

**DLF**

# User Guide

**Issue** 6  
**Date** 2020-08-12



**HUAWEI TECHNOLOGIES CO., LTD.**

**Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

# Contents

<b>1 Preparations.....</b>	<b>1</b>
<b>2 IAM Permissions Management.....</b>	<b>2</b>
2.1 Creating a User and Granting Permissions.....	2
<b>3 Data Management.....</b>	<b>4</b>
3.1 Overview.....	4
3.2 Data Connections.....	4
3.2.1 Creating a Data Connection.....	4
3.2.2 Editing a Data Connection.....	12
3.2.3 Deleting a Data Connection.....	12
3.2.4 Exporting a Data Connection.....	13
3.2.5 Importing a Data Connection.....	14
3.3 Databases.....	15
3.3.1 Creating a Database.....	15
3.3.2 Modifying a Database.....	16
3.3.3 Deleting a Database.....	17
3.4 Namespaces.....	17
3.4.1 Creating a Namespace.....	17
3.4.2 Deleting a Namespace.....	18
3.5 Database Schemas.....	18
3.5.1 Creating a Database Schema.....	19
3.5.2 Modifying a Database Schema.....	19
3.5.3 Deleting a Database Schema.....	20
3.6 Data Tables.....	20
3.6.1 Creating a Data Table (Visualized Mode).....	20
3.6.2 Creating a Data Table (DDL Mode).....	27
3.6.3 Viewing Data Table Details.....	28
3.6.4 Deleting a Data Table.....	29
3.7 Columns.....	29
<b>4 Data Integration.....</b>	<b>30</b>
4.1 Managing CDM Clusters.....	30
4.2 Managing DIS Streams.....	30
4.3 Managing CS Jobs.....	30

<b>5 Data Development.....</b>	<b>31</b>
5.1 Script Development.....	31
5.1.1 Creating a Script.....	31
5.1.2 Developing an SQL Script.....	33
5.1.3 Developing a Shell Script.....	37
5.1.4 Renaming a Script.....	39
5.1.5 Moving a Script.....	41
5.1.6 Exporting and Importing a Script.....	43
5.1.7 Deleting a Script.....	45
5.1.8 Copying a Script.....	46
5.2 Job Development.....	46
5.2.1 Creating a Job.....	46
5.2.2 Developing a Job.....	48
5.2.3 Renaming a Job.....	57
5.2.4 Moving a Job.....	59
5.2.5 Exporting and Importing a Job.....	60
5.2.6 Deleting a Job.....	63
5.2.7 Copying a Job.....	64
<b>6 Solution.....</b>	<b>66</b>
<b>7 O&amp;M and Scheduling.....</b>	<b>68</b>
7.1 Overview.....	68
7.2 Job Monitoring.....	68
7.2.1 Monitoring a Batch Job.....	68
7.2.2 Monitoring a Real-Time Job.....	75
7.2.3 Monitoring Real-Time Subjobs.....	78
7.3 Instance Monitoring.....	80
7.4 PatchData Monitoring.....	81
7.5 Notification Management.....	82
7.5.1 Managing a Notification.....	82
7.5.2 Cycle Overview.....	85
7.6 Backing Up and Restoring Assets.....	87
<b>8 Configuration and Management.....</b>	<b>89</b>
8.1 Managing Host Connections.....	89
8.2 Managing Resources.....	91
<b>9 Specifications.....</b>	<b>95</b>
9.1 Workspace.....	95
9.2 Managing Enterprise Projects.....	96
9.3 Environment Variables.....	97
9.4 Configuring a Log Storage Path.....	99
9.5 Configuring Agencies.....	100

<b>10 Usage Tutorials.....</b>	<b>108</b>
10.1 Developing a Spark Job.....	108
10.2 Developing a Hive SQL Script.....	111
<b>11 References.....</b>	<b>115</b>
11.1 Nodes.....	115
11.1.1 Node Overview.....	115
11.1.2 CDM Job.....	116
11.1.3 DIS Stream.....	119
11.1.4 DIS Dump.....	121
11.1.5 DIS Client.....	123
11.1.6 Rest Client.....	125
11.1.7 Import GES.....	131
11.1.8 MRS Kafka.....	134
11.1.9 Kafka Client.....	135
11.1.10 CS Job.....	137
11.1.11 DLI SQL.....	141
11.1.12 DLI Spark.....	146
11.1.13 DWS SQL.....	148
11.1.14 MRS SparkSQL.....	153
11.1.15 MRS Hive SQL.....	155
11.1.16 MRS Presto SQL.....	157
11.1.17 MRS Spark.....	159
11.1.18 MRS Spark Python.....	161
11.1.19 MRS Flink Job.....	163
11.1.20 MRS MapReduce.....	165
11.1.21 CSS.....	167
11.1.22 Shell.....	169
11.1.23 RDS SQL.....	171
11.1.24 ETL Job.....	173
11.1.25 OCR.....	178
11.1.26 Create OBS.....	179
11.1.27 Delete OBS.....	181
11.1.28 OBS Manager.....	182
11.1.29 Open/Close Resource.....	184
11.1.30 Data Quality Monitor.....	186
11.1.31 Subjob.....	188
11.1.32 SMN.....	189
11.1.33 Dummy.....	192
11.1.34 For Each.....	193
11.2 EL.....	195
11.2.1 Expression Overview.....	195
11.2.2 Basic Operators.....	196

11.2.3 Date and Time Mode..... 197

11.2.4 Env Embedded Objects..... 198

11.2.5 Job Embedded Objects..... 199

11.2.6 StringUtil Embedded Objects..... 201

11.2.7 DateUtil Embedded Objects..... 201

11.2.8 JSONUtil Embedded Objects..... 202

11.2.9 Loop Embedded Objects..... 203


11.2.10 Expression Use Example..... 203

**A Change History..... 207**

# 1 Preparations

---

To access Data Development, perform the following steps:

- Step 1** Visit the HUAWEI CLOUD console.
- Step 2** Click  in the upper left corner of the page to select a region and project.
- Step 3** On the **All Services** tab page, choose **EI Enterprise Intelligence** > **Data Lake Factory** to access the **Dashboard** page of Data Development.

