

О руководстве

Quick start

- Вход в программу
- Создание нового проекта
- Создание и настройка трансформации
 - Просмотр результата преобразования данных
- Создание и настройка потока работ
- Инструменты запуска приложений и мониторинга в Neoflex Datagram

Работа с программой

Подключение Neoflex Datagram к базам данных внешних систем

- Объекты "JDBC Connection"
- Объекты "Software System"
- Объекты "Deployment"
- Объекты "Scheme"
- Объекты "Jdbc Context"

Подключение Neoflex Datagram к исполняющим средам

- Объекты "Oozie"
- Объекты "Workflow Deployment"
- Объекты "Livy Server"
- Объекты "Transformation Deployment"
- Объекты "Coordinator Deployment"
- Мастер импорта. Настройка и создание

Работа в мастере импорта

Трансформация данных

- Объекты "Transformation"
- Элементы диаграмм трансформаций
- Группа элементов SOURCES

Local source

CSV source

XML source

Avro source

Expression source

SQL source

Hive source

HBase source

Table source

Kafka source

Группа элементов DATA TRANSFORM

Join

Aggregation

Selection

Projection

Sequence

Sort

Group with state

Union

Drools

Model based analysis

Spark SQL

Explode fields

Группа элементов TARGETS

- Local target*
- Table target*
- Procedure target*
- CSV target*
- XML target*
- Streaming target*
- Hive target*
- HBase target*
- Kafka target*

- Агрегатные пользовательские функции
- Объекты "Scheme data set"

- Поток работ

- Объекты "Workflow"
- Элементы диаграмм объектов Workflow
- Группа элементов POINT

- Start*
- End*
- Kill*

- Группа элементов RULE

- Fork*
- Join*
- Decision*

- Группа элементов ACTION

- Transformation*
- Workflow*
- Execute shell*
- Execute Java*

- Запуск на исполнение настроенных объектов Transformation и Workflow

- Запуск исполнения настроенного объекта Transformation
- Запуск исполнения настроенного объекта Workflow
- Запуск исполнения настроенного объекта Workflow по расписанию
- Объекты CoJob

- Подсистема переноса метаданных

- Объекты Project. Группировка объектов программы
- Операции экспорта/импорта метаданных

- Адаптация подсистемы Meta Server для работы с разворачиваемым репозиторием

- Объекты Environment

- Sandbox. Инструмент анализа данных

- Создание объекта Cluster

- Workspace. Рабочее пространство для анализа данных

- Наполнение Workspace
 - Импорт схемы в Jdbc workspace*
 - Импорт датасетов*
 - Создание датасетов и child workspaces*

- Виды датасетов

- Dataset*
- Hive dataset*
- Hive external dataset*
- Jdbc dataset*
- Jdbc table dataset*
- Linked dataset*
- Reference dataset*
- Notebook*

Работа с notebook

Работа с датасетами

Просмотр данных датасета

Просмотр метаданных датасета

Редактирование атрибутов датасета

Business process

Дополнительные возможности программы

Подключение к серверу Zeppelin

Синхронизация с сервером Atlas

Перенос данных из CSV файлов в базу данных внешней системы

Объекты Staging Area

Запуск операций объектов по расписанию

Streaming. Поточковая обработка данных

Объекты Events processor

Настройка потоков данных и правил анализа

Объекты Function

SLA мониторинг задач Oozie

Приложения

Приложение 1. Соответствие типов полей в дизайнера трансформаций классам языка Scala

Приложение 2. Встроенные функции редактора выражений

Приложение 3. API Meta Server

О руководстве

В руководстве пользователя описаны возможности Neoflex Datagram по разработке приложений для преобразования данных.

В главе «[Quick start](#)» представлен пример основных сценариев работы с программой:

- Запуск Neoflex Datagram;
- Создание нового проекта;
- Проектирование схемы преобразования данных;
- Создание потока управления преобразованием данных и запуск приложения;
- Контроль исполнения приложения и просмотр результата.

В главе «[Работа с программой](#)» описаны:

- Подготовка запуска приложения;
- Интерфейс Neoflex Datagram;
- Подключение Neoflex Datagram к базам данных внешних систем и настройки взаимодействия с исполняющими средами;
- Элементы диаграмм трансформаций и потоков работ;
- Запуск исполнения объектов Workflow по расписанию;
- Дополнительные возможности программы.

В связи с непрерывно ведущимися работами по усовершенствованию программы, следует отметить, что описание программы может отличаться от того, что Вы увидите на экране.

Quick start

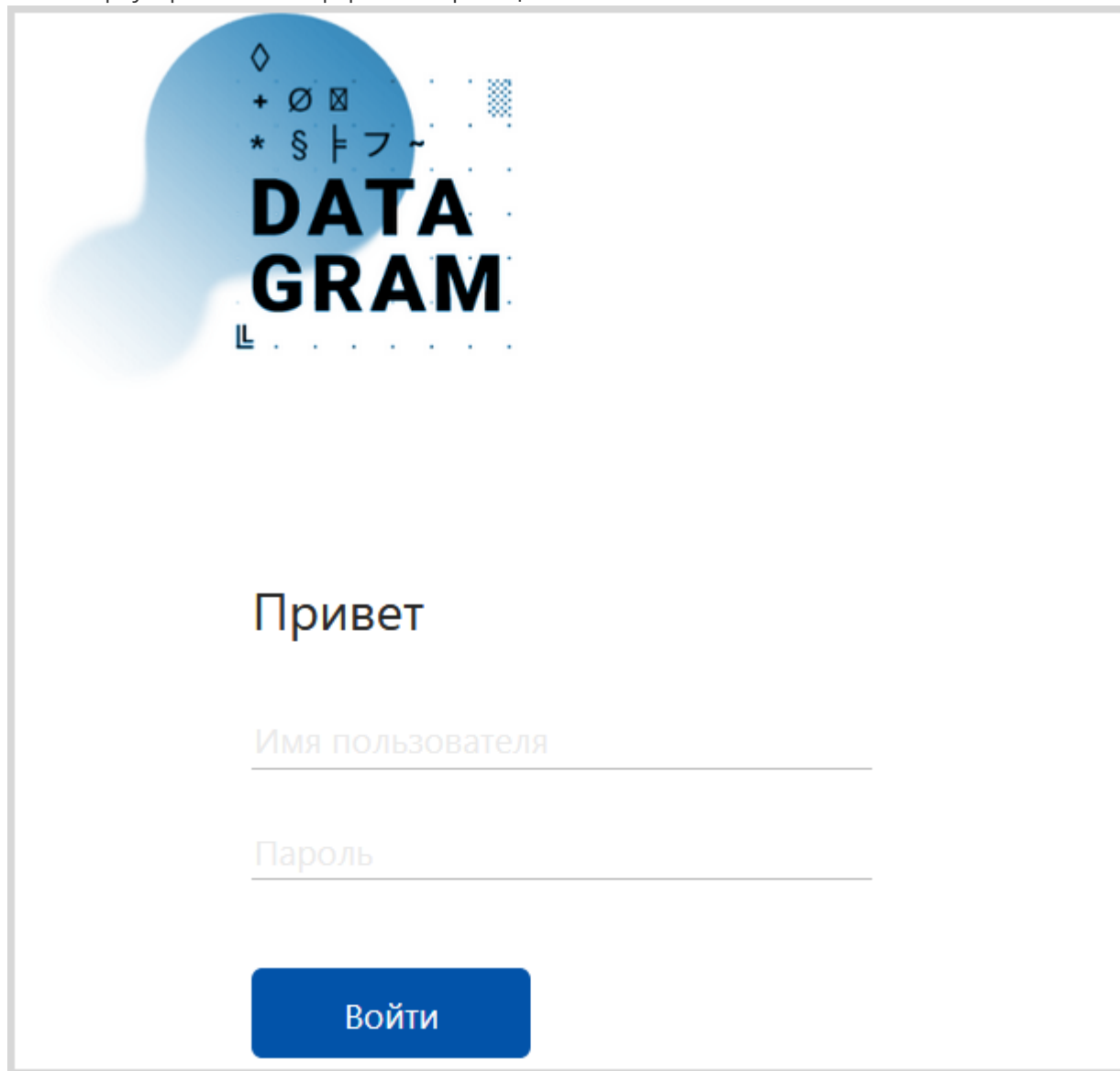
Вход в программу

Запустите браузер и в адресной строке введите:

<http://host:port/cim/ddesigner/build/index.html?>

,где **host** - хост сервера, на котором установлена программа, **port** - номер порта сервера.

В окне браузера появится форма авторизации пользователя.



DATA
GRAM

Привет

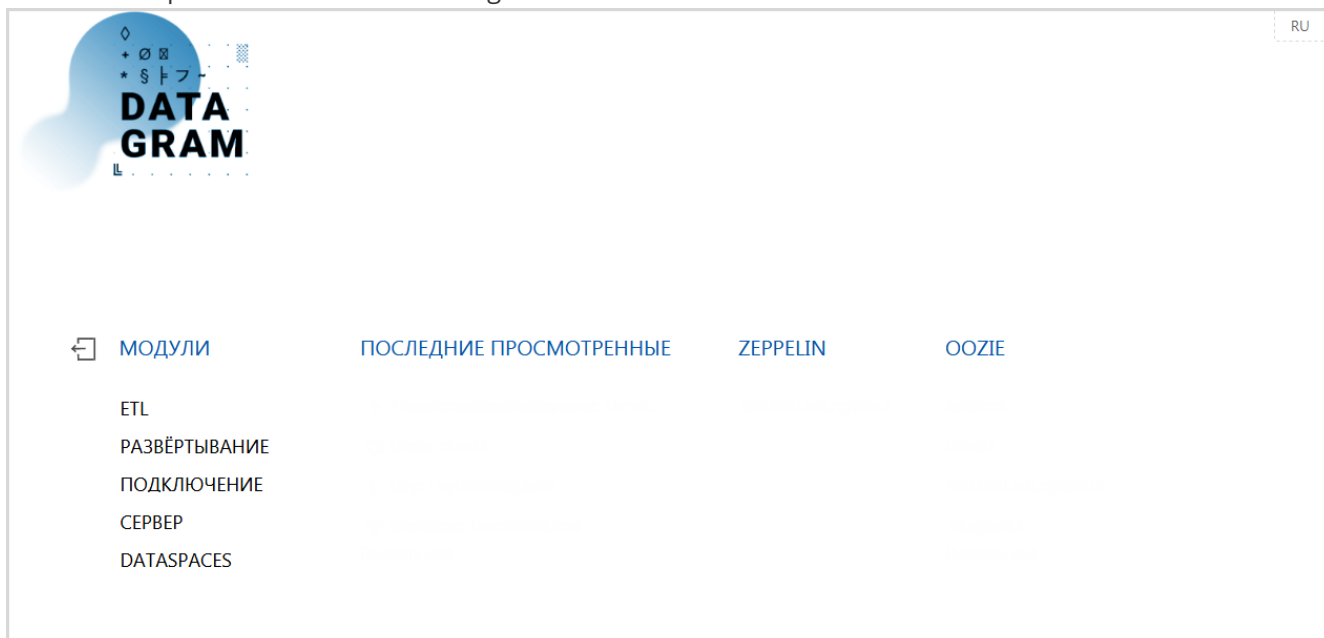
Имя пользователя

Пароль

Войти

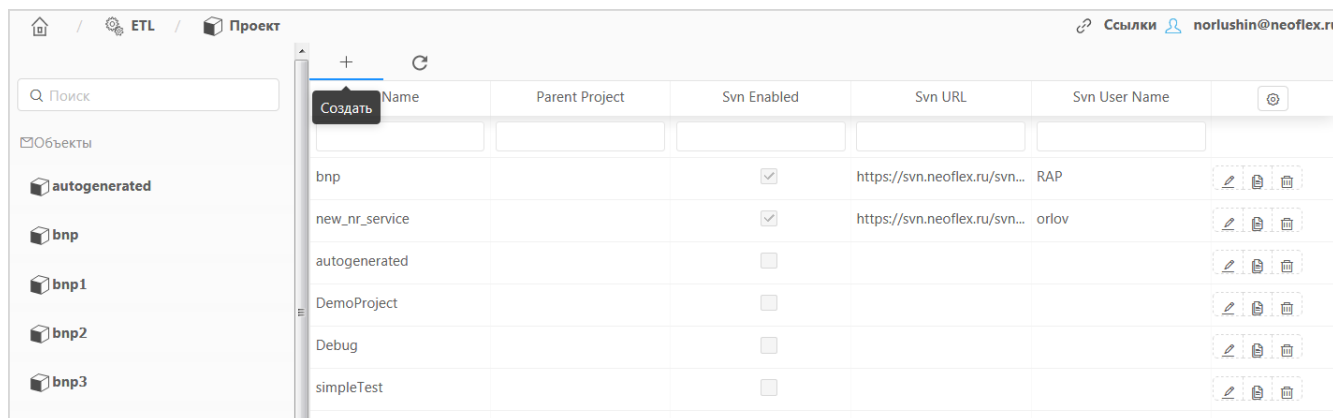
Для входа в программу укажите имя пользователя, пароль и нажмите кнопку **«Вход»**. На экране

появится стартовое окно Neoflex Datagram.



Создание нового проекта

В разделе интерфейса **«ETL/Project»** создайте объект **Project** с названием **«DemoProject»** (остальные поля оставьте пустыми).



Примечание.

Атрибуты объектов **Project** описаны в разделе руководства [«Группировка объектов программы»](#).

Объект **«DemoProject»** необходимо создать, чтобы в дальнейшем привязать к нему объекты **Transformation** и **Workflow**, тем самым объединив их в группу. В последующем данную группу объектов можно будет переносить между программными средами.

Создание и настройка трансформации

Внимание!

Для выполнения действий, описанных далее, необходимо чтобы в программе были настроены [подключения к базе данных внешней системы](#) и [исполняющим средам](#).

В разделе рассмотрен пример создания объекта **Transformation**, для переноса данных из CSV файла в файлы формата JSON.

Предварительно, в файловой системе HDFS, создайте файл-источник данных Demo.txt:

1,Иванов,1000.00

2,Петров,1200.00

3,Сидоров,1250.00


В разделе **"ETL/Transformation"** создайте объект с атрибутами:

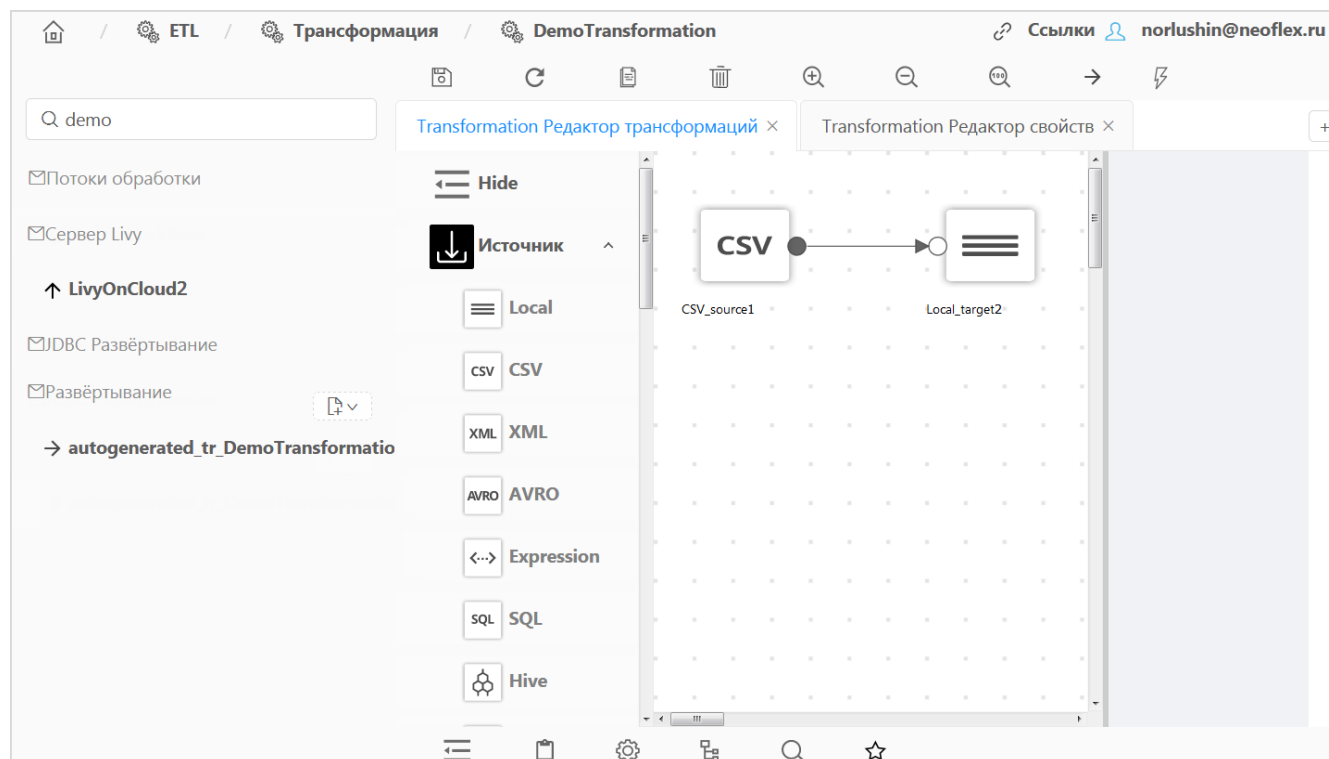
- **Name** - **DemoTransformation**;
- **Label** - **DemoLabelTransformation**;
- **Project** - из списка выберите ранее созданный объект **«DemoProject»**.

Примечание.

Атрибуты объектов **Transformation** описаны в разделе [«Объекты Transformation»](#).

Настройка и запуск трансформации

Кнопкой  откройте дизайнер трансформаций. В дизайнере трансформаций перетащите элементы **CSV source** и **Local target** в рабочую область и укажите направление процесса переноса данных, как показано на рисунке ниже.



На заметку.

При переносе элементов в рабочую область, программа автоматически задаст им названия.

Кликните по элементу «**CSV_source1**», чтобы открыть панель свойств.

Для элемента «**CSV_source1**» установите следующие настройки:



- В поле **Path** укажите путь к файлу, из которого будут переноситься данные (например: /user/Demo.txt);
- В группе **CSV** в поле **Delimiter** введите символ, который служит разделителем значений в файле-источнике (в примере это запятая);
- В группе **OUT PORT->Fields** добавьте три поля:
id - Integer
name - string
salary - decimal
- Для остальных настроек оставьте дефолтные значения.

Настройте элемент «**Local_target2**»:

- В списке **Local file format** выберите **JSON**;
- В поле **local file name** укажите каталог, в который будет записан результат трансформации (пример: /user/demo/demo.json);
- В поле **Input fields mapping** настройте соответствие полей:
ID - id
NAME - name
SALARY - salary
- В поле **Partitions** добавьте каталог **id**, который будет входить в каталог **demo.json**;
- Для остальных настроек оставьте значения, которые были заданы программой автоматически.

Примечание.

Элементы диаграмм трансформаций описаны в разделе «[Элементы диаграмм трансформаций](#)».

Сохраните настройки кнопкой . Откройте список операций кнопкой  и запустите операцию "**Проверить**" для автоматической проверки синтаксиса трансформации. Если действия инструкции выполнены правильно, то программа не обнаружит ошибок.

На заметку.

В случае обнаружения программой ошибок проверьте код трансформации, для этого воспользуйтесь [редактором исходного кода](#).

Выполните операцию "**Запустить**" для запуска исполнения трансформации. После запуска объекта на исполнение, в разделе интерфейса "**Развертывание/Transformation Deployment**" будет создан объект с названием «**autogenerated_tr_DemoTransformation**», который выполнит развертывание и запуск исполнения «**DemoTransformation**» на исполняющей среде Livy Server.

На заметку.


По кнопке  можно создать новый объект Transformation Deployment.

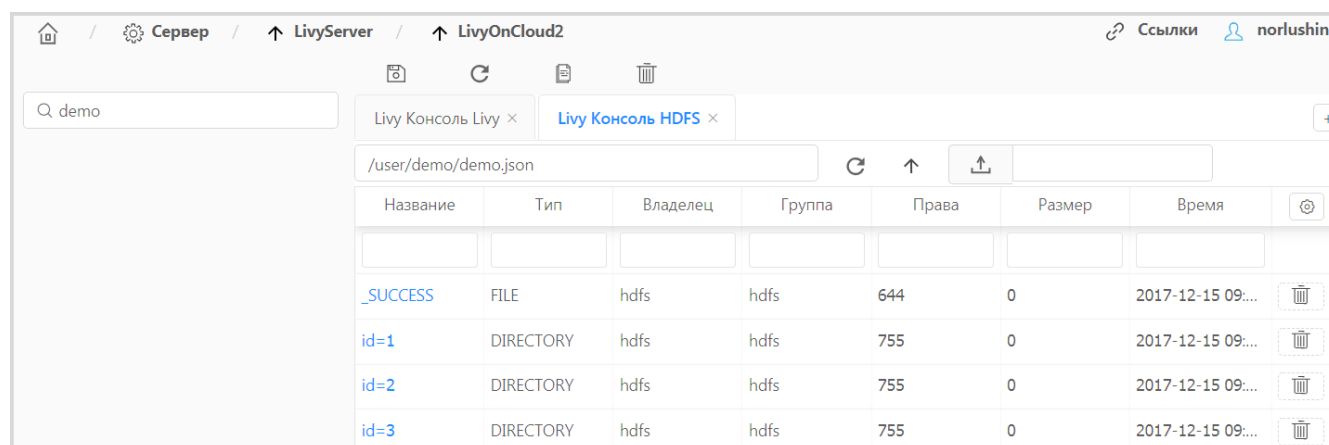
Атрибуты объектов **Transformation Deployment** описаны в разделе «[Объекты Transformation Deployment](#)».





Дождитесь окончания исполнения трансформации - на экране появится сообщение:

- В случае успешного исполнения трансформации, на экране появится окно с сообщением: «Ок».
- Если в ходе исполнения объекта возникнет ошибка, то на экране появится окно с текстом, описывающим ошибку.

Просмотр результата преобразования данных

Чтобы просмотреть файлы, полученные в результате исполнения трансформации, перейдите в интерфейс **консоли HDFS**. Для этого перейдите в раздел **"Сервер/Livy"**, в списке выберите сервер на котором разворачивалась трансформация. На открывшейся странице нажмите кнопку  и в списке выберите пункт **"Консоль HDFS"**. В консоли перейдите в каталог, который создан в результате выполнения трансформации (данный каталог указывался в настройках элемента "Local_target2" в поле "local file name").



Название	Тип	Владелец	Группа	Права	Размер	Время	
._SUCCESS	FILE	hdfs	hdfs	644	0	2017-12-15 09:...	
id=1	DIRECTORY	hdfs	hdfs	755	0	2017-12-15 09:...	
id=2	DIRECTORY	hdfs	hdfs	755	0	2017-12-15 09:...	
id=3	DIRECTORY	hdfs	hdfs	755	0	2017-12-15 09:...	

Создание и настройка потока работ

Создаваемый объект **Workflow** будет выполнять следующие действия:

1. Запускать трансформацию «DemoTransformation»;
 2. Анализировать ее выполнение:
- При успешном выполнении трансформации процесс будет завершен;
 - При обнаружении ошибки процесс будет прерван.


В разделе **"ETL/Workflow"** создайте объект с атрибутами:

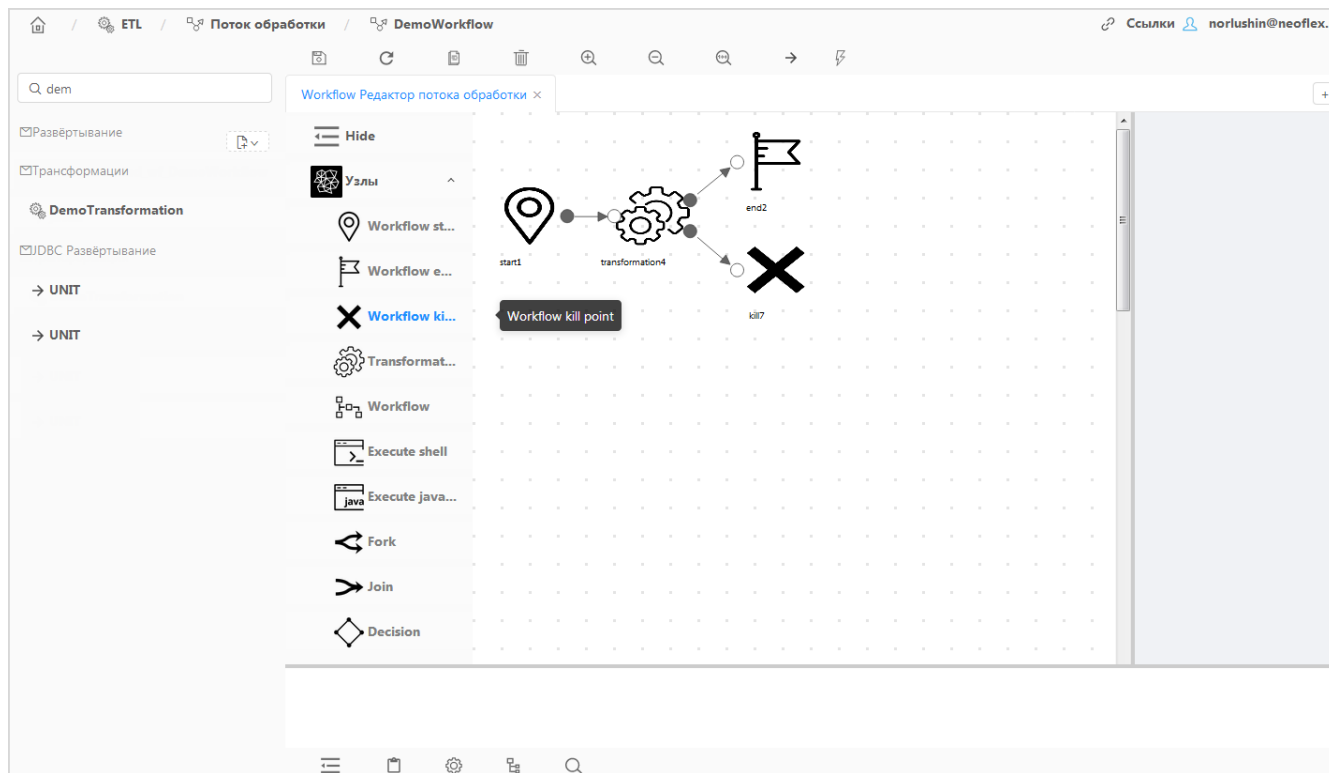
- **Name** - **DemoWorkflow**;
- **Label** - **DemoLabel**;
- **Project** - из списка выберите ранее созданный объект **DemoProject**.

Примечание.

Атрибуты объектов **Workflow** описаны в разделе «[Объекты Workflow](#)».

