

# СИСТЕМА ФОРМИРОВАНИЯ ДОКУМЕНТОВ

Руководство по эксплуатации

Версия 1.4.0

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](mailto:office@nexign.com); [www.nexign.com](http://www.nexign.com)

# Содержание

|  |           |
|--|-----------|
| <b>1. Общие сведения</b>   | <b>5</b>  |
| <b>2. Системные требования</b>   | <b>6</b>  |
| 2.1. Программное обеспечение и аппаратные средства   | 6         |
| 2.2. Дополнительные рекомендации по аппаратному обеспечению                                    | 6         |
| 2.3. Рекомендации по квалификации персонала  | 7         |
| <b>3. Функциональная архитектура продукта</b>  | <b>8</b>  |
| <b>4. Настройка</b>  | <b>9</b>  |
| <b>5. Особенности эксплуатации</b>   | <b>10</b> |
| 5.1. Запуск продукта   | 10        |
| 5.2. Особенности кеширования внешних запросов  | 10        |
| 5.3. Работа с RPT_CONFIG   | 10        |
| 5.4. Работа с RPT_INTERFACE – запросы на построение отчета                                     | 11        |
| 5.4.1. Синхронный режим работы   | 11        |
| 5.4.2. Асинхронный режим работы  | 11        |
| 5.5. Работа с RPT_FRNET_EDIT   | 12        |
| 5.5.1. Запуск редактора отчетов  | 12        |
| 5.5.2. Графический интерфейс редактора FRX-отчетов   | 12        |
| 5.5.3. Редактирование схемы данных отчета  | 15        |
| <b>6. Журналирование</b>   | <b>18</b> |
| 6.1. Интеграция с Kubernetes   | 18        |
| 6.2. Настройки логирования модулей DGS   | 18        |
| 6.3. Структура файлов конфигурации журналирования  | 19        |
| 6.4. Уровни журналирования   | 19        |
| 6.5. Настройка логирования REPORT_ENGINE средствами ELOG                                       | 20        |
| <b>7. Мониторинг</b>   | <b>21</b> |
| 7.1. Мониторинг модулей DGS  | 21        |
| 7.1.1. Метрики клиента базы данных   | 21        |
| 7.1.2. Метрики соединения с базой данных   | 22        |
| 7.1.3. Метрики соединения с RabbitMQ   | 23        |
| 7.1.4. Метрики HTTP-сервера  | 24        |
| 7.1.5. Параметры работы GarbageCollector   | 24        |
| 7.1.6. Прочие метрики  | 26        |
| 7.2. Мониторинг модулей REPORT_ENGINE  | 27        |
| 7.2.1. Мониторинг состояния модуля   | 28        |
| 7.2.2. Мониторинг времени работы сервиса   | 29        |
| 7.2.3. Мониторинг соединения с RabbitMQ  | 30        |
| 7.2.4. Мониторинг Storages   | 32        |
| 7.2.5. Мониторинг соединений с серверами конфигурации и метаданных (внутреннее взаимодействие) | 35        |
| 7.2.6. Мониторинг ошибок построения данных   | 38        |
| 7.2.7. Мониторинг производительности системы   | 40        |
| 7.2.8. Общее количество обработанных заявок на формирование отчетов                            | 43        |

7.2.9. Мониторинг узлов SFTP-кластера ..... 46

# 1. Общие сведения

Продукт «Система формирования документов» (DGS) предназначен для обработки разовых и массовых заказов на генерацию отчетов.

Данное руководство содержит описание сведений, необходимых для эксплуатации продукта.

Описание установки продукта представлено в Руководстве по установке [DGS-DOC\_INSTALL].

## 2. Системные требования

Для эксплуатации продукта необходимо наличие программного и аппаратного обеспечения, перечисленного в данной главе.

### 2.1. Программное обеспечение и аппаратные средства

Для корректной работы оборудование должно обладать следующими характеристиками:

- сервер базы данных:
  - ОС – РЕД ОС;
  - количество ядер процессора – не менее 4;
  - оперативная память – не менее 32 Гб;
  - объем свободного дискового пространства – не менее 250 Гб;
- сервер для установки модулей DGS:
  - ОС – РЕД ОС;
  - количество ядер процессора – не менее 4;
  - оперативная память – не менее 8 Гб;
  - объем свободного дискового пространства – не менее 50 Гб;
- серверы приложений для REPORT\_ENGINE:
  - Tomcat – не менее 2 экземпляров;
  - ОС – РЕД ОС;
  - количество ядер процессора – не менее 2;
  - оперативная память – не менее 4 Гб;
  - объем свободного дискового пространства – не менее 50 Гб на узел;
  - канал связи – не менее 1 Гбит/сек.



#### Примечание.

Примерная производительность сервера приложений минимальной конфигурации – 20000 отчётов в час в формате PDF.

Для работоспособности продукта необходимо следующее ПО:

- ОС: РЕД ОС версии 7.0;
- Java версии 11 или выше;
- Java JDK версии 1.8;
- PostgreSQL версии 13.1 или выше (NORD версии 2.2.0);
- Kubernetes версии 1.19
- RabbitMQ версии 3.6 или выше;
- Apache ZooKeeper версии 3.8 или выше.
- Apache Tomcat версии 8.2.0, который предоставляет АО «Нэксайн».

Перед установкой продукта убедитесь в корректной конфигурации и настройке окружения.

### 2.2. Дополнительные рекомендации по аппаратному обеспечению

1. Пропускная способность канала между файловым хранилищем и соответствующими серверами получателей отчетов должна быть достаточной для загрузки необходимых массивов данных.

2. Для ведения журналов может потребоваться дополнительный объем дискового пространства, который зависит от установленного уровня журналирования; по умолчанию используется уровень ERROR.
3. При использовании минимальной конфигурации для компонента REPORT\_ENGINE на одном экземпляре Tomcat разверните сервер конфигурации (RPT\_CONFIG), на другом – остальные модули REPORT\_ENGINE.

Для повышенной надёжности рекомендуется разнести модули компонента REPORT\_ENGINE по серверам Tomcat в следующей конфигурации:

- RPT\_CONFIG – 2 сервера Tomcat низкой производительности;
- RPT\_METADATA и RPT\_INTERFACE – 2 сервера Tomcat средней производительности;
- RPT\_DATA\_BUILD – 2 сервера Tomcat высокой производительности.

Данная схема обеспечивает аппаратное резервирование узлов системы.

4. Для REPORT\_ENGINE рекомендуется выделить отдельные серверы Tomcat и не устанавливать на них другие компоненты и продукты.

## 2.3. Рекомендации по квалификации персонала

Администратор, программист и системный программист должны иметь высшее профильное образование и сертификаты компании – производителя операционной системы, а также понимать основы функционирования продукта и иметь опыт работы с Kubernetes.

## 3. Функциональная архитектура продукта

DGS – клиент-серверное решение, в архитектуру которого входят компоненты:

- обработчик заказов на формирование отчетов; состоит из модулей:
  - DGS\_API – предоставляет API-интерфейс для взаимодействия внешних систем с продуктом;
  - DGS\_PROCESSOR – обеспечивает обработку заказов на формирование отчетов и взаимодействие с компонентом REPORT\_ENGINE;
  - DGS\_COMBINER – выполняет архивацию и формирование единого файла-отчета из нескольких документов или комплектов документов;
  - DGS\_DBADMIN – обеспечивает партиционирование таблиц базы данных;
- REPORT\_ENGINE – генератор отчетов с разделением процессов построения данных и визуализации отчетов; состоит из модулей:
  - RPT\_COMMON – общие библиотеки для построения отчетов;
  - RPT\_CONFIG – сервер конфигурации;
  - RPT\_METADATA – сервер получения метаданных для отчетов;
  - RPT\_DATA\_BUILD – сервер подготовки данных для отчетов;
  - RPT\_INTERFACE – интерфейс для построения отчетов;
  - RPT\_FRNET\_BUILD – сервер построения отчетов в графических форматах;
  - RPT\_FRNET\_EDIT – редактор FRX-отчетов; предоставляет пользовательский графический интерфейс для редактирования шаблонов FRX-отчетов и просмотра подготовленных отчетов.

## 4. Настройка

Настройка продукта DGS выполняется в процессе установки.

Для изменения настроек продукта во время работы установите новые значения конфигурационных параметров на сервере ZooKeeper.

Подробное описание установки и конфигурационных параметров приведено в Руководстве по установке [DGS-DOC\_INSTALL].

## 5. Особенности эксплуатации

Для реализации функциональных возможностей DGS взаимодействует с внешними системами – получает от них запросы на создание разового или массового заказа на формирование отчетов.