-----End

# 2 IAM Permissions Management

---

## 2.1 Creating a User and Granting Permissions

This chapter describes how to use IAM to implement fine-grained permissions control for your DLF resources. With [IAM](#), you can:

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials, providing access to DLF resources.
- Grant only the permissions required for users to perform a task.
- Entrust a HUAWEI CLOUD account or cloud service to perform efficient O&M on your DLF resources.

If your HUAWEI CLOUD account does not require individual IAM users, skip this section.

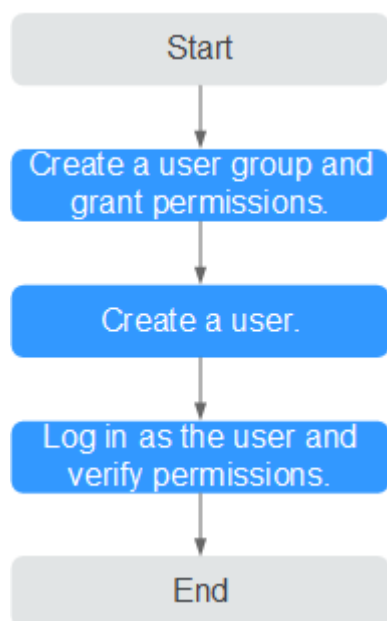
This section describes the procedure for granting permissions. [Figure 2-1](#) shows the procedure.

### Prerequisites

Learn about the permissions supported by DLF and choose policies or roles according to your requirements. For details about the permissions supported by DLF, see [Permissions Management](#). For the system-defined policies of other services, see [System Permissions](#).

## Process Flow

**Figure 2-1** Process for granting DLF permissions



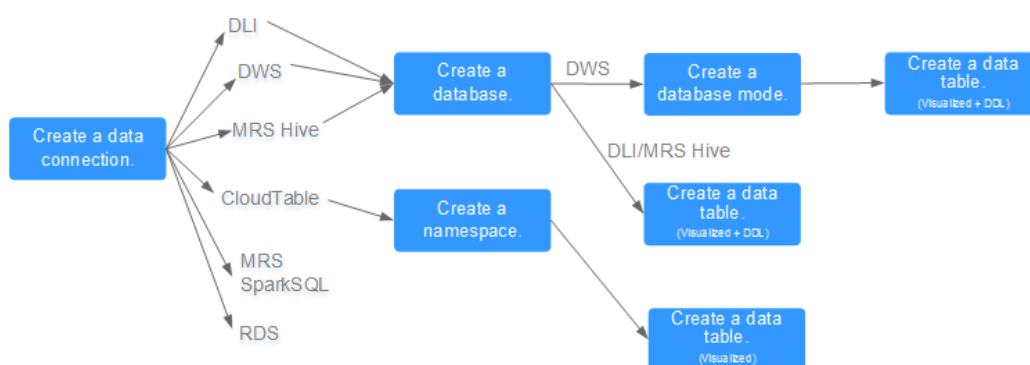
1. **Create a user group and assign permissions** to it.  
Create a user group on the IAM console and assign the **DLF OperationAndMaintenanceAccess** policy to the group.
2. **Create an IAM user**.  
Create a user on the IAM console and add the user to the group created in 1.
3. **Log in** and verify permissions.  
Log in to the DLF console as the created user, and verify that it has management permissions for DLF.
  - a. Choose **Service List > Data Lake Factory**. Then click **Buy DLF** on the DLF console. If no message appears indicating insufficient permissions to perform the operation, the "**DLF OperationAndMaintenanceAccess**" policy has already taken effect.
  - b. Choose any other service in the **Service List**. If a message appears indicating insufficient permissions to access the service, the **DLF OperationAndMaintenanceAccess** policy has already taken effect.

# 3 Data Management

## 3.1 Overview

The data management function helps users quickly establish data models and provides users with data entities for script and job development. The process for using the data management function is as follows:

**Figure 3-1** Data management process



1. DLF communicates with another HUAWEI CLOUD service by building a data connection.
2. After the data connection is built, you can perform data operations on DLF, for example, manage databases, namespaces, database schema, and data tables.

## 3.2 Data Connections

### 3.2.1 Creating a Data Connection

A data connection is storage space used to save data entities managed by Data Development, along with their connection information. With just one data connection, you can run multiple jobs and develop multiple scripts. If the

connection information saved in the data connection changes, you only need to modify the corresponding information in Connection Management.

The following types of data connections can be created:

- DLI
- DWS
- MRS Hive
- MRS SparkSQL
- RDS

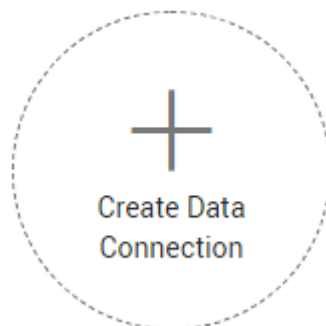
## Prerequisites


- The corresponding cloud service has been enabled.  
For example, before creating an RDS data connection, you need to create a database instance in RDS.
- The quantity of data connections is less than the maximum quota (20).