Настройка диаграммы потока работ

Кнопкой  откройте дизайнер потока работ. В рабочую область дизайнера перетащите элементы **Start**, **End**, **Transformation** и **Kill**. Укажите направление потоков управления, как показано на рисунке ниже.





Настройте элементы схемы:

- Для элемента **«transformation»** в поле **«transformation»** выберите значение **«DemoTransformation»**, в остальных полях оставьте дефолтные значения;
- В настройках элемента **«kill»** в поле **«message»** укажите текст сообщения об ошибке, например: **«Ошибка! Процесс остановлен»**. В остальных полях оставьте значения, которые были заданы программой автоматически.

Примечание.

Элементы диаграмм потоков работ описаны в разделе [«Элементы диаграмм объектов Workflow»](#).

Сохраните настройки кнопкой . Откройте список операций кнопкой  и запустите операцию **"Проверить"**, программа автоматически проверит синтаксис настроенной диаграммы объекта **«DemoWorkflow»**. Если проверка прошла успешно (это должно быть именно так, если действия, описанные в инструкции, выполнены правильно), то запустите объект на исполнение (операция **"Запустить"**)

После запуска объекта на исполнение, в разделе интерфейса **«Развертывание/Workflow Deployment»** будет создан объект **«autogenerated_wf_DemoWorkflow»**, который развернет и запустит исполнение **«DemoWorkflow»** на исполняющей среде Oozie.

На заметку.

По кнопке  можно создать новый объект Workflow Deployment.

Атрибуты объектов **Workflow Deployment** описаны в разделе «[Объекты Workflow Deployment](#)».

Дождитесь окончания исполнения объекта - на экране появится сообщение:

- В случае успешного исполнения, на экране появится окно с сообщением: «Ок».
- Если в ходе исполнения объекта возникнет ошибка, то на экране появится окно с текстом, описывающим ошибку.

Инструменты запуска приложений и мониторинга в Neoflex Datagram

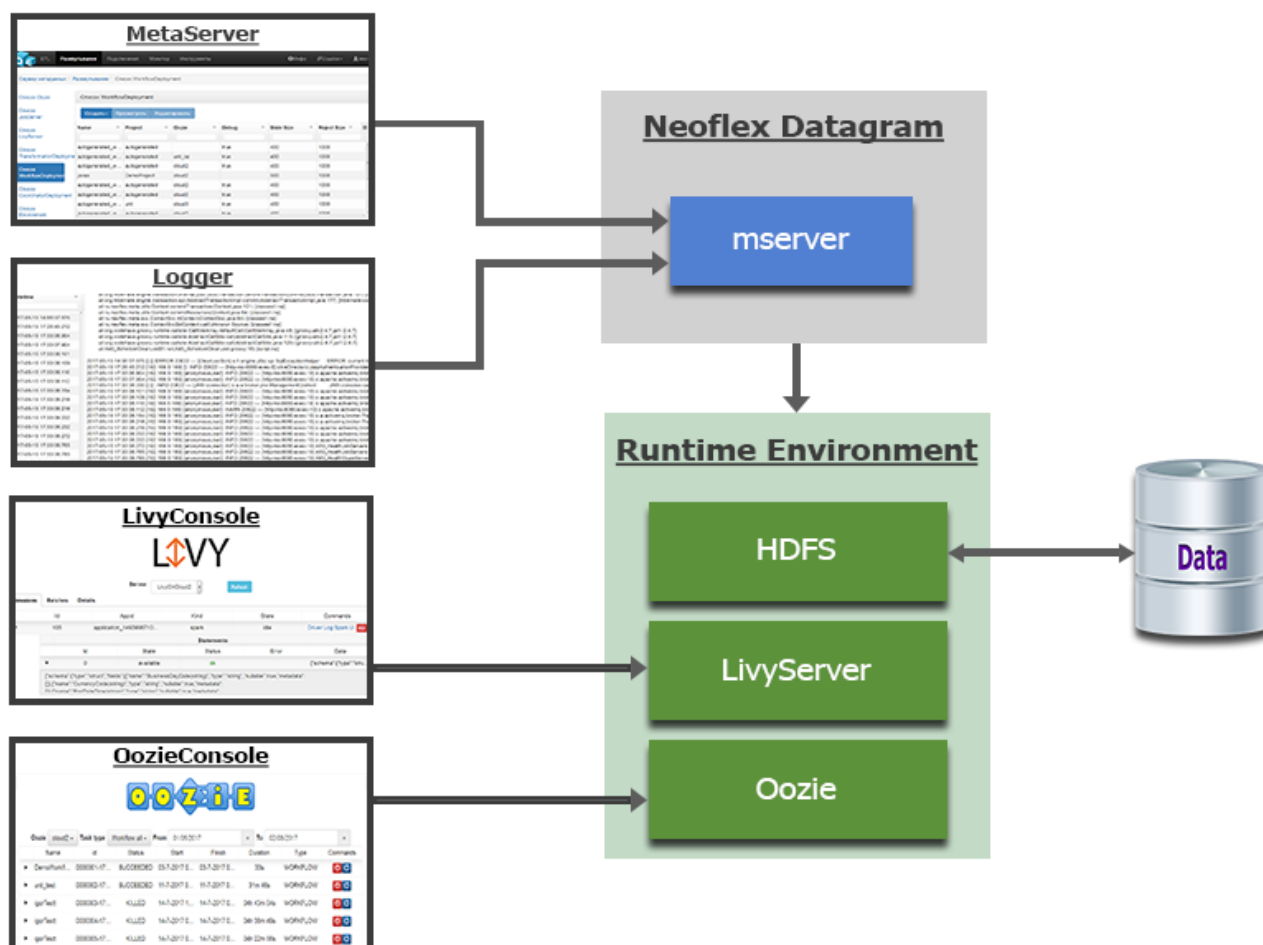
Запуск исполнения объектов Workflow или Transformation выполняется приложением Meta Server и состоит из фаз:

1. Генерация исходных файлов;
2. Компиляция задачи;
3. Развертывание задачи на исполняющей среде;
4. Запуск исполнения задачи через планировщик задач (Oozie или Livy Server);
5. Выполнение задачи на исполняющей среде.

Фазы 1, 2, 3 выполняются приложением Meta Server .

Фаза 4 выполняется приложением Meta Server через планировщик задач Spark - Oozie или Livy Server.

Фаза 5 выполняется в Hadoop.



Для мониторинга процессов в Neoflex Datagram используются инструменты:

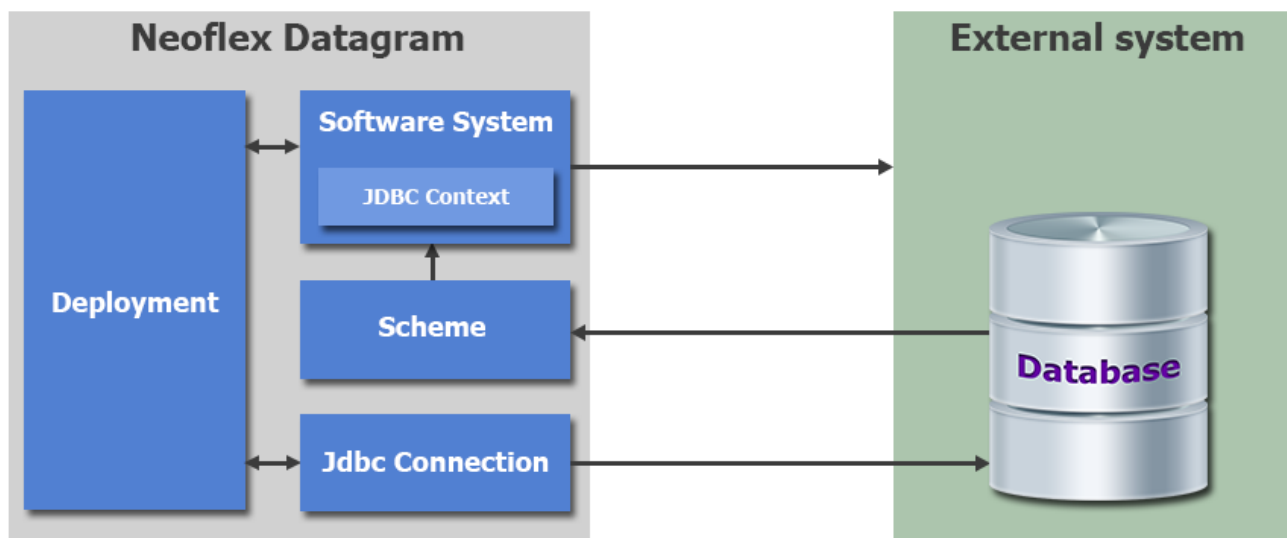
- **Logger** - отображает в реальном времени логи работы приложения Meta Server (фазы 1, 2, 3 и начало 4).
- **Консоли Livy и Oozie** - в интерфейсах консолей выводятся данные о результатах исполнения конкретных объектов Transformation и Workflow на исполняющих средах Hadoopworks Data Platform (фаза 5).

Работа с программой

Подключение Neoflex Datagram к базам данных внешних систем

Для подключения Neoflex Datagram к базе данных внешней системы необходимо создать и настроить объекты:

- [JDBC Connection](#);
- [Software System](#);
- [Deployment](#);
- [Scheme](#);
- [JDBC Context](#).



Объекты "JDBC Connection"

Объекты "**JDBC Connection**" хранят параметры подключения к базам данных внешних систем.

Настройка объектов выполняется в разделе интерфейса «Подключение/JDBC Connection».

Описание атрибутов объектов "JDBC Connection"

Атрибут	Обязательно заполнение	Описание
Name	Да	<p>Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java</p> <p><i>Пример: DemoJdbcConnection</i></p>
Project	Нет	<p>Проект, к которому привязан объект "JDBC Connection"</p> <p><i>Пример: DemoProject</i></p>
Url	Да	<p>Url-адрес для подключения к базе данных внешней системы</p> <p><i>Пример: jdbc:oracle:thin:@192.168.0.198:1522</i></p>
Scheme	Да	<p>scheme в терминах JDBC</p>
Catalog	Нет	<p>catalog в терминах JDBC</p>
User	Да	<p>Имя пользователя, используемое для подключения к базе данных внешней системы</p> <p><i>Пример: system</i></p>
Password	Нет	<p>Пароль, используемый для подключения к базе данных внешней системы.</p> <p><i>Рекомендуется использовать скрытый способ хранения паролей (см. раздел «Хранение паролей в системе»)</i></p>
Driver	Да	<p>Название драйвера для подключения к базе данных. Название используемого драйвера зависит от типа и версии БД, к которой выполняется подключение</p> <p><i>Пример: oracle.jdbc.driver.OracleDriver</i></p>

Операции, доступные для объектов "JDBC Connection"

Название операции	Описание
Протестировать	Операция выполняет проверку соединения с базой данных внешней системы

Объекты "Software System"

"Software System" - объекты, описывающие внешние системы, к базам данных которых подключается Neoflex Datagram.

Действия с объектами **"Software System"** выполняются в разделе интерфейса **«Подключение/Software System»**.

Описание атрибутов объектов "Software System"

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java и не содержать подчеркиваний <i>Пример: DemoSoftwareSystem</i>
Project	Нет	Проект, к которому привязан объект "Software System" <i>Пример: DemoProject</i>
Scheme	Нет	Название объекта "Scheme", описывающего базу данных внешней системы (см. раздел " Объекты "Scheme" ")
Default Deployment	Нет	Объект "Deployment", к которому привязан объект "Software System" <i>Пример: DemoDeployment</i>

Операции, доступные для объектов "Software System"

Название операции	Описание
Обновить схему	При запуске операции считываются метаданные БД внешней системы. На основе полученных метаданных обновляется привязанный объект "Scheme", либо генерируется новый и сохраняется в разделе интерфейса «Подключение/Scheme», одновременно данный объект "Scheme" привязывается к объекту "Software System", из которого запускалась операция

Объекты "Deployment"

"Deployment" - это объекты, обеспечивающие связь между объектами **"Software System"** и **"JDBC Connection"**.

Действия с объектами **"Deployment"** выполняются в разделе интерфейса **«Подключение/Deployment»**.

Описание атрибутов объектов "Deployment"

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java <i>Пример: DemoDeployment</i>
Project	Нет	Проект, к которому привязан объект "Deployment" <i>Пример: DemoProject</i>
Connection	Нет	Объект "JDBC Connection", который привязан к объекту "Deployment" <i>Пример: DemoJdbcConnection</i>
Software System	Нет	Объект "Software System", который привязан к объекту "Deployment" <i>Пример: DemoSoftwareSystem</i>
Load Stored Procs	Нет	При включении параметра, программа будет выполнять загрузку метаданных по хранимым процедурам

Операции, доступные для объектов "Deployment"

Название операции	Описание
Обновить схему	Операция посредством объекта "JDBC Connection", который указан в атрибутах объекта "Deployment", считывает метаданные БД внешней системы. На основе полученных метаданных обновляет имеющийся объект "Scheme", либо генерирует новый и сохраняет его в разделе интерфейса «Подключение/Scheme». Далее привязывает (обновленный или созданный) объект "Scheme" к объекту "Software System", выбранному в настройках объекта "Deployment"

Объекты "Scheme"

Объекты "**Scheme**" хранят следующую информацию о базе данных внешней системы:

- **views** - список динамически формируемых таблиц;

- **tables** - список таблиц базы данных;
- **Stored Procedures** - список хранимых процедур БД.

Данные объекты могут быть созданы:

- **Автоматически** - при выполнении команды "Обновить схему" объектов "Software System" или "Deployment";
- **Вручную** - в разделе интерфейса "Подключение/Scheme".

Объекты "Jdbc Context"

"Jdbc Context" - это объекты, обеспечивающие выбор подключения к базе данных внешней системы в настройках элементов дизайнера трансформаций (см. раздел «[Элементы диаграмм трансформаций](#)»).

Действия с объектами **Jdbc Context** выполняются в разделе интерфейса «**ETL/Jdbc Context**».

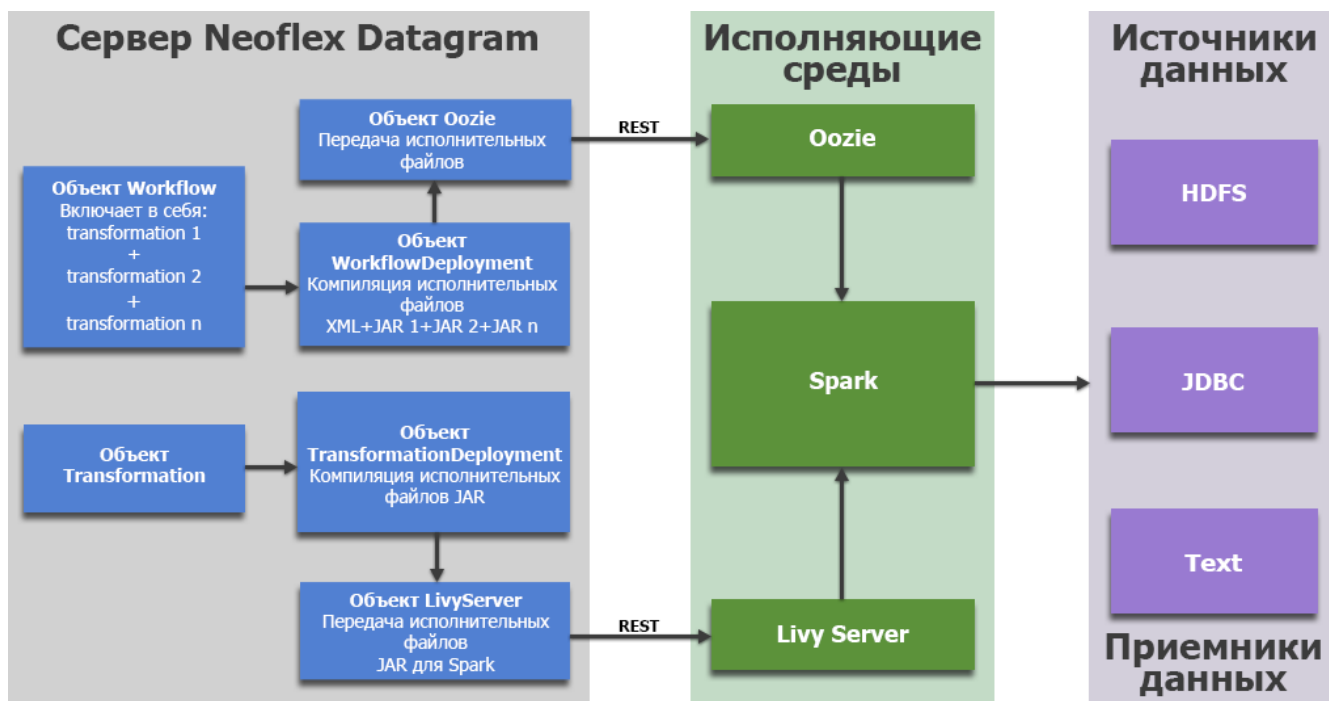
Описание атрибутов объектов "Jdbc Context"

Атрибут	Обязательно заполнение	Описание
Name	Да	<p>Название объекта.</p> <p>При указании имени необходимо учитывать, что оно должно соответствовать названию объекта SoftwareSystem и не должно содержать подчеркиваний</p> <p><i>Пример: DemoSoftwareSystem</i></p>

Подключение Neoflex Datagram к исполняющим средам

В зависимости от выполняемой задачи (выполнение преобразований или управление потоком работ по преобразованию данных), Neoflex Datagram может взаимодействовать с одной из исполняющих сред:

- [Oozie](#) - исполняет объекты Workflow;
- [Livy Server](#) - исполняет объекты Transformation.



Для взаимодействия с исполняющими средами в программе Neoflex Datagram должны быть созданы и настроены объекты:

- [Oozie](#);
- [Workflow Deployment](#);
- [Livy Server](#) (работает со Spark версии 2.X и выше);
- [Transformation Deployment](#);
- [Coordinator Deployment](#) - для развертывания заданий [планировщика задач Oozie](#).

Объекты "Oozie"

Объекты "**Oozie**" хранят параметры подключения к серверу Oozie и управления исполняющей средой.

Действия с объектами "**Oozie**" выполняются в разделе интерфейса **«Сервер/Oozie»**.

Описание атрибутов объектов "Oozie"

Атрибут	Обязательно заполнение	Описание
Name	Да	<p>Название объекта "Oozie". При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java</p> <p><i>Пример: DemoOozie</i></p>
Project	Нет	<p>Проект, к которому привязан объект "Oozie"</p> <p><i>Пример: DemoProject</i></p>
Job Tracker	Да	<p>Url-адрес Job Tracker</p> <p><i>Пример: cloud.neo.ru:8050</i></p>
Name Node	Да	<p>Параметры доступа к файловой системе HDFS</p> <p><i>Пример: hdfs://cloud.neo.ru:8020</i></p>
Master	Да	<p>Url-адрес для подключения к кластеру (подробное описание)</p> <p><i>Пример: spark://192.168.2.65:5310, local[4]</i></p>
Mode	Нет	<p>Атрибут определяет вариант развертывания драйвера Spark:</p> <p>client (по умолчанию) - на локальной машине в качестве внешнего клиента;</p> <p>cluster - на рабочем узле;</p> <p>yarn - YARN кластер. Конфигурация кластера задается переменными окружения</p>
is Default	Нет	<p>При включенном параметре, объект "Oozie" будет использоваться по умолчанию при запуске на исполнение объектов "Workflow".</p> <p><i>В случае если в Neoflex Datagram создано несколько объектов "Oozie", и у всех включен параметр is default, то для исполнения объекта "Workflow", объект "Oozie" будет выбран случайно</i></p>
Spark 2	Нет	<p>Параметр требует включения, если используется Spark версии 2.1 и выше</p>
Аутентификация	Нет	<p>При включенном параметре для подключения к серверу Spark будет использоваться алгоритм аутентификации Kerberos</p>

Атрибут	Обязательно заполнение	Описание
Путь к keytab	Нет	Путь к файлу, в котором хранятся пароли principal
User Principal	Нет	Principal, под которым авторизуется Meta Server
HCAT URL	Нет	Url-адрес Hive metastore
HCAT Principal	Нет	Principal, под которым авторизуется Hive
Num Executors	Да	Количество исполняющих процессов Spark <i>Пример: 5</i>
Executor Cores	Да	Количество ядер, задействованных для реализации исполняющего процесса Spark <i>Пример: 2</i>
Driver Memory	Да	Объем памяти, используемый для инициализации SparkContext <i>Пример: 512m, 2g</i>
Executor Memory	Да	Объем памяти, используемый для каждого исполняющего процесса Spark <i>Пример: 512m, 2g</i>
Queue	Нет	YARN очередь, в которой будет выполняться задача
Retry Max	Нет	Количество попыток запуска на исполнение задачи Spark <i>Пример: 5</i>
Retry Interval	Нет	Временной интервал между попытками запуска на исполнение задачи Spark <i>Пример: 10</i>
Cred	Нет	Учетные данные реализаций (подробнее)
Sftp	Нет	Параметр не используется. Sftp-адрес сервера, на котором разворачиваются артефакты объектов "Workflow" <i>Пример: sftp://192.168.2.44</i>

Атрибут	Обязательно заполнение	Описание
Http	Да	Url-адрес Oozie API <i>Пример:</i> http://cloud.neo.ru:12000
Webhdfs	Нет	Url-адрес HDFS API <i>Пример:</i> http://cloud3.neo.ru:50070/webhdfs/v1
Home	Да	Каталог в HDFS, используемый для развертывания "Workflow" <i>Пример:</i> /user
User	Да	Пользователь HDFS, от имени которого разворачиваются "Workflow" <i>Пример:</i> hdfs
Files Browser Util	Нет	Url web-консоли Hadoop

Объекты "Workflow Deployment"

Объекты **"Workflow Deployment"** создают комплект файлов, описывающих "Workflow", передают их на сервер Oozie и запускают исполнение.

Действия с объектами **"Workflow Deployment"** выполняются в разделе интерфейса **«Развертывание/Workflow Deployment»**.

На заметку.

*В списке могут присутствовать объекты **"Workflow Deployment"** с названием **«autogenerated_[workflow name]»**. Данные объекты создаются автоматически при запуске на исполнение объектов **"Workflow"** без указания конкретного объекта **"Workflow Deployment"**.*

Описание атрибутов объектов "Workflow Deployment"

Атрибут	Обязательно заполнение	Описание
Name	Да	<p>Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java</p> <p><i>Пример: DemoWorkflowDeployment</i></p>
Project	Нет	<p>Проект, к которому привязан объект "Workflow Deployment"</p> <p><i>Пример: DemoProject</i></p>
Oozie	Нет	<p>Объект "Oozie", описывающий подключение к серверу на котором выполняется объект "Workflow Deployment"</p> <p><i>Пример: DemoOozie</i></p>
Deployments	Нет	<p>Список объектов "Deployment", при помощи которых осуществляется доступ к базам данных внешних систем</p> <p><i>Пример: DemoDeployment</i></p>
Start	Нет	Разворачиваемый объект Workflow
Debug	Нет	При включенном параметре создаются файлы с промежуточным результатом трансформации
Slide Size	Нет	<p>Количество данных, которое единовременно записывается в Jdbc приемник данных</p> <p><i>Пример: 500</i></p>
Reject Size	Нет	<p>Максимально допустимое количество ошибок при записи данных в Jdbc приемник данных. При превышении установленного значения, выполнение трансформации будет принудительно завершено</p> <p><i>Пример: 1000</i></p>
Fetch Size	Нет	<p>Количество данных, которое единовременно считывается из Jdbc источника данных</p> <p><i>Пример: 100000</i></p>
Partition Num	Нет	<p>Количество рабочих процессов, выполняемых при записи данных в Jdbc приемник данных</p> <p><i>Пример: 4</i></p>

Атрибут	Обязательно заполнение	Описание
Master	Да	<p>Url-адрес для подключения к кластеру (подробное описание) <i>Данная настройка объекта "Workflow Deployment" имеет приоритет над аналогичной настройкой объекта "Oozie"</i></p> <p><i>Пример: spark://192.168.2.65:5310, local[4]</i></p>
Mode	Нет	<p>Атрибут определяет вариант развертывания драйвера Spark: client (по умолчанию) - на локальной машине в качестве внешнего клиента; cluster - на рабочем узле; yarn - YARN кластер. Конфигурация кластера задается переменными окружения <i>Данная настройка объекта "Workflow Deployment" имеет приоритет над аналогичной астройкой объекта "Oozie"</i></p>
Num Executors	Нет	<p>Количество исполняющих процессов Spark <i>Данная настройка объекта "Workflow Deployment" имеет приоритет над аналогичной настройкой объекта "Oozie"</i></p> <p><i>Пример: 5</i></p>
Executor Cores	Нет	<p>Количество ядер, задействованных для реализации исполняющего процесса Spark <i>Данная настройка объекта "Workflow Deployment" имеет приоритет над аналогичной настройкой объекта "Oozie"</i></p> <p><i>Пример: 2</i></p>
Driver Memory	Нет	<p>Объем памяти, используемый для инициализации SparkContext <i>Данная настройка объекта "Workflow Deployment" имеет приоритет над аналогичной настройкой объекта "Oozie"</i></p> <p><i>Пример: 512m, 2g</i></p>
Executor Memory	Нет	<p>Объем памяти, используемый для каждого исполняющего процесса Spark <i>Данная настройка объекта "Workflow Deployment" имеет приоритет над аналогичной настройкой объекта "Oozie"</i></p> <p><i>Пример: 512m, 2g</i></p>
Jvm Opts	Нет	Опции Java Virtual Machine
Persist On Disk	Нет	Если параметр включен, то при выполнении операции Check Point будет происходить сохранение промежуточных данных на диск, а не в память

Атрибут	Обязательно заполнение	Описание
Dynamic Allocation	Нет	При помощи данного параметра можно включить механизм распределения ресурсов в зависимости от рабочей нагрузки (по умолчанию выключено)

Описание параметров объектов "Workflow Deployment"

Параметр	Обязательно заполнение	Описание
Name	Да	Название параметра
Expression	Нет	Не используется для объектов "Workflow Deployment"
Description	Нет	Описание параметра

Описание опций Spark

Параметр	Обязательно заполнение	Описание
Name	Да	Название опции
Value	Нет	Значение опции

Операции, доступные для объектов "Workflow Deployment"

Название операции	Описание
Проверить	Neoflex Datagram выполняет проверку корректности объекта "Workflow" и привязанных объектов "Transformation"
Сгенерировать	Операция генерирует файлы XML, описывающие объект "Workflow", и код на языке Scala, описывающий связанные с объектом "Workflow" объекты "Transformation"
Собрать	При выполнении операции происходит компиляция JAR-файлов из кода языка Scala, описывающего объекты "Transformation" и формирование каталогов с XML и JAR-файлами для передачи на Oozie
Скопировать	При выполнении операции файлы XML и JAR копируются с сервера Neoflex Datagram в файловую систему ОС Linux сервера, на котором работает Oozie, далее выполняется копирование файлов в HDFS
Сгенерировать и скопировать	Последовательно выполняет операции: сгенерировать, собрать, скопировать
Запустить	Операция запускает исполнение файлов XML и JAR на Oozie
Сгенерировать и запустить	Последовательно выполняет операции: сгенерировать, собрать, скопировать и запустить
Собрать и запустить	Последовательно выполняет операции: собрать, скопировать и запустить

Объекты "Livy Server"

Объекты **"Livy Server"** хранят параметры подключения к серверу Livy и управления исполняющей средой Spark версии 2.X. и выше.

