для соединения с RabbitMQ, через запятую;



Примечание. Необходимо указывать серверы одного кластера.

- `virtualHost` – виртуальный хост;
- `userName` – имя пользователя RabbitMQ;
- `password` – пароль пользователя RabbitMQ; зашифрованное значение, если задано значение параметра `secretKey`;
- `request` – параметры доступа к заявкам на построение отчетов:
 - `prefetchCount` – количество сообщений, считываемых за одну операцию из очереди разовой печати;

- `exchangeName` – наименование точки обмена для получения сообщений;
- `routingKeyPrefix` – префикс ключа маршрутизации;
- `statistic` – параметры для статистических сообщений;
- `exchangeName` – наименование точки обмена для статистических сообщений;
- `routingKeyPrefix` – префикс ключа маршрутизации;
- `result` – параметры доступа к результату обработки:
 - `exchangeName` – наименование точки обмена для получения результата;
 - `routingKeyPrefix` – префикс ключа маршрутизации;
- `documentRequest` – параметры доступа к запросам на построение выходных документов:
 - `exchangeName` – наименование точки обмена для запросов построения документов;
 - `routingKeyPrefix` – префикс ключа маршрутизации запросов на формирование выходных документов;
- `external` – параметры доступа к ответам о готовности данных от внешних сервисов:
 - `exchangeName` – наименование точки обмена для получения сообщений;
 - `routingKeyPrefix` – префикс ключа маршрутизации.

Параметры конфигурации RabbitMQ:

- `result` – параметры конфигурации RabbitMQ для отсылки результата выполнения массовых заявок на формирование документов:
 - `exchangeName` – наименование точки обмена для получения ответов;
 - `routingKeyPrefix` – префикс ключа маршрутизации.

Параметры доступа на SFTP-сервер:

- `engineResource`:
 - `localStorages` – именованные хранилища выходных документов и файлов данных:
 - `test_storage` – имя тега является наименованием хранилища; имена хранилищ регистрозависимы; описание хранилища:
 - `nix` – список путей к папкам в Unix/Linux-системе (через запятую);
 - `local`:
 - `path` – локальный путь к каталогу;
 - `id` – имя или IP-адрес машины, для которой это хранилище является локальным;
 - `storageClusters` – конфигурация кластеров SFTP-серверов:
 - `key` – ключ шифрования для парольной фразы (`PassPhrase`);
 - `clusterReports` – имя тега является именем кластера (имена регистрозависимы), описание кластера:
 - `servers` – описание узлов кластера:
 - `server1` – описание первого узла кластера, имя тега является именем узла; параметры узла:
 - `protocol` – протокол для подключения к узлу (например, `sftp`);
 - `host` – хост для подключения по протоколу SSH;
 - `login` – пользователь для подключения по протоколу SSH;
 - `password` – пароль пользователя; зашифрованное значение, если задано значение параметра `secretKey`;
 - `sshKeyPath` – относительный путь к файлу с SSH-ключом (от каталога, который является домашней директорией для пользователя);

- `sshKeyPassphrasePath` – относительный путь к файлу с зашифрованной парольной фразой (`PassPhrase`);
- `path` – путь к каталогу на сервере, в котором будут храниться файлы отчетов;
- `resources` – конфигурация ресурсов для внешних сервисов:
 - `report` – имя тега является наименованием ресурса (их может быть несколько), описание ресурса:
 - `storageCluster` – ссылка на имя кластера (имена регистрозависимы).

Параметры модуля RPT_DATA_BUILD

- `rpt_data_build`:
 - `httpRequestsCacheLifeTime` – время хранения в кеше ответов от внешних сервисов;
 - `externalRequestsProcessorsCount` – число обработчиков ответов от внешних сервисов;
 - `resourceCleanersCount` – число обработчиков для очистки ресурсов (SFTP, локальной файловой системы и т. п.);
 - `builderCount` – число обработчиков для формирования отчетов;
 - `ccmExportBasePath` – базовый путь для выгрузки информации по абонентским профилям; обязательно укажите точку монтирования каталогов, если внешний сервис использует выгрузку с относительными путями; для выгрузки с абсолютными путями оставьте параметр пустым;
 - `camExportBasePath` – базовый путь для выгрузки иерархии клиентов из внешнего сервиса в выделенный каталог на серверах SFTP;
 - `request` – параметры заявки:
 - `ttl` – время жизни заявки;
 - `cancelTryCount` – количество попыток отменить заявку на выгрузку данных из внешней системы (интервал между попытками – 3 минуты); значение по умолчанию: 1.

Параметры модуля RPT_INTERFACE

- `rpt_interface`:
 - `syncMode` – параметры синхронного режима запросов:
 - `maxTime` – максимальное время ожидания ответа, в секундах;
 - `result` – параметры результата:
 - `lifetime` – время жизни конечного результата на сервере, в секундах;
 - `asBackend` – признак необходимости регистрации в ZooKeeper; возможные значения: `true`, `false`.

Параметры модуля RPT_METADATA

- `rpt_metadata`:
 - `baseUrl` – URL сервера метаданных;
 - `reportPath` – абсолютный путь к каталогу с отчетами.