## Procedure

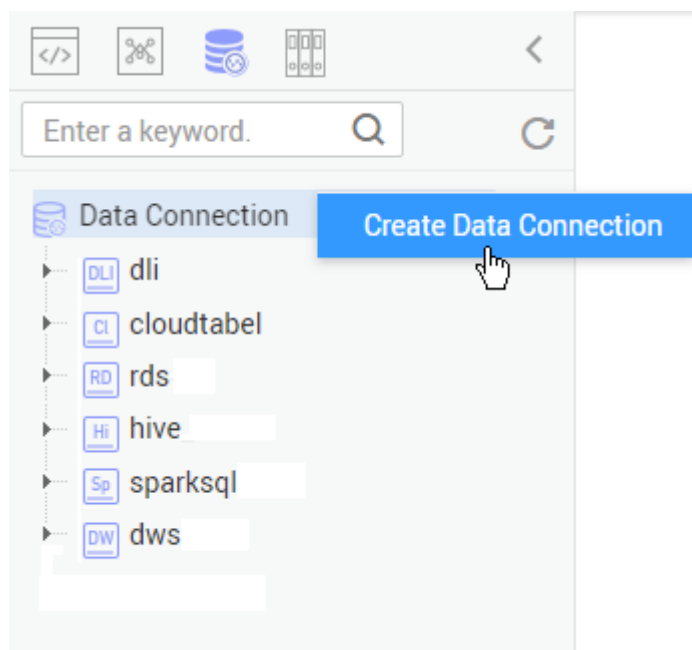
- Step 1** Choose either of the entrances to create a data connection: **Connection Management** page and area on the right.
- **Connection Management** page
    - a. In the navigation tree of the Data Development console, choose **Connection > Connection Management**.
    - b. In the upper right corner of the page, click **Create Data Connection**.
  - Area on the right
    - a. In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.
    - b. Create a data connection in the area on the right using one of the following three methods:  
Method 1: Click **Create Data Connection**.


**Figure 3-2** Creating a data connection (method 1)



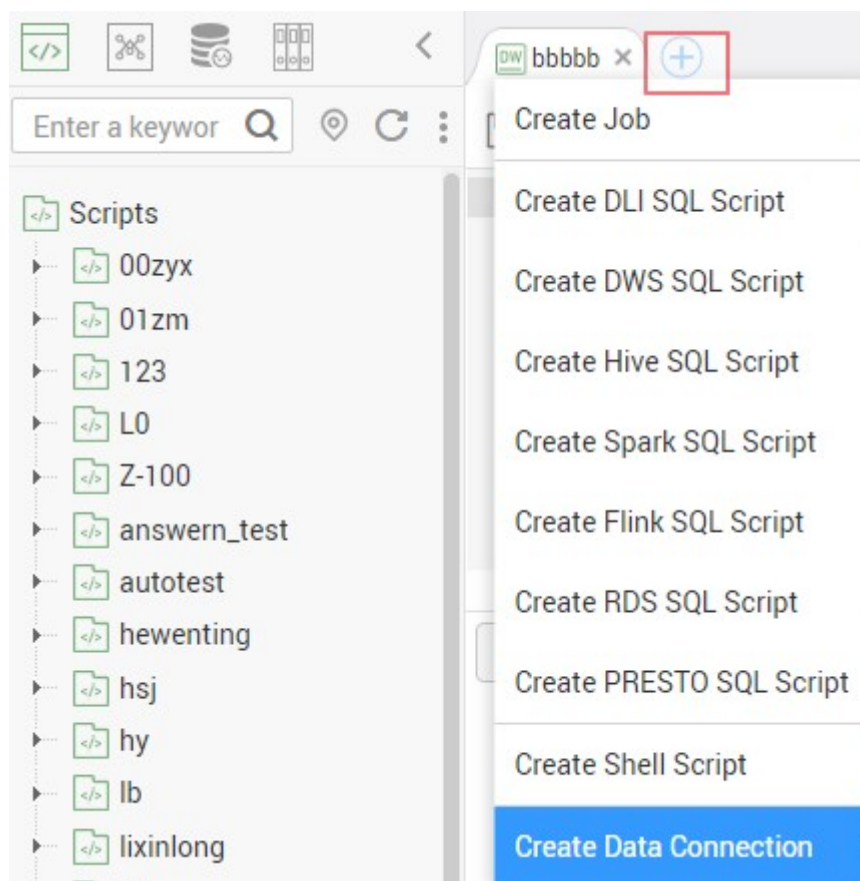
Method 2: In the menu on the left, click , right-click root directory **Data Connection**, and choose **Create Data Connection**.

**Figure 3-3** Creating a data connection (method 2)



Method 3: Open a script or job, click , and choose **Create Data Connection**.

**Figure 3-4** Creating a data connection (method 3)



**Step 2** In the displayed dialog box, select a data connection type and configure data connection parameters. [Table 3-1](#) describes the data connection parameters.

**Table 3-1** Data connection parameters

Data Connection Type	Parameter	Description
DLI	For details, see <a href="#">Table 3-2</a> .	Only one DLI data connection can be created.
DWS	For details, see <a href="#">Table 3-3</a> .	-
MRS Hive	For details, see <a href="#">Table 3-4</a> .	-
MRS SparkSQL	For details, see <a href="#">Table 3-5</a> .	-
RDS	For details, see <a href="#">Table 3-6</a> .	-

**Step 3** Click **Test** to test connectivity to the data connection. If the connectivity is verified, the data connection has been successfully created.

**Step 4** Click **OK**.

----End

## Parameter Description

**Table 3-2** DLI data connection

Parameter	Mandatory	Description
Data Connection Name	Yes	Name of the data connection to be created. Must consist of 1 to 100 characters and contain only letters, digits, and underscores (_).

**Table 3-3** DWS data connection

Parameter	Mandatory	Description
Data Connection Name	Yes	Name of the data connection to be created. Must consist of 1 to 100 characters and contain only letters, digits, and underscores (_).
Cluster Name	No	Name of the DWS cluster. If you do not select a DWS cluster, then configure the access address and port number.