Действия с объектами **"Livy Server"** выполняются в разделе интерфейса **«Сервер/Livy Server»**.

Описание атрибутов объектов "Livy Server"

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта "Livy Server". При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Project	Нет	Объект "Project", к которому привязан объект LivyServer
Http	Да	Url-адрес Livy Server API <i>Пример:</i> http://cloud.company.ru:8090
Home	Нет	Каталог, используемый для развертывания "Transformation" <i>Пример:</i> /user
User	Нет	Пользователь HDFS, от имени которого разворачиваются "Transformation" <i>Пример:</i> hdfs
WebHDFS	Нет	Url-адрес HDFS API <i>Пример:</i> http://cloud3.company.ru:50070/webhdfs/v1
Аутентификация Kerberos	Нет	При включенном параметре для подключения к серверу Spark будет использоваться алгоритм аутентификации Kerberos
Путь к keytab	Нет	Путь к файлу, в котором хранятся пароли для principal
User Principal	Нет	Principal, под которым авторизуется Meta Server
Num Executors	Да	Количество исполняющих процессов Spark <i>Пример:</i> 5
Executor Cores	Да	Количество ядер, задействованных для реализации исполняющего процесса Spark <i>Пример:</i> 2
Driver Memory	Да	Объем памяти, используемый для инициализации SparkContext <i>Пример:</i> 512m, 2g

Атрибут	Обязательно заполнение	Описание
Executor Memory	Да	Объем памяти, используемый для каждого исполняющего процесса Spark <i>Пример: 512m, 2g</i>
is default	Нет	При включенном параметре, объект "Livy Server" будет использоваться по умолчанию при запуске на исполнение объектов "Transformation". <i>Если в программе создано несколько объектов "Livy Server", и у всех включен параметр "is default", то для исполнения объекта Transformation объект "Livy Server" будет выбран случайно</i>

Объекты "Transformation Deployment"

Объекты **"Transformation Deployment"** создают JAR-файлы, описывающие объекты "Transformation", и разворачивают их на сервере Livy.

Действия с объектами **"Transformation Deployment"** выполняются в разделе интерфейса **«Развертывание/Transformation Deployment»**.

Примечание.

*В списке могут присутствовать объекты **"Transformation Deployment"** с названием **«autogenerated_[transformation name]»**. Данные объекты создаются автоматически при запуске на исполнение объектов **"Transformation"** без указания объекта **"Transformation Deployment"**.*

Описание атрибутов объектов "Transformation Deployment"

Атрибут	Обязательно заполнение	Описание
Name	Да	<p>Название объекта "Transformation Deployment". Имя объекта должно удовлетворять правилам формирования идентификаторов в языке Java</p> <p><i>Пример: DemoTransformationDeployment</i></p>
Project	Нет	<p>Объект "Project", к которому привязан объект "Transformation Deployment"</p> <p><i>Пример: DemoProject</i></p>
Livy Server	Нет	<p>Объект "Livy Server", который обеспечивает подключение к исполняющей среде</p> <p><i>Пример: DemoLivyServer</i></p>
Transformation	Нет	<p>Объект "Transformation", который обрабатывается объектом "Transformation Deployment"</p> <p><i>Пример: DemoTransformation</i></p>
Deployments	Нет	<p>Объект "Deployment" для доступа к базам данных внешних систем</p> <p><i>Пример: DemoDeployment</i></p>
Debug	Нет	<p>При включенном параметре создаются файлы с промежуточным результатом трансформации</p>
Slide Size	Нет	<p>Количество данных, которое одновременно записывается в Jdbc приемник данных</p> <p><i>Пример: 500</i></p>
Reject Size	Нет	<p>Максимально допустимое количество ошибок при записи данных в Jdbc приемник данных. При превышении установленного значения, выполнение трансформации будет принудительно завершено</p> <p><i>Пример: 1000</i></p>
Fetch Size	Нет	<p>Максимальный объем данных, одновременно захватываемый из Jdbc источника данных</p> <p><i>Пример: 100000</i></p>

Атрибут	Обязательно заполнение	Описание
Partition Num	Нет	Количество рабочих процессов, исполняемых при записи данных в Jdbc приемник данных <i>Пример: 4</i>
Master	Да	Url-адрес для подключения к кластеру (подробное описание) <i>Пример: spark://192.168.2.65:5310, local[4]</i>
Mode	Нет	Атрибут определяет вариант развертывания драйвера Spark: client (по умолчанию) - на локальной машине в качестве внешнего клиента; cluster - на рабочем узле; yarn - YARN кластер. Конфигурация кластера задается переменными окружения
Num Executors	Да	Количество исполняющих процессов Spark <i>Пример: 5</i>
Executor Cores	Да	Количество ядер, задействованных для реализации исполняющего процесса Spark <i>Пример: 2</i>
Driver Memory	Да	Объем памяти, используемый для инициализации SparkContext <i>Пример: 512m, 2g</i>
Executor Memory	Да	Объем памяти, используемый для каждого исполняющего процесса Spark <i>Пример: 512m, 2g</i>
Persist on disk	Нет	Если параметр включен, то при выполнении операции "Check Point" будет происходить сохранение промежуточных данных на диск, а не в память
is default	Нет	При включенном параметре, объект "Livy Server" будет использоваться по умолчанию при запуске на исполнение объектов "Transformation". <i>Если в программе создано несколько объектов "Livy Server", и у всех включен параметр "is default", то для исполнения объекта Transformation объект "Livy Server" будет выбран случайно</i>

Описание параметров объектов "Transformation Deployment"

Параметр	Обязательно заполнение	Описание
Name	Да	Название параметра объекта
Expression	Нет	Поле не используется для объектов "Transformation Deployment"
Description	Нет	Описание параметра

Операции, доступные для объектов "Transformation Deployment"

Название операции	Описание
Проверить	Neoflex Datagram выполняет проверку корректности привязанного объекта "Transformation"
Сгенерировать	Операция генерирует код на языке Scala, описывающий привязанный объект "Transformation"
Собрать	При выполнении операции происходит компиляция JAR-файлов из кода языка Scala, описывающего объекты "Transformation" для передачи на Livy Server
Скопировать	При выполнении операции JAR-файлы копируются с сервера Neoflex Datagram на исполняющую среду Livy Server
Сгенерировать и скопировать	Последовательно выполняет операции: сгенерировать, собрать, скопировать
Запустить	Операция запускает исполнение файлов JAR на Livy Server
Сгенерировать и запустить	Последовательно выполняет операции: сгенерировать, собрать, скопировать и запустить
Собрать и запустить	Последовательно выполняет операции: собрать, скопировать и запустить

Объекты "Coordinator Deployment"

Объекты "**Coordinator Deployment**" создают файлы, описывающие объект "**Co Job**", передают их на сервер Oozie и запускают исполнение задачи.

Действия с объектами "**Coordinator Deployment**" выполняются в разделе интерфейса **«Развертывание/Coordinator Deployment»**.

Большинство атрибутов, параметров и операций объектов "**Coordinator Deployment**" аналогичны [атрибутам](#), [параметрам](#) и [операциям](#) объектов "**Workflow Deployment**".

Описание уникальных атрибутов объектов "Coordinator Deployment"

Атрибут	Обязательно заполнение	Описание
Coordinator	Нет	Название привязанного объекта "Co Job" <i>Пример: DemoCoJob</i>
Job Id	Нет	Идентификатор привязанного объекта "Co Job" (задается программой автоматически)

Уникальные операции объектов "Coordinator Deployment"

Название операции	Описание
Текущее состояние	Выдает сообщение, описывающее текущее состояние исполнения объекта "Co Job"

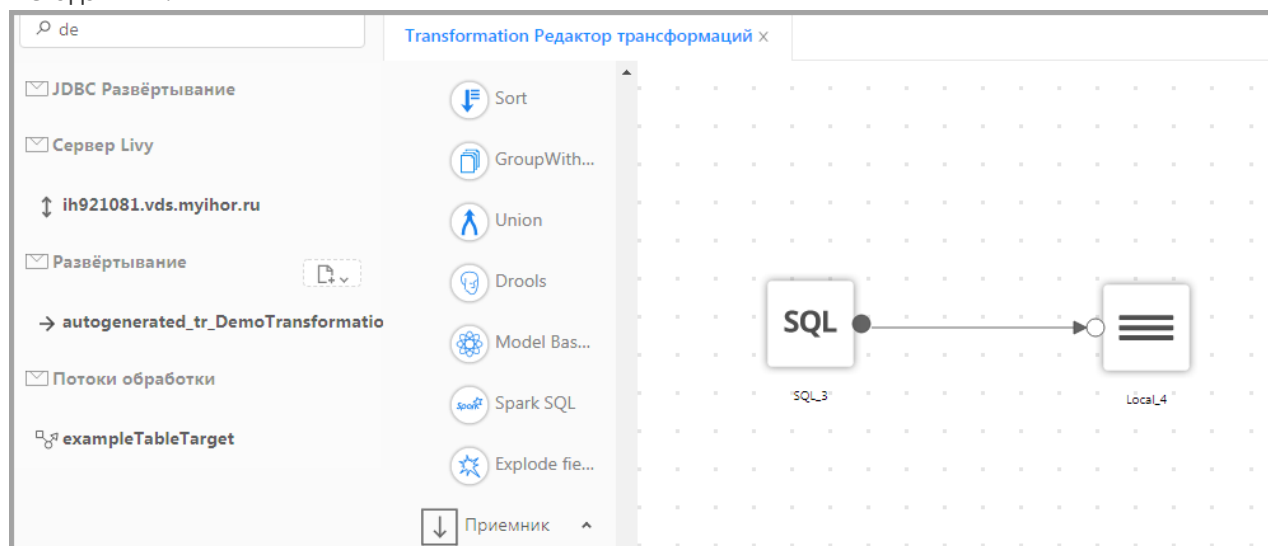
Мастер импорта. Настройка и создание

Мастер импорта автоматизирует процесс создания и настройки схемы Workflow для массовой загрузки метаданных из базы данных в кластер Hadoop, а также объектов Workflow deployment.

Для настройки мастера массовой загрузки метаданных необходимо выполнить действия:

1. Создать [трансформацию](#), при помощи которой будет выполняться считывание метаданных из базы и их сохранение в кластер Hadoop.

Пример простейшей трансформации, обеспечивающей требования считывания и сохранения метаданных:



2. В разделе "Развертывание/Мастер импорта" выполнить настройку мастера импорта;

3. Настроить объект "Entities", который описывает импортируемые сущности и параметры управления Spark.

Описание атрибутов мастера импорта

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Project	Нет	Объект "Project", к которому привязан объект
Jdbc Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Oozie	Да	Объект "Oozie", который описывает настройки подключения к серверу Oozie
Parallelism	Да	Максимальное количество одновременно исполняемых процессов импорта
HDFS path	Да	Путь к каталогу для хранения импортированных данных
Register Hive table	Нет	Если параметр включен, то результат будет регистрироваться в Hive meta store
Database	Нет	Название базы данных Hive
Table prefix	Да, если существуют совпадающие имена таблиц и представлений	Символ(ы), добавляемые к началу названия для идентификации таблицы при импорте
Table suffix	Да, если существуют совпадающие имена таблиц и представлений	Символ(ы), добавляемые к концу названия для идентификации таблицы при импорте
View prefix	Да, если существуют совпадающие имена таблиц и представлений	Символ(ы), добавляемые к началу названия для идентификации представлений при импорте
View suffix	Да, если существуют совпадающие имена таблиц и представлений	Символ(ы), добавляемые к началу названия для идентификации представлений при импорте
Show tables	Нет	При включенном чекбоксе в форме просмотра мастера импорта будут отображены таблицы
Show view	Нет	При включенном чекбоксе в форме просмотра мастера импорта будут отображены представления

Workflow parameters

Параметр	Описание
Name	Название параметра объекта
Value	Значение параметра
Expression	Если флаг включен, то значение Value является выражением языка Scala. В обратном случае - текстовое значение
Description	Описание параметра

Описание атрибутов Entities

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Scheme	Да	Объект Skheme, описывающий базу данных
Import entity type	Да	Тип импортируемых объектов: Table или View
Active	Нет	Если флаг установлен, то в интерфейсе мастера импорта все объекты будут выделены для импорта
Deleted	Нет	Если флаг установлен, то в интерфейсе мастера импорта объекты, которые существуют в модели, но отсутствуют в базе, будут выделены красным
Partition field	Да	Список названий партиций в HDFS для сохранения данных из БД. В качестве разделителя используется символ ","
Partition expression	Нет	Запрос на языке Spark SQL, который применяется к данным из таблицы перед их записью в партицию. Данное выражение может быть использовано, например, для изменения типа записываемых данных
Prestatement	Нет	Запрос на языке Spark SQL для создания представления (view) из которого будут импортироваться данные
WHERE condition	Нет	Выражение с условием WHERE для фильтрации импортируемых данных
ID Field	Нет	Название поля, по которому будет определено количество данных для считывания в каждом параллельном потоке <i>Пример:</i> <i>В ID field указано название поля ID. При этом программа определит максимальное и минимальное значения в поле ID и поделит его на количество потоков считывания, указанное в поле ID Parallelism</i>
ID Parallelism	Нет	Количество параллельных потоков считывания данных


Атрибут	Обязательно заполнение	Описание
Num Executors	Да, если данная настройка Oozie не задана, или требуется изменение значения	Количество исполняющих процессов Spark <i>Данная настройка объекта имеет приоритет над аналогичной настройкой объекта "Oozie"</i> <i>Пример: 5</i>
Executor cores	Да, если данная настройка Oozie не задана, или требуется изменение значения	Количество ядер, задействованных для реализации исполняющего процесса Spark <i>Данная настройка имеет приоритет над аналогичной настройкой объекта "Oozie"</i> <i>Пример: 2</i>
Driver memory	Да, если данная настройка Oozie не задана, или требуется изменение значения	Объем памяти, используемый для инициализации SparkContext <i>Данная настройка объекта имеет приоритет над аналогичной настройкой объекта "Oozie"</i> <i>Пример: 512m, 2g</i>
Executor memory	Да, если данная настройка Oozie не задана, или требуется изменение значения	Объем памяти, используемый для каждого исполняющего процесса Spark <i>Данная настройка объекта имеет приоритет над аналогичной настройкой объекта "Oozie"</i> <i>Пример: 512m, 2g</i>
Spark options	Нет	Опции Spark: Name - название опции Spark; Value - значение опции

Работа в мастере импорта

После сохранения настроек создаваемого мастера импорта на экране появится форма обзора, которая позволяет изменить настройки мастера и сформировать список импортируемых объектов.

Работа в мастере импорта поделена на три этапа:

Этап 1. Проверка/редактирование настроек Context, Workflow, указанных при создании мастера импорта.

После создания мастера импорта, список доступных для импорта объектов будет пуст. Для формирования списка нажмите кнопку "Обновить" или запустите операцию "Загрузить метаданные" из списка, открываемого по кнопке .

На данном этапе выберите объекты, которые необходимо импортировать. Выбранные объекты подсвечиваются голубым цветом.



















Для удобства поиска объектов для импорта, используйте фильтры в списке объектов, а также настройки отображения объектов.

Примечание.

Объекты, которые существуют в модели, но отсутствуют в базе, будут выделены красным.

ImportTeneo

Select objects to import

Schema	Object	Schema ↓	Type
public	 dwh_csvoptions	public	
	 dwh_defaultfield	public	
	 dwh_mapping	public	
	 dwh_stagingarea	public	
	 dwh_stagingarea_deployments	public	
	 dwh_stagingtable	public	
	 dwh_stagingtable_defaultfields	public	
	 etl_aggregation_groupbyfieldname	public	
	 etl_aggregationparameter	public	
	 etl_avroexplodefield	public	
	 etl_coaction	public	
	 etl_cocontrols	public	
	 etl_codataset	public	
	 etl_coinputevent	public	
	 etl_cojob	public	
	 etl_configurationproperty	public	
	 etl_context	public	
	 etl_cooutputevent	public	

View settings

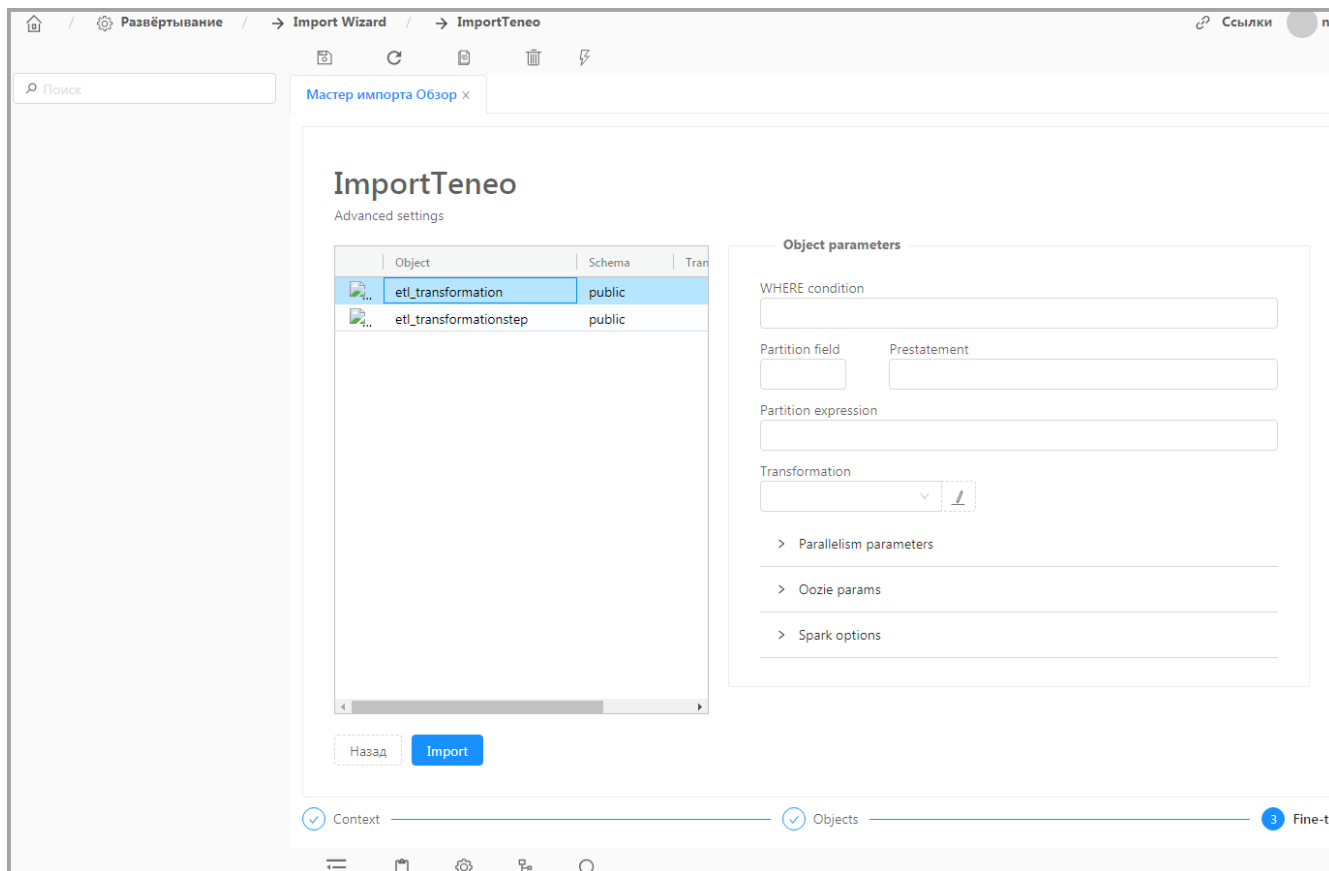
- ☒ Show tables
- ☒ Show views

Selected objects: 145

После выбора объектов для импорта, если не требуются дополнительные настройки/редактирование атрибутов импортируемых объектов, можно [запустить генерацию Workflow](#), который будет выполнять импорт данных.

Если атрибуты каких-либо импортируемых объектов требуют дополнительных настроек, то по кнопке "Вперед" перейдите на следующий этап.

Этап 3. Настройка атрибутов импортируемых объектов



На третьем этапе для каждого отдельного импортируемого объекта можно задать/отредактировать атрибуты. Данные атрибуты аналогичны тем, что описаны в таблице ["Описание атрибутов Entities"](#). Также данная форма позволяет для каждого импортируемого объекта задать отдельную трансформацию для считывания метаданных из базы и их сохранения в кластер Hadoop.

После выполнения всех необходимых настроек запустите генерацию Workflow, которая позволит выполнять импорт данных.

Генерация Workflow для импорта данных

Создание Workflow запускается операцией "Сгенерировать Workflow" из списка, доступного по кнопке



Сгенерированному Workflow будет присвоено название по правилу: "[Import wizard name]workflow". Для запуска или редактирования схемы сгенерированного workflow перейдите в раздел интерфейса «ETL/Workflow» (см. раздел ["Поток работ"](#)).

Трансформация данных

Объекты "Transformation"

"Transformation" - это объекты, описывающие логику преобразований данных.

Действия с объектами "Transformation" выполняются в разделе интерфейса «ETL/Transformation».

Описание атрибутов объектов "Transformation"

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта "Transformation". При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java <i>Пример: DemoTransformationDeployment</i>
Label	Нет	Примечание или короткий комментарий (например: DemoLabel). Допускается использовать кириллицу
Project	Нет	Объект "Project", к которому привязан объект "Transformation"
Json View	Нет	Представление объекта в формате json
Sources	Нет	Описание источников данных (sources), используемых в схеме трансформации
Targets	Нет	Описание приемников данных (targets), используемых в схеме трансформации
Transformation steps	Нет	Описание элементов преобразующих данные (data transformation), используемых в схеме трансформации
Transitions	Нет	Описание переходов (data flows) между элементами схемы трансформации
Parameters	Нет	Параметры объекта "Transformation". Name - название параметра объекта; Value - значение параметра; Expression - если флаг включен, то значение Value является выражением языка Scala. В обратном случае - текстовое значение; Description - описание параметра

На заметку.

Для отправки измененных параметров объекта "Transformation" необходимо выполнить операцию **«Запустить»** для соответствующего объекта **"Transformation Deployment"**.

Операции объектов "Transformation"

Название операции	Описание
Импорт	Из каталога проекта, имя которого совпадает с именем объекта "Project", к которому привязан выбранный объект "Transformation", импортируются данные объекта "Transformation"
Экспорт	В каталог проекта, имя которого совпадает с именем объекта "Project", к которому привязан выбранный объект "Transformation", экспортируются данные объекта "Transformation"
Проверить	Операция выполняет проверку корректности настроек объекта и логики его работы
Запустить	Операция запускает исполнение трансформации

Элементы диаграмм трансформаций

Группа элементов SOURCES

Local source

В качестве источника данных используется файл, хранимый в файловой системе HDFS сервера Oozie. При помощи данного элемента может быть создана схема потоковой обработки исходных данных.

Описание атрибутов элемента Local source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Local file name	Да	Путь к файлу, используемому в качестве источника данных <i>Пример: /user/hdfs/demo/demo.txt</i>
Local File Format	Да	Формат записи данных в файле-источнике: JSON; PARQUET; ORC; JDBC; CSV
Streaming	Нет	При включенном параметре элемент трансформации отслеживает появление новых данных и запускает исполнение трансформации (поточная обработка данных)
Options	Нет	Опции элемента трансформации: key - название опции; value - значение опции <i>Пример настройки опции для чтения данных из CSV файла, в котором в качестве разделителя используется символ «;»:</i> sep - значение поля <i>key</i> ; ; - значение поля <i>value</i>
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

CSV source

В качестве источника данных может быть использован CSV файл или таблица Excel.

Описание атрибутов элемента CSV source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
HDFS	Нет	Включение параметра указывает на то, что файл-источник хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Path	Да	Путь к файлу-источнику данных
Format	Да	Формат файла-источника данных: CSV (описание атрибутов CSV); EXCEL (описание атрибутов Excel)
Header	Нет	Если параметр включен, то при извлечении данных из файла будет пропускаться первая строка (используется, если в необходимо пропустить заголовок при считывании данных)
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Атрибуты формата CSV

Атрибут	Обязательно заполнение	Описание
Charset	Да	Кодировка, используемая в файле-источнике
Delimiter	Да	Символ, используемый в качестве разделителя между значениями в CSV
Quote	Нет	Символы, предназначенные для выделения значения, содержащего символы Delimiter <i>Пример:</i> <i>Если в качестве разделителя используется символ [,], то значение 2,5 должно быть обозначено: "2,5"</i>
Escape	Нет	Символы, предназначенные для выделения значения, содержащего символы Quote
Comment	Нет	Символ, предназначенный для обозначения комментария. Строки, помеченные таким символом, игнорируются при извлечении данных
Date format	Нет	Описание формата Date <i>Пример: dd.mm.yyyy</i>
Null value	Нет	Текстовое значение, которое интерпретируется как Null при чтении данных из файла-источника

Атрибуты формата Excel

Атрибут	Обязательно заполнение	Описание
Data address	Да	Адрес данных для начала считывания (по умолчанию: A1) <i>Пример: 'My Sheet'!B3:C35</i>
Add color columns	Да	Окрашивание колонок (по умолчанию: false)
Treat empty values as null	Нет	Если параметр включен, то при чтении пустые значения будут определены как Null
Timestamp format	Нет	Описание формата Timestamp <i>Пример: mm-dd-yyyy hh:mm:ss</i>
Max rows in memory	Нет	Если значение установлено, то будет задействован streaming reader. Используется для считывания данных из больших файлов <i>Пример: 20</i>

XML source

В качестве источника используется файл, содержащий данные в формате XML ([более подробное описание](#)). Работа элемента поддерживана в версии Spark 2.X и выше.