В процессе работы DGS использует инфраструктурные узлы:

- базу данных, реализованную на СУБД с открытым исходным кодом PostgreSQL, – предоставляет продукт «Система управления объектно-реляционной базой данных Nexign» (NORD);
- брокер сообщений RabbitMQ – для организации взаимодействия отдельных модулей продукта между собой и публикации уведомлений о готовности отчетов для получателей отчетов;
- сервер ZooKeeper – для хранения собственной конфигурации и настроек;
- файловый сервер – сервер на стороне клиента для хранения готовых отчетов; DGS обеспечивает обмен файлами по протоколу SFTP.

### 5.1. Запуск продукта

Порядок запуска продукта описан в Руководстве по установке [DGS-DOC\_INSTALL].

### 5.2. Особенности кеширования внешних запросов

Кеш для хранения внешних запросов находится в памяти приложения.

В случае если включено кеширование (`cacheable = true`), то в память будет помещаться тело ответа.

Уникальность синхронных и асинхронных запросов определяют:

- `path`;
- тело запроса;
- URL-параметры;
- Query-параметры.

Пример запроса:

```
dictionaries/common/branches/{branchId}?macrId=100
```

где:

- `branchId` – URL-параметр;
- `macrId` – Query-параметр.

Не рекомендуется включать кеширование для данных, связанных с информацией по клиенту.

Следует включать кеширование для запросов справочной информации, данных по операторам и т. п.

В случае кеширования асинхронных запросов нельзя очищать каталог с файлами–ответами от внешних сервисов.

### 5.3. Работа с RPT\_CONFIG

Для получения основной информации о работе сервера конфигураций используйте консоль RPT\_CONFIG.

Для доступа к консоли наберите в адресной строке браузера:

```
http://<хост>:<порт>/rptconfig/console/main
```

Консоль содержит следующую информацию:

- перечень конфигурационных веток в ZooKeeper и их состояние;

В случае ошибки выводится информационное сообщение в поле **Состояние** с описанием возникшей проблемы, например: «*Ошибка разбора файла конфигурации по причине: XML document structures must start and end within the same entity*».

- перечень рабочих узлов с подробной информацией:
  - идентификатор узла;
  - тип узла;
  - расположение узла;
  - дата регистрации;
  - дата обновления;
  - планируемое время перерегистрации узла.

## 5.4. Работа с RPT\_INTERFACE – запросы на построение отчета

Построение отчета возможно в синхронном и асинхронном режиме работы.

### 5.4.1. Синхронный режим работы



**Примечание.** Синхронный режим работы REPORT\_ENGINE используется только для отладки отчетов.

В синхронном режиме построение отчета включает в себя следующие шаги:

1. Запрос данных отчета у сервера метаданных (RPT\_METADATA).
2. Проверка переданного формата на допустимость (входит ли в список возможных форматов).
3. Помещение запроса на построение отчета в очередь.
4. Ожидание сообщения о готовности отчета в течение времени, заданного в параметре `syncMode/maxTime`; по истечении этого времени, если отчет не получен, отправка сообщения об ошибке.
5. Удаление файла отчета после его получения, если при запросе через браузер не задано внешнее файловое хранилище в параметре `fileLocation`.

### 5.4.2. Асинхронный режим работы

В асинхронном режиме построение отчета включает в себя следующие шаги:

1. Запрос данных отчета у сервера метаданных (RPT\_METADATA).
2. Проверка переданного формата на допустимость (входит ли в список возможных форматов).
3. Помещение запроса на построение отчета в очередь.
4. Возвращение идентификатора заказа; проверка статуса формирования отчета.
5. Удаление файла после выполнения запроса по истечении времени, указанного в параметре `result/lifeTime`. Файл не удаляется, если в параметре `fileLocation` задано расположение файла во внешнем файловом хранилище.

## Пример:

Пример вызова отчета в асинхронном режиме по технологии FRX в формате pdf с id=reportBill:

```
http://host:9090/rptinterface/api/v1/order/run?reportId=reportBill&rptFormat=pdf&userLogin=admin&rptNativeTechnology=false&dataSource=bis&rptParams={«BILL_ID»:«7654»}
```

Сформированный отчет помещается в хранилище, указанное в параметре defaultStorage в файле конфигурации RptFrNetBuilder.xml.

## 5.5. Работа с RPT\_FRNET\_EDIT

Модуль «Редактор FRX-отчетов» (RPT\_FRNET\_EDIT) предоставляет пользовательский графический интерфейс, который позволяет:

- создавать и редактировать шаблоны отчетов;
- создавать и редактировать схемы данных для формирования отчетов;
- просматривать в режиме редактирования шаблона будущий отчет с пользовательскими данными;
- просматривать подготовленные отчеты в формате FPX.

### 5.5.1. Запуск редактора отчётов

Для запуска редактора используйте стартовое меню операционной системы или командную строку.

При запуске через командную строку в качестве параметра укажите имя обрабатываемого файла, тогда будет активирован режим работы в соответствии с расширением файла:

- `xsd` (схема данных) – активирует режим создания и редактирования схемы данных;
- `xml` (данные) – активирует режим добавления и изменения данных;
- `frx` (визуальное представление отчета) – активирует режим редактирования визуального представления отчета;
- `fpx` (подготовленный к просмотру и печати отчет с данными) – активирует интерфейс для просмотра готового отчета, все другие режимы при этом недоступны.

### 5.5.2. Графический интерфейс редактора FRX-отчетов

Главное окно редактора FRX-отчетов (см. [Рис. 1](#)) содержит:

- панель инструментов;
- панель структуры схемы;
- панель данных.

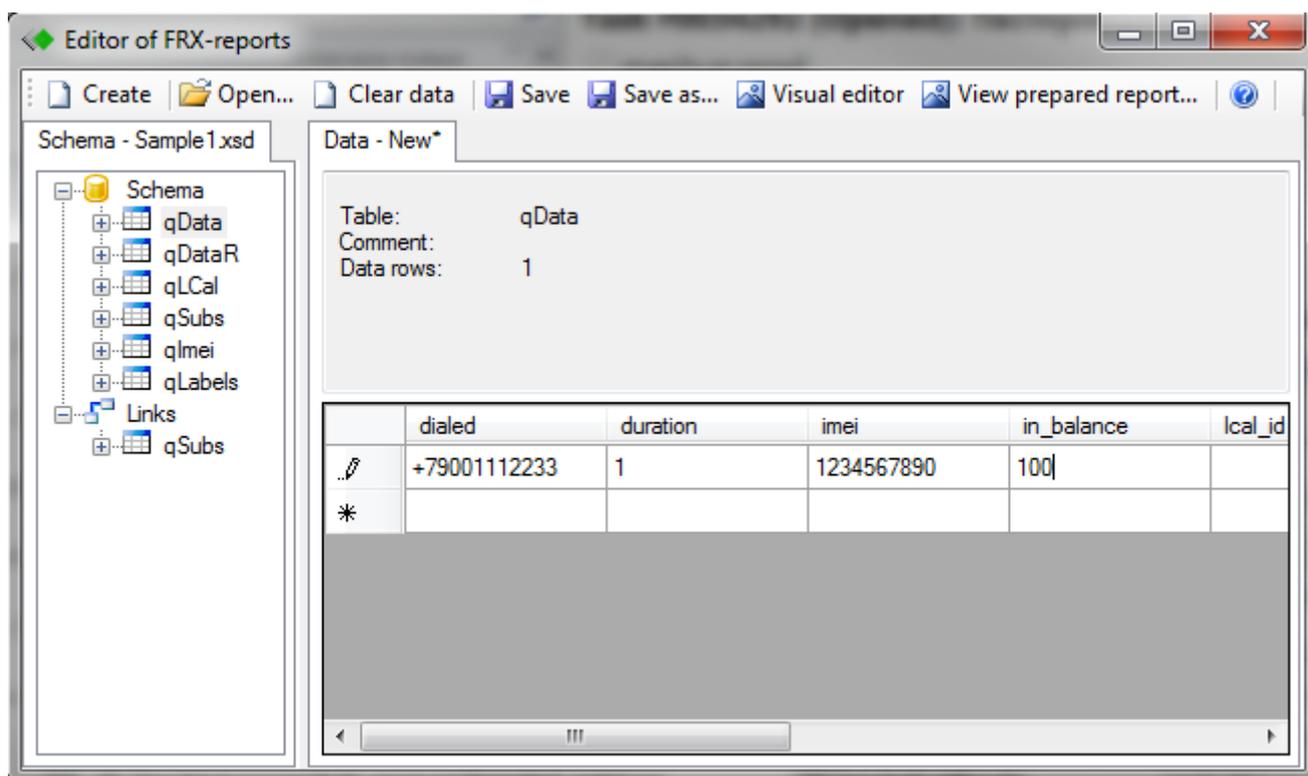


Рис. 1. Общий вид главного окна редактора FRX-отчетов

## Панель инструментов

Для работы со схемами данных и отчетами используйте кнопки на панели инструментов или соответствующие «горячие» клавиши:

- **Create** (Ctrl + N) – создать новую схему данных;
- **Open...** (Ctrl + O) – открыть существующую схему данных для редактирования;
- **Clear data** (Ctrl + D) – очистить данные;
- **Save** (Ctrl + S) – сохранить схему;
- **Save as...** – сохранить схему в другой файл (открывает диалоговое окно для задания параметров сохранения файла);
- **Visual editor** – открыть окно визуального редактора отчетов;

Подробнее о работе с визуальным редактором отчетов см. в документации FastReport (<http://www.fast-report.com/ru/download/documentation/>).