Parameter	Mandatory	Description
Access Address	Yes/No	IP address for accessing the DWS cluster. <ul style="list-style-type: none"><li>• If you select the DWS cluster in the cluster name, the system automatically sets this parameter to the access address of the DWS cluster.</li><li>• If the DWS cluster is not selected, you need to enter the DWS cluster access address.</li></ul>
Port	Yes/No	Port for accessing the DWS cluster. <ul style="list-style-type: none"><li>• If you select the DWS cluster in the cluster name, the system automatically sets this parameter to the port of the DWS cluster.</li><li>• If the DWS cluster is not selected, you need to enter the port of the DWS cluster.</li></ul>
Username	Yes	Administrator name for logging in to the DWS cluster.
Password	Yes	Administrator password for logging in to the DWS cluster.
SSL Connection	Yes/No	DWS supports connections in SSL authentication mode so that data transmitted between the DWS client and the database can be encrypted. The SSL connection mode delivers a higher security than the common mode. For security purposes, you are advised to enable SSL connection.
KMS Key	Yes	Key created on Key Management Service (KMS) and used for encrypting and decrypting user passwords and key pairs. You can select a created key from KMS.
Agent	Yes	Data Warehouse Service (DWS) is not a fully managed service and thus cannot be directly connected to Data Development. A CDM cluster can provide an agent for Data Development to communicate with non-fully-managed services. Therefore, you need to select a CDM cluster when creating a DWS data connection. If no CDM cluster is available, create one.

**Table 3-4** MRS Hive data connection

Parameter	Mandatory	Description
Data Connection Name	Yes	Name of the data connection to be created. Must consist of 1 to 100 characters and contain only letters, digits, and underscores (_).
Cluster Name	Yes	Name of the MRS cluster. Select the MRS cluster to which Hive belongs.
Connection Mode	Yes	<p>Select the mode for DLF to connect to MRS.</p> <p><b>Proxy Connection</b></p> <p>Use the communication proxy function of the CDM cluster to connect DLF to MRS. This mode is recommended.</p> <p>If you select this mode, configure the following parameters:</p> <ul style="list-style-type: none"><li>• <b>Username</b> (optional): administrator of MRS. The username does not need to be configured for some MRS clusters.</li><li>• <b>Password</b> (optional): administrator password of MRS. The username does not need to be configured for some MRS clusters.</li><li>• <b>KMS Key</b> (optional): used to encrypt and decrypt the passwords of user passwords and key pairs. Select a key created in KMS.</li><li>• <b>Connection Proxy</b> (mandatory): Select an available CDM cluster.</li></ul> <p><b>Direct Connection</b></p> <p>If you select this mode, the Hive data tables and fields cannot be viewed. When the Hive SQL script is developed online, the execution result can be viewed only in logs.</p>

**Table 3-5** MRS SparkSQL data connection

Parameter	Mandatory	Description
Data Connection Name	Yes	Name of the data connection to be created. Must consist of 1 to 100 characters and contain only letters, digits, and underscores (_).
Cluster Name	Yes	Name of the MRS cluster. Select the MRS cluster to which SparkSQL belongs.
Connection Mode	Yes	<p>Select the mode for DLF to connect to MRS.</p> <p><b>Proxy Connection</b></p> <p>Use the communication proxy function of the CDM cluster to connect DLF to MRS. This mode is recommended.</p> <p>If you select this mode, configure the following parameters:</p> <ul style="list-style-type: none"><li>• <b>Username</b> (optional): administrator of MRS. The username does not need to be configured for some MRS clusters.</li><li>• <b>Password</b> (optional): administrator password of MRS. The username does not need to be configured for some MRS clusters.</li><li>• <b>KMS Key</b> (optional): used to encrypt and decrypt the passwords of user passwords and key pairs. Select a key created in KMS.</li><li>• <b>Connection Proxy</b> (mandatory): Select an available CDM cluster.</li></ul> <p><b>Direct Connection</b></p> <p>If you select this mode, the Hive data tables and fields cannot be viewed. When the SparkSQL script is developed online, the execution result can be viewed only in logs.</p>


**Table 3-6** RDS data connection

Parameter	Mandatory	Description
Data Connection Name	Yes	Name of the data connection to be created. Must consist of 1 to 100 characters and contain only letters, digits, and underscores (_).
IP Address	Yes	IP address for logging in to the RDS instance.
Port	Yes	Port for logging in to the RDS instance.
Driver Name	Yes	Name of the driver. Possible values: <ul style="list-style-type: none"><li>• <b>com.mysql.jdbc.Driver</b></li><li>• <b>org.postgresql.Driver</b></li></ul>
Username	Yes	Username for logging in to the RDS instance. Default value: <b>root</b>
Password	Yes	Password for logging in to the RDS instance.
KMS Key	Yes	Key created on Key Management Service (KMS) and used for encrypting and decrypting user passwords and key pairs. You can select a created key from KMS.
Driver Path	Yes	Path to the JDBC driver. Download the JDBC driver from the MySQL and PostgreSQL official websites as required and upload the JDBC driver to the Object Storage Service (OBS) bucket. <ul style="list-style-type: none"><li>• If <b>Driver Name</b> is set to <b>com.mysql.jdbc.Driver</b>, use the <b>mysql-connector-java-5.1.21.jar</b> driver.</li><li>• If <b>Driver Name</b> is set to <b>org.postgresql.Driver</b>, use the <b>postgresql-42.2.2.jar</b> driver.</li></ul>
Agent	Yes	Relational Database Service (RDS) is not a fully managed service and thus cannot be directly connected to Data Development. A CDM cluster can provide an agent for Data Development to communicate with non-fully-managed services. Therefore, you need to select a CDM cluster when creating an RDS data connection. If no CDM cluster is available, create one.

## 3.2.2 Editing a Data Connection

After creating a data connection, you can modify data connection parameters.