Описание атрибутов элемента XML source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
HDFS	Нет	Включение параметра указывает на то, что файл-источник хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Path	Да	Путь к файлу-источнику данных
Charset	Да	Кодировка, используемая в файле
Row Tag	Нет	Тег XML-файла, который будет определен как строка
Sampling Ratio	Да	<p>Процент строк для определения типа данных в полях.</p> <p><i>Пример:</i> так как xml может быть создан без проверки, то его содержимое может выглядеть следующим образом:</p> <pre><rows> <row><field1>1</field1><row> <row><field1>2</field1><row> <row><field1>3</field1><row> <row><field1>январь</field1><row> </rows></pre> <p>В результате, если Sampling Ratio установить 75% (без последней строки), то тип данных для field1 определится как INTEGER. Если 100%, то уже STRING</p>
Exclude Attribute	Нет	Если параметр включен, то при чтении атрибуты элементов будут исключены
Treat Empty Values As Nulls	Нет	Если параметр включен, то при чтении пустые значения будут определены как Null

Атрибут	Обязательно заполнение	Описание
Mode	Да	Выбор режима обработки поврежденных записей: PERMISSIVE (по умолчанию) - при обнаружении поврежденной записи в строке устанавливается значение Null. Текст поврежденной строки сохраняется в новое поле, указанное в параметре Column Name Of Corrupt Record; DROP MALFORMED - игнорирует поврежденную запись; FAILFAST - при обнаружении поврежденной записи выводит сообщение с предупреждением
Column Name Of Corrupt Record	Да	Название поля, в котором сохраняются поврежденные строки в режиме PERMISSIVE
Attribute Prefix	Да	Символ, используемый для обособления атрибутов
Value Tag	Да	Тег, используемый в качестве метки для значения атрибута элемента, не имеющего наследников
Ignore Surrounding Spaces	Нет	Если параметр включен, то при чтении данных пробелы, окружающие значение будут игнорироваться

Атрибут	Обязательно заполнение	Описание
Explode Fields	Нет	<p>Список полей, по которым будут развернуты строки, т.е. для каждого элемента внутри указанного массива будет создана строка во всем наборе данных.</p> <p>Список формируется при помощи параметров: alias - псевдоним поля; path - путь к полю.</p> <p><i>Пример:</i> Содержимое xml файла:</p> <pre><rows> <dep name="Бухгалтерия"> <employers> <fio>Иванов</fio> <fio>Петров</fio> <fio>Сидоров</fio> </employers> </dep> <dep name="HR"> <employers> <fio>Иванов</fio> <fio>Петров</fio> <fio>Сидоров</fio> </employers> </dep> <employers></pre> <p>Если не указывать <i>Explode Fields</i>, то будет сформирован набор данных:</p> <pre>Dep - employers Бухгалтерия - нечитаемая структура</pre> <p>Если настроить <i>Explode Fields</i>:</p> <p>alias – <i>employers</i>; path – <i>dep.employers</i>, то сформируется набор данных:</p> <pre>Dep - employers.fio Бухгалтерия - Иванов Бухгалтерия - Петров Бухгалтерия - Сидоров HR - Иванова HR - Петрова HR - Сидорова</pre>

Атрибут	Обязательно заполнение	Описание
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Avro source

В качестве источника используется файл в формате *.avro.

Описание атрибутов элемента Avro source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
HDFS	Нет	Включение параметра указывает на то, что файл-источник хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Path	Да	Путь к файлу-источнику данных
Schema HDFS	Нет	Включение параметра указывает на то, что файл схемы Avro хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Schema Path	Нет	Путь к файлу схемы Avro, в котором описан формат сообщения
Charset	Да	Кодировка, используемая в файле-источнике
Explode Fields	Да	Список полей, по которым будут развернуты строки. Список формируется при помощи параметров: alias - псевдоним поля; fields - название поля
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Expression source

В качестве источника данных используется массив (Array) элементов типа Map на языке Scala.

Описание атрибутов элемента Expression source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Expression	Нет	Выражение на языке Scala
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

SQL source

В качестве источника данных используются результат запроса к реляционной базе данных внешней системы.

Описание атрибутов элемента SQL source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size (Количество возвращаемых строк)	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Context From String	Нет	Если чекбокс включен, то контекст выбирается не из списка, а из строкового поля. При включении чекбокса появится поле Context from string, в котором необходимо указать строковое наименование контекста
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Sql Options	Нет	Дополнительные параметры SQL
Statement	Да	Запрос на языке SQL, в результате выполнения которого формируется таблица-источник данных <i>Пример: select*from system.help</i>
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Scheme on read	Нет	Включение флага необходимо в случае, когда схема источника заранее не известна и будет определена в момент чтения данных. Используется в запросах типа SELECT*FROM
Is parallel	Нет	При включенном параметре будет производится параллельное чтение данных из источника

Атрибут	Обязательно заполнение	Описание
Partition column	Да, если включено параллельное чтение данных	Колонка, по которой выполняется разбиение на интервалы при параллельном чтении данных
Num partitions	Да, если включено параллельное чтение данных	Количество параллельных потоков чтения данных из источника
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Hive source

В качестве источника данных используются результат выполнения запроса на языке SQL.

Описание атрибутов элемента Hive source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Context	Да	Объект "Jdbc Context", используемый при считывании данных в режиме "design time"
Explain	Нет	Включение функции EXPLAIN (Apache Hive), которая предоставляет данные о выполнении запроса, указанного в поле Statement
Statement	Да	Запрос на языке SQL, в результате выполнения которого формируется таблица-источник данных <i>Пример: select*from system.help</i>
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

HBase source

В качестве источника данных используется таблица базы данных HBase.

Описание атрибутов элемента HBase source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Name space	Да	Имя атрибута space, где располагается таблица
Table name	Да	Название таблицы, используемой в качестве источника данных <i>Пример: Users_purchase</i>
Row Key	Да	Значение row key таблицы
Min stamp	Нет	Минимальная версия, возвращаемая в запросе
Max stamp	Нет	Максимальная версия, возвращаемая в запросе
Max version	Нет	Максимальное количество версий для каждого ключа
Merge to latest	Нет	Объединяет все версии в одну, если включено
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Table source

В качестве источника данных используется одиночная таблица базы данных внешней системы.

Описание атрибутов элемента Table source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк выводимых в окне просмотра данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Table name	Да	Название таблицы, используемой в качестве источника данных <i>Пример: Users_purchase</i>
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Kafka source

В качестве источника данных использует сообщения системы [Apache Kafka](#), которые поступают по схеме «издатель-подписчик». Элемент Kafka source реализует потоковое чтение данных (read stream).

Описание атрибутов элемента Kafka source

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Bootstrap servers	Нет	Имена хостов и номера портов bootstrap серверов Kafka <i>Пример: host1:port1,host2:port2</i>
Kafka consume type	Нет	Способ приема сообщений: ASSIGN - принимать отдельные партиции; SUBSCRIBE - подписка; SUBSCRIBE_PATTERN - шаблон названий топиков
Consume option value	Нет	В поле устанавливается значение опции в зависимости от выбранного способа приема сообщений: ASSIGN - Список принимаемых партиций. <i>Пример: json string {"topicA":[0,1],"topicB":[2,4]}</i> SUBSCRIBE - Через запятую перечисляется список топиков. <i>Пример: test или test1, test2</i> SUBSCRIBE_PATTERN - Шаблон задается регулярным выражением на языке Java, по которому будет определяться название топика

Атрибут	Обязательно заполнение	Описание
Options	Нет	<p>Опции элемента: key - название опции; value - значение опции.</p> <p>Примеры опций:</p> <p>1. Стартовая точка для запуска запроса. Значение поля key: startingOffsets; Значение поля value: earliest, latest, также можно задать начальные партиции каждого топика, например: json string {"topicA":{"0":23,"1":-1},"topicB":{"0":-2}}.</p> <p>2. Прерывание выполнения запроса в случае ошибки. Значение поля key: failOnDataLoss; Значение поля value: true или false.</p> <p>3. Время ожидания выполнения запроса к системе Kafka. Значение поля key: kafkaConsumer.pollTimeoutMs; В поле value устанавливается количество миллисекунд.</p> <p>4. Количество попыток отправки повторного запроса к системе Kafka. Значение поля key: fetchOffset.numRetries; В поле value устанавливается количество попыток.</p> <p>5. Время ожидания между повторными отправками запроса к системе Kafka. Значение поля key: fetchOffset.retryIntervalMs; В поле value устанавливается количество миллисекунд.</p> <p>6. Ограничение количества смещений, обрабатываемых за один запуск. Указанное количество пропорционально распределяется между партициями топиков разного объема. Значение поля key: maxOffsetsPerTrigger; В поле value устанавливается максимальное количество смещений</p>
Value type	Нет	Формат сообщений: AVRO; XML; JSON
Value Scheme	Нет	Объект « Scheme data set », используемый для конвертации данных из сообщений
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »)

Атрибут	Обязательно заполнение	Описание
Debug list	Нет	Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Структура сообщений Apache Kafka

Поле	Тип данных
key	BINARY
value	BINARY
topic	STRING
partition	INTEGER
offset	LONG (разрядности просто integer'a недостаточно)
timestamp	Дата-время
timestampMap	INTEGER

Группа элементов DATA TRANSFORM

Join

Элемент выполняет функции оператора JOIN языка SQL и поддерживает следующие типы объединения таблиц:

- INNER JOIN;
- LEFT JOIN;
- RIGHT JOIN;
- FULL JOIN.

Описание атрибутов элемента Join

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Join type	Да	Тип операции JOIN: INNER; LEFT; RIGHT; FULL
Key fields	Да	Название поля таблицы, поступающей на вход in элемента трансформации
Join key field	Да	Название поля таблицы, поступающей на вход join элемента трансформации
Watermark field	Нет	Названия полей, которые используются при потоковой обработке
Watermark threshold	Нет	Пороговое значение
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Для элемента **Join** доступна функция визуальной настройки правила объединения таблиц.

Описание полей формы создания правила объединения таблиц

Название поля	Обязательно заполнение	Описание
Name	Да	Название правила
Field operation type	Да	<p>Тип операции:</p> <p>SQL - позволяет указать произвольное SQL выражение в поле Expression, которое будет применено к полям, указанным в Source fields;</p> <p>TRANSFORM - к полям, указанным в Source fields будет применяться выражение на языке Scala, которое указывается в поле Expression;</p> <p>ADD - сохраняет значения полей без применения к ним выражений;</p> <p>PACK - операция формирует набор переменных различного типа (структуру). Результирующее значение имеет тип STRUCT</p>
Data type domain	Да	Тип данных полей, указанных в Source fields
Source fields	Да	Список настраиваемых полей

Aggregation

Элемент выполняет функции операторов агрегации.

Описание атрибутов элемента Aggregation

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Group by field name	Да	Название поля по которому выполняется группировка
Aggregation parameters	Да	Параметры агрегации: result field name - название результирующего поля; field name - название поля по которому выполняется агрегация; aggregation function - тип функции агрегации: 1. AVG -возвращает среднее значение столбца таблицы. Результирующее значение имеет тип DECIMAL; 2. SUM - возвращает сумму значений столбца таблицы. Результирующее значение имеет тип DECIMAL; 3. MIN - возвращает минимальное значение столбца таблицы, в результате тип данных не меняется; 4. MAX - возвращает максимальное значение столбца таблицы, в результате тип данных не меняется; 5. FIRST - возвращает первое значение столбца таблицы, в результате тип данных не меняется; 6. LAST - возвращает последнее значение столбца таблицы, тип данных не меняется; 7. COUNT - возвращает количество записей (строк) в таблице. Результирующее значение имеет тип LONG; 8. LIST - функция формирует массив данных. Результирующее значение имеет тип ARRAY
Pivot field		Колонки, добавляемые к результирующему набору данных
Pivot parameters		Значения, которые будут преобразовываться в колонки (подробнее)

Атрибут	Обязательно заполнение	Описание
User aggregation	Нет	<p>Если флаг выключен, то используются стандартные функции агрегации (из списка Aggregation parameters») и применяются ко всем полям, выбранным в списке Group by field name</p> <p><i>Пример: (поле1, поле 2, ...).agg(max(полеN) as resultFieldN, count(полеN+1) as resultFieldN1, ...)).</i></p> <p>Соответственно, в каждом элементе списка Aggregation parameters следует определить все три поля: result field name, field name, aggregation function.</p> <p>Если флаг включен, то агрегация будет определяться пользовательскими выражениями без учета выбранной функции агрегации в поле Aggregation function. Важно учитывать следующее:</p> <p>Модель определяет список полей на выходе по Group by field name и Result field name. Соответственно, при написании выражений необходимо пользоваться набором названий, определенных в списках Group by field name и Result field name.</p> <p>Таким образом, если выражение пользовательской агрегации применяется к большому количеству полей, то все они должны быть определены - созданы элементы в Group by field name и Result field name</p>
Expression	Да, если включен User aggregation	Выражение, выполняемое для каждой строки входящих данных внутри группы. Данное выражение действует после выражения Init Expression
Init Expression	Да, если включен User aggregation	Инициализирующее выражение. Выполняется один раз для каждой группы данных
Final Expression	Да, если включен User aggregation	Выражение финализации. Данное выражение выполняется в конце один раз для каждой группы данных для получения результата
Merge Expression	Да, если включен User aggregation	Выражение используется для объединения промежуточных результатов агрегации
Output port	Да	<p>Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение «Соответствие типов полей в дизайнера трансформаций классам языка Scala»).</p> <p>Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента</p>

Атрибут	Обязательно заполнение	Описание
---------	---------------------------	----------

Selection

Элемент выполняет функцию фильтрации потока данных при помощи выражения на языке SQL:

- если результат применения выражения - true, то данные пропускаются;
- если результат применения выражения - false, то данные отфильтровываются.

Описание атрибутов элемента Selection

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Expression	Да	Выражение на языке SQL
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Projection

Элемент позволяет изменять формат входящего потока данных в соответствии с заданными условиями.

Описание атрибутов элемента Projection

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Watermark field	Нет	Названия полей, которые используются при потоковой обработке
Watermark threshold	Нет	Пороговое значение
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Для элемента **Projection** доступна функция настройки преобразования.

Описание полей формы создания правила преобразования

Название поля	Обязательно заполнение	Описание
Name	Да	Название правила
Field operation type	Да	<p>Тип операции:</p> <p>SQL - произвольное SQL выражение, которое будет применено для форматирования данных;</p> <p>TRANSFORM - к полям будет применяться выражение на языке Scala, которое указывается в поле Expression;</p> <p>ADD - сохраняет значения полей без применения к ним выражений;</p> <p>PACK - операция формирует набор переменных различного типа (структуру). Результирующее значение имеет тип STRUCT</p>
Data type domain	Да	Тип данных полей (см. приложение Соответствие типов полей в дизайнера трансформаций классам языка Scala)
Source fields	Да	Список настраиваемых полей

Sequence

Элемент выполняет функции генерации последовательности значений.

Описание атрибутов элемента Sequence

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Field name	Да	Название поля, в котором записывается сгенерированная последовательность значений
Sequence type	Да	Тип последовательности: ORACLE - последовательность формируется при помощи объекта Sequence БД Oracle; LOCAL - формирует последовательность от 1 до ∞
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Sequence name	Используется при работе с БД Oracle	Название генератора последовательности значений
Batch size	Используется при работе с БД Oracle	Количество значений в формируемой последовательности
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Sort

Элемент, позволяющий выполнить сортировку входящего потока данных.

Описание атрибутов элемента Sort

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Sort features	Да	Список полей, по которым выполняется сортировка: Field name - название поля, для которого выполняется сортировка; Ascending - при включенном параметре, сортировка записей выполняется по возрастанию
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Group with state

Элемент позволяет обрабатывать входящий поток данных при помощи обработчика, написанного пользователем на языке Scala и/или процессора анализа событий (см. раздел ["Потоковая обработка данных"](#)).

Описание атрибутов элемента Group with state

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента
Input port	Да	Описание входящего потока данных
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов " Transformation Deployment ")
List of group by key	Нет	Список полей, по которым выполняется группировка данных
Internal state definition	Нет	Описание внутренней структуры данных, которая используется для обработки группы данных
Flat map groups with state	Нет	Текст функции обработки данных на языке Scala
Events processor	Нет	Объект Events processor , используемый для обработки входящего потока данных
Refresh timeout	Нет	Время ожидания обновления данных в группе. По истечении установленного времени, если данные в группе не изменились, группа будет закрыта
Sort group by field	Нет	Название поля, по которому сортируется входящий поток данных перед началом обработки
Output mode	Нет	Параметр записи данных (подробнее)

Union

Элемент выполняет функции оператора UNION языка SQL.

Описание атрибутов элемента Union

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Правила объединения полей создаются в соответствующем окне.

Описание полей формы создания правила объединения полей

Название поля	Обязательно заполнение	Описание
Name	Да	Название правила
Data type domain	Да	Тип данных полей (см. приложение Соответствие типов полей в дизайнера трансформаций классам языка Scala)
Input port field	Да	Список полей, поступающих на вход "In" элемента Union
Union port field	Да	Список полей, поступающих на вход "Union" элемента

Drools

Элемент позволяет анализировать входящий поток данных средствами [Drools Expert](#).

Описание атрибутов элемента Drools

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Rule Files	Нет	Список файлов, описывающих правила обработки входящего потока данных: File url - Url-адрес файла; File type - тип файла: CSV, XLS, DRL, PKG, JAR, OTHER - тип файла определяется по расширению; Hdfs - включение параметра указывает на то, что файл хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Input fact type name	Да	Полное имя типа, передаваемого на вход правила для каждой строки (обычно описывается в файле *.drl)
Result fact type name	Да	Полное имя типа, передаваемого на выход правила для каждой строки
Result query name	Да	Имя результирующего запроса
Result fact object name	Да	Имя объекта, полученного после выполнения запроса Result query name
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Model based analysis

Элемент позволяет выполнять статистический анализ данных, основанный на применении прошедшей обучение модели, при помощи [apache.spark.mllib](https://spark.apache.org/docs/latest/mllib-guide.html). В настоящее время поддерживается классификация на основе деревьев принятия решений, метода свободных векторов, регрессионного анализа, наивного байесовского классификатора. Полный список методов:

- Decision Tree;
- Gradient Boosted Trees;
- Random Forest Trees;
- SVM;
- Naive Bayes;
- Logistic Regression;
- Linear Regression;
- Isotonic Regression.

Для настройки элемента Model based analysis необходимо заранее создать модель анализа данных и провести ее обучение. Создание, настройку и обучение модели можно выполнить, например, при помощи инструмента Zeppelin, входящего в поставку Ambari.

Описание атрибутов элемента Model based analysis

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Model file	Да	Путь файлу модели анализа данных
Label field name	Да	Название поля (типа Decimal), в которое будет записан результат анализа входящих данных
Model features fields	Да	Список полей типа Decimal, на основе которых делается анализ (модель делает предположение и записывает результат в поле, название которого задано в Label field name)
Method name	Да	Методы анализа данных: DecisionTree; GradientBoostedTrees; RandomForestTrees; SVM; NaiveBayes; LogisticRegression; LinearRegression; IsotonicRegression
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

На выходе элемента будет получен набор данных поданных на вход с добавлением нового поля, название которого задавалось атрибутом «Label field name», куда записано предположение (prediction).

Spark SQL

Элемент позволяет применять запросы на языке SQL к RDD ([подробнее](#)).

Описание атрибутов элемента Spark SQL

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Explain	Нет	Включение функции EXPLAIN, которая предоставляет данные о выполнении запроса, указанного в поле Statement
Custom SQL	Нет	Используется для пользовательской кастомизации. Значение выражения не перезаписывается при обновлении версии метаданных
Количество возвращаемых строк	Нет	Количество возвращаемых строк
Statement	Да	Запрос на языке SQL, который будет применяться к RDD
Sql ports	Да	<p>Названия принимающих портов и их псевдонимы. Псевдонимы портов используются при написании запросов.</p> <p><i>Примечание.</i> <i>Элемент может не иметь входящих портов вообще, тогда он будет выступать, как источник данных</i></p>

Атрибут	Обязательно заполнение	Описание
Output port	Да	<p>Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение «Соответствие типов полей в дизайнера трансформаций классам языка Scala»).</p> <p>Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента</p> <p><i>Внимание!</i> Созданное автоматически описание формата исходящего потока данных требует проверки и, возможно, корректировки, т.к. в случаях, когда на принимающие порты элемента поступают данные от нескольких элементов, и эти данные имеют одинаковые названия полей, то в Output port программа создаст только одно поле с таким названием. Также необходимо вручную задавать новые поля, если таковые создаются в результате запроса (например: <code>SELECT MAX(id) AS max_id</code>)</p>

Примечание.

Для элемента **Spark SQL** доступна функция редактирования запроса. Особенностью элемента является то, что данные могут быть обработаны только в режиме «run time».

Explode fields

Элемент предоставляет доступ к данным из массива данных. Элемент разворачивает массив и делает из одной записи столько строк, сколько элементов в массиве.

Описание атрибутов элемента Explode fields

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Checkpoint	Нет	Включает/выключает функцию сохранения результата, полученного при выполнении данного элемента диаграммы трансформации (см. описание атрибута persistOnDisk объектов "Transformation Deployment")
Explode Fields	Нет	Alias - псевдоним поля; Field - список полей, по которым будут развернуты строки, т.е. для каждого элемента внутри указанного массива будет создана строка во всем наборе данных; Data type domain - тип данных
Output port	Да	Описание формата исходящего потока данных, полученных в результате выполнения данного элемента диаграммы трансформации и передаваемых следующему элементу (см. приложение « Соответствие типов полей в дизайнера трансформаций классам языка Scala »).
		Debug list Список отладочных файлов, в которых содержится промежуточный результат выполнения элемента

Группа элементов TARGETS

Local target

В качестве приемника данных используется файл, хранимый в файловой системе HDFS сервера Oozie.

Описание атрибутов элемента Local target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Local file format	Да	Формат файла, в котором сохраняются данные: JSON; PARQUET; ORC; JDBC; CSV
Save mode	Да	Режим операции записи результатов выполнения элемента: APPEND - добавляет только новые данные; OVERWRITE - перезаписывает все данные; DISCARD - удаляет партиции, указанные в подменю PARTITIONS
Local file name	Да	Указывается путь и название файла, который будет создан в результате выполнения трансформации
Delete before save	Нет	Если включено, то выполняется удаление данных, сформированных в результате предыдущего выполнения данного элемента
Register table	Нет	Если параметр включен, то результат будет регистрироваться в Hive meta store
Hive table name	Нет	Имя таблицы в Hive metastore
Options	Нет	key - название опции; value - значение опции
Partition From String	Нет	Если чекбокс включен, то партиции будут выбраны из текстового поля, а не из списка партиций. При включении чекбокса появится поле Partitions String, в котором необходимо перечислить партиции. В качестве разделителя могут использоваться символы "," или ";". Допускается использование параметров
Partitions	Да	Список партиций (каталогов). При разбиении на партиции, данные сортируются по каталогам. Количество партиций должно быть меньше, чем количество атрибутов. Например, таблицу со столбцами ID и NAME можно разбить по какому-нибудь одному атрибуту: либо ID, либо NAME

Table target

В качестве приемника данных используется одиночная таблица базы данных внешней системы.

Описание атрибутов элемента Table target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Table name	Да	Название таблицы, используемой в качестве приемника данных
Target type	Да	Тип операции записи результатов выполнения элемента: INSERT - добавляет новые данные в таблицу; UPDATE - обновляет данные в таблице; DELETE - удаляет записи по ключу; MERGE - работает только для БД Oracle. Если в момент записи данных в таблицу в ней присутствуют старые записи, то выполнится операция Update. Если данные отсутствуют, то выполнится операция Insert
Clear	Нет	При включенном параметре выполняется очистка значений в таблице, сформированной в результате предыдущего выполнения данного элемента
Check If Exists	Нет	Если параметр включен, то при отсутствии какого-либо поля в таблице процесс будет завершаться ошибкой (будет выполняться reject вместо генерации записей)
Pre SQL statement	Нет	SQL запрос, выполняемый на входе данного элемента диаграммы трансформации
Post SQL statement	Нет	SQL запрос, выполняемый на выходе данного элемента
Input fields mapping	Да	Маппинг входных и выходных данных

Procedure target

В качестве приемника данных используется вызов хранимой процедуры.

Описание атрибутов элемента **Procedure target**

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Catalog name	Да	Catalog name в терминах Jdbc
Stored procedure	Да	Procedure name в терминах Jdbc
Pre SQL statement	Нет	SQL запрос, выполняемый на входе данного элемента диаграммы трансформации
Post SQL statement	Нет	SQL запрос, выполняемый на выходе данного элемента диаграммы трансформации
Input fields mapping	Да	Маппинг входных и выходных данных

Для элемента **Procedure target** доступна функция загрузки процедуры, используемой в качестве приемника данных. Функция запускается двойным кликом по пиктограмме элемента, находящейся в рабочей области. Обновление/загрузка списка процедур может занимать продолжительное время.

CSV target

В качестве приемника данных может быть использован CSV файл или таблица Excel.

Описание атрибутов элемента **CSV target**

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sample Size	Нет	Ограничение количества строк в окне просмотра массива данных, поступающих на вход элемента
HDFS	Нет	Включение параметра указывает на то, что файл-приемник хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Path	Да	Путь к файлу-приемнику данных
Format	Да	Формат файла-приемника данных: CSV (описание атрибутов CSV); EXCEL (описание атрибутов Excel)

Атрибуты формата CSV

Атрибут	Обязательно заполнение	Описание
Charset	Да	Кодировка, используемая в файле-приемнике
Delimiter	Да	Символ, используемый в качестве разделителя между значениями в CSV
Quote	Нет	Символы, предназначенные для выделения значения, содержащего символы Delimiter <i>Пример:</i> <i>Если в качестве разделителя используется символ [,], то значение 2,5 должно быть обозначено: "2,5"</i>
Escape	Нет	Символы, предназначенные для выделения значения, содержащего символы Quote
Comment	Нет	Символ, предназначенный для обозначения комментария. Строки, помеченные таким символом, игнорируются при извлечении данных
Date format	Нет	Описание формата Date <i>Пример: dd.mm.yyyy</i>
Null value	Нет	Текстовое значение, которое интерпретируется как Null при записи данных
Codec	Нет	Название кодека, используемого для сжатия данных. Если значение не указано, то сжатие данных не используется
Quote mode	Нет	Режим обособления значений символом Quote: ALL - обособляются все значения; MINIMAL - обособляются только те значения, которые содержат символ Delimiter; NON_NUMERIC - обособляются только текстовые данные; NONE - не обособляются

Атрибуты формата Excel

Атрибут	Обязательно заполнение	Описание
Save mode	Да	Тип операции записи результатов выполнения элемента: APPEND – добавляет данные в таблицу; OVERWRITE – перезаписывает данные в таблице; DISCARD - не используется
Data address	Да	Адрес данных для начала записи (по умолчанию: A1) <i>Пример: 'My Sheet'!B3:C35</i>
Date format	Нет	Описание формата Date <i>Пример: dd.mm.yyyy</i>
Timestamp format	Нет	Описание формата Timestamp <i>Пример: mm-dd-yyyy hh:mm:ss</i>

XML target

В качестве приемника данных используется XML-файл ([более подробное описание](#)). Работа элемента поддерживается в версии Spark 2.X и выше.