- **View prepared report...** – просмотреть подготовленный отчет;
- **?** – открыть справочное окно с описанием вызова основных операций.

## Панель структуры схемы

На панели в левой части окна (см. [Рис. 2](#)) отображается структурное дерево открытой схемы данных отчёта. Дерево содержит список таблиц с перечнем полей и список связей (Links) между полями таблиц.

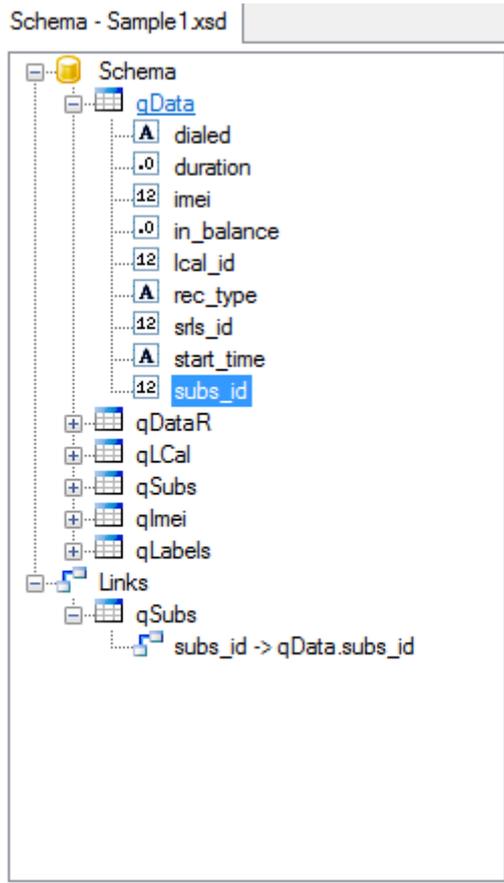


Рис. 2. Общий вид панели структуры схемы данных

## Панель данных

На панели в правой части окна (см. [Рис. 3](#)) отображаются данные таблицы, выделенной на панели структуры схемы.

|   | dialed       | duration | imei       | in_balance | lcal_id |
|---|--------------|----------|------------|------------|---------|
| ▶ | +79001112233 | 1        | 1234567890 | 100        |         |

Рис. 3. Общий вид панели данных

### 5.5.3. Редактирование схемы данных отчёта

Для работы со структурой схемы данных используйте «горячие» клавиши:

- **Insert** – вставить в выделенную ветку дерева схемы новый элемент;
- **Delete** – удалить из дерева схемы выбранный элемент;
- **Enter** – начать редактирование выбранного элемента;
- **+** – развернуть выбранную ветку дерева;
- **-** – свернуть выбранную ветку дерева;
- **Ctrl +** – развернуть все ветки дерева;
- **Ctrl -** – свернуть все ветки дерева.

#### Добавление таблицы в схему данных

1. На панели структуры схемы (см. [Рис. 2](#)) выделите элемент `Schema` и нажмите на клавишу **Insert**.
2. В открывшемся окне (см. [Рис. 4](#)) заполните поля:
  - `Name` – имя таблицы;
  - `Description` – описание таблицы.

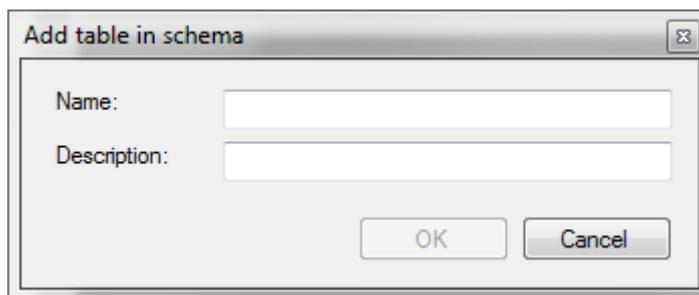


Рис. 4. Окно добавления таблицы в схему

## Добавление столбца в таблицу

1. На панели структуры схемы (см. [Рис. 2](#)) выделите таблицу и нажмите на клавишу **Insert**.
2. В открывшемся окне (см. [Рис. 5](#)) заполните поля:
  - Name – имя столбца;
  - Description – описание столбца;
  - Type – тип столбца; из раскрывающегося списка выберите значение:
    - Integer – целое число;
    - Currency – валюта;
    - Float – вещественное число;
    - Text – текстовое значение;
    - Date – дата;
    - Picture – рисунок.

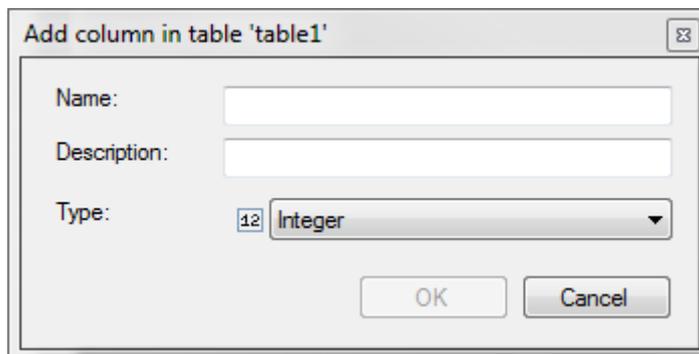


Рис. 5. Окно добавления столбца в таблицу

## Создание связей между полями таблиц

1. На панели структуры схемы (см. [Рис. 2](#)) выделите элемент `Links` и нажмите на клавишу **Insert**.
2. В открывшемся окне (см. [Рис. 6](#)) создайте связи между столбцами таблиц схемы данных.

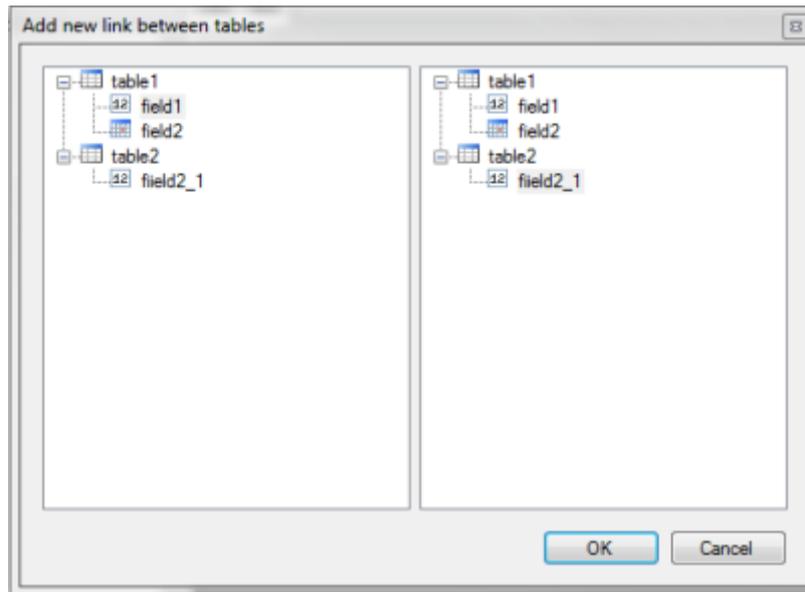


Рис. 6. Окно создания связей между столбцами таблицы

### Редактирование содержимого таблицы

1. На панели структуры схемы выберите таблицу.
2. На панели данных (см. [Рис. 3](#)) отредактируйте значения полей таблицы; для активации ячейки используйте мышь или клавишу F2 – начать редактирование ячейки; для очистки ячейки – клавишу Delete.

## 6. Журналирование

Основное журналирование продукта осуществляется в текстовый файл. Файл рекомендуется анализировать с помощью утилит фильтрации (find, grep и т.д.) и регулярных выражений.

Упростить процесс диагностики и поиска ошибок позволяет интегрирование DGS с продуктом «Сервис централизованной обработки логов» (ELOG).