Параметры модуля RPT_FRNET_BUILD

- `rpt_frnet_build`:
 - `logLevel` – уровень журналирования модуля (TRACE, DEBUG, INFO, WARN, ERROR, FATAL); значение по умолчанию: ERROR;
 - `logFile` – имя журнального файла с указанием пути; если указан неполный путь или

- пустое значение, то журнал сохраняется в каталоге запуска;
- `maxLogSizeKb` – максимально допустимый размер журнального файла, в Kb; значение по умолчанию: 10240 (10 Mb);
 - `listenPort` – порт слушателя; значение по умолчанию: 9393;
 - `maxBuilders` – максимальное количество одновременно работающих потоков построения отчетов; значение по умолчанию: 16;
 - `tmpPath` – каталог для временного хранения, путь может быть относительным или абсолютным; значение по умолчанию: пусто (не использовать);
 - `maxRptInMemory` – максимальное количество отчетов (пар FRX+XSD), кешируемых в памяти; значение по умолчанию: 1000;
 - `maxXmlSizeKb` – максимально допустимый размер файла данных (*.XML), в Kb; значение по умолчанию: 2097152 (2 Gb);
 - `dataSizeKbToUseFileCache` – размер файла данных, при котором происходит переключение на файловый кеш с целью экономии оперативной памяти, в Kb; значение по умолчанию: 32768 (32 Mb);
 - `cleanData` – признак удаления файла данных после обработки отчета; возможные значения: `true/false`; значение по умолчанию: `true`;
 - `export` – параметры экспорта для различных форматов со значениями по умолчанию:
 - `txt`:
 - `encoding`: 'utf-8'
 - `dataOnly`: 'false'
 - `pageBreaks`: 'true'
 - `emptyLines`: 'false'
 - `frames`: 'true'
 - `textFrames`: 'false'
 - `html`:
 - `layers`: 'true'
 - `pictures`: 'true'
 - `wysiwyg`: 'true'
 - `pageBreaks`: 'false'
 - `mht`:
 - `pictures`: 'true'
 - `wysiwyg`: 'true'
 - `rtf`:
 - `imageFormat`: 'emf'
 - `pageBreaks`: 'true'
 - `pictures`: 'true'
 - `wysiwyg`: 'true'
 - `xls`:
 - `dataOnly`: 'false'
 - `pageBreaks`: 'true'
 - `wysiwyg`: 'false'
 - `seamlessTable`: 'false'
 - `pdf`:
 - `embeddingFonts`: 'false'
 - `background`: 'true'

- printOptimized: 'true'
- title: ''
- author: ''
- subject: ''
- keywords: ''
- creator: ''
- producer: ''
- outline: 'true'
- hideToolbar: 'false'
- hideMenubar: 'false'
- hideWindowUI: 'false'
- fitWindow: 'false'
- centerWindow: 'true'
- printScaling: 'false'
- userPassword: '' – зашифрованное значение пароля, если задано значение параметра secretKey;
- ownerPassword: '' – зашифрованное значение пароля, если задано значение параметра secretKey;
- allowPrint: 'true'
- allowModify: 'true'
- allowCopy: 'true'
- allowAnnotate: 'true'
- jpegQuality: 95
- xml:
 - dataOnly: 'false'
 - pageBreaks: 'true'
 - wysiwyg: 'true'
- csv:
 - dataOnly: 'false'
 - encoding: 'default'
 - separator: ''
 - noQuotes: 'false'

3. Запуск и остановка продукта

Порядок запуска продукта:

1. Запустите модули DGS – выставите количество реплик подов в кластере Kubernetes; последовательность запуска модулей не имеет значения.
2. Запустите генератор отчетов REPORT_ENGINE от имени пользователя, который имеет необходимые права, в следующем порядке:
 - службу RabbitMQ;
 - сервер конфигурации (RPT_CONFIG);
 - серверы приложений (TOMCAT) с установленными модулями REPORT_ENGINE;
 - модуль RPT_FRNET_BUILD (как автономное приложение).
3. Если требуется редактирование FRX-отчетов, запустите модуль RPT_FRNET_EDIT (как автономное приложение).

Для остановки продукта выполните обратные запуску действия. Последовательность остановки модулей DGS не имеет значения.

4. Обновление

Для обновления продукта последовательно выполните следующие действия:

1. Остановите запущенные модули продукта.
2. Установите новую версию продукта, как описано в главе [«Установка продукта»](#).
3. Запустите продукт.

5. Проверка работоспособности

Для проверки установки продукта DGS убедитесь, что в log-файлах установочных скриптов нет сообщений об ошибках.

Для проверки работоспособности компонентов после старта убедитесь, что:

- log-файлы не содержат ошибок;
- в ZooKeeper зарегистрированы узлы:
 - ps/instances/apps/dgs_api/nodes/<узел с именем хоста и порта>;
 - ps/instances/apps/dgs_api/nodes/versionedNode/{current-version}/<узел с именем хоста и порта>.

Для проверки работоспособности компонента REPORT_ENGINE:

- для модулей, которые установлены на сервер Tomcat, – вызовите любой API-метод;
- для REPORT_ENGINE в целом – вызовите любой установленный отчет.