Описание атрибутов элемента XML target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
HDFS	Нет	Включение параметра указывает на то, что файл-приемник хранится в файловой системе HDFS, в обратном случае - используется файловая система хоста
Path	Да	Путь к файлу-приемнику данных
Charset	Да	Кодировка, используемая в файле
Row Tag	Нет	Тег XML-файла, который будет определен, как строка
Root Tag		Корневой тег XML-файла
Null Value	Нет	Значение, которое интерпретируется как Null
Attribute Prefix	Да	Символ, используемый для обособления атрибутов
Value Tag	Да	Тег, используемый в качестве метки для значения атрибута элемента, не имеющего наследников
Compression	Нет	Название кодека, используемого для сжатия файла при сохранении

Streaming target

Приемник данных, при помощи которого можно организовать потоковую обработку данных. В зависимости от настроек элемент может записывать результат трансформации в файл, хранимый в файловой системе HDFS или в таблицу базы данных внешней системы.

Описание атрибутов элемента Stream target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Output mode	Да	Режим записи результатов выполнения элемента: APPEND – (используется по умолчанию) добавляет только новые данные в таблицу результатов; COMPLETE – перезаписывает все данные в таблице результатов. Данный режим используется только для трансформаций с агрегациями
Checkpoint Location	Да	Путь к файлу в который записывается результат, полученный при выполнении данного элемента диаграммы
Local File Format	Да	Формат файла-приемника данных. <i>При выборе формата HBASE, в настройках добавятся дополнительные поля, необходимые для сохранения данных в таблице HBASE. Данные поля описаны в элементе трансформации Hbase target</i>
Local File Name	Да	Название файла-приемника данных
Trigger	Нет	Интервал времени между запусками проверки наличия необработанных данных в источнике. Если значение не задано, то проверка будет выполняться постоянно
Trigger units	Нет	Единица измерения атрибута Trigger
Timeout	Нет	Интервал между выполнениями запроса. Если параметр не задан, то запрос работает бесконечно
Refresh timeout	Нет	Интервал между обновлениями справочников
Options	Нет	Опции элемента: key – название опции; value – значение опции

Атрибут	Обязательно заполнение	Описание
Partitions	Да	Список партиций (каталогов). При разбиении на партиции, данные сортируются по каталогам. Количество партиций должно быть меньше, чем количество атрибутов. Например, таблицу со столбцами ID и NAME можно разбить по какому-нибудь одному атрибуту, либо ID, либо NAME

Hive target

Элемент **Hive target**, как и **Table target**, записывает данные в таблицы баз данных. Отличительной особенностью элемента **Hive target** является то, что при работе использует Hive (систему управления базами данных на основе Hadoop).

Примечание.

*Для использования элемента **Hive target** в схеме трансформации необходимо загрузить метаданные таблицы, которая будет являться приемником данных.*

Описание атрибутов элемента Hive target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Context	Да	Объект "Jdbc Context", который определяет настройки подключения к базе данных внешней системы
Table name	Да	Название таблицы, используемой в качестве приемника данных
Hive Target Type	Да	Тип операции записи результатов выполнения элемента: APPEND – добавляет данные в таблицу; OVERWRITE – перезаписывает данные в таблице; IGNORE – если точно такие же записи уже существуют, они не будут перезаписаны (проигнорированы); ERROR – выдает ошибку, если точно такие же записи уже существуют
Pre SQL statement	Нет	SQL запрос, выполняемый на входе данного элемента диаграммы трансформации
Post SQL statement	Нет	SQL запрос, выполняемый на выходе данного элемента
Partitions	Да	Список партиций (каталогов). При разбиении на партиции, данные сортируются по каталогам. Количество партиций должно быть меньше, чем количество атрибутов. Например, таблицу со столбцами ID и NAME можно разбить по какому-нибудь одному атрибуту, либо ID, либо NAME
Input fields mapping	Да	Маппинг входных и выходных данных

HBase target

В качестве приемника данных используется таблица базы данных HBase.

Описание атрибутов элемента HBase target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Name space	Да	Имя атрибута space, где располагается таблица
Table name	Да	Название таблицы, используемой в качестве приемника данных <i>Пример: Users_purchase</i>
Row Key	Да	Значение row key таблицы
Regions for new table	Да	Значение атрибута Region для создаваемой таблицы в приемнике данных. Region должен быть ≥ 5
Version column	Нет	Колонка, значение которой будет использоваться в качестве версии
Input fields mapping	Да	Маппинг входных и выходных данных

Kafka target

Элемент из поступающих на вход данных формирует сообщения и отправляет их в систему [Apache Kafka](#). Данный элемент не является стриминговым.

Описание атрибутов элемента Kafka target

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Bootstrap servers	Да	Имя хоста и номер порта bootstrap сервера Kafka
Topic	Нет	Название топика
Message key	Нет	Название поля, которое будет служить ключом сообщения
Message value	Нет	Название поля, которое будет служить значением сообщения
Properties	Нет	Дополнительные свойства отправки сообщений: key – название свойства; value – значение свойства. Для ознакомления с полным списком свойств перейдите в раздел Configuration settings на странице
Value type	Нет	Формат сообщений: AVRO; XML; JSON
Value Scheme	Нет	Объект « Scheme data set », используемый для конвертации данных, отправляемых в Kafka

Агрегатные пользовательские функции

В программе реализована возможность применения агрегатных пользовательских функций ([UDAF](#)) в SQL и Spark SQL запросах при настройке элементов трансформаций, потоков работ и т.д.

Создание и настройка агрегатных пользовательских функций выполняется в разделе интерфейса "ETL/User defined function".

Атрибуты User defined function

Атрибут	Обязательно заполнение	Описание
UDF name	Да	Название объекта User defined function. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Code	Нет	Выражение (поддерживаются языки: Scala, Java)
UDF class name	Нет	Имя класса UDF
Project	Нет	Объект "Project", к которому привязан объект "User defined function"

Объекты "Scheme data set"

Объекты **"Scheme data set"** хранят настройки для сериализации/десериализации данных при работе с бинарными сообщениями системы [Apache Kafka](#). Данные объекты используются в работе элементов схем трансформации [Kafka source](#) и [Kafka target](#).

Работа с объектами "Scheme data set" выполняется в разделе: **"ETL/Scheme data set"**.

Атрибуты Scheme data set

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта Scheme data set. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Scheme	Нет	Описание схемы сообщений, получаемых из Apache Kafka
Scheme type	Нет	Формат схемы сообщений: AVRO; XML; JSON
Dataset	Нет	Описание структуры в которую конвертируются данные из сообщений

Поток работ

Объекты "Workflow"

Объекты **"Workflow"** - это объекты, описывающие алгоритм управления потоком работ по преобразованию исходных данных.

Действия с объектами **"Workflow"** выполняются в разделе интерфейса «ETL/Workflow».

Описание атрибутов объектов "Workflow"

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта "Workflow". При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java <i>Пример: DemoTransformationDeployment</i>
Label	Нет	Примечание или короткий комментарий. Допускается использовать кириллицу
Project	Нет	Объект "Project", к которому привязан объект "Workflow"
Nodes	Нет	Описание элементов схемы управления
Json View	Нет	Представление объекта в формате json

Атрибуты вложенного объекта SLA (см. раздел «[SLA мониторинг задач Oozie](#)»)

Атрибут	Обязательно заполнение	Описание
Nominal Time	Да	Точка отсчета для SLA. Переменная nominal_time создается автоматически при запуске задачи из Datagram. В нее записывается текущее время
Should Start	В зависимости от отслеживаемых событий	Количество времени, относительно Nominal Time, в течение которого должна запуститься задача. Может быть задана в: Минутах - MINUTES; Часах - HOURS; Днях - DAYS <i>Пример: 10*MINUTES</i>
Should End	В зависимости от отслеживаемых событий	Количество времени, в течение которого должно быть закончено исполнение задачи относительно Nominal Time. Может быть задана в: Минутах - MINUTES; Часах - HOURS; Днях - DAYS <i>Пример: 2*HOURS</i>
Max Duration	В зависимости от отслеживаемых событий	Максимальная продолжительность исполнения задачи относительно Nominal Time. Может быть задана в: Минутах - MINUTES; Часах - HOURS; Днях - DAYS <i>Пример: 3*HOURS</i>
Alert Events	Да	Список отслеживаемых событий, для которых необходимо отправлять отчет по электронной почте: start_miss; end_miss; duration_miss. В качестве разделителя используется запятая <i>Пример: start_miss, end_miss, duration_miss</i>
Alert Contact	Да	Список адресов электронной почты, на которые будут отправляться оповещения. В качестве разделителя используется запятая <i>Пример: example@box.com</i>

Операции, доступные для объектов "Workflow"

Название операции	Описание
Проверить	Операция выполняет проверку корректности настроек объекта и логики его работы
Создать представление	Операция создает представление объекта в формате json
Сгенерировать и скопировать	При выполнении операции генерируются файлы JAR и XML, описывающие объект, и копируются с сервера Neoflex Datagram в файловую систему HDFS
Импорт	Из каталога проекта, имя которого совпадает с именем объекта Project, к которому привязан выбранный объект Workflow, импортируются данные объекта Workflow
Экспорт	В каталог проекта, имя которого совпадает с именем объекта Project, к которому привязан выбранный объект Workflow, экспортируются данные объекта Workflow
Запустить	Операция запускает исполнение объекта

Элементы диаграмм объектов Workflow

Дизайнер диаграмм объектов **Workflow** предназначен для построения диаграмм, описывающих поток работ и обеспечивающих запуск, управление и контроль над выполнением преобразований исходных данных в исполняющей среде Oozie. [Описание исполняющей среды Oozie](#)

Группа элементов POINT

В группе собраны элементы, обеспечивающие запуск, завершение исполнения потока работ и прерывание исполнения в случае ошибки.

Start

Элемент обозначает начало диаграммы потока работ.

На заметку.

*Диаграмма, описывающая поток работ, может содержать только один элемент **start**.*

Атрибуты элемента Start

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента

End

Элемент используется для обозначения успешного окончания диаграммы потока работ.

На заметку.

*Диаграмма, описывающая поток работ, может содержать только один элемент **end**. Если в момент достижения элемента **end** один или несколько элементов диаграммы еще выполнялись, то выполнение элементов будет принудительно остановлено, и это не будет считаться ошибкой.*

Атрибуты элемента End

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента

Kill

Элемент используется для контроля исполнения отдельных элементов или части диаграммы потока работ. В случае обнаружения ошибки, элемент принудительно останавливает исполнение диаграммы, при этом может выполняться запись заданного текста сообщения об ошибке в лог Oozie.

Описание атрибутов элемента kill

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Properties Message	Нет	Текст сообщения, которое записывается в лог Oozie в случае выполнения данного элемента

Группа элементов RULE

Элементы данной группы используются для определения последовательности выполнения заданий:

- параллельность или последовательность выполнения заданий определяется элементами ветвления (fork) и соединения (join);
- последовательность выполнения заданий в зависимости от доступности определенных данных или завершения каких-либо других событий определяется элементами принятия решений (decision).

Fork

Элементы **fork** позволяют настроить параллельное выполнение неограниченного количества заданий.

Описание атрибутов элемента Fork

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Paths	Да	Количество и названия передающих портов элемента

Join

Элементы **Join** позволяют объединять неограниченное количество параллельных заданий.

Описание атрибутов элемента Join

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
To	Да	Элемент схемы к которому подключен порт Out элемента Join

Decision

Элементы **Decision** управляют запуском выполнения заданий в зависимости от условий, описываемых в виде предикатов.

Описание атрибутов элемента Decision

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Cases	Да	Описание исходящих портов: Label - название порта; Predicate - предикат времени или событий, записанный на языке Expression Language , результат вычисления которого может иметь одно из значений: true или false
Default	Да	Переход по умолчанию

Группа элементов ACTION

Элементы данной группы определяют задания по преобразованию исходных данных в диаграммах потока работ.

Transformation

Элементы **Transformation** предназначены для включения в диаграмму потока работ объектов **Transformation**.

Атрибуты элемента Transformation

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Transformation	Да	Из списка выбирается объект "Transformation", который должен использоваться в схеме
Jvm opts	Нет	Опции Java Virtual Machine
Prepare	Нет	Действия, выполняемые перед началом работы элемента: Delete dir - список каталогов, удаляемых при запуске элемента; Mk dir - список каталогов, которые будут создаваться при запуске элемента
Parameters	Нет	Параметры: Name - название параметра; Value - значение параметра; Expression - если флаг включен, то значение Value является выражением языка Scala. В обратном случае - текстовое значение; Description - описание параметра
Sla	Нет	Настройки SLA мониторинга (см. таблицу "Атрибуты вложенного объекта SLA")
Ok	Да	Название элемента подключенного к порту Ok
Error	Да	Название элемента подключенного к порту Error

Workflow

Элементы **Workflow** предназначены для включения в диаграмму потока работ вложенных объектов **Workflow**.

Атрибуты элемента Workflow

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Sub Workflow	Да	Название объекта "Workflow", привязанного к данному элементу
Propagate Configuration	Нет	Если параметр включен, то конфигурация основного Workflow применяется к вложенным Sub Workflow
Properties	Нет	Список свойств конфигурации окружения (hadoop config), которые требуется переопределить: Name - название свойства; value - новое значение
Sla	Нет	Настройки SLA мониторинга (см. таблицу "Атрибуты вложенного объекта SLA")
Ok	Да	Название элемента подключенного к порту Ok
Error	Да	Название элемента подключенного к порту Error

Execute shell

Элемент **Execute shell** позволяет выполнять в диаграмме потока работ shell скрипты.

Атрибуты элемента Execute shell

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Exec	Да	Имя файла, содержащего shell скрипт. <i>Пример: script.sh</i>
Args	Да	Аргументы shell скрипта
File	Да	Путь к файлу, содержащему shell скрипт. <i>Пример: /home/hdfs/data/script.sh</i>
Capture Output	Нет	Включение параметра позволяет сохранять данные, полученные в результате выполнения данного элемента, для использования в диаграмме в дальнейшем. Данные должны соответствовать требованиям: 1. Формат результата должен соответствовать формату файла свойств Java; 2. Размер результирующего файла не должен превышать 2 kB
Sla	Нет	Настройки SLA мониторинга (см. таблицу "Атрибуты вложенного объекта SLA")
Ok	Да	Название элемента подключенного к порту Ok
Error	Да	Название элемента подключенного к порту Error

Execute Java

Элемент **Execute Java** позволяет выполнять в диаграмме потока работ java скрипты.

Атрибуты элемента Execute Java

Атрибут	Обязательно заполнение	Описание
Name	Да	Название элемента. Название должно удовлетворять правилам формирования идентификаторов в языке Java
Label	Нет	Краткое описание элемента
Jar files	Нет	Путь к файлам, к которым Spark должен предоставить доступ во время работы элемента Execute java
Prepare	Нет	Действия, выполняемые перед началом работы элемента: Delete dir - список каталогов, удаляемых при запуске элемента; Mk dir - список каталогов, которые будут создаваться при запуске элемента
Main class	Да	Полное имя класса, содержащего метод public static void main(args), который будет выполнен
Args	Нет	Список аргументов, передаваемых в функцию main
Javaopts	Нет	Параметры Java Virtual Machine
Capture Output	Нет	Включение параметра позволяет сохранять данные, полученные в результате выполнения данного элемента, для использования в диаграмме в дальнейшем. Данные должны соответствовать требованиям: 1. Формат результата должен соответствовать формату файла свойств Java; 2. Размер результирующего файла не должен превышать 2 kB
File	Нет	Путь к файлу, к которому Spark должен предоставить доступ во время работы элемента Execute java
Archive	Нет	Путь к архиву, к которому Spark должен предоставить доступ во время работы элемента Execute java
Properties	Нет	Список свойств конфигурации окружения (hadoop config), которые требуется переопределить: Name - название свойства; Value - новое значение
Sla	Нет	Настройки SLA мониторинга (см. таблицу "Атрибуты вложенного объекта SLA")
Ok	Да	Название элемента подключенного к порту Ok
Error	Да	Название элемента подключенного к порту Error

Запуск на исполнение настроенных объектов Transformation и Workflow

В разделе ["Quick start"](#) был рассмотрен вариант запуска на исполнение объектов **Transformation** и **Workflow** из интерфейса дизайнера диаграмм. Данный способ удобен при создании либо редактировании диаграмм объектов. В случаях, когда необходимо периодически выполнять запуск уже настроенных объектов, удобнее это делать из формы просмотра объектов [Transformation Deployment](#) (запускает объекты Transformation) или [Workflow Deployment](#) (запускает объекты Workflow).

Запуск исполнения настроенного объекта Transformation

Для запуска перейдите в раздел **«Развертывание/Transformation Deployment»**. Найдите в списке или создайте новый объект **Transformation Deployment** (см. раздел [«Объекты Transformation Deployment»](#)), при помощи которого будет запущен объект **Transformation**.

На заметку.

Объект **Transformation Deployment** будет запускать объект **Transformation**, который привязан к нему. Таким образом, в списке необходимо выбрать объект **TransformationDeployment**, у которого в столбце **«Transformation»** указано название объекта **Transformation**, запуск которого нужно выполнить.

Откройте форму просмотра выбранного объекта **Transformation Deployment**. В форме просмотра в списке **«Выполнить операцию»** выберите пункт **«Запустить»** (см. раздел ["Объекты Transformation Deployment"](#)).

Запуск исполнения настроенного объекта Workflow

Процедура запуска объектов **Workflow** схожа с процедурой запуска объектов **Transformation**, описанной в предыдущем разделе. Разница заключается в том, что запуск объектов **Workflow** выполняется из формы просмотра объекта **Workflow Deployment** (см. раздел [«Объекты WorkflowDeployment»](#)). Список объектов **WorkflowDeployment** доступен в разделе интерфейса **«Развертывание/Workflow Deployment»**.

На заметку.

Объект **WorkflowDeployment** запускает объект **Workflow**, который привязан к нему. Привязка объекта может быть определена в форме редактирования объекта **WorkflowDeployment** в поле **«Workflows»**.

Запуск исполнения настроенного объекта Workflow по расписанию

В программе поддерживается работа с модулем [Oozie Coordinator](#), что позволяет выполнять запуск исполнения объектов **Workflow** по расписанию.

Для запуска исполнения объектов Workflow по расписанию в программе должны быть созданы и настроены объекты:

- [Oozie](#), описывающий параметры подключения к серверу Oozie и управления исполняющей средой;
- [Coordinator Deployments](#), отвечающий за создание комплекта файлов, описывающих объект **CoJob** и связанный с ним **Workflow**, передачу файлов на сервер Oozie и запуск исполнения;
- **CoJob**, описывающий параметры работы Oozie Coordinator.

Объекты CoJob

Объекты **CoJob**, а также вложенные объекты, хранят настройки работы модуля Oozie Coordinator.

Действия с объектами **CoJob** выполняются в разделе интерфейса «**ETL/CoJob**».

Атрибуты объекта CoJob

Атрибут	Обязательно заполнение	Описание
Name	Да	<p>Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java</p> <p><i>Пример: DemoOozie</i></p>
Project	Нет	<p>Проект, к которому привязан объект</p> <p><i>Пример: DemoProject</i></p>
Frequency	Нет	<p>Периодичность запусков исполнения объекта Workflow в заданном интервале времени. Может быть задана в минутах, или выражением на языке Expression Language.</p> <p><i>Пример в минутах: 10;</i> <i>Пример выражения: \${coord:days(1)}</i></p>
Start	Нет	<p>Начало временного интервала, в течение которого происходят запуски исполнения объекта Workflow.</p> <p><i>Пример: 2016-09-02T08:05Z</i></p>
End	Нет	<p>Конец временного интервала, в течение которого происходят запуски исполнения объекта Workflow.</p> <p><i>Пример: 2016-09-03T10:25Z</i></p>
Timezone	Нет	<p>Часовой пояс (стандарт UTC или GMT).</p> <p><i>Пример: Etc/GMT-4</i></p>

Атрибуты вложенного объекта Controls (описывает параметры управления исполнением экземпляров объекта Workflow)

Атрибут	Обязательно заполнение	Описание
Timeout	Нет	<p>Время ожидания возникновения дополнительных условий для запуска очередного экземпляра объекта Workflow (задается в минутах).</p> <p>Значение «0» - означает, что во время исполнения объекта должны сохраняться дополнительные условия запуска. Если по каким-либо причинам дополнительные условия запуска прекратят существовать, то исполнение объекта будет отложено;</p> <p>Значение «-1» - означает, что запуск исполнения объекта Workflow будет выполняться без учета дополнительных условий. <i>Если поле не заполнено, то передается значение «-1»</i></p>
Concurrency	Нет	<p>Количество параллельно исполняемых экземпляров объекта Workflow, относящихся к одному CoJob</p> <p><i>Пример: 3</i></p>
Execution	Нет	<p>Порядок исполнения нескольких экземпляров объекта Workflow, относящихся к одному CoJob:</p> <p>FIFO; LIFO; LAST_ONLY; NONE</p> <p>Варианты описаны в документации Oozie Coordinator</p>
Throttle	Нет	<p>Количество экземпляров объекта Workflow, находящихся в состоянии ожидания (по умолчанию: 12)</p>

Атрибуты вложенного объекта Datasets (описывает набор данных, который может быть использован как входное или выходное событие ([подробнее](#))).

Атрибут	Обязательно заполнение	Описание
Name	Да	Название набора данных. <i>Пример: DemoDataset</i>
Frequency	Нет	Периодичность проверок наличия набора данных. Может быть задана в минутах, или выражением на языке Expression Language <i>Пример в минутах: 10;</i> <i>Пример выражения: \${coord:hours(1)}</i>
Initial instance	Нет	Дата и время (в формате UTC) возникновения первого экземпляра набора данных. <i>Пример: 2001-01-01T00:01Z</i>
Timezone	Нет	Часовой пояс (стандарт UTC или GMT). <i>Пример: Etc/GMT-4</i>
Uri Template	Нет	Шаблон Uri для идентификации набора данных. Может быть задан константой или переменным значением. <i>Пример константы: <code>hdfs://cloud:5531/data/\${YEAR}/\${MONTH}</code>;</i> <i>Пример переменной:</i> <i><code>hdfs://cloud:5531/place/\${market}/\${language}</code></i>
Done Flag	Нет	Флаг, обозначающий завершение формирования набора данных

Атрибуты вложенного объекта Input Events (входные события)

Атрибут	Обязательно заполнение	Описание
Name	Да	Название входного события. <i>Пример: DemoInputEvent</i>
Dataset	Нет	Название объекта dataset, который описывает набор данных, используемый в качестве входного события. <i>Пример: DemoDataset</i>
Instance	Нет	Начальная точка для определения Start Instance и End Instance. <i>Пример: \${coord:current(0)}</i>
Start Instance	Нет	Время создания первого контролируемого набора данных. <i>Пример: \${coord:current(-23)}</i>
End Instance	Нет	Время создания последнего контролируемого набора данных. <i>Пример: \${coord:current(0)}</i>

Атрибуты вложенного объекта **Output Events** (выходные события)

Атрибут	Обязательно заполнение	Описание
Name	Да	Название входного события. <i>Пример: DemoOutputEvent</i>
Dataset	Нет	Название объекта dataset, который описывает набор данных, используемый в качестве выходного события. <i>Пример: DemoDataset</i>
Instance	Нет	Начальная точка для определения Start Instance и End Instance. <i>Пример: \${coord:current(0)}</i>
Start Instance	Нет	Время создания первого контролируемого набора данных. <i>Пример: \${coord:current(-23)}</i>
End Instance	Нет	Время создания последнего контролируемого набора данных. <i>Пример: \${coord:current(0)}</i>

Атрибуты вложенного объекта **Action**

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта Action. <i>Пример: DemoAction</i>
Workflow	Да	Привязанный объект Workflow. <i>Пример: DemoWorkflow</i>
SLA	Нет	Настройки SLA мониторинга (см. "Атрибуты вложенного объекта SLA")
Configuration	Нет	Параметры объекта, передаваемые на исполнительную среду: Name - название параметра; Expression - включение параметра означает, что значение является выражением языка Scala. В обратном случае – текстовое значение; Description - описание параметра

Подсистема переноса метаданных

Подсистема переноса метаданных решает задачи экспорта/импорта объектов программы между программными средами. Подсистема переноса может выполнять экспорт/импорт, как всех объектов программы, так и групп объектов.

При работе в режиме экспорта, подсистема переноса метаданных создает комплект файлов, описывающих экспортируемые объекты и внешние связи между ними.

Neoflex Datagram может работать с системами контроля версий данных [TortoiseSVN](#) или [GIT](#), параметры подключения к которым задаются в настройках объектов [Project](#).

Объекты Project. Группировка объектов программы

Группировка объектов программы для экспорта/импорта метаданных выполняется посредством их привязки к одному объекту **Project**. Другими словами, чтобы сгруппировать объекты для переноса, достаточно в форме настройки каждого из объектов в поле **Project** из списка выбрать один и тот же объект **Project**.

На заметку.

*При использовании функции группировки объектов для переноса метаданных необходимо учитывать, что программа, помимо объектов, привязанных к одному объекту **Project**, автоматически добавит связанные с ними объекты. Например, если в группу для переноса входит объект **Workflow**, диаграмма которого управляет исполнением нескольких объектов **Transformation**, то данные объекты **Transformation** также попадут в группу независимо от их привязки к объектам **Project**.*

Работа с объектами **Project** выполняются в разделе интерфейса «**ETL/Project**».

Описание атрибутов объектов Project

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта Project. Необходимо учитывать, что название должно удовлетворять правилам формирования идентификаторов Java
Parent Project	Нет	Родительский проект
VCS Enabled	Нет	Включение параметра означает, что для учета версий файлов, описывающих экспортируемые объекты и внешние связи между ними, используется система контроля версий данных
VCS type	Да, если используется система контроля версий данных	Тип системы контроля версий данных SVN или GIT
VCS URL	Обязательно при включении VCS Enabled	Url-адрес сервера системы контроля версий данных
Connect As Logged In User	Нет	Если параметр включен, то для подключения к SVN/GIT используется логин/пароль текущего пользователя
VCS User Name	Обязательно при включении VCS Enabled	Имя пользователя для аутентификации Neoflex Datagram в системе контроля версий данных
VCS Password	Обязательно при включении VCS Enabled	Пароль пользователя для входа в систему контроля версий данных. <i>Рекомендуется использовать скрытый способ хранения паролей (см. раздел «Хранение паролей в системе»)</i>
VCS Commit Message	Нет	Текст сообщения, которое будет записываться при выполнении операции Commit

Операции экспорта/импорта метаданных

Операции экспорта/импорта метаданных объектов доступны из форм просмотра и редактирования объектов Project по кнопке **«Выполнить операцию»**.