ELOG обеспечивает централизованное хранение журнальных файлов (log-файлов) и управление трассировками, использует общие механизмы логирования работы IT-решений АО «Нэксайн».

### 6.1. Интеграция с Kubernetes

Для интеграции ELOG с Kubernetes на каждый узел с Docker-контейнерами установите сервис Filebeat. Он выводит данные при помощи потока (STDOUT) в Logstash, откуда они могут перенаправляться, например, в базу хранения типа Elasticsearch. При интеграции с Kubernetes используется стандартный механизм логирования при помощи библиотеки logback (см. документацию на сайте <https://logback.qos.ch/>).

### 6.2. Настройки логирования модулей DGS

В составе дистрибутива поставляется файл logback.xml с настройками сбора информации с модулей продукта.

Продукт DGS поддерживает унифицированный формат log-файлов CEF (Common Event Format).

Формат CEF определен стандартом и представляет собой список полей, разделенных символом «|». Запись состоит из следующих полей:

- стандартные поля:
  - Version: «CEF:0»;
  - Device Vendor: «nexign»;
  - Device Product: «dgs-api»;
  - Device Version: версия DGS, например «1.3.0»;
  - Device Event Class ID: «RecipientTypeManagerImpl.java:39»;
  - Name: сообщение об ошибке;
  - Severity: уровень события = (ERROR|WARN|INFO|DEBUG|TRACE);
- расширенные поля:
  - eventTime: время события в формате «yyyy-MM-dd'T'HH:mm:ssZ»;
  - threadName: имя потока, который инициировал запись в log-файл;
  - stackTrace: стек вызовов до функции, вызвавшей исключение.

Пример записи:

```
CEF:0|nexign|dgs-api|1.3.0|RecipientTypeManagerImpl.java:39|Recipient
types caches are cleared|DEBUG|eventTime=2022-08-
02T11:03:37+0000|threadName=scheduling-1
```

Журнальные файлы дополнительно выводятся в консоль.

## 6.3. Структура файлов конфигурации журналирования

Общая структура файла конфигурации журналирования представлена на [Рис. 7](#):

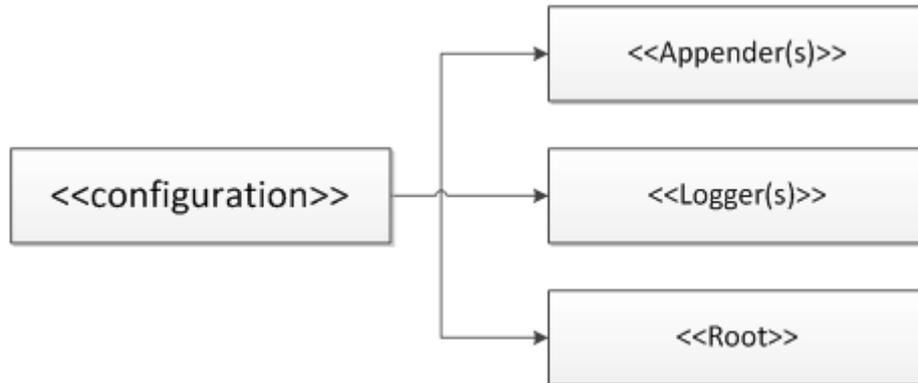


Рис. 7. Структура файла конфигурации журналирования

Конфигурационный файл состоит из компонентов:

1. **Appender** – отвечает за вывод сообщений в файл, консоль, e-mail и т. д. Конфигурация компонента включает настройку направления вывода, формат выводимых сообщений, параметры ротации.
2. **Logger** – отвечает за уровень и направление выводимых сообщений. Конфигурация компонента включает настройку атрибутов:
  - `name` – наименование logger'a; обязательный атрибут;
  - `level` – уровень выводимых сообщений (уровень журналирования);
  - `additivity` – возможность дополнительного вывода сообщений в основной журнал; возможные значения: `true` или `false`.
3. **Root** – задает базовую конфигурацию журналирования и используется для logger'ов, у которых не задана конфигурация; содержит атрибут `level` – уровень вывода сообщений.

## 6.4. Уровни журналирования

Поддерживаются следующие уровни журналирования:

- **ALERT** (наличие уровня определяется конфигурацией журналирования отдельного модуля) – вывод сообщения требует немедленной реакции;
- **FATAL** – информация о критических ошибках, которые возникли в работе модулей и привели к аварийной остановке приложения или к невозможности запуска (например, недостаточно памяти для JVM);
- **ERROR** – информация об ошибках, возникших в работе модулей продукта; содержит описание технических проблем, которые необходимо решить для надлежащего функционирования приложения (например, сообщение о невозможности подключить модуль к базе данных);
- **WARN** – информация о событиях, которые потенциально могут привести к ошибкам в работе модулей: временные проблемы или неожиданное поведение, которое не препятствует функционированию приложения (например, неправильный ввод логина пользователя);
- **INFO** – подтверждения или информация о событиях, не приводящих к ошибкам в работе модулей; сообщения о выполнении очередной операции (например, о подключении к очередям RabbitMQ);
- **DEBUG** – подробное и детальное журналирование всей системной информации для последующего использования в отладке (например, начало выполнения метода);
- **TRACE** – подробный вывод информации обо всех операциях приложения; предназначен для

случаев, когда `DEBUG` не позволяет локализовать ошибку.

## 6.5. Настройка логирования `REPORT_ENGINE` средствами `ELOG`

Компонент `REPORT_ENGINE` использует общие механизмы протоколирования работы в журнальных файлах.

В режиме интеграции с «Сервисом централизованной обработки логов» (`ELOG`) собираются данные об узлах и этапе обработки заказов разовой печати. Сообщения записываются в JSONL-формате и содержат поля:

- `eventTime` – время отправки трассировки заказа;
- `product` – название продукта;
- `productVersion` – версия продукта;
- `component` – название компонента, отправившего трассировку;
- `componentVersion` – версия компонента, отправившего трассировку;
- `pstxid` – идентификатор исходного запроса (если возможно определить);
- `custom_app_has_srrdId` – идентификатор заказа на разовую печать документа;
- `message` – сообщение с описанием состояния заказа на узле-обработчике.

Для использования `ELOG`:

1. Настройте продукт `ELOG` и внесите в его конфигурацию файл `${catalina.home}/logs/report_engine.jsonl` сервера `Tomcat`, на котором установлен модуль `RPT_INTERFACE`.
2. Установите уровень журналирования `INFO` в файле `${catalina.home}/conf/report_engine_logback.xml` сервера `Tomcat`, на котором установлен модуль `RPT_INTERFACE` (возможные значения: `OFF` – трассировка отключена, `INFO` – трассировка включена).

## 7. Мониторинг

В DGS реализована возможность мониторинга работы отдельных модулей продукта.

Для удобства мониторинга рекомендуется использовать продукт «Централизованная платформа мониторинга» (EMON). Настройку интеграции с продуктом EMON выполните в соответствии с документацией на этот продукт.

Описание особенностей мониторинга и метрик мониторинга приведено в разделах:

- [Мониторинг модулей DGS](#);
- [Мониторинг модулей REPORT\\_ENGINE](#).

### 7.1. Мониторинг модулей DGS

В текущем разделе представлено описание технических метрик, которые отражают состояние модулей с префиксом DGS\_ (DGS\_API, DGS\_COMBINER, DGS\_DBADMIN, DGS\_PROCESSOR) и достаточны для определения факта их неисправности при эксплуатации.

Расчет метрик производится каждые 10 секунд, сохраняется в ОП сервера и доступен для выгрузки в формате Prometheus.

Метрики мониторинга отличаются по уровню важности:

- высокий – наступление события влечет приостановку основной функциональности модуля продукта;
- средний – событие не влияет на уровень предоставления основной функциональности, но влечет за собой сбой в работе модуля или его части;
- низкий – информационное сообщение, указывающее на восстановление основной функциональности модуля.

Период опроса метрик регулируется на уровне агента EMON и по умолчанию равен 10 секундам.

Для определения категории метрики используются теги:

- COUNTER – счетчик элементов, полученных из запроса; отражает возрастающее значение, которое не может уменьшаться (количество вызовов, количество отправленных сообщений и т. д.);
- CALCULABLE – вычисляемое значение; рассчитывается за последние N секунд до опроса значений метрик, где N – настраиваемая величина; предоставляются минимальное и максимальное значения;
- GAUGE – числовое значение;
- SUMMARY – суммарное значение;
- TIME – показатель времени; предоставляется в виде количества секунд, прошедших с момента события.

#### 7.1.1. Метрики клиента базы данных

##### **jdbc\_connections\_idle**

Количество установленных, но неактивных соединений.