### Procedure

- Step 1** Choose either of the entrances to edit a data connection: **Connection Management** page and area on the right.
- **Connection Management** page
    - a. In the navigation tree of the Data Development console, choose **Connection > Connection Management**.
    - b. In the **Operation** column of the data connection that you want to edit, click **Edit**.
  - Area on the right
    - a. In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.
    - b. In the menu on the left, click , right-click the data connection that you want to edit, and choose **Edit** from the shortcut menu.
- Step 2** In the displayed dialog box, modify data connection parameters by referring to parameter configuration in [Parameter Description](#).
- Step 3** Click **Test** to test connectivity to the data connection. If the connectivity is verified, the data connection has been successfully created.
- Step 4** Click **Yes**.
- End

## 3.2.3 Deleting a Data Connection

If you do not need to use a data connection any more, perform the following operations to delete it.

---

### NOTICE

If you forcibly delete a data connection that is being associated with a script or job, ensure that services are not affected by going to the script or job development page and reassociating an available data connection with the script or job.

---

### Procedure

- Step 1** Choose either of the entrances to delete a data connection: **Connection Management** page and area on the right.
- **Connection Management** page
    - a. In the navigation tree of the Data Development console, choose **Connection > Connection Management**.
    - b. In the **Operation** column of the data connection that you want to delete, click **Delete**.

- Area on the right
  - a. In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.
  - b. In the menu on the left, click , right-click the data connection that you want to delete, and choose **Delete** from the shortcut menu.

**Step 2** In the displayed dialog box, click **OK**.

----End

## 3.2.4 Exporting a Data Connection

You can export a created data connection.

The existing host connections can be synchronously exported.



### Prerequisites

You have enabled the corresponding cloud service and [created a data connection](#).

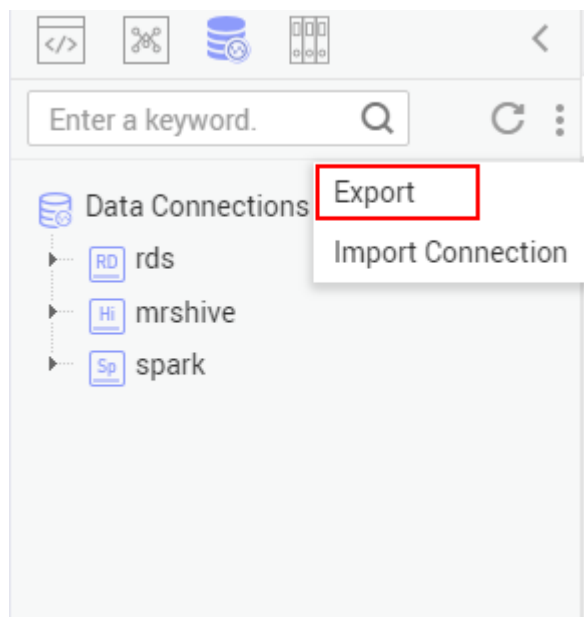
### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** Click  and choose  > **Export**.

**Figure 3-5** Exporting the data connection



----End

## 3.2.5 Importing a Data Connection

Importing a data connection is a process of importing data from OBS to DLF.



### Prerequisites

- You have obtained the username and password for accessing the desired data source.
- OBS has been enabled and a folder has been created in OBS.
- Data has been uploaded from the local host to the OBS folder.
- The quantity of data connections is less than the maximum quota (20).

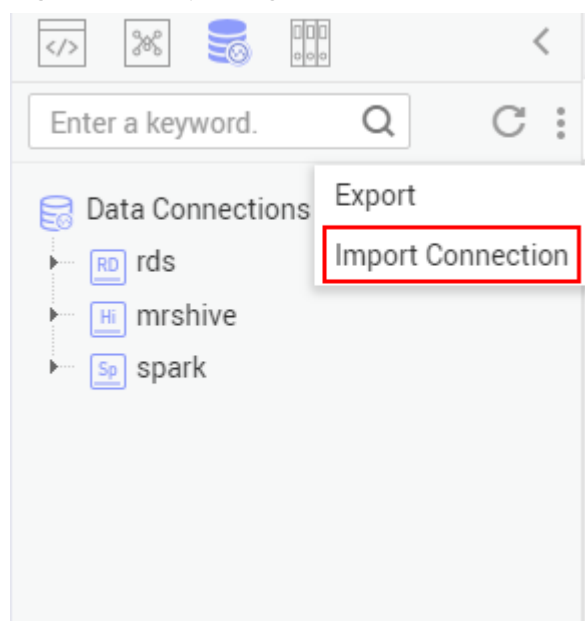
### Procedure

**Step 1** Log in to the DLF console.

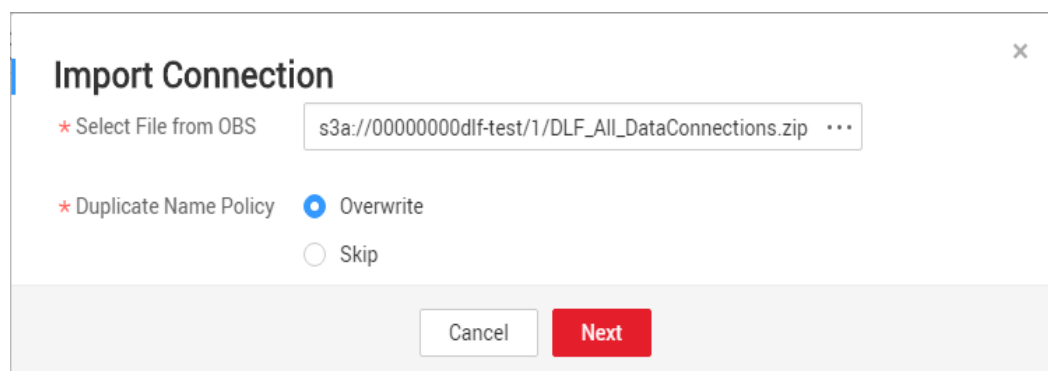
**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** Click  and choose  > **Import Connection**.