Операции, доступные для объектов Project

Название операции	Описание
Загрузить проект	Выполняет загрузку метаданных из удалённого репозитория в каталог: <code>\${deploy.dir}/cim/MetaServer/data/projects/\${project.name}</code> , затем загружает их в БД postgres
Выгрузить проект	Выполняет выгрузку всех метаданных проекта из БД postgres в каталог: <code>\${deploy.dir}/cim/MetaServer/data/projects/\${project.name}</code> и далее в удалённый репозиторий
Загрузить архив	Выполняет загрузку метаданных проекта из ZIP архива
Выгрузить архив	Выполняет выгрузку всех метаданных проекта из БД postgres в виде ZIP архива
Загрузить репозиторий	Загружает все объекты из каталога <code>\${deploy.dir}/cim/MetaServer/data/all</code>
Выгрузить репозиторий	Подсистема переноса метаданных создает комплект файлов, описывающих все объекты программы, независимо от привязки к объекту Project, из формы просмотра/редактирования которого запущена операция. Данные файлы сохраняются в каталоге: <code>\${deploy.dir}/cim/MetaServer/data/all</code> и могут быть использованы для импорта в другую программную среду
Загрузить данные проекта	Операция загружает данные проекта из каталога: <code>\${deploy.dir}/cim/MetaServer/data/projects/\${project.name}/_data</code>
Выгрузить данные	Операция выгружает любые данные из HDFS
Восстановить ссылки	Операция восстанавливает внешние связи между объектами программы
Удалить потерянные объекты	Системная функция
VCS Checkout	В SVN выполняет команду Checkout. В GIT выполняет команду git clone, если задан url. Если url не задан, команду git init
VCS Update	В SVN выполняет команду Update. В GIT выполняет команду git pull, если задан url. В обратном случае - git checkout
VCS Checkout or Update	Выполняет команду Checkout или Update в зависимости от наличия файлов

Название операции	Описание
VCS Commit	<p>Операция фиксирует внесенные изменения с созданием новой ревизии в репозитории SVN.</p> <p>В GIT операция Commit со списком файлов (вызывается из трансформации) выполняет команду git add для каждого файла, git commit и, если задан url, то git push.</p> <p><i>При выполнении операции commit из центрального репозитория SVN/GIT удаляются файлы, удалённые локально</i></p>
VCS Cleanup	<p>Очистка lock-файлов SVN.</p> <p>В GIT выполняет команду git gc--aggressive</p>

Пример работы с подсистемой переноса метаданных

В главе ["Quick start"](#) в описании процесса создания объектов [Transformation](#) и [Workflow](#) указывалось, что в форме создания каждого из объектов в поле **Project** необходимо выбрать объект **"DemoProject"**. Таким образом была выполнена привязка объектов **Transformation** и **Workflow** к объекту **"DemoProject"** и формирование группы объектов для переноса.

Чтобы создать комплект файлов экспорта объектов программы в другую программную среду, перейдите в форму просмотра объекта **"DemoProject"** и нажмите кнопку **«Выполнить операцию»**, в появившемся списке кликните по операции **«Выгрузить проект»**.

После завершения операции будет сформирован комплект файлов, описывающих объекты программы и связи между ними, в каталоге:

```

${deploy.dir}/cim/MetaServer/data/projects/${project.name}

```

Для того чтобы импортировать объекты, скопируйте каталог **\${project.name}**, полученный в результате выполнения операции **«Выгрузить проект»**, в каталог:

```

${deploy.dir}/cim/MetaServer/data/projects/${project.name}

```

находящийся на другом сервере, где установлена программа.

В интерфейсе программной среды, в которую импортируются объекты создайте объект **Project** с названием, соответствующим названию каталога **\${project.name}**, и выполните операцию **«Загрузить проект»**.

Пример настройки подключения к системе контроля версий данных

Далее приведен пример подключения к системе контроля версий данных GIT. Подключение к системе SVN выполняется аналогичным образом.

Для подключения к системе контроля версий данных GIT выполните следующие действия:

1. В форме создания/редактирования Project включите флаг "VCS Enabled";

2. В поле "VCS Type" установите тип системы контроля версий данных - GIT;
3. В поле "VCS URL" укажите URL внешнего репозитория GIT, например: https://github.com/arch7tect/dg_blueprint.git;
4. Если для доступа Neoflex Datagram к внешнему репозиторию GIT создан отдельный пользователь, тогда в полях "VCS User Name" и "VCS Password" укажите его имя и пароль.
Если необходимо использовать имя и пароль текущего пользователя для доступа к внешнему репозиторию, то включите флаг "Connect As Logged In User";
5. Сохраните настройки;
6. Выполнить команду VCS Checkout Or Update. По этой команде выполнится клонирование внешнего репозитория в каталог `${deploy.dir}/cim/MetaServer/data/projects/${project.name}` сервера DG.

The screenshot shows the 'Project Редактор свойств' (Project Properties Editor) window. The left sidebar contains a search bar and a list of project items: 'Projects.caption', 'Трансформации', and 'Потоки обработки'. The main area displays the configuration for the 'Test project'. The 'Name' field is 'Test project'. The 'Parent Project' field is empty. The 'VCS Enabled' checkbox is checked. The 'VCS Type' dropdown is set to 'GIT'. The 'VCS URL' field contains 'https://github.com/arch7tect/dg_blueprint.git'. The 'Connect As Logged In User' checkbox is checked. The 'VCS User Name' and 'VCS Password' fields are empty. Two red callout boxes highlight the 'Сохранить' (Save) button and the 'Список команд' (List of commands) button.

Адаптация подсистемы Meta Server для работы с разворачиваемым репозиторием

Иногда для разворачивания репозитория может потребоваться адаптация подсистемы **Meta Server**. Адаптация выполняется при помощи настройки объектов **Environment** и вложенных объектов **Environment Parameter**.

Объекты Environment

Данные объекты содержат список параметров среды, которые необходимо изменить для корректной работы подсистемы **Meta Server** с разворачиваемым репозиторием.

Действия с объектами **Environment** выполняются в разделе интерфейса «Развертывание/Environment».

Атрибуты объектов Environment

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java. Внимание! <i>Название объекта Environment должно совпадать со значением параметра запуска customer_code, указанном в файле Install_env</i>
Description	Нет	Описание объекта Environment

Список параметров, которые необходимо отредактировать формируется при помощи вложенных объектов **Environment Parameter**.

Атрибуты вложенных объектов Parameter

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Description	Нет	Описание объекта Parameter
Object class	Да	Класс объекта. <i>Пример: rt.Oozie</i>
Object name	Да	Название объекта. <i>Пример: ambarioozie</i>
Attribute path	Да	Путь к атрибуту, параметр которого необходимо изменить. Задается на языке Expression Language . <i>Пример: nameNode</i>
Parameter value	Да	Новое значение параметра. Строковые значения должны выделяться одинарными кавычками. Любой другой тип значений не выделяется символами. <i>Пример: 'hdfs://cloud3.neo.ru:8020'</i>

Для завершения адаптации подсистемы Meta Server необходимо выполнить одну из операций, доступных для объекта **Environment**.

Операции, доступные для объектов Environment

Название операции	Описание
Перезаписать параметры	Операция устанавливает перезаписываемые параметры, указанные в выбранном объекте Environment
Перезаписать параметры текущей среды	Операция устанавливает перезаписываемые параметры из объекта Environment, название которого соответствует текущему значению параметра запуска customer_code
Зашифровать строку	При помощи данной операции можно зашифровать любой текст, например, чтобы не указывать пароль в явном виде
Расшифровать строку	Операция расшифровывает текстовое значение, зашифрованное при помощи операции "Зашифровать строку"

Sandbox. Инструмент анализа данных

Sandbox предназначен, в первую очередь, для аналитиков, работающих с разнородными данными в том числе и в области BigData. Инструмент позволяет упростить процессы анализа, инжиниринга данных и бизнес-аналитики при помощи визуализации данных, полученных из источников:

- **JDBC** - ссылка на таблицу или результат выполнения SQL запроса (View);
- **Hive** - ссылка на Hive таблицу или ссылка на данные, хранимые в HDFS, для которых предоставляется доступ через Hive;
- **Результат пользовательского запроса** к данным из workspace.

Основная сущность, с которой работает пользователь в Sandbox - это датасеты (Dataset), которые создаются в рамках рабочей области (workspace), при этом:

- для базы данных каждая таблица, или результат выполнения запроса, представляется в виде отдельного датасета;
- если в качестве источника данных используются файлы из HDFS, то содержимое каждого файла представляется в виде отдельного датасета.

Для работы с **Sandbox** необходимо настроить объекты:

- **Cluster**;
- **Workspace**.

Создание объекта Cluster

Объекты **Cluster** хранят параметры подключений к серверу Livy, Hive Metastore, источникам данных и используется при создании Workspaces.

Действия с объектами **Cluster** выполняются в разделе интерфейса «**Sandbox/Cluster**».

Атрибуты объектов Cluster

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Code	Да	Код объекта
Cluster description	Нет	Описание кластера Hadoop
Livy server	Нет	Из списка выбирается сервер Livy
Connection to Hive Thrift Server	Нет	Строка для jdbc подключения к Hive Thrift Server <i>Пример: jdbc:hive2://nchd-rep69-h.moex.com:2181,nchd-rep68-h.moex.com:2181,nchd-rep67-h.moex.com:2181;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2</i>
Hive Metastore URI	Нет	URI Hive Metastore <i>Пример: thrift://nchd-rep69-h.moex.com:9083</i>
References Connection	Нет	Jdbc соединение для БД, в которой будут храниться справочники, структура и наполнение которых определяются пользователями. Так как справочники могут модифицироваться и дополняться позаписно, хранить их в HDFS невозможно

Workspace. Рабочее пространство для анализа данных

Workspace - это основной интерфейс системы Sandbox в котором отображается структура данных, содержимое сформированных датасетов и хранятся инструменты для анализа и инжиниринга данных.

В системе существуют следующие типы Workspace:

- **Analytic workspace** - рабочая область, предназначенная для анализа данных;
- **Jdbc workspace** - позволяет разработчикам загрузить данные из источника и ограничить права доступа к отдельным датасетам;
- **Model pipeline workspace** - рабочее пространство, в рамках которого создается модель обработки данных. Workspace данного типа имеет дополнительные инструменты, позволяющие отслеживать состояния разрабатываемой модели, получать рабочий код для обработки данных и использовать его в разработке, тестировании и продакшене.

Действия с объектами **Workspace** выполняются в разделе интерфейса **«Sandbox/Workspace»**.

Атрибуты Analytic и Model Pipeline workspace

Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Название рабочей области
Shortname	Да	Код рабочей области
Workspace description	Нет	Описание рабочей области
User - Owner	Заполняется автоматически	Имя пользователя, создавшего рабочую область
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочей области, которая является родительской по отношению к редактируемой
Cluster	Нет	Из списка выбирается объект cluster
Project	Нет	Проект, к которому привязан объект
Parameters	Нет	Параметры workspace. Name - Название параметра объекта; Expression - Включение параметра означает, что значение является выражением языка Scala. В обратном случае - текстовое значение; Description - Описание параметра

Jdbc workspace имеет дополнительные атрибуты.

Атрибуты Jdbc workspace

Атрибут	Обязательно заполнение	Описание
Jdbc connection	Нет	Из списка выбирается объект Jdbc connection, который необходимо использовать для подключения к базе данных
Default schema to work	Нет	Из списка выбирается объект Scheme, описывающий базу данных


Наполнение Workspace

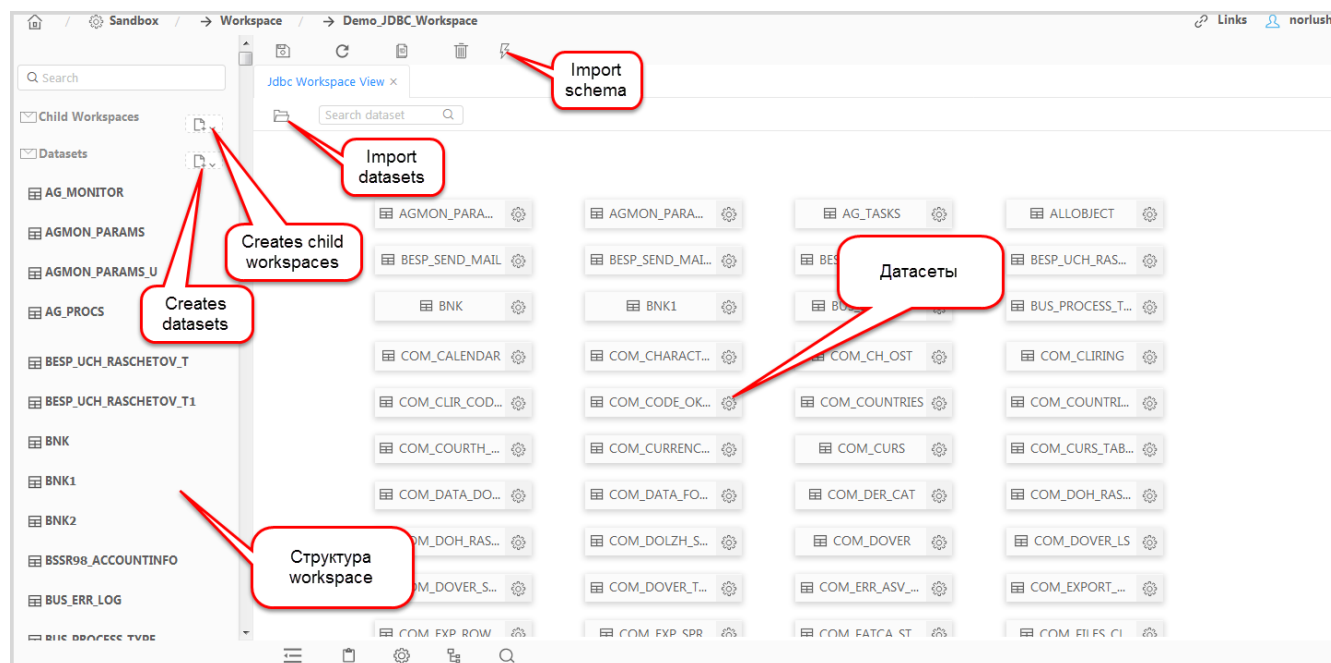
После создания workspace (analytic или Jdbc), ее необходимо наполнить данными, другими словами создать датасеты и, если это необходимо, создать дочерние workspace.

Датасеты - это основная единица работы с данными в рамках приложения, соединяет в себе непосредственно набор данных, способ их получения, и структуру с бизнес-описанием атрибутов.


Child workspaces - дочерние рабочие среды, данные из которых будут доступны в рабочей workspace. Независимо от типа родительской workspace, Child workspaces могут быть любого типа.

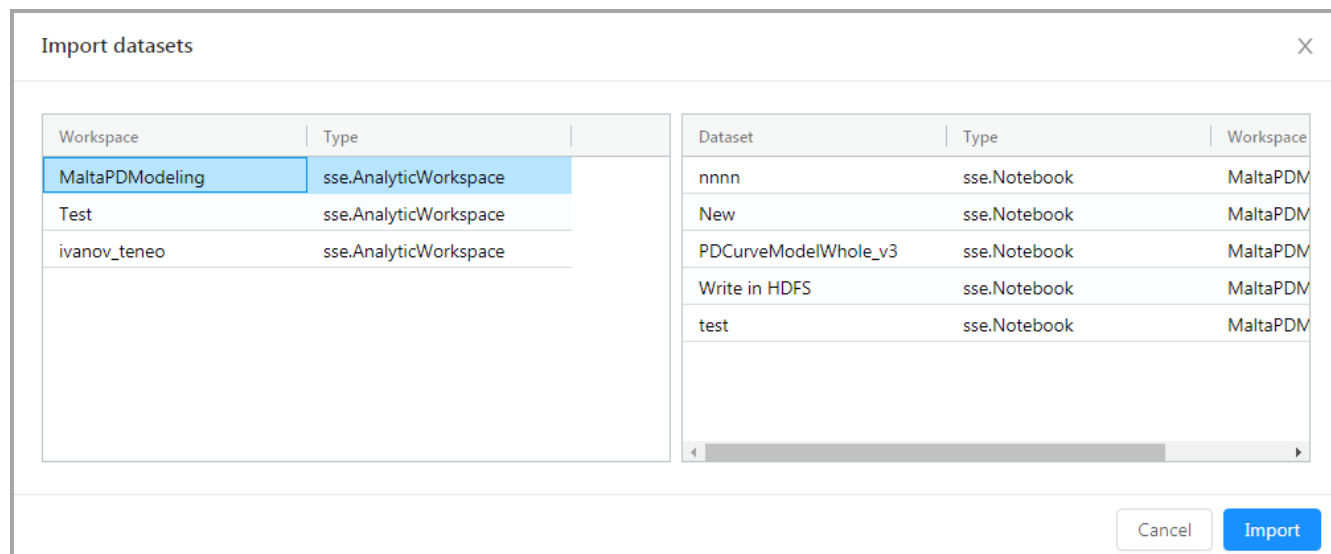
Импорт схемы в Jdbc workspace

В **Jdbc workspace** существует возможность автоматически создать датасеты для всей базы данных по кнопке  (Import schema). При этом, в настройках Jdbc workspace обязательно должны быть указаны атрибуты **Jdbc connection** и **Default schema to work**. После загрузки базы данных, на экране будут показаны датасеты в виде блоков, а в левой части экрана представлена структура Workspace.




Импорт датасетов

Во всех типах Workspace можно импортировать уже созданные датасеты из других workspace. Для этого кликните по кнопке , на экране появится окно выбора датасетов для импорта.



В окне выберите workspace, в которой находятся нужные датасеты, а затем отметьте датасеты, которые необходимо импортировать. Для завершения импорта нажмите кнопку "Import".

Создание датасетов и child workspaces

Чтобы создать датасет и/или child workspace, нажмите кнопку , расположенную рядом с соответствующим пунктом на панели со структурой workspace. Описание видов датасетов и их атрибуты описаны в главе "[Виды датасетов](#)"

Виды датасетов

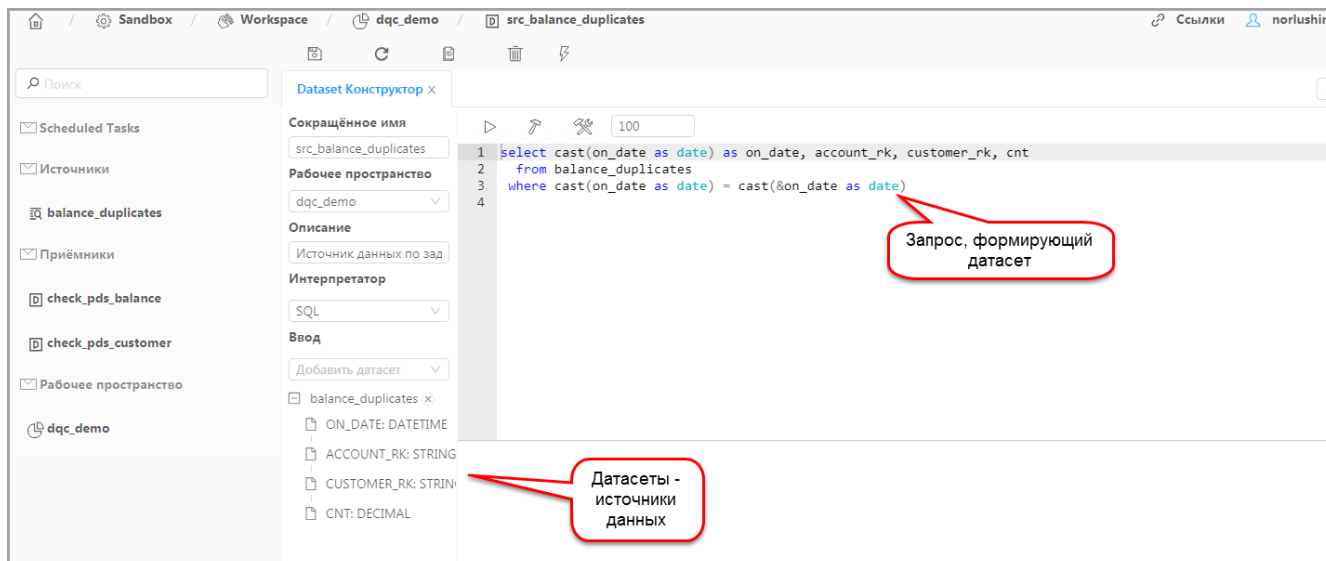
Dataset

Данный тип на основании различных источников данных, описывает логику трансформации данных и получившуюся структуру.





Атрибуты Dataset

Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Used datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Used Interpreter	Да	Интерпретатор выражения: Spark; Python; R; SQL
Expression	Нет	Выражение, для формирования датасета

На рисунке ниже представлен пример настройки датасета.



Для формирования данных в датасете, используйте одну из команд:

-  (Run) - запускает выполнение запроса и отображает результат в пространстве интерфейса под запросом (см. рисунок выше). Используется для отладки;
-  (Build) - запускает выполнение запроса и сохраняет результат в HDFS;
-  (Full rebuild) - пересчитывает все датасеты - источники данных, выполняет запрос настраиваемого датасета и сохраняет результат в HDFS;
-  (Schedule full rebuild) - запускает операцию "Full rebuild" и создает объект "[ScheduledTask](#)" для последующего запуска операции по расписанию.

По кнопке  доступны формы:

- Конструктор - форма создания/изменения свойств датасета (показана на рисунке выше);
- Dataset метаданные - форма просмотра метаданных;
- Dataset обзор - форма для визуализации данных;
- Редактор свойств - расширенная форма, которая позволяет отредактировать свойства датасета, задать/отредактировать настройки доступа к датасету, просмотреть историю изменений.

Hive dataset

Hive dataset – описывает существующую таблицу в Hive Metastore, ее структуру, и предоставляет доступ к данным через Hive Thrift Server.


Атрибуты Hive dataset

Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Hive Database	Нет	Название базы данных
Hive Table Name	Нет	Название таблицы

По кнопке  доступны формы:

- Метаданные - форма просмотра метаданных;
- Обзор - форма для визуализации данных;
- Редактор свойств.

Операции Hive dataset

- Import metadata - данная операция импортирует метаданные из источника данных. Операция доступна по кнопке .

Hive external dataset

Hive external dataset – описывает набор данных, расположенных в файлах на HDFS (в настоящий момент поддерживаются форматы parquet, orc, json, xml, csv). Предоставляет доступ к данным, как к external таблице Hive через Hive Thrift Server.

Атрибуты Hive external dataset

Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Hive Database	Нет	Название базы данных
Hive Table Name	Нет	Название таблицы
HDFS Dataset Location	Нет	Путь к файлу-источнику данных в HDFS
File type	Нет	Из списка выбирается формат файла-источника данных

По кнопке  доступны формы:

- Метаданные - форма просмотра метаданных;
- Обзор - форма для визуализации данных;
- Редактор свойств.

Операции Hive external dataset

Операции доступны по кнопке .

- Import metadata - данная операция импортирует метаданные из источника данных.
- Build hive table - создает external table в Hive.

Jdbc dataset

Jdbc dataset – аналог view над реляционными данными из jdbc совместимого источника.

Атрибуты Jdbc dataset

Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Dataset read connection	Нет	Название объекта Jdbc Connection, используемого для подключения к базе данных
Dataset read query	Нет	SQL запрос, для формирования датасета

В текстовом поле записывается выражение, которое сформирует датасет.

Для запуска выполнения запроса нажмите кнопку "Run".

По кнопке  доступны формы:

- Конструктор - форма быстрого создания/изменения свойств датасета;
- Метаданные - форма просмотра метаданных;
- Обзор - форма для визуализации данных;
- Редактор свойств - расширенная форма, которая позволяет отредактировать свойства датасета, задать/отредактировать настройки доступа к датасету, просмотреть историю изменений.

Jdbc table dataset

Jdbc table dataset – описывает существующую таблицу в jdbc источнике данных.


Атрибуты Jdbc table dataset

Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Dataset read connection	Нет	Из списка выбирается объект Jdbc Connection
Database schema	Нет	Из списка выбирается объект Scheme
Database table	Нет	Название таблицы - источника данных

По кнопке  доступны формы:

- Метаданные - форма просмотра метаданных;
- Обзор - форма для визуализации данных;
- Редактор свойств.

Операции Jdbc table dataset

- Import metadata - данная операция импортирует метаданные из источника данных. Операция доступна по кнопке .

Linked dataset

Linked dataset – вспомогательная сущность. Позволяет ссылаться на датасеты, которые подготовили и опубликовали другие аналитики или разработчики.

Атрибуты Linked dataset


Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Linked dataset	Нет	Название датасетов из которого берутся данные

По кнопке  доступны формы:

- Метаданные - форма просмотра метаданных;

- Обзор - форма для визуализации данных;
- Редактор свойств.

Операции Linked dataset

- Import metadata - данная операция импортирует метаданные из источника данных. Операция доступна по кнопке .

Reference dataset

Reference dataset – вспомогательная сущность. Позволяет аналитику создавать и, в дальнейшем, дополнять справочники для классификации данных в существующих датасетах.

Атрибуты Reference dataset


Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя датасета
Short name	Да	Код датасета
Dataset description	Нет	Описание датасета
User - Owner	Заполняется автоматически	Имя пользователя, создавшего датасет
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан датасет
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к датасету
Dataset columns	Нет	Array Scalar Struct
Partition columns	Нет	Список колонок, по которым будут партицироваться данные
Primary key	Нет	Значение Primary key

По кнопке  доступны формы:

- Метаданные - форма просмотра метаданных;
- Обзор - форма для визуализации данных;

- Редактор свойств.

Операции Reference dataset

Операции доступны по кнопке .

- Recreate table - пересоздает таблицу с данными.
- Load CSV file - загружает данные из CSV файла в датасет.

Notebook

Notebook - аналог Zeppelin Notebook.