Тип метрики – gauge.

##### **jdbc\_connections\_active**

Текущее количество активных соединений.

Тип метрики – gauge.

### **jdbc\_connections\_max**

Максимальное количество активных соединений, которые могут быть выделены одновременно.

Тип метрики – gauge.

### **jdbc\_connections\_min**

Минимальное количество простаивающих соединений в пуле.

Тип метрики – gauge.

## 7.1.2. Метрики соединения с базой данных

### **db\_connections\_active**

Количество активных соединений.

Тип метрики – gauge.

### **db\_connections\_creation\_seconds\_max**

Максимальное время создания соединения.

Тип метрики – gauge.

### **db\_connections\_creation\_seconds\_count, db\_connections\_creation\_seconds\_sum**

Суммарное время создания соединений.

Тип метрики – summary.

### **db\_connections\_usage\_seconds\_count, db\_connections\_usage\_seconds\_sum**

Суммарное время использования соединения.

Тип метрики – summary.

### **db\_connections\_usage\_seconds\_max**

Максимальное время использования соединения.

Тип метрики – gauge.

### **db\_connections**

Общее количество подключений.

Тип метрики – gauge.

### **db\_connections\_status**

Статус соединения.

Возможные значения:

- 1 – соединение установлено;
- 0 – нет подключения к базе данных.

Тип метрики – gauge.

### **db\_connections\_timeout\_total**

Общее время ожидания соединения.

Тип метрики – counter.

#### **db\_connections\_min**

Минимальное количество соединений.

Тип метрики – gauge.

#### **db\_connections\_acquire\_seconds\_max**

Максимальное время установления соединения.

Тип метрики – gauge.

#### **db\_connections\_acquire\_seconds\_count, db\_connections\_acquire\_seconds\_sum**

Суммарное время установления соединения.

Тип метрики – summary.

#### **db\_connections\_max**

Максимальное количество соединений.

Тип метрики – gauge.

#### **db\_connections\_idle**

Количество неактивных соединений.

Тип метрики – gauge.

#### **db\_connections\_pending**

Количество ожидающих соединений.

Тип метрики – gauge.

### 7.1.3. Метрики соединения с RabbitMQ

#### **rabbitmq\_not\_acknowledged\_published\_total**

Общее количество неподтвержденных публикаций.

Тип метрики – counter.

#### **rabbitmq\_unrouted\_published\_total**

Общее количество неизвестных соединений.

Тип метрики – counter.

#### **rabbitmq\_channels**

Количество каналов соединений.

Тип метрики – gauge.

#### **rabbitmq\_consumed\_total**

Общее количество потребленных сообщений.

Тип метрики – counter.

#### **rabbitmq\_acknowledged\_total**

Количество успешных соединений.

Тип метрики – counter.

#### **rabbitmq\_failed\_to\_publish\_total**

Общее количество неуспешных публикаций.

Тип метрики – counter.

#### **rabbitmq\_rejected\_total**

Общее количество отклоненных соединений.

Тип метрики – counter.

#### **rabbitmq\_connections**

Количество соединений.

Тип метрики – gauge.

#### **rabbitmq\_acknowledged\_published\_total**

Общее количество подтвержденных публикаций.

Тип метрики – counter.

#### **rabbitmq\_published\_total**

Общее количество публикаций.

Тип метрики – counter.

### 7.1.4. Метрики HTTP-сервера

Метрики используются только в модуле DGS\_API.

Каждая метрика дополняется ярлыками uri, exception, outcome, status и method.

#### **http\_server\_requests\_seconds\_count, http\_server\_requests\_seconds\_sum**

Суммарное количество вызовов целевого объекта.

Тип метрики – summary.

#### **http\_server\_requests\_seconds\_max**

Максимальное время вызова целевого объекта.

Тип метрики – gauge.

### 7.1.5. Параметры работы GarbageCollector

В подразделе приведено описание метрик специального процесса, называемого сборщиком мусора (GarbageCollector), который периодически освобождает память, удаляя из неё ставшие ненужными объекты. Сборка мусора в программировании — одна из форм автоматического управления памятью.

#### **jvm\_threads\_peak\_threads**

Пиковое количество активных потоков с момента запуска виртуальной машины Java или сброса пикового значения.

Тип метрики – gauge.

### **jvm\_memory\_max\_bytes**

Максимальный объем памяти, доступный для управления, в байтах.

Тип метрики – gauge.

### **jvm\_buffer\_total\_capacity\_bytes**

Оценка общей емкости буферов в данном пуле.

Тип метрики – gauge.

### **jvm\_gc\_live\_data\_size\_bytes**

Размер долгоживущего пула динамической памяти после высвобождения.

Тип метрики – gauge.

### **vm\_threads\_daemon\_threads**

Текущее количество активных потоков демона.

Тип метрики – gauge.

### **jvm\_gc\_memory\_promoted\_bytes\_total**

Количество сборок мусора, приведших к увеличению области памяти для Old Generation.

Тип метрики – counter.

### **jvm\_gc\_memory\_allocated\_bytes\_total**

Количество произведенных сборок мусора в Young Generation.

Тип метрики – counter.

### **jvm\_threads\_live\_threads**

Количество рабочих потоков.

Тип метрики – gauge.

### **jvm\_classes\_loaded\_classes**

Количество классов, которые в данный момент загружены в виртуальную машину Java.

Тип метрики – gauge.

### **jvm\_threads\_states\_threads**

Количество потоков.

Метрика имеет ярлык `state` – состояние потока; возможные значения:

- `runnable` – запущен;
- `blocked` – заблокирован;
- `waiting` – в ожидании;
- `timed-waiting` – в ожидании по времени;
- `new` – новый;
- `terminated` – удален.

Тип метрики – gauge.

### **jvm\_gc\_max\_data\_size\_bytes**

Максимальный размер долгоживущего пула динамической памяти.

Тип метрики – `gauge`.

### **jvm\_buffer\_memory\_used\_bytes**

Оценка памяти, которую виртуальная машина Java использует для данного пула буферов.

Тип метрики – `gauge`.

### **jvm\_memory\_used\_bytes**

Объем используемой памяти (RAM).

Имеет ярлыки:

- `area` – область памяти; возможные значения: `nonheap`, `heap`;
- `id` – идентификатор пространства.

Тип метрики – `gauge`.

### **jvm\_gc\_pause\_seconds\_count, jvm\_gc\_pause\_seconds\_sum**

Количество сборок мусора.

Имеет ярлык `action` – причина запуска сборщика.

Тип метрики – `summary`.

### **jvm\_gc\_pause\_seconds\_max**

Максимальное время сборки мусора.

Имеет ярлык `action` – причина запуска сборщика.

Тип метрики – `gauge`.

### **jvm\_buffer\_count\_buffers**

Оценка количества буферов в пуле.

Тип метрики – `gauge`.

### **jvm\_memory\_committed\_bytes**

Объем памяти, выделенный для использования виртуальной машиной Java, в байтах.

Тип метрики – `gauge`.

### **jvm\_classes\_unloaded\_classes\_total**

Общее количество классов, выгруженных с момента запуска виртуальной машины Java.

Тип метрики – `counter`.

## 7.1.6. Прочие метрики

### **system\_cpu\_usage**

«Недавнее использование процессора» для всей системы.

Тип метрики – `gauge`.

### **system\_load\_average\_1m**

Сумма количества исполняемых сущностей, поставленных в очередь на доступные процессоры, и среднего количества исполняемых сущностей, работающих на доступных процессорах, за определенный период времени.

Тип метрики – gauge.

#### **process\_uptime\_seconds**

Время работы приложения с момента запуска.

Тип метрики – gauge.

#### **process\_files\_open\_files**

Количество открытых файлов.

Тип метрики – gauge.

#### **process\_cpu\_usage**

Данные об использовании процессора виртуальной машиной Java.

Тип метрики – gauge.

#### **system\_cpu\_count**

Количество процессорных ядер, доступных Java-машине.

Тип метрики – gauge.

#### **process\_start\_time\_seconds**

Дата и время запуска приложения в формате Unix-времени.

Тип метрики – gauge.

#### **process\_files\_max\_files**

Максимальное количество файловых дескрипторов.

Тип метрики – gauge.

## 7.2. Мониторинг модулей REPORT\_ENGINE

Компонент `REPORT_ENGINE` предоставляет статистические сведения о своей работе средствами протокола JMX.

Для просмотра метрик JMX используйте приложение `jconsole`, входящее в состав JDK (Java Development Kit) – [Рис. 8](#).