**Figure 3-6** Importing a data connection



**Step 4** On the **Import Connection** page, select the file that has been uploaded to the OBS folder and set a duplicate name policy.

**Figure 3-7** Importing a data connection

**Step 5** Click **Next** and proceed with the following operations as prompted. For details about the parameters of each data connection, see [Parameter Description](#).

----End

## 3.3 Databases

### 3.3.1 Creating a Database

After creating a data connection, you can manage the databases under the data connection in the area on the right.

The following types of databases can be created:

- DLI
- DWS
- MRS Hive


#### Prerequisites

A data connection has been created. For details, see [Creating a Data Connection](#).

#### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** In the menu on the left, click , right-click the data connection for which you want to create a database, and choose **Create Database** from the shortcut menu. Set database parameters. [Table 3-7](#) describes the database parameters.

#### NOTE

You can create a maximum of 10 databases for a DLI data connection. No quantity limit is set on other types of data connections.

**Table 3-7** Creating a database

Parameter	Mandatory	Description
Database Name	Yes	Name of the database. The naming rules are as follows: <ul style="list-style-type: none"><li>• DLI: The value must consist of 1 to 128 characters and contain only letters, digits, and underscores (_). It must start with a digit or letter and cannot contain only digits.</li><li>• DWS: The value must consist of 1 to 63 characters and contain only letters, digits, underscores (_), and dollar signs (\$). It must start with a letter or underscore and cannot contain only digits.</li><li>• MRS Hive: The value must consist of 1 to 128 characters and contain only letters, digits, and underscores (_). It must start with a digit or letter and cannot contain only digits.</li></ul>
Description	No	Descriptive information about the database. The requirements are as follows: <ul style="list-style-type: none"><li>• DLI: The value contains a maximum of 256 characters.</li><li>• DWS: The value contains a maximum of 1024 characters.</li><li>• MRS Hive: The value contains a maximum of 1024 characters.</li></ul>

**Step 4** Click **OK**.

----End


### 3.3.2 Modifying a Database

After creating a database, you can modify the description of the DWS or MRS Hive database as required.

#### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** In the menu on the left, click , right-click the database that you want to edit, and choose **Edit** from the shortcut menu.

**Step 4** In the displayed dialog box, modify the description of the database.

**Step 5** Click **Yes**.

----End

### 3.3.3 Deleting a Database

If you do not need to use a database any more, perform the following operations to delete it.


#### Prerequisites

The database that you want to delete is not used and is not associated with any data tables.

#### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** In the menu on the left, click , right-click the database that you want to delete, and choose **Delete** from the shortcut menu.

**Step 4** In the displayed dialog box, click **OK**.

----End

## 3.4 Namespaces

### 3.4.1 Creating a Namespace

After creating a CloudTable data connection, you can manage the namespaces under the CloudTable data connection in the area on the right.

#### Prerequisites

A CloudTable data connection has been created. For details, see [Creating a Data Connection](#).

#### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

- Step 3** In the menu on the left, click , right-click the CloudTable data connection name, and choose **Create Namespace** from the shortcut menu. Set namespace parameters. [Table 3-8](#) describes the namespace parameters.

**Table 3-8** Namespace parameters

Parameter	Mandatory	Description
Namespace Name	Yes	Name of the namespace to be created. Must consist of 1 to 200 characters and contain only letters, digits, and underscores (_).
Description	No	Descriptive information about the namespace. Can contain a maximum of 1024 characters.

- Step 4** Click **OK**.

----End


## 3.4.2 Deleting a Namespace

If you do not need to use a namespace any more, perform the following operations to delete the namespace.

### Prerequisites

The namespace that you want to delete is not used and is not associated with any data tables.

### Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.
- Step 3** In the menu on the left, click , right-click the namespace that you want to delete, and choose **Delete** from the shortcut menu.
- Step 4** In the displayed dialog box, click **OK**.

----End

## 3.5 Database Schemas

## 3.5.1 Creating a Database Schema

After creating a DWS data connection, you can manage the database schemas under the DWS data connection in the area on the right.

### Prerequisites

- A DWS data connection has been created. For details, see [Creating a Data Connection](#).
- A DWS database has been created. For details, see [Creating a Database](#).

### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** In the menu on the left, click . Click a DWS data connection and choose a desired database. Right-click **schemas**, and choose **Create Schema** from the shortcut menu.

**Step 4** In the displayed dialog box, configure schema parameters. [Table 3-9](#) describes the database schema parameters.

**Table 3-9** Creating a database schema

Parameter	Mandatory	Description
Schema Name	Yes	Name of the database schema.
Description	No	Descriptive information about the database schema.

**Step 5** Click **OK**.

----End


## 3.5.2 Modifying a Database Schema

After creating a database schema, you can modify the description of the database schema as required.

### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** In the menu on the left, click , right-click the database schema that you want to modify, and choose **Modify** from the shortcut menu.

**Step 4** In the displayed dialog box, modify the description of the database.

**Step 5** Click **Yes**.

----End

### 3.5.3 Deleting a Database Schema

If you do not need to use a database schema any more, perform the following operations to delete it.


#### Prerequisites

The default database schema cannot be deleted.

#### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.

**Step 3** In the menu on the left, click , right-click the database schema that you want to delete, and choose **Delete** from the shortcut menu.

**Step 4** In the displayed dialog box, click **OK**.

----End

## 3.6 Data Tables

### 3.6.1 Creating a Data Table (Visualized Mode)

You can create permanent data tables in visualized mode. After creating a data table, you can use it for job and script development.