Атрибуты Notebook

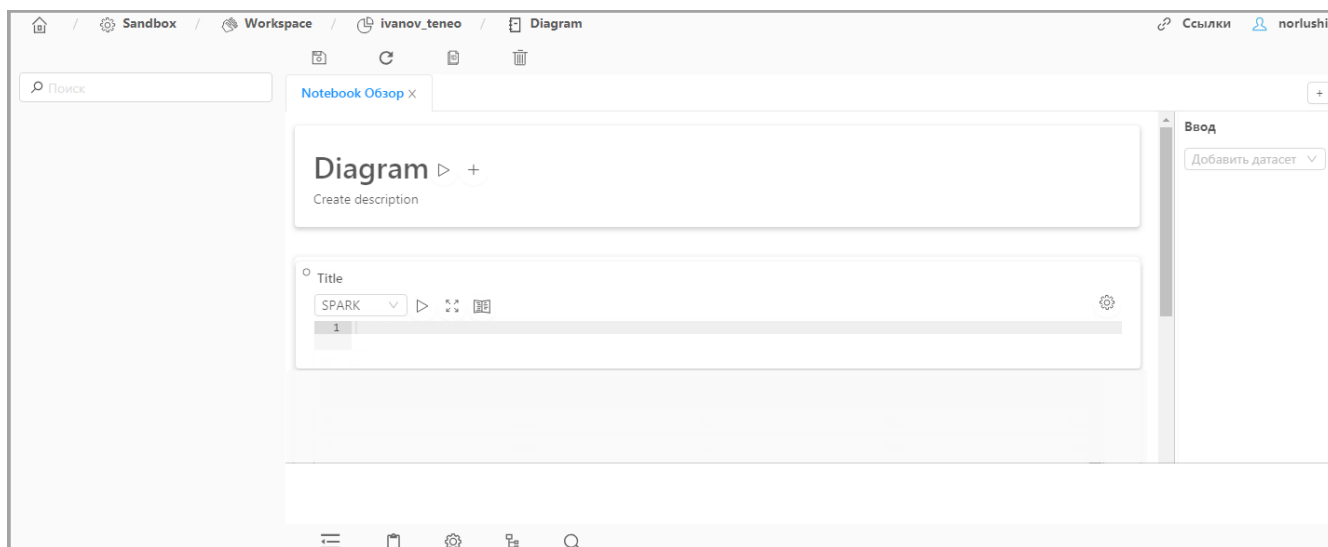
Атрибут	Обязательно заполнение	Описание
Name	Заполняется автоматически	Имя notebook
Short name	Да	Код notebook
Dataset description	Нет	Описание notebook
User - Owner	Заполняется автоматически	Имя пользователя, создавшего notebook
Group - Owner	Заполняется автоматически	Название группы пользователей
Parent Workspace	Заполняется автоматически	Имя рабочего пространства, в котором существует датасет
Project	Нет	Проект, к которому привязан notebook
Datasets	Нет	Список зависимых датасетов
Dataset access permissions	Нет	Настройки доступа к notebook
Notebook paragraphs	Нет	Список параграфов в notebook
Parameters	Нет	Параметры объекта notebook. Name - Название параметра; Value - Значение параметра; Expression - Включение параметра означает, что значение является выражением языка Scala. В обратном случае - текстовое значение; Description - Описание параметра



По кнопке  доступны формы:

- Обзор - форма для визуализации данных;
- Редактор свойств.

Работа с notebook



После создания notebook на экране отобразится форма Notebook view. Если при создании notebook были указаны параграфы (поле "Notebook paragraphs"), то данные параграфы также отобразятся в форме.



Для добавления новых параграфов используйте кнопки , расположенную в заголовке notebook, или , которая появляется при подведении курсора к центру пространства под параграфом.

После создания необходимого количества параграфов, для каждого из них выберите интерпретатор кода и напечатайте запрос или код на соответствующем языке. Если в коде должны быть использованы данные из других датасетов, то их необходимо добавить в notebook, для этого кликните по полю "Добавить датасет", расположенном на панели в правой части экрана.

Для формирования данных в notebook запустите выполнение кода одним из вариантов:

- По кнопке , расположенной в заголовке notebook будет запущено выполнение всех параграфов;
- По кнопке , расположенной в блоке параграфа будет запущено выполнение данного параграфа.

После запуска в параграфах отобразятся результаты выполнения кода.

Diagram

Create description

SQL


```
1 select * from cfit
```

time	n_risk	n_event	n_censor	surv
1	5501	70	488	0.9902
2	4943	53	486	0.9820
3	4404	52	486	0.9731
4	3866	49	521	0.9636
5	3296	48	495	0.9526

SPARK

```
1 spark.sql("show tables").show()
```

database | tableName | isTemporary

По кнопке , расположенной в блоке параграфа, доступен список настроек вывода данных и редактора кода данного параграфа.

Работа с датасетами

Просмотр данных датасета

Переход на вкладку "Dataset view" для просмотра данных возможен двумя способами:

1. Откройте датасет, нажмите кнопку  и выберите пункт "View".

Dataset Constructor

Short name: src_balance_duplicates

Workspace: dqc_demo

Description: Источник данных по зад

Interpreter: SQL

Input: Add dataset


balance_duplicates

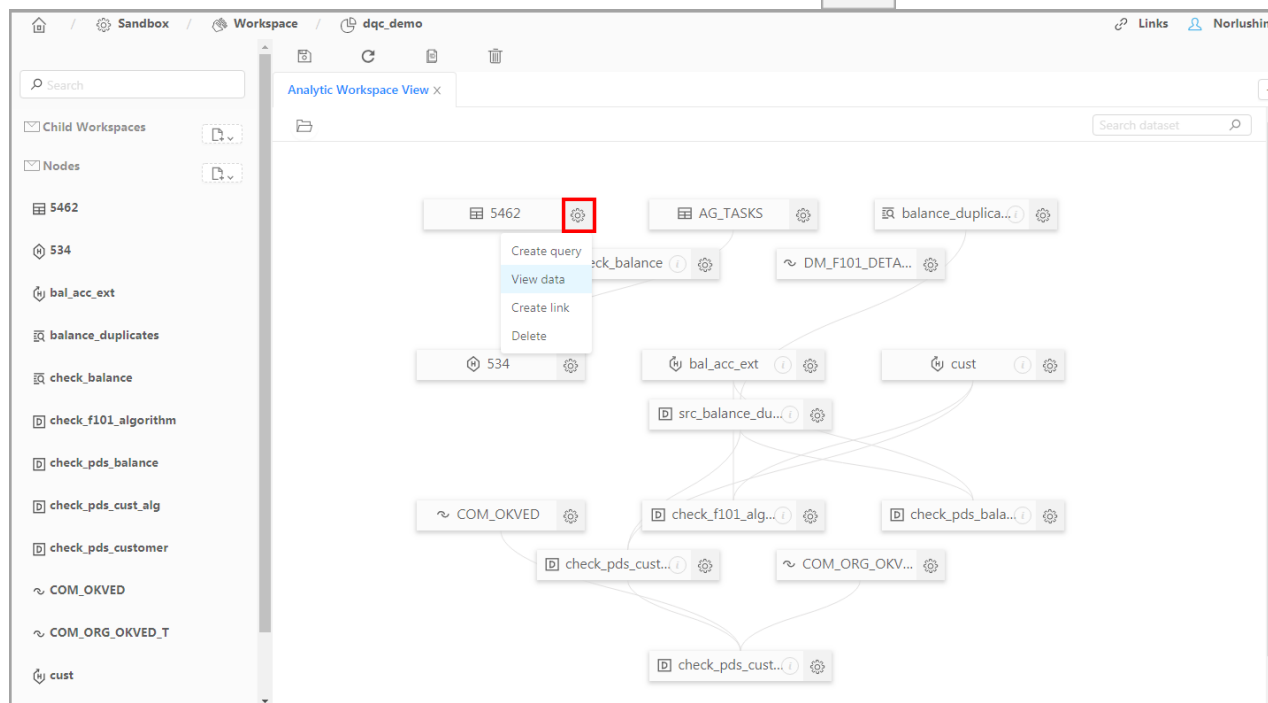
- ON_DATE: DATETIME
- ACCOUNT_RK: STRING
- CUSTOMER_RK: STRING
- CNT: DECIMAL

```
1 select cast(on_date as date) as on_date, account_rk, customer_rk, cnt
2 from balance_duplicates
3 where cast(on_date as date) = cast(@on_date as date)
4
```

Constructor

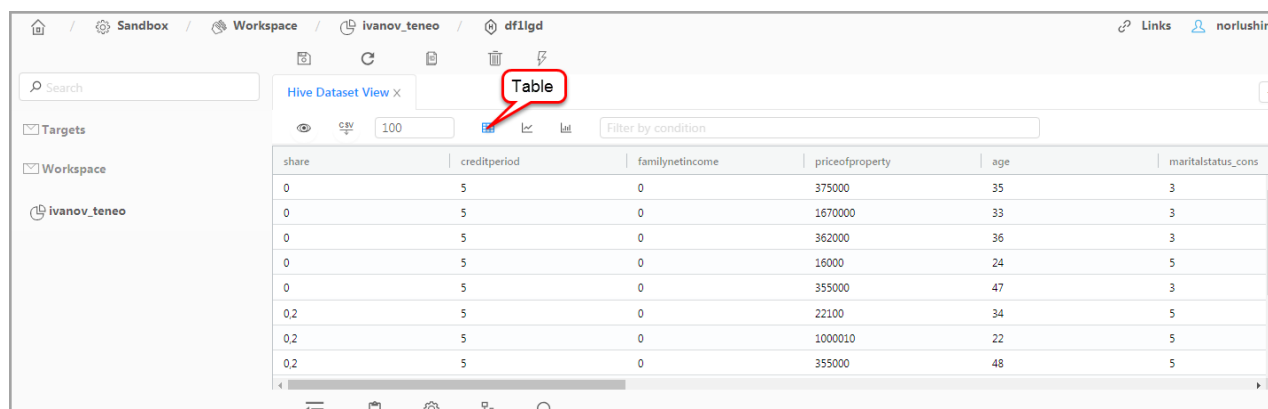
- Metadata
- View
- Properties Editor

2. Откройте workspace, найдите нужный датасет, нажмите кнопку  и выберите пункт "View data".



На вкладке "View" данные могут быть представлены в виде:

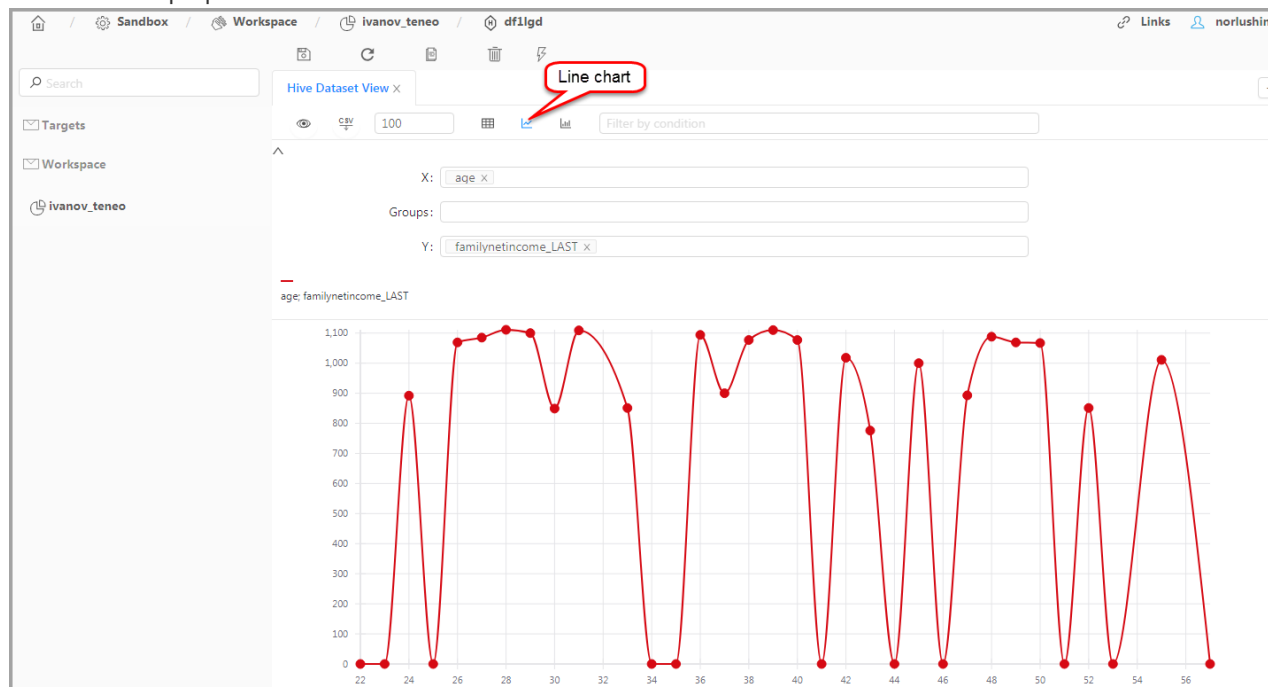
- Таблицы.



The screenshot shows the 'Hive Dataset View' for the 'df11gd' dataset. The view is set to 'Table' (indicated by a red box and a red arrow). The table displays data with columns: share, creditperiod, familynetincome, priceofproperty, age, and maritalstatus_cons. The data is filtered by condition.



share	creditperiod	familynetincome	priceofproperty	age	maritalstatus_cons
0	5	0	375000	35	3
0	5	0	1670000	33	3
0	5	0	362000	36	3
0	5	0	16000	24	5
0	5	0	355000	47	3
0.2	5	0	22100	34	5
0.2	5	0	1000010	22	5
0.2	5	0	355000	48	5

- Линейного графика.



- Гистограммы.

Просмотр метаданных датасета

Для просмотра метаданных, перейдите на вкладку "Dataset metadata". Для этого выберите пункт "Metadata" в списке, открываемом кнопкой  в открытом датасете, или кнопкой  в открытом workspace.


На вкладке "Dataset metadata" представлены:

- Common - метаданные датасета.



The screenshot shows the 'Dataset metadata' view for a 'Jdbc Table Dataset'. The 'Common' tab is active, displaying various metadata fields. A red callout box labeled 'List of operations' points to a button in the top right corner of the interface.

- Dataset columns - на вкладке представлены метаданные, импортированные из источника данных датасета.

Column name	Column type	Column description	Allow nulls
ID	DECIMAL		<input type="checkbox"/>
GID	STRING		<input checked="" type="checkbox"/>
ARC	DECIMAL		<input checked="" type="checkbox"/>
S_MDT	DATETIME		<input checked="" type="checkbox"/>
NAME	STRING		<input type="checkbox"/>
UNITOID	DECIMAL		<input type="checkbox"/>

Для импорта/обновления метаданных в датасете используйте операцию "Import metadata", которая доступна по кнопке  (данная операция доступна не для всех датасетов).

Редактирование атрибутов датасета

Редактирование атрибутов датасета выполняется на вкладке "Properties editor". Для перехода на вкладку выберите пункт "Properties editor" в списке, открываемом кнопкой  в открытом датасете, или кнопкой  в открытом workspace. Атрибуты датасетов описаны в соответствующих пунктах раздела ["Виды датасетов"](#).

Business process

Атрибуты Business process

Атрибут	Обязательно заполнение	Описание
Business process	Да	Название бизнес-процесса
Target object class	Нет	Объект, над которым осуществляется бизнес-процесс. Этот объект будет менять состояния в соответствии с описанием бизнес-процесса. Например, бизнес-процесс "Разработка модели" может содержать состояния: Разработка, Тестирование, Продакшн
States	Нет	Состояния, в которых может находиться target object в ходе выполнения бизнес-процесса (атрибуты States описаны в таблице ниже). Например, объект "Модель" может находиться в состояниях: Разработка, Тестирование, Продакшн
Description	Нет	Описание бизнес-процесса
Start state	Нет	Начальное состояние target object в бизнес-процессе, таким образом, при создании бизнес-процесса и связи его с target object, последний всегда будет находиться в этом состоянии. Например, объект "Модель" при создании бизнес-процесса "Разработка модели", всегда в начале находится в состоянии Разработка

Атрибуты States

Атрибут	Обязательно заполнение	Описание
State name	Да	Название состояния бизнес-процесса
Transitions	Нет	Возможные переходы из состояния. Например, из состояния Разработка target object может перейти только в Тестирование, а из Тестирования можно вернуть в разработку или перевести в Продакшн. Transition - название перехода в бизнес-процессе; Description - описание перехода; Next state - следующее состояние target object в бизнес-процессе; Action - действие, которое будет выполнено при выполнении перехода. Действие описывается на Groovy; Allowers to user roles - список ролей, пользователям из состава которых доступна функция перевода target object в следующее состояние; Guard condition - условие, которое должно быть выполнено для перехода. Условие описывается на Groovy
Description	Нет	Описание состояния бизнес-процесса

Дополнительные возможности программы

Подключение к серверу Zeppelin

Для подключения к серверу, в разделе "Сервер/Zeppelein" создайте объект типа Zeppelin.

Атрибуты объектов Zeppelin

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Project	Нет	Проект, к которому привязан объект
Http	Да	Url-адрес сервера Zeppelin
User name	Да	Имя пользователя для входа на сервер Zeppelin
Password	Да	Пароль для входа на сервер Zeppelin

Для перехода на главную страницу сервера нажмите кнопку , расположенную в строке "Http".


Синхронизация с сервером Atlas

Для подключения к серверу, в разделе "Сервер/Atlas" создайте объект типа Atlas.

Атрибуты объектов Atlas

Атрибут	Обязательно заполнение	Описание
Atlas service	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Project	Нет	Проект, к которому привязан объект
Http url	Да	Url-адрес сервера Atlas
User name	Да	Имя пользователя для входа на сервер Atlas
Password	Да	Пароль для входа на сервер Atlas

Для перехода на главную страницу сервера нажмите кнопку  , расположенную в строке "Http".

Для синхронизации данных с сервером Atlas выполните операцию "Опубликовать схемы", которая доступна по кнопке .

Перенос данных из CSV файлов в базу данных внешней системы

В ситуациях, когда необходимо выполнить перенос данных из большого количества CSV файлов в базу данных внешней системы для каждого CSV файла нужно настроить свой объект **Transformation**, а для управления трансформациями создать объект **Workflow**. Выполнение такого объема настроек может занять значительное количество времени. Для решения данной проблемы в программе разработаны объекты **StagingArea**.

Объекты Staging Area

Атрибуты объектов **Staging Area**, а также атрибуты вложенных объектов, описывают параметры переноса данных из CSV файлов в базу данных и позволяют автоматизировать создание объектов **Transformation** и **Workflow** для выполнения переноса.

Действия с объектами **Staging Area** выполняются в разделе интерфейса «**Инструменты/Staging Area**».

Атрибуты объектов Staging Area

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта. При указании имени необходимо учитывать, что оно должно удовлетворять правилам формирования идентификаторов в языке Java
Oozie	Нет	Объект Oozie, для подключения к серверу Oozie
Deployments	Нет	Список объектов Deployment, при помощи которых выполняется доступ к базам данных внешних систем
Project	Нет	Проект, к которому привязан объект
Context	Нет	В поле устанавливается объект JdbcContext, который определяет настройки подключения к нужной базе данных
Software system	Нет	Объект Software system, который описывает внешнюю систему, к которой будет выполнено подключение
Catalog	Нет	Путь к каталогу, в котором хранятся CSV файлы
CSV Options	Нет	Delimiter - символ, используемый в качестве разделителя между значениями в CSV; Date Format - формат даты, используемый в файле; Charset - Кодировка, используемая в файле; Header - Если параметр включен, то пропускается первая строка при чтении данных из CSV файла

Параметры объектов Staging Area

Параметр	Обязательно заполнение	Описание
Name	Да	Название параметра объекта
Value	Нет	Значение параметра
Expression	Нет	Включение параметра означает, что значение является выражением языка Scala. В обратном случае - текстовое значение
Description	Нет	Описание параметра

Атрибуты вложенного объекта Default Fields

Параметр	Обязательно заполнение	Описание
Name	Да	Название объекта Default Field
Field	Нет	Название поля столбца в таблице
Expression	Нет	Выражение на языке Scala, используемое для формирования значения, которое записывается в столбец таблицы
Static	Нет	Если параметр включен, то значение рассчитывается один и записывается во все строки столбца. В обратном случае значение рассчитывается для каждой строки

Атрибуты вложенного объекта Tables

Параметр	Обязательно заполнение	Описание
Name	Да	Название объекта Default Field
Table	Нет	Название таблицы, в которую переносятся данные из CSV файла
File Name	Нет	Название файла, из которого переносятся данные
Columns	Нет	Список полей с данными в CSV файле
Default Fields	Нет	Список столбцов в таблице
Mappings (используется в случае несоответствия названий полей в CSV файле и таблице)	Нет	From - название поля в CSV; To - название столбца таблицы, в который переносятся данные
CSV Options (используется, если в конкретном файле параметры CSV отличаются от заданных для объекта Staging Area)	Нет	Delimiter - символ, используемый в качестве разделителя между значениями в CSV; Date Format - формат даты, используемый в файле; Charset - Кодировка, используемая в файле; Header - Если параметр включен, то пропускается первая строка при чтении данных из CSV файла

Операции, доступные для объектов Staging Area

Название операции	Описание
Сгенерировать Workflow	<p>При выполнении операции программа автоматически генерирует объекты Transformation, Workflow и WorkflowDeployment, необходимые для исполнения трансформации на сервере Oozie.</p> <p>Названия создаваемых объектов формируются по правилам:</p> <pre>workflow.name = stagingArea.name + _workflow; workflowDeployment.name = stagingArea.name + _workflowDeployment; transformation.name = stagingArea.name + _stagingTable.table.name</pre>

Запуск операций объектов по расписанию

Объекты "Scheduled task" позволяют настроить запуск любых операций любых объектов программы по расписанию.

Действия с объектами "Scheduled task" выполняются в разделе интерфейса «Сервер/Scheduled task».

Атрибуты объектов Scheduled task

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта "Scheduled task"
Project	Нет	Объект "Project", к которому привязан объект "Scheduled task"
Enable	Нет	Включить/выключить работу "Scheduled task"
Entity Type	Нет	Тип сущности (см. таблицу)
Entity Name	Нет	Название сущности для которой будет применяться запуск по расписанию (имя объекта)
Method name	Нет	Имя метода (см. таблицу)
Scheduler	Нет	<p>Типы расписания:</p> <p>Chron - устанавливается расписание запусков заданий при помощи cron выражения (подробнее);</p> <p>Delay - запуск задания через указанный период времени. Началом отсчета интервала времени для каждого последующего периода запуска является окончание предыдущего запуска задания;</p> <p>Once - единичный запуск задания по времени, указанном в поле "Start time". Если "Start time" не задан, то запуск будет произведен единожды при выполнении операции "Refresh scheduler". Параметр Disable After Run позволяет настроить отключение объекта "Scheduled task" после запуска задания;</p> <p>Period - запуск задания через указанный период времени. Данный тип расписания не отслеживает завершенность предыдущего запуска задания, это необходимо учитывать, т.к. может произойти наложение запусков</p>
Backoff Policy	Нет	<p>Поддерживаемые типы Backoff Policy (подробнее):</p> <p>Exponential Backoff Policy;</p> <p>Exponential Random Backoff Policy;</p> <p>Fixed Backoff Policy;</p> <p>No Backoff Policy;</p> <p>Uniform Random Backoff Policy</p>
Retry Policy	Нет	<p>Поддерживаемые типы Retry Policy (подробнее):</p> <p>Always Retry Policy;</p> <p>Never Retry Policy;</p> <p>Simple Retry Policy;</p> <p>Timeout Retry Policy</p>
Last Schedule Time	Заполняется автоматически	Расчетное время последнего запуска по расписанию

Атрибут	Обязательно заполнение	Описание
Last Run Time	Заполняется автоматически	Реальное время последнего запуска
Last Error Time	Заполняется автоматически	Время возникновения ошибки запуска
Last Error	Заполняется автоматически	Описание ошибки

Операции, доступные для объектов Scheduled task

Название операции	Описание
Refresh Scheduler	Операция запускает выполнение Scheduled task с измененными настройками, если такие вносились

Соответствие типов объектов и Entity type

Тип объекта	Entity name
JDBC Connection	rt.JdbcConnection
Software System	rt.SoftwareSystem
Deployment	rt.Deployment
Workflow Deployment	rt.WorkflowDeployment
Transformation Deployment	rt.TransformationDeployment
Coordinator deployment	rt.CoordinatorDeployment
Transformation	etl.Transformation
Workflow	etl.Workflow
Project	etl.Project
Workspace	sse.Workspace
Environment	rt.Environment
Scheduled task	rt.ScheduledTask

Соответствие операций и Method name

Название операции	Method name
Протестировать	test
Обновить схему	refreshScheme
Проверить	validate
Сгенерировать	generate
Собрать	build
Скопировать	deploy
Сгенерировать и скопировать	install
Запустить	run runit (для объектов Workflow)
Сгенерировать и запустить	generateAndRun
Собрать и запустить	buildAndRun
Текущее состояние	getStatus
Импорт	importTransformation (для объектов Transformation) importWorkflow (для объектов Workflow)
Экспорт	exportTransformation (для объектов Transformation) exportWorkflow (для объектов Workflow)
Зависимости	dependencies
Создать представление	setJsonValue
Загрузить проект	importProject
Выгрузить проект	exportProject
Загрузить архив	uploadArchive
Выгрузить архив	downloadArchive
Загрузить репозиторий	importRepo
Выгрузить репозиторий	exportRepo
Очистить проект	clear
Восстановить ссылки	importScripts
Удалить потерянные объекты	clearLost
SVN Checkout	svnCheckout
SVN Update	svnUpdate

Название операции	Method name
SVN Checkout or Update	svnCheckoutOrUpdate
SVN Commit	svnCommit
SVN Cleanup	svnCleanup
Перезаписать параметры	rewriteParameters
Перезаписать параметры текущей среды	rewriteCurrent
Зашифровать строку	encryptString
Расшифровать строку	decryptString

Streaming. Потокковая обработка данных

Streaming - система потоковой обработки данных, которая позволяет анализировать входящий поток данных, получаемых из Kafka или из других систем в форматах *.json или *.avro.

Данная система может быть использована в различных сферах бизнеса, где требуется беспрерывный анализ входящего потока событий, например: анализ поведения пользователя на страницах сайта или контроль перемещения автомобилей при организации перевозок и т.д.

Для организации потоковой обработки данных необходимо создать и настроить объекты:

- Events processor;
- [Transformation](#) (схема трансформации обязательно должна содержать элемент [Group with state](#) при помощи которого в схему включаются правила обработки входящего потока данных).

Объекты Events processor


Объекты Events processor хранят описания структур входящего и исходящего потока данных, а также правила их обработки. Настроенные объекты Events processor используются в схемах трансформации. Включение Events processor в схему трансформации выполняется при помощи элемента [Group with state](#).

Действия с объектами **Events processor** выполняются в разделе интерфейса **«Streaming/Events processor»**.