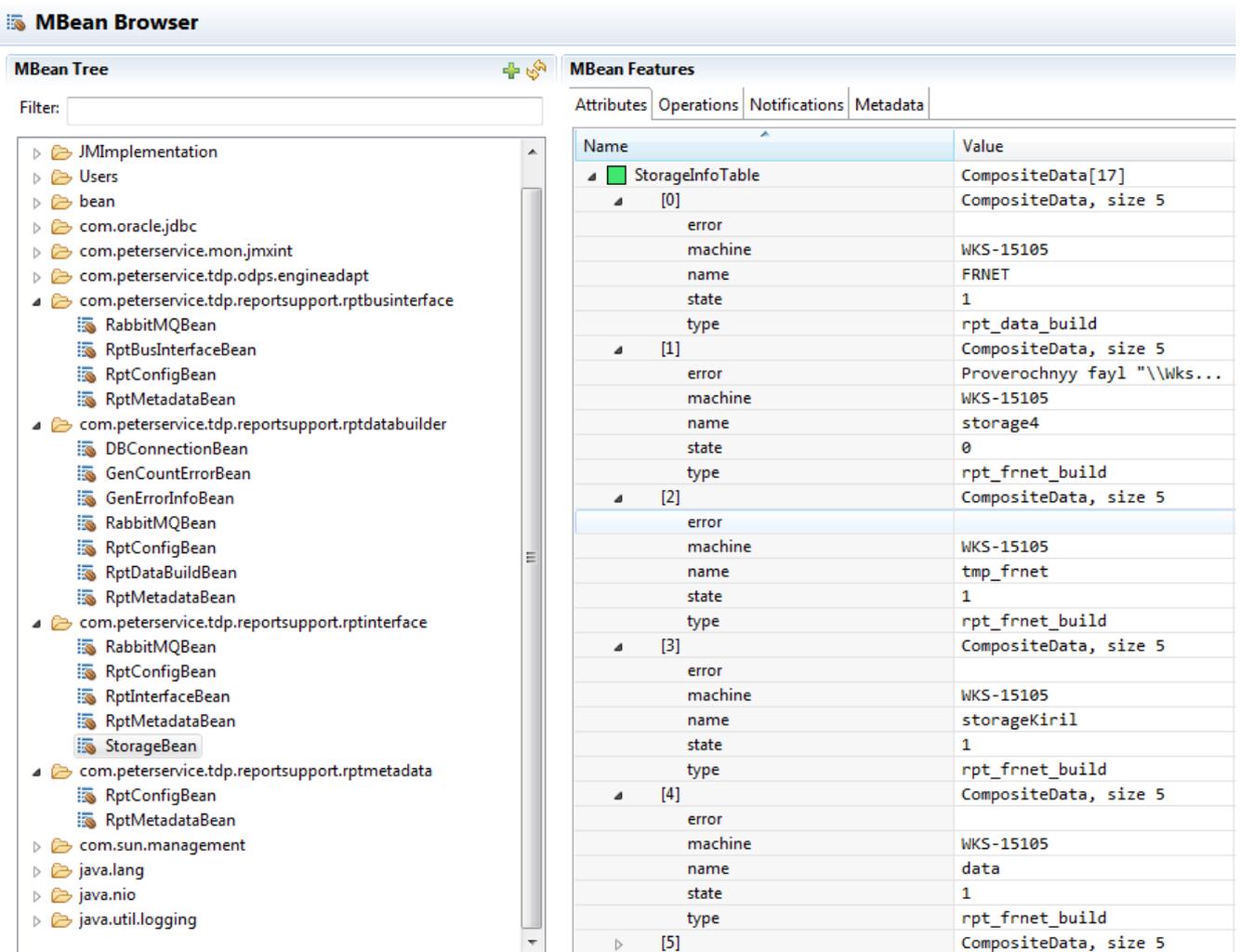


Рис. 8. Просмотр JMX-параметров мониторинга

Метрики мониторинга статуса и времени работы сервиса объединены в одну таблицу.

Параметры мониторинга разделены на три группы по уровню критичности их показателей:

- **Level1** (`subsystemParamLevel1Group`) – набор параметров мониторинга, критичных для работы приложения, где `subsystem` – значение на основе префикса модуля (например, `rptConfig`);
- **Level2** (`subsystemParamLevel2Group`) – параметры, накапливающие количество ошибок, уровень критичности средний;
- **Level3** (`subsystemParamLevel3Group`) – статистические параметры, уровень критичности низкий.

### 7.2.1. Мониторинг состояния модуля

| Свойство               | Значение  |
|------------------------|---|
| Наименование параметра | <code>subsystem State</code> , где <code>subsystem</code> – значение на основе префикса модуля <code>REPORT_ENGINE</code> (например, <code>rptConfig</code> ) |

| Свойство                            | Значение   |
|-------------------------------------|--|
| Назначение параметра                | Отслеживание состояния модуля.<br>Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра                  | Integer32  |
| Интервал опроса по умолчанию        | 1 минута   |
| Пороги по умолчанию                 | <ul style="list-style-type: none"> <li>• 0 – сервис не запущен;</li> <li>• 1 – сервис запущен</li> </ul>   |
| Действия в случае превышения порога | При появлении значения "0" система мониторинга формирует оповещение, текст по умолчанию: <b>"Service SUBSYSTEM is not started"</b> , где SUBSYSTEM – префикс модуля REPORT_ENGINE, например, RPT_CONFIG  |
| Уровень критичности параметра       | Level1 (высокий)   |
| Примечание                          | Запрос не кешируется   |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=SUBSYSTEMBean (cols=State),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptconfig:name=RptConfigBean) |

## 7.2.2. Мониторинг времени работы сервиса

| Свойство                     | Значение  |
|------------------------------|---|
| Наименование параметра       | subsystemUptime,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptConfig)    |
| Назначение параметра         | Отслеживание времени работы сервиса.<br>Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра           | TimeTicks   |
| Интервал опроса по умолчанию | 1 минута  |

| Свойство                      | Значение  |
|-------------------------------|---|
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=SUBSYSTEMBean (cols=Uptime),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptconfig:name=RptConfigBean) |

### 7.2.3. Мониторинг соединения с RabbitMQ

Метрики мониторинга соединения с RabbitMQ объединены в одну таблицу.

#### Строка соединения с RabbitMQ

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | subsystemRabbitMQServer,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptDataBuild)  |
| Назначение параметра          | Отслеживание информации о строке соединения с сервером RabbitMQ. Параметр предназначен для автоматического опроса системой мониторинга   |
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RabbitMQBean (cols=Server),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=RabbitMQBean) |

## Статус соединения с RabbitMQ

| Свойство                            | Значение  |
|-------------------------------------|---|
| Наименование параметра              | subsystemRabbitMQState, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptDataBuild)   |
| Назначение параметра                | Отслеживание состояния соединения с сервером RabbitMQ. Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра                  | Integer32   |
| Интервал опроса по умолчанию        | 1 минута  |
| Пороги по умолчанию                 | <ul style="list-style-type: none"> <li>• 0 – сервис RabbitMQ недоступен;</li> <li>• 1 – сервис RabbitMQ доступен</li> </ul>   |
| Действия в случае превышения порога | При появлении значения "0" система мониторинга формирует оповещение, текст по умолчанию: <i>"Service RabbitMQ is not available for SUBSYSTEM"</i> , где SUBSYSTEM – префикс модуля REPORT_ENGINE, например, RPT_DATA_BUILD                          |
| Уровень критичности параметра       | Level1 (высокий)  |
| Примечание                          | Запрос не кешируется  |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RabbitMQBean (cols=State), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=RabbitMQBean) |

## Сообщение об ошибке при попытке соединения с RabbitMQ

| Свойство               | Значение  |
|------------------------|---|
| Наименование параметра | subsystemRabbitMQError, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptDataBuild)                                   |
| Назначение параметра   | Отслеживание сообщений об ошибках при попытке соединения с сервером RabbitMQ. Параметр предназначен для автоматического опроса системой мониторинга |

| Свойство                      | Значение  |
|-------------------------------|---|
| SNMP тип параметра            | DisplayString   |
| Интервал опроса по умолчанию  | 1 минута  |
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RabbitMQBean (cols=Error), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=RabbitMQBean) |

## 7.2.4. Мониторинг Storages

Метрики с информацией о соединениях с хранилищами отчетов объединяются в одну таблицу.