The following types of data tables can be created:

- DLI
- DWS
- MRS Hive
- CloudTable

#### Prerequisites

- A corresponding cloud service has been enabled and a database has been created in the cloud service. For example, before creating a DLI table, DLI has been enabled and a database has been created in DLI.
- A data connection that matches the data table type has been created in Data Development. For details, see [Creating a Data Connection](#).

## Procedure

**Step 1** Perform the following steps:

1. In the navigation tree of the DLF console, choose **Development > Develop Script/Development > Develop Job**.
2. In the menu on the left, click , right-click **tables**, and choose **Create Data Table** from the shortcut menu.

**Step 2** On the displayed page, configure basic properties. Specific settings vary depending on the data connection type you select. [Table 3-10](#) lists the links for viewing property parameters of each type of data connection.

**Table 3-10** Basic property parameters

Data Connection Type	Parameter Description
DLI	For details, see the <b>Basic Property</b> part in <a href="#">Table 3-12</a> .
DWS	For details, see the <b>Basic Property</b> part in <a href="#">Table 3-13</a> .
MRS Hive	For details, see the <b>Basic Property</b> part in <a href="#">Table 3-14</a> .
CloudTable	For details, see the <b>Basic Property</b> part in <a href="#">Table 3-15</a> .

**Step 3** Click **Next**. On the **Configure Table Structure** page, configure table structure parameters. [Table 3-11](#) describes the table structure parameters.

**Table 3-11** Table structure

Data Connection Type	Parameter Description
DLI	For details, see the <b>Table Structure</b> part in <a href="#">Table 3-12</a> .
DWS	For details, see the <b>Table Structure</b> part in <a href="#">Table 3-13</a> .
MRS Hive	For details, see the <b>Table Structure</b> part in <a href="#">Table 3-14</a> .
CloudTable	For details, see the <b>Table Structure</b> part in <a href="#">Table 3-15</a> .


**Step 4** Click **OK**.

----End

## Parameter Description

Table 3-12 DLI data table


Parameter	Mandatory	Description
<b>Basic Property</b>		
Table Name	Yes	Name of the data table. Must consist of 1 to 63 characters and contain only lowercase letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Alias	No	Alias of the data table. Must consist of 1 to 63 characters and contain only letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Data Connection	Yes	Data connection to which the data table belongs.
Database	Yes	Database to which the data table belongs.
Data Location	Yes	Location to save data. Possible values: <ul style="list-style-type: none"><li>• OBS</li><li>• DLI</li></ul>
Data Format	Yes	Format of data. This parameter is available only when <b>Data Location</b> is set to <b>OBS</b> . Possible values: <ul style="list-style-type: none"><li>• <b>parquet</b>: DLF can read non-compressed parquet data and parquet data compressed using Snappy or gzip.</li><li>• <b>csv</b>: DLF can read non-compressed CSV data and CSV data compressed using gzip.</li><li>• <b>orc</b>: DLF can read non-compressed ORC data and ORC data compressed using Snappy.</li><li>• <b>json</b>: DLF can read non-compressed JSON data and JSON data compressed using gzip.</li></ul>
Path	Yes	OBS path where the data is stored. This parameter is available only when <b>Data Location</b> is set to <b>OBS</b> .
Table Description	No	Descriptive information about the data table.

Parameter	Mandatory	Description
<b>Table Structure</b>		
Column Name	Yes	Name of the column. Must be unique.
Type	Yes	Type of data. For details about the data types, see <i>Data Lake Insight SQL Syntax Reference</i> .
Column Description	No	Descriptive information about the column.
Operation	No	To add a column, click  .

**Table 3-13** DWS data table


Parameter	Mandatory	Description
<b>Basic Property</b>		
Table Name	Yes	Name of the data table. Must consist of 1 to 63 characters and contain only letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Alias	No	Alias of the data table. Must consist of 1 to 63 characters and contain only letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Data Connection	Yes	Data connection to which the data table belongs.
Database	Yes	Database to which the data table belongs.
Schema	Yes	Schema of the database.
Table Description	No	Descriptive information about the data table.

Parameter	Mandatory	Description
Advanced Settings	No	<p>The following advanced options are available:</p> <ul style="list-style-type: none"><li>• Storage method of a data table. Possible values:<ul style="list-style-type: none"><li>– <b>Row store</b></li><li>– <b>Column store</b></li></ul></li><li>• Compression level of a data table<ul style="list-style-type: none"><li>– Possible values if the storage method is row store: <b>YES</b> or <b>NO</b>.</li><li>– Possible values if the storage method is column store: <b>YES</b>, <b>NO</b>, <b>LOW</b>, <b>MIDDLE</b>, or <b>HIGH</b>. For the same compression level in column store mode, you can configure compression grades from 0 to 3. Within any compression level, the higher the grade, the greater the compression ratio.</li></ul></li></ul>
<b>Table Structure</b>		
Column Name	Yes	Name of the column. Must be unique.
Data Classification	Yes	<p>Classification of data. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Value</b></li><li>• <b>Currency</b></li><li>• <b>Boolean</b></li><li>• <b>Binary</b></li><li>• <b>Character</b></li><li>• <b>Time</b></li><li>• <b>Geometric</b></li><li>• <b>Network address</b></li><li>• <b>Bit string</b></li><li>• <b>Text search</b></li><li>• <b>UUID</b></li><li>• <b>JSON</b></li><li>• <b>OID</b></li></ul>
Data Type	Yes	Type of data. For details about the data types, see <i>Data Warehouse Service Developer Guide</i> .
Column Description	No	Descriptive information about the column.