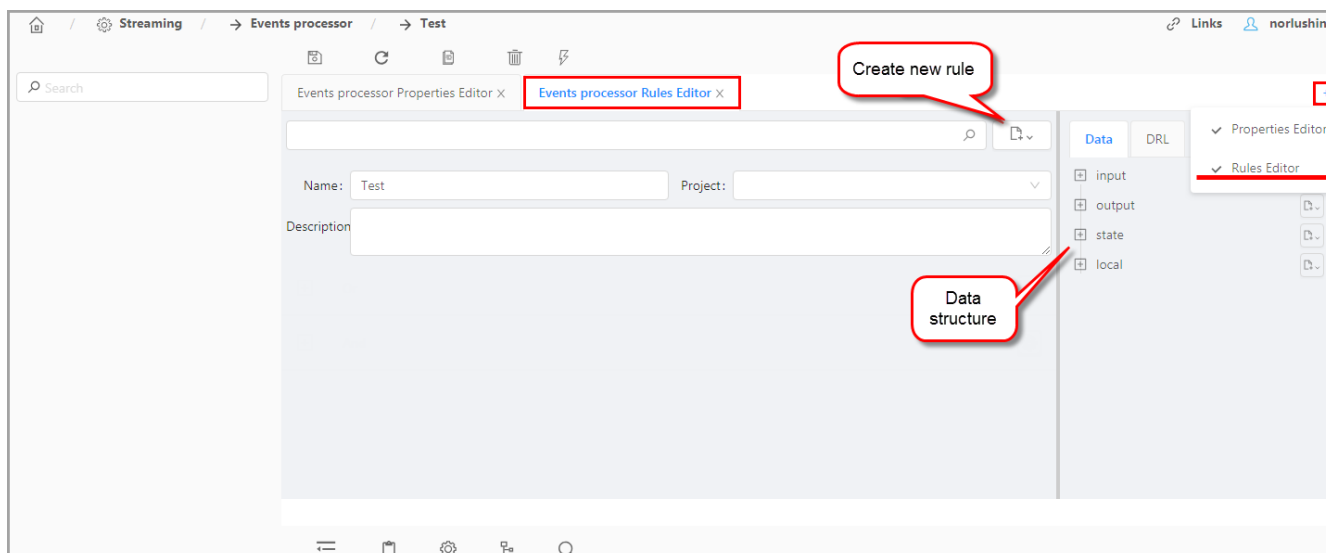
Атрибуты объектов Events processor

Атрибут	Обязательно заполнение	Описание
Name	Да	Название объекта
Description	Нет	Описание объекта
Project	Нет	Объект "Project", к которому привязан объект "Events processor"
Input structure	Да	Описание структуры входящего потока данных
Internal state structure	Нет	Структура данных с промежуточными результатами обработки входящего потока, рассчитанными между событиями. Internal state structure сохраняются между вызовами
Output structure	Нет	Описание структуры исходящего потока данных
Local variables	Нет	Структура данных с промежуточными результатами обработки входящего потока, рассчитанными между правилами. Local variables не сохраняются между вызовами
Rules	Нет	Список правил анализа событий <i>Между собой все правила объединяются логическим оператором AND</i>

Настройка потоков данных и правил анализа

После создания объекта Events processor перейдите на вкладку "Rule editor", которая открывается по кнопке , расположенной в правой части экрана.

Если при создании нового Event processor не были созданы структуры данных, то их необходимо создать. Создание структуры данных выполняется на панели, расположенной в правой части экрана.



После того, как структура данных создана, можно перейти к созданию правил анализа событий.

Новые правила создаются по кнопке  (см. рисунок выше).

Примечание.
Количество создаваемых правил не ограничено. Между собой правила верхнего уровня объединяются логическим оператором AND.

Каждое правило состоит из условия, которое задается в строке When и действия, которое задается в строке Then.


Name

☐

Modify fact


When

All of this are true



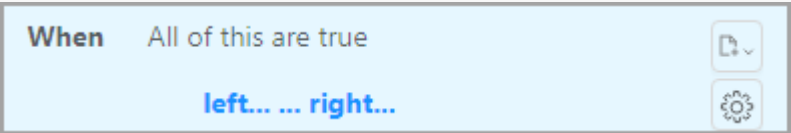
Then

Fire all this actions



Примечание.
Если в Drools используется функция "Modify", тогда включите функцию Modify fact.

Типы условий

Название условия	Описание
AND	Создает условие "All of this are true" внутри условия
OR	Создает условие "Any of this are true" внутри условия
Expression	Позволяет записать условие в виде выражения на языке Mvel
Params	<div>Условие сравнивает два параметра.</div> <div></div> <div>Для настройки условия необходимо выбрать два параметра, кликнув по left и right, и выбрать знак сравнения, кликнув по "..."</div>
Params-Expression	Условие позволяет сравнивать значение параметра с результатом выполнения выражения на языке Mvel . Настройка условия схожа с настройкой условия Params, только вместо параметра right записывается выражение

Типы действий

Название действия	Описание
Function	Вызов пользовательской функции
Set value	Установка значения переменной
Map value	Присваивает переменной значение другой переменной
Timeout	Установка таймаута для группы
Add output	Добавляет выходной факт
Remove state	Прекращает обработку потока событий с указанным значением состояния
Initialize fact	

Объекты Function

Объекты Function предназначены для создания пользовательских функций, которые можно использовать в качестве действий в правилах Events processor.

Действия с объектами **Function** выполняются в разделе интерфейса **«Streaming/Function»**

Атрибуты объектов Function

Атрибут	Обязательно заполнение	Описание
Library	Нет	Название библиотеки, к которой привязан объект Function. Библиотеки создаются в разделе интерфейса: "Streaming/Functions library"
Text pattern	Нет	Описание объекта, которое отображается в окне поиска функции при настройке правил анализа событий
Name	Да	Имя объекта
Input parameters	Да	Описание входящих параметров
Output parameters	Да	Описание исходящих параметров
Scala source code	Нет	Код функции на языке Scala
Description	Нет	Описание кода функции

SLA мониторинг задач Oozie

SLA позволяет ([подробнее](#)):

1. Посылать сообщения, если в ходе выполнения задача не уложилась во временные рамки, указанные в ее настройках (задачи Oozie - это Coordinator Jobs и Workflow, а также Task запущенные в ходе Workflow);
2. Показывать результат на закладке «SLA» в Oozie web Console.

Для того чтобы появилась закладка, настройте Oozie, для этого в site.xml укажите:

1. `<property>`
2. `<name>oozie.services.ext</name>`
3. `<value>`
4. `org.apache.oozie.service.EventHandlerService,org.apache.oozie.sla.service.SLAService`
`</value>`
5. `</property>`
6. `<property>`
7. `<name>oozie.service.EventHandlerService.event.listeners</name>`
8. `<value>`
9. `org.apache.oozie.sla.listener.SLAJobEventListener,org.apache.oozie.sla.listener.SLAEmailE`
`ventListener`
10. `</value>`
11. `</property>`

Пример настройки Workflow для SLA.

Sla

SlaDefinition

Nominal Time
nominal_time

Should Start
10 * MINUTES

Should End
20 * MINUTES

Max Duration
30 * MINUTES

Alert Events
start_miss,end_miss,duration_miss

Alert Contact
psimanihin@neoflex.ru

1. Точка отчета для SLA - это переменная nominal_time, такая переменная создается автоматически при запуске задачи из Datagram, туда записывается текущее время;
2. Задача должна запуститься в течение 10 минут после nominal_time;
3. Задача должна завершиться через 20 минут после nominal_time;
4. Задача максимально должна длиться не более 30 минут начиная с nominal_time;
5. Информировать в случае если начало, конец и длительность не уложились в указанные рамки;
6. Выслать отчет об ошибке на psimanihin@neoflex.ru.

Приложения

Приложение 1. Соответствие типов полей в дизайнера трансформаций классам языка Scala

Тип данных в дизайнере	Соответствующий класс языка Scala
STRING	java.lang.String
DECIMAL	java.math.BigDecimal
INTEGER	java.lang.Integer
DATE	java.sql.Date
TIME	java.sql.Timestamp
DATETIME	java.sql.Timestamp
BINARY	Array[Byte]
BOOLEAN	java.lang.Boolean
LONG	java.lang.Long
FLOAT	java.lang.Float
STRUCT	-
ARRAY	-

Приложение 2. Встроенные функции редактора выражений

ABS(arg: java.math.BigDecimal): java.math.BigDecimal

Функция принимает значение одного аргумента и возвращает его абсолютное значение.

Аргументы:

arg - число, абсолютное значение которого необходимо определить.

Возвращаемое значение:

абсолютное значение аргумента

ABS(arg: java.lang.Integer): java.lang.Integer

Функция принимает значение одного аргумента и возвращает его абсолютное значение.

Аргументы:

arg - число, абсолютное значение которого необходимо определить.

Возвращаемое значение:

Абсолютное значение аргумента

SIGN(arg: java.math.BigDecimal): java.lang.Integer

Функция принимает значение одного аргумента и возвращает результат применения signum-функции.

Аргументы:

arg - значение аргумента к которому применяется signum-функция.

Возвращаемое значение:

-1, если $\text{arg} < 0$;

1, если $\text{arg} > 0$;

0, если $\text{arg} = 0$

SIGN(arg: java.lang.Integer): java.lang.Integer

Функция принимает значение одного аргумента и возвращает результат применения signum-функции.

Аргументы:

arg - значение аргумента к которому применяется signum-функция.

Возвращаемое значение:

-1, если $\text{arg} < 0$;

1, если $\text{arg} > 0$;

0, если $\text{arg} = 0$

NVL(expr: AnyRef, arg1: AnyRef): AnyRef

Функция получает два аргумента (expr и arg1). Если expr не равен null (Null), то функция возвращает его значение. Если expr равен null (Null), то возвращает значение arg1.

Аргументы:

expr - значение, сравниваемое с null;

arg1 - аргумент.

Возвращаемое значение:

expr, если $\text{expr} \neq \text{null (Null)}$;

arg1, если $\text{expr} = \text{null (Null)}$

NVL2(expr: AnyRef, arg1: AnyRef, arg2: AnyRef): AnyRef

Функция получает три аргумента (expr, arg1 и arg2). Если expr не равен null (Null), то функция возвращает значение аргумента arg1. Если expr равен null (Null), то возвращает значение arg2.

Аргументы:

expr - значение, сравниваемое с null;

arg1 и **arg2** - аргументы.

Возвращаемое значение:

arg1, если expr ≠ null;

arg2, если expr = null

COALESCE(arg: AnyRef*): AnyRef

Функция принимает произвольное количество аргументов и возвращает значение первого аргумента, который не равен null (Null). Если значения всех аргументов определены как null (Null) или они отсутствуют, то функция вернет null (Null).

Аргументы:

arg - значения аргументов, которые сравниваются с null.

Возвращаемое значение:

Значение первого аргумента, который не равен null (Null);

null (Null), если значения всех аргументов определены как null (Null)

DEFINED(arg: AnyRef): Boolean

Функция принимает один аргумент и проверяет определено ли его значение.

Аргументы:

arg - значение, сравниваемое с null.

Возвращаемое значение:

False - если значение аргумента не определено (равно null (Null));

True - если значение аргумента определено

DECODE(expr: AnyRef, search1: AnyRef, result1: AnyRef,...,searchN: AnyRef, resultN: AnyRef,...[,default: AnyRef]): AnyRef

Функция последовательно сравнивает значение аргумента `expr` со значениями аргументов `search1, search2,..., searchN`. При обнаружении равенства значений аргументов, функция возвращает значение соответствующего аргумента `result`. Если равенств не найдено, то функция вернет значение аргумента `default` или `null`, если аргумент `default` не задан.

Аргументы:

expr - значение аргумента с которым функция сравнивает значения аргументов `search`;

search1,...,searchN - значения аргументов, сравниваемые со значением аргумента `expr`;

result1,...,resultN - значения аргументов, которые функция может вернуть в случае равенства значений аргументов `expr` и соответствующего `search`.

Возвращаемое значение:

resultX - если значение аргумента `expr=searchX`;

default - если равенств не найдено;

null - если равенств не найдено и `default` не задан.

Пример:

DECODE (warehouse_id,1,'Southlake',2,'San Francisco',3,'New Jersey',4,'Seattle', , 'Non domestic')

В приведенном примере функция вернет значение:

Southlake, если `warehouse_id = 1`;

San Francisco, если `warehouse_id =2`;

Non domestic, если `warehouse_id` не равно 1, 2, 3, или 4

DATETIME(date_part: java.sql.Timestamp, hourOfDay: Int, minute: Int, second: Int, millisecond: Int): java.sql.Timestamp

Функция принимает объект `java.sql.Timestamp` и четыре значения типа `Int`, возвращает объект `java.sql.Timestamp`.

Аргументы:

date_part - объект `java.sql.Timestamp` из которого выбираются компоненты даты [год, месяц, день];

hourOfDay - количество часов в диапазоне от 0 до 23;

minute - количество минут в диапазоне от 0 до 59;

second - количество секунд в диапазоне от 0 до 59;

millisecond - количество миллисекунд в диапазоне от 0 до 999.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргументов

DATETIME(date_part: java.sql.Timestamp, time_part: java.sql.Timestamp): java.sql.Timestamp

Функция принимает два объекта: java.sql.Timestamp. Из первого объекта выбирает компоненты даты, из второго объекта – компоненты времени и возвращает объект java.sql.Timestamp с выбранными значениями.

Аргументы:

date_part - объект java.sql.Timestamp из которого выбираются компоненты даты [год, месяц, день];

time_part - объект java.sql.Timestamp из которого выбираются компоненты времени [часы, минуты, секунды, миллисекунды].

Возвращаемое значение:

java.sql.Timestamp, полученный из аргументов

DATETIME(date: java.sql.Date, time_part: java.sql.Timestamp): java.sql.Timestamp

Функция принимает два объекта: java.sql.Date и java.sql.Timestamp. Из первого объекта выбирает значения даты, из второго объекта - компонент времени и возвращает объект java.sql.Timestamp с выбранными значениями.

Аргументы:

date - значения [год, месяц, день], полученные из объекта java.sql.Date;

time_part - значения [часы, минуты, секунды, миллисекунды], полученные из объекта java.sql.Timestamp.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргументов

DATETIME(year: Int, month: Int, dayOfMonth: Int, time_part: java.sql.Timestamp): java.sql.Timestamp

Функция принимает четыре значения типа Int, объект java.sql.Timestamp, из которого выбирает компонент времени и возвращает объект java.sql.Timestamp.

Аргументы:

year - значение типа Int [год];

month - значение типа Int [месяц];

dayOfMonth - значение типа Int [день];

time_part - компоненты времени [часы, минуты, секунды, миллисекунды], выбранные из объекта java.sql.Timestamp.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргументов

DATETIME(date: java.sql.Date): java.sql.Timestamp

Функция преобразует значения объекта java.sql.Date в объект java.sql.Timestamp.

Аргументы:

date - значения, полученные из объекта java.sql.Date.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргумента

DATETIME(year: Int, month: Int, dayOfMonth: Int, hourOfDay: Int, minute: Int, second: Int, millisecond: Int): java.sql.Timestamp

Функция принимает семь значений типа Int и возвращает объект java.sql.Timestamp.

Аргументы:

year - значение типа Int [год];

month - значение типа Int [месяц];

dayOfMonth - значение типа Int [день];

hourOfDay - значение типа Int [часы];

minute - значение типа Int [минуты];

second - значение типа Int [секунды];

millisecond - значение типа Int [миллисекунды].

Возвращаемое значение:

java.sql.Timestamp, полученный из аргументов

DATETIME_STRING(timestamp: java.sql.Timestamp): String

Функция преобразует значения объекта java.sql.Timestamp в строковую величину вида: **y-m-d n:m:s.s** (пример: 2016-04-15 13:38:45.999).

Аргументы:

timestamp - значение, которое необходимо отформатировать.

Возвращаемое значение:

Строковое представление **timestamp**

DATETIME(value: String): java.sql.Timestamp

Функция преобразует строковую величину вида: `y-M-d H:m:s.S` (пример: 2016-04-15 13:38:45.999) в объект `java.sql.Timestamp`.

Аргументы:

value - строка, которую необходимо преобразовать.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргумента

DATETIME(): java.sql.Timestamp

Функция возвращает значение [1900, 1, 1, 0, 0, 0, 0] в объекте `java.sql.Timestamp`.

Возвращаемое значение:

java.sql.Timestamp со значением [1900, 1, 1, 0, 0, 0, 0]

DATE_STRING(date: java.sql.Date): String

Функция преобразует значения объекта `java.sql.Date` в строковую величину вида: `y-M-d` (пример: 2016-04-15).

Аргументы:

date - значение, которое необходимо отформатировать.

Возвращаемое значение:

Строковое представление **date**

DATE_PART(timestamp: java.sql.Timestamp): java.sql.Timestamp

Функция принимает объект `java.sql.Timestamp` и возвращает объект `java.sql.Timestamp` со значением [год, месяц, день, 0, 0, 0, 0].

Аргументы:

timestamp - компонент даты [год, месяц, день] из объекта `java.sql.Timestamp`.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргумента

DATE_PART(value: String): java.sql.Timestamp

Функция принимает значение типа String (y-M-d) и возвращает объект java.sql.Timestamp.

Аргументы:

value - значение типа String (пример: 2016-04-15).

Возвращаемое значение:

java.sql.Timestamp, полученный из аргумента

TIME_PART(timestamp: java.sql.Timestamp): java.sql.Timestamp

Функция принимает объект java.sql.Timestamp, из которого выбирает компонент времени и возвращает объект java.sql.Timestamp.

Аргументы:

timestamp - значения, полученные из объекта java.sql.Timestamp.

Возвращаемое значение:

java.sql.Timestamp, полученный из аргумента

TIME_STRING(timestamp: java.sql.Timestamp): String

Функция преобразует значения объекта java.sql.Timestamp в строковую величину вида: **Н:m:s.S** (пример: 13:38:45.999).

Аргументы:

timestamp - значения, полученные из объекта java.sql.Timestamp.

Возвращаемое значение:

Строковое представление **timestamp**

TIME(hourOfDay: Int, minute: Int, second: Int, millisecond: Int): java.sql.Timestamp

Функция принимает четыре значения типа Int и преобразует их в объект java.sql.Timestamp.

Аргументы:

hourOfDay - значение типа Int [часы];

minute - значение типа Int [минуты];

second - значение типа Int [секунды];

millisecond - значение типа Int [миллисекунды].

Возвращаемое значение:

Объект **java.sql.Timestamp**, полученный из аргументов

TIME(value: String): java.sql.Timestamp

Функция преобразует принятое значение типа String (H^MS.S) в объект java.sql.Timestamp.

Аргументы:

value - значение типа String (пример: 13:38:45.999).

Возвращаемое значение:

java.sql.Timestamp, полученный из аргумента

DATE(timestamp: java.sql.Timestamp): java.sql.Date

Функция выбирает компонент даты из объекта java.sql.Timestamp и возвращает объект java.sql.Date с этими значениями.

Аргументы:

timestamp - значение из объекта java.sql.Timestamp, которое необходимо преобразовать.

Возвращаемое значение:

java.sql.Date, полученный из аргумента

DATE(value: String): java.sql.Date

Функция принимает значение типа String (y-M-d) и преобразует его в объект java.sql.Date.

Аргументы:

value - значение типа String (пример: 2016-04-15), которое необходимо преобразовать.

Возвращаемое значение:

java.sql.Date, полученный из аргумента

DATE(year: Int, month: Int, dayOfMonth: Int): java.sql.Date

Функция принимает три значения типа Int и возвращает их в объекте java.sql.Date.

Аргументы:

year - значение типа Int [год];

month - значение типа Int [месяц];

dayOfMonth - значение типа Int [день].

Возвращаемое значение:

java.sql.Date, полученный из аргумента

NOW(): java.sql.Timestamp

Функция возвращает значение текущей даты и времени в объекте java.sql.Timestamp.

Возвращаемое значение:

Текущая дата и время

TODAY(): java.sql.Date

Функция возвращает значение текущей даты в объекте java.sql.Date.

Возвращаемое значение:

java.sql.Date, полученный из текущей даты

YEAR(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [год] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

MONTH(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [месяц] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

DAY_OF_MONTH(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [день] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

HOURL_OF_DAY(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [часы] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

MINUTE(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [минуты] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

SECOND(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [секунды] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

MILLISECOND(timestamp: java.sql.Timestamp): Int

Функция выбирает из объекта java.sql.Timestamp значение [миллисекунды] и возвращает значение типа Int.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

Значение типа **Int**

ADD(timestamp: java.sql.Timestamp, shift: java.math.BigDecimal): java.sql.Timestamp

Функция принимает два аргумента. К значению [миллисекунды] из java.sql.Timestamp прибавляет значение аргумента shift и формирует объект java.sql.Timestamp.

Аргументы:

timestamp - значения из объекта java.sql.Timestamp;

shift - значение java.math.BigDecimal.

Возвращаемое значение:

java.sql.Timestamp с измененным значением [миллисекунды]

ADD_DAYS(timestamp: java.sql.Timestamp, days: Int): java.sql.Timestamp

Функция принимает два аргумента. К значению [дни] из java.sql.Timestamp прибавляет значение аргумента days и возвращает объект java.sql.Timestamp.

Аргументы:

timestamp - значения из объекта java.sql.Timestamp;

days - значение типа Int.

Возвращаемое значение:

java.sql.Timestamp с измененным значением [дни]

ADD_HOURS(timestamp: java.sql.Timestamp, hours: Int): java.sql.Timestamp

Функция принимает два аргумента. К значению [часы] из java.sql.Timestamp прибавляет значение аргумента hours и формирует объект java.sql.Timestamp.

Аргументы:

timestamp - значения из объекта java.sql.Timestamp;

hours - значение типа Int.

Возвращаемое значение:

java.sql.Timestamp с измененным значением [часы]

ADD_SECONDS(timestamp: java.sql.Timestamp, seconds: Int): java.sql.Timestamp

Функция принимает два аргумента. К значению [секунды] из java.sql.Timestamp прибавляет значение аргумента seconds и формирует объект java.sql.Timestamp.

Аргументы:

timestamp - значения из объекта java.sql.Timestamp;

seconds - значение типа Int.

Возвращаемое значение:

java.sql.Timestamp с измененным значением [секунды]

SUBTRACT(t1: java.sql.Timestamp, t2: java.sql.Timestamp): java.math.BigDecimal

Функция принимает два аргумента (t1, t2) и вычисляет в миллисекундах количество времени между датами, возвращает это значение в объекте java.math.BigDecimal.

Аргументы:

t1 - значение времени и даты из объекта java.sql.Timestamp;

t2 - значение времени и даты из объекта java.sql.Timestamp.

Возвращаемое значение:

java.math.BigDecimal, полученный из аргумента

INTEGER(value: Int): java.lang.Integer

Функция преобразует значение аргумента типа Int в объект java.lang.Integer.

Аргументы:

value - значение типа Int, которое необходимо преобразовать.

Возвращаемое значение:

java.lang.Integer, полученный из аргумента

INTEGER(value: java.math.BigDecimal): java.lang.Integer

Функция преобразует объект java.math.BigDecimal в объект java.lang.Integer.

Аргументы:

value - значение, которое необходимо преобразовать.

Возвращаемое значение:

Целая часть аргумента **value**

INTEGER(value: Double): java.lang.Integer

Функция преобразует значение аргумента типа Double в объект java.lang.Integer.

Аргументы:

value - значение типа Double, которое необходимо преобразовать.

Возвращаемое значение:

java.lang.Integer, полученный из аргумента

INTEGER(value: String): java.lang.Integer

Функция преобразует значение аргумента типа String в объект java.lang.Integer.

Аргументы:

value - значение типа String, которое необходимо преобразовать.

Возвращаемое значение:

java.lang.Integer, полученный из аргумента

INTEGER(value: String, radix: Int): java.lang.Integer

Функция преобразует значение аргумента value в заданной системе счисления в объект java.lang.Integer.

Аргументы:

value - значение, которое необходимо преобразовать;

radix - основание системы счисления (2 - двоичная, 8 - восьмеричная и т.д.).

Возвращаемое значение:

java.lang.Integer, полученный из аргумента

DECIMAL(value: Int): java.math.BigDecimal

Функция преобразует значение аргумента типа Integer в объект java.math.BigDecimal.

Аргументы:

value - значение типа Integer, которое необходимо преобразовать.

Возвращаемое значение:

java.math.BigDecimal, полученный из аргумента

DECIMAL(value: String): java.math.BigDecimal

Функция преобразует значение аргумента типа String в объект java.math.BigDecimal.

Аргументы:

value - значение типа String, которое необходимо преобразовать.

Возвращаемое значение:

java.math.BigDecimal, полученный из аргумента

DECIMAL(value: Double): java.math.BigDecimal

Функция преобразует значение аргумента типа Double в объект java.math.BigDecimal.

Аргументы:

value - значение типа Double, которое необходимо преобразовать.

Возвращаемое значение:

java.math.BigDecimal, полученный из аргумента

DECIMAL(timestamp: java.sql.Timestamp): java.math.BigDecimal

Функция возвращает объект java.math.BigDecimal со значением количества миллисекунд прошедших с 01.01.1970 до даты, указанной в объекте java.sql.Timestamp.

Аргументы:

timestamp - значения объекта java.sql.Timestamp.

Возвращаемое значение:

java.math.BigDecimal с вычисленным количеством миллисекунд

Приложение 3. API Meta Server

Каждый объект, хранящийся в программе, содержит поля `_type_` и `e_id`, например: `{"type": "etl.Project", "e_id": 123456, ...}`.

Поле `_type_` содержит строку вида: `{package}.{EntityType}`

Все типы объектов описаны в emf описаниях (примеры описания для пакетов [etl](#) и [rt](#)).

Тип аутентификации для всех запросов – basic.

Запросы

Тип запроса	Описание
Delete /api/teneo/{package}.{EntityType}/{e_id}	Удаляет объект
Get/api/deep/{package}.{EntityType}/{id}	Возвращает объект со всеми вложенными подобъектами. Для ссылочных вложенных объектов возвращаются только атрибуты без вложенных подобъектов.
Get/api/teneo/{package}.{EntityType}	Возвращает список объектов данного типа
Get/api/teneo/select/{select ... }	Произвольный HSQL запрос
Post /api/teneo/{package}.{EntityType}	Изменяет или создает объект. Тело запроса – json представление объекта со всеми его вложенными подобъектами. Если присутствует атрибут <code>e_id</code> , то происходит update, если нет, то insert
Get/api/operation/MetaServer/{package}/{EntityType}/{name}/{method}?param1=value1&...	Вызов операции по объекту. Name – имя объекта (значение атрибута name) Method – статический метод в groovy классе, расположенном по пути: mserver-2.0-SNAPSHOT.jar\BOOT-INF\lib\MetaServer-2.0-SNAPSHOT.jar\cim\MetaServer\pim\scripts\psm\groovy\MetaServer{package}{EntityType}

Операции

Операция	Описание
etl.Project.downloadArchive	Выгрузка архива проекта в виде аттачмента {project name}.zip. Передать параметр export=true
etl.Project.uploadArchive	<p>Загрузка архива. Ожидает zip архив в multipart параметре file.</p> <p>Пример:</p> <pre><form method="POST" enctype="multipart/form-data" action="/upload"> File to upload: <input type="file" name="file"> Name: <input type="text" name="name"> <input type="submit" value="Upload"> Press here to upload the file! </form></pre>
rt. TransformationDeployment.generate	Генерация исходного кода
rt. TransformationDeployment.build	Компиляция
rt. TransformationDeployment.deploy	Копирование в hdfs jar файла и файла параметров
rt. TransformationDeployment.run	Запуск, возвращает: {"data": "sessionId": sessionId}} - не ждёт завершения
rt. TransformationDeployment.waitJob	Ожидает завершения, получает sessionId, timeout (интервал проверки, по умолчанию – 500мс)
rt. TransformationDeployment.install	Последовательно выполняет: generate, build
rt. TransformationDeployment.generateAndRun	Последовательно выполняет: install, deploy, run, waitJob
rt. TransformationDeployment.buildAndRun	Последовательно выполняет: build, deploy, run, waitJob