### Информация об имени хранилища файлов

| Свойство                      | Значение  |
|-------------------------------|---|
| Наименование параметра        | subsystemStorageName, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface) |
| Назначение параметра          | Имя хранилища файлов.<br>Параметр предназначен для автоматического опроса системой мониторинга                  |
| SNMP тип параметра            | DisplayString   |
| Интервал опроса по умолчанию  | 1 минута  |
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |

| Свойство | Значение   |
|----------|--|
| API      | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=StorageBean (cols=Name), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptinterface:name=StorageBean) |

### Информация о компьютере, где работает модуль

| Свойство                      | Значение  |
|-------------------------------|---|
| Наименование параметра        | subsystemStorageMachine, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)  |
| Назначение параметра          | Имя компьютера, где работает модуль.<br>Параметр предназначен для автоматического опроса системой мониторинга   |
| SNMP тип параметра            | DisplayString   |
| Интервал опроса по умолчанию  | 1 минута  |
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=StorageBean (cols=Machine), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptinterface:name=StorageBean) |

### Информация о типе проверяемого модуля

| Свойство               | Значение  |
|------------------------|---|
| Наименование параметра | subsystemStorageType, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface) |
| Назначение параметра   | Тип проверяемого модуля.<br>Параметр предназначен для автоматического опроса системой мониторинга               |

| Свойство                      | Значение  |
|-------------------------------|---|
| SNMP тип параметра            | DisplayString   |
| Интервал опроса по умолчанию  | 1 минута  |
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=StorageBean (cols=Type),<br>где subsystem – значение на основе префикса модуля продукта (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=StorageBean) |

### Статус работы модуля с хранилищами файлов

| Свойство                            | Значение   |
|-------------------------------------|--|
| Наименование параметра              | subsystem StorageState, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)                        |
| Назначение параметра                | Отслеживание состояния соединения модуля с хранилищем файлов. Параметр предназначен для автоматического опроса системой мониторинга      |
| SNMP тип параметра                  | Integer32  |
| Интервал опроса по умолчанию        | 1 минута   |
| Пороги по умолчанию                 | <ul style="list-style-type: none"> <li>• 0 – некорректная работа;</li> <li>• 1 – корректная работа</li> </ul>                            |
| Действия в случае превышения порога | При появлении значения "0" система мониторинга формирует оповещение, текст по умолчанию: <i>"Service doesn't have access to storage"</i> |
| Уровень критичности параметра       | Level1 (высокий)   |

| Свойство   | Значение   |
|------------|--|
| Примечание | Запрос не кешируется   |
| API        | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=StorageBean (cols=State),<br>где subsystem – значение на основе префикса компонента продукта (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=StorageBean) |

### Информация об ошибке доступа к хранилищу файлов

| Свойство                      | Значение  |
|-------------------------------|---|
| Наименование параметра        | subsystem StorageError,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)  |
| Назначение параметра          | Отслеживание ошибок доступа к хранилищу файлов.<br>Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра            | DisplayString   |
| Интервал опроса по умолчанию  | 1 минута  |
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=StorageBean (cols=Error),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptInterface:name=StorageBean) |

### 7.2.5. Мониторинг соединений с серверами конфигурации и метаданных (внутреннее взаимодействие)

Информация о сервере конфигурации и сервере метаданных разносится в отдельные таблицы.

## Информация о доступности серверов конфигурации и метаданных

| Свойство                            | Значение  |
|-------------------------------------|---|
| Наименование параметра              | [RptConfig RptMetadata]State  |
| Назначение параметра                | Отслеживание состояния соединения с сервером конфигурации или метаданных.<br>Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра                  | Integer32   |
| Интервал опроса по умолчанию        | 1 минута  |
| Пороги по умолчанию                 | <ul style="list-style-type: none"> <li>• 0 – сервер конфигурации метаданных недоступен;</li> <li>• 1 – сервер конфигурации метаданных доступен</li> </ul>   |
| Действия в случае превышения порога | При появлении значения "0" система мониторинга формирует оповещение, текст по умолчанию: <i>"Service [RPT_CONFIG RPT_METADATA] is not available"</i>  |
| Уровень критичности параметра       | Level1 (высокий)  |
| Примечание                          | Запрос не кешируется  |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=[RptConfig RptMetadata]Bean (cols=State),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=[RptConfig RptMetadata]Bean) |

## Информация о сервере конфигурации или метаданных

| Свойство               | Значение  |
|------------------------|---|
| Наименование параметра | [RptConfig RptMetadata]Server   |
| Назначение параметра   | Адрес сервера конфигурации метаданных.<br>Параметр предназначен для автоматического опроса системой мониторинга |

| Свойство                      | Значение   |
|-------------------------------|--|
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=[RptConfig RptMetadata]Bean (cols=Server),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=[RptConfig RptMetadata]Bean) |

### Информация о причине недоступности сервера конфигурации или метаданных

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | [RptConfig RptMetadata]Error   |
| Назначение параметра          | Причина недоступности сервера конфигурации метаданных (ошибка при попытке получить информацию о сервере).<br>Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |

| Свойство | Значение  |
|----------|---|
| API      | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=[RptConfig RptMetadata]Bean (cols=Error),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=[RptConfig RptMetadata]Bean) |

## 7.2.6. Мониторинг ошибок построения данных

В разделе приведены метрики для отслеживания ошибок при построении данных.

### Количество ошибок за последний опрос

| Свойство                            | Значение   |
|-------------------------------------|--|
| Наименование параметра              | rptDataBuildGenCountError  |
| Назначение параметра                | Отслеживание количества ошибок построения данных с момента последнего опроса.<br>Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра                  | Counter32  |
| Интервал опроса по умолчанию        | 10 минут   |
| Пороги по умолчанию                 | >= 10 – количество ошибок превышает допустимый порог   |
| Действия в случае превышения порога | При появлении значения >= 10 система мониторинга формирует оповещение, текст по умолчанию: <i>"The number of errors exceeds the limit"</i>             |
| Уровень критичности параметра       | Level1 (высокий)   |
| Примечание                          | Запрос не кешируется   |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=GenCountErrorBean (cols=CountError)  |

## Информация о порядковом номере ошибки

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | rptDataBuildGenErrorNumber   |
| Назначение параметра          | Порядковый номер ошибки.<br>Параметр предназначен для автоматического опроса системой мониторинга    |
| SNMP тип параметра            | Counter32  |
| Интервал опроса по умолчанию  | 10 минут   |
| Уровень критичности параметра | Level1 (высокий)   |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=GenErrorBean (cols=Number) |

## Информация о времени ошибки

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | rptDataBuildGenErrorTime   |
| Назначение параметра          | Время появления ошибки.<br>Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 10 минут   |
| Уровень критичности параметра | Level1 (высокий)   |
| Примечание                    | Запрос не кешируется   |

| Свойство | Значение   |
|----------|--|
| API      | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=GenErrorBean (cols=Time) |

### Информация об ошибке

| Свойство                      | Значение  |
|-------------------------------|---|
| Наименование параметра        | rptDataBuildGenError  |
| Назначение параметра          | Информация об ошибке.<br>Параметр предназначен для автоматического опроса системой мониторинга      |
| SNMP тип параметра            | DisplayString   |
| Интервал опроса по умолчанию  | 10 минута   |
| Уровень критичности параметра | Level1 (высокий)  |
| Примечание                    | Запрос не кешируется  |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.rptdatabuilder:name=GenErrorBean (cols=Error) |

### 7.2.7. Мониторинг производительности системы

Метрики оценки производительности системы – это метрики количества обработанных заявок и метрики среднего времени формирования каждого отчета в отдельности.

Метрики реализованы в каждом из нагруженных модулей REPORT\_ENGINE.

#### Количество обрабатываемых заявок на формирование отчета

| Свойство               | Значение   |
|------------------------|--|
| Наименование параметра | subsystem RequestsProcessing,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface) |

| Свойство                      | Значение   |
|-------------------------------|--|
| Назначение параметра          | Количество заявок на формирование отчетов, обрабатываемых модулем REPORT_ENGINE в момент опроса.<br>Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра            | Unsigned32   |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Requests (cols=Processing), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptinterface:name=Requests) |
| Сценарий использования        | Для информации   |

### Количество успешно обработанных заявок на формирование отчета

| Свойство                      | Значение  |
|-------------------------------|---|
| Наименование параметра        | subsystemRequestsSucceeded,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)                          |
| Назначение параметра          | Количество заявок на формирование отчетов, успешно обработанных модулем.<br>Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра            | Counter32   |
| Интервал опроса по умолчанию  | 1 минута  |
| Уровень критичности параметра | Level3 (низкий)   |

| Свойство               | Значение  |
|------------------------|---|
| Примечание             | Запрос не кешируется  |
| API                    | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Requests (cols=Succeeded),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=Requests) |
| Сценарий использования | Для информации  |