Parameter	Mandatory	Description
Create ES Index	No	If you click the check box, an ES index needs to be created. When creating the ES index, select the created CSS cluster from the <b>CloudSearch Cluster Name</b> drop-down list. For details about how to create a CSS cluster, see <i>Cloud Search Service User Guide</i> .
Index Data Type	No	Data type of the ES index. The options are as follows: <ul style="list-style-type: none"><li>• text</li><li>• keyword</li><li>• date</li><li>• long</li><li>• integer</li><li>• short</li><li>• byte</li><li>• double</li><li>• boolean</li><li>• binary</li></ul>
Operation	No	To add a column, click  .


**Table 3-14** Basic property parameters of an MRS Hive data table

Parameter	Mandatory	Description
<b>Basic Property</b>		
Table Name	Yes	Name of the data table. Must consist of 1 to 63 characters and contain only lowercase letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Alias	No	Alias of the data table. Must consist of 1 to 63 characters and contain only letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Data Connection	Yes	Data connection to which the data table belongs.

Parameter	Mandatory	Description
Database	Yes	Database to which the data table belongs.
Table Description	No	Descriptive information about the data table.
<b>Table Structure</b>		
Column Name	Yes	Name of the column. Must be unique.
Data Classification	Yes	Classification of data. Possible values: <ul style="list-style-type: none"><li>• <b>Original</b></li><li>• <b>ARRAY</b></li><li>• <b>MAP</b></li><li>• <b>STRUCT</b></li><li>• <b>UNION</b></li></ul>
Data Type	Yes	Type of data.
Column Description	No	Descriptive information about the column.
Operation	No	To add a column, click  .

**Table 3-15** Basic property parameters of a CloudTable data table

Parameter	Mandatory	Description
<b>Basic Property</b>		
Table Name	Yes	Name of the data table. Must consist of 1 to 63 characters and contain only letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Alias	No	Alias of the data table. Must consist of 1 to 63 characters and contain only letters, digits, and underscores (_). Cannot contain only digits or start with an underscore.
Data Connection	Yes	Data connection to which the data table belongs.
Namespace	Yes	Namespace to which the data table belongs.

Parameter	Mandatory	Description
Table Description	No	Descriptive information about the data table.
<b>Table Structure</b>		
Column Family Name	Yes	Name of the column family. Must be unique.
Column Family Description	No	Descriptive information about the column family.
Operation	No	To add a column, click  .

### 3.6.2 Creating a Data Table (DDL Mode)

You can create permanent and temporary data tables in DDL mode. After creating a data table, you can use it for job and script development.

The following types of data tables can be created:

- DLI
- DWS
- MRS Hive

#### Prerequisites

- A corresponding cloud service has been enabled and a database has been created in the cloud service. For example, before creating a DLI table, DLI has been enabled and a database has been created in DLI.
- A data connection that matches the data table type has been created in Data Development. For details, see [Creating a Data Connection](#).

#### Procedure

**Step 1** Perform the following steps:

1. In the navigation tree of the Data Development console, choose **Data Development > Develop Script/Data Development > Develop Job**.
2. In the menu on the left, click , right-click **tables**, and choose **Create Data Table** from the shortcut menu.

**Step 2** Click **DDL-based Table Creation**, configure parameters described in [Table 3-16](#), and enter SQL statements in the editor in the lower part.

**Table 3-16** Data table parameters

Parameter	Description
Data Connection Type	Type of data connection to which the data table belongs. <ul style="list-style-type: none"><li>• DLI</li><li>• DWS</li><li>• HIVE</li></ul>
Data Connection	Data connection to which the data table belongs.
Database	Database to which the data table belongs.

**Step 3** Click **OK**.

----End

### 3.6.3 Viewing Data Table Details

After creating a data table, you can view the basic information, storage information, field information, and preview data of the data table.

#### Procedure

**Step 1** Perform the following steps:

1. In the navigation tree of the DLF console, choose **Development > Develop Script/Development > Develop Job**.
2. In the menu on the left, click , right-click the data table that you want to view, and choose **View Details** from the shortcut menu.

**Step 2** In the displayed dialog box, view the data table information.

**Table 3-17** Table details page

Tab Name	Description
Table Information	Displays the basic information and storage information about the data table.
Field Information	Displays the field information about the data table.
Data Preview	Displays 10 records about the data table.
DDL	Displays the DDL of the DLI or DWS data table.


----End

## 3.6.4 Deleting a Data Table

If you do not need to use a data table any more, perform the following operations to delete it.

### Procedure

**Step 1** Perform the following steps:

1. In the navigation tree of the DLF console, choose **Development > Develop Script/Development > Develop Job**.
2. In the menu on the left, click , right-click the data table that you want to delete, and choose **Delete** from the shortcut menu.

**Step 2** In the displayed dialog box, click **OK**.

----End


## 3.7 Columns

You can view the column information of a data table in the area on the right.

### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the DLF console, choose **Development > Develop Script/Development > Develop Job**.

**Step 3** In the menu on the left, click , and expand the data connection directory to view column information under a desired data table.

----End

# 4 Data Integration

---

## 4.1 Managing CDM Clusters

To help users quickly migrate data, Data Development is integrated with Cloud Data Migration (CDM). You can go to the CDM console by choosing **Data Integration** from the console drop-down list in the upper left corner of the page, and selecting **CDM** in the navigation tree. Alternatively, you can directly access the CDM console to perform operations.

For details about how to use CDM, see the *Cloud Data Migration User Guide*.

## 4.2 Managing DIS Streams

To help users transfer data to the cloud in real time, Data Development is integrated with Data Ingestion Service (DIS). You can go to the DIS console by choosing **Data Integration** from the console drop-down list in the upper left corner of the page, and selecting **DIS** in the navigation tree. Alternatively, users can directly access the DIS console to perform operations.

For details about how to use DIS, see the *Data Ingestion Service User Guide*.

## 4.3 Managing CS Jobs

To help users quickly analyze streaming data, Data Development is integrated with