### Количество неуспешно обработанных заявок на формирование отчета

| Свойство                            | Значение   |
|-------------------------------------|--|
| Наименование параметра              | subsystemRequestsFailed,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)                                  |
| Назначение параметра                | Количество заявок на формирование отчетов, завершённых с ошибкой. Параметр предназначен для автоматического опроса системой мониторинга                |
| SNMP тип параметра                  | Counter32  |
| Интервал опроса по умолчанию        | 1 минута   |
| Пороги по умолчанию                 | <ul style="list-style-type: none"> <li>в RPT_INTERFACE <math>\geq 1</math>;</li> <li>в RPT_DATA_BUILD и RPT_FRNET_BUILD порог не установлен</li> </ul> |
| Действия в случае превышения порога | При превышении порога система мониторинга формирует оповещение, текст по умолчанию: « <i>Report generation error</i> ». Уровень важности: «error»      |
| Уровень критичности параметра       | Level2 (средний)   |
| Примечание                          | Запрос не кешируется   |

| Свойство               | Значение   |
|------------------------|--|
| API                    | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Requests (cols=Failed), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptinterface:name=Requests) |
| Сценарий использования | Если произошла ошибка генерации отчета из массовой печати, будет выставлен флаг  |

## 7.2.8. Общее количество обработанных заявок на формирование отчетов

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | subsystem RequestsTotal,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)  |
| Назначение параметра          | Общее количество заявок на формирование отчетов в модуле REPORT_ENGINE.<br>Параметр предназначен для автоматического опроса системой мониторинга   |
| SNMP тип параметра            | Counter32  |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Requests (cols=Total),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptinterface:name=Requests) |
| Сценарий использования        | Для информации   |

## Наименование отчета

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | subsystem ReportName, где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)  |
| Назначение параметра          | Наименование отчета.<br>Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Reports (cols=name),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=Reports) |
| Сценарий использования        | Для информации   |

## Среднее время формирования отчета

| Свойство               | Значение   |
|------------------------|--|
| Наименование параметра | subsystem ReportTime,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)                               |
| Назначение параметра   | Среднее время формирования отчета %REPORT_NAME% в модуле REPORT_ENGINE.<br>Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра     | TimeTicks  |

| Свойство                            | Значение   |
|-------------------------------------|--|
| Интервал опроса по умолчанию        | 1 минута   |
| Пороги по умолчанию                 | <ul style="list-style-type: none"> <li>в RPT_INTERFACE &gt;= 120 с;</li> <li>в RPT_DATA_BUILD и RPT_FRNET_BUILD порог не установлен</li> </ul>   |
| Действия в случае превышения порога | При превышении порога система мониторинга формирует оповещение, текст по умолчанию: « <i>Report generation time is exceeded</i> ». Уровень важности: «warning»   |
| Уровень критичности параметра       | Level2 (средний)   |
| Примечание                          | Запрос не кешируется   |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Reports (cols=time), где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, ds=com.peterservice.tdp.reportsupport.rptinterface:name=Reports) |
| Сценарий использования              | Если время генерации отчета из массовой печати превысит две минуты, будет выставлен флаг   |

### Количество запросов отчета

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | subsystem ReportCalls,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface)                    |
| Назначение параметра          | Количество запросов отчета %REPORT_NAME% в модуле REPORT_ENGINE. Параметр предназначен для автоматического опроса системой мониторинга |
| SNMP тип параметра            | Unsigned32   |
| Интервал опроса по умолчанию  | 1 минута   |
| Уровень критичности параметра | Level3 (низкий)  |

| Свойство                  | Значение   |
|---------------------------|--|
| Примечание                | Запрос не кешируется   |
| API                       | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Re<br>ports (cols=calls),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE<br>(например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=<br>Reports) |
| Сценарий<br>использования | Для информации   |

## 7.2.9. Мониторинг узлов SFTP-кластера

Описание метрик для оценки текущего состояния узлов SFTP-кластера.

### Имя удалённого кластера SFTP-серверов

| Свойство                            | Значение   |
|-------------------------------------|--|
| Наименование<br>параметра           | subsystem RemoteStorageCluster,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE<br>(например, rptInterface)  |
| Назначение<br>параметра             | Имя удалённого кластера SFTP-серверов.<br>Параметр предназначен для автоматического опроса системой<br>мониторинга   |
| SNMP тип<br>параметра               | DisplayString  |
| Интервал опроса<br>по умолчанию     | 600 секунд   |
| Уровень<br>критичности<br>параметра | Level3 (низкий)  |
| Примечание                          | Запрос не кешируется   |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=Re<br>moteStorage (cols=Cluster),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE<br>(например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=Clu<br>ster) |

| Свойство               | Значение       |
|------------------------|----------------|
| Сценарий использования | Для информации |

### Хост удалённого кластера SFTP-серверов

| Свойство                      | Значение  |
|-------------------------------|---|
| Наименование параметра        | <code>subsystem RemoteStorageHost</code> ,<br>где <code>subsystem</code> – значение на основе префикса модуля <code>REPORT_ENGINE</code> (например, <code>rptInterface</code> )   |
| Назначение параметра          | Имя хоста удалённого кластера SFTP-серверов.<br>Параметр предназначен для автоматического опроса системой мониторинга   |
| SNMP тип параметра            | <code>DisplayString</code>  |
| Интервал опроса по умолчанию  | 600 секунд  |
| Уровень критичности параметра | Level3 (низкий)   |
| Примечание                    | Запрос не кешируется  |
| API                           | <code>GET-REQ-JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RemoteStorage(cols=Host)</code> ,<br>где <code>subsystem</code> – значение на основе префикса модуля <code>REPORT_ENGINE</code> (например, <code>ds=com.peterservice.tdp.reportsupport.rptinterface:name=Host</code> ) |
| Сценарий использования        | Для информации  |

### Тип удалённого кластера SFTP-серверов

| Свойство               | Значение   |
|------------------------|--|
| Наименование параметра | <code>subsystemRemoteStorageType</code> ,<br>где <code>subsystem</code> – значение на основе префикса модуля <code>REPORT_ENGINE</code> (например, <code>rptInterface</code> ) |

| Свойство                      | Значение   |
|-------------------------------|--|
| Назначение параметра          | Тип удалённого кластера SFTP-серверов.<br>Параметр предназначен для автоматического опроса системой мониторинга  |
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 600 секунд   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |
| API                           | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RemoteStorage(cols=Type),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=Type) |
| Сценарий использования        | Для информации   |

### Состояние доступа к SFTP-ресурсу

| Свойство                     | Значение   |
|------------------------------|--|
| Наименование параметра       | subsystem RemoteStorageState,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface) |
| Назначение параметра         | Состояние доступа к SFTP-ресурсу.<br>Параметр предназначен для автоматического опроса системой мониторинга                 |
| SNMP тип параметра           | Unsigned32   |
| Интервал опроса по умолчанию | 600 секунд   |
| Пороги по умолчанию          | <ul style="list-style-type: none"> <li>• =0 – SFTP-ресурс недоступен;</li> <li>• =1 – SFTP-ресурс доступен</li> </ul>      |

| Свойство                            | Значение  |
|-------------------------------------|---|
| Действия в случае превышения порога | <ul style="list-style-type: none"> <li>• При пересечении порога =0 система мониторинга формирует оповещение, текст по умолчанию: <i>"Remote storage is not available"</i>;</li> <li>• при пересечении порога =1 система мониторинга формирует оповещение, текст по умолчанию: <i>"Remote storage is available"</i></li> </ul> |
| Уровень критичности параметра       | Level1 (высокий)  |
| Примечание                          | Запрос не кешируется  |
| API                                 | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RemoteStorage(cols=State),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=State)  |
| Сценарий использования              | При пересечении порога “=0” система предупреждает о проблеме доступа к SFTP-серверу. Выясните причину недоступности ресурса и устраните её  |

### Ошибка соединения с кластером SFTP-серверов

| Свойство                      | Значение   |
|-------------------------------|--|
| Наименование параметра        | subsystem RemoteStorageError,<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например, rptInterface) |
| Назначение параметра          | Ошибка соединения с кластером SFTP-серверов.<br>Параметр предназначен для автоматического опроса системой мониторинга      |
| SNMP тип параметра            | DisplayString  |
| Интервал опроса по умолчанию  | 600 секунд   |
| Уровень критичности параметра | Level3 (низкий)  |
| Примечание                    | Запрос не кешируется   |

| Свойство               | Значение   |
|------------------------|--|
| API                    | GET-REQ-<br>JMX:ds=com.peterservice.tdp.reportsupport.subsystem:name=RemoteStorage(cols=Error),<br>где subsystem – значение на основе префикса модуля REPORT_ENGINE (например,<br>ds=com.peterservice.tdp.reportsupport.rptinterface:name=Error) |
| Сценарий использования | Для информации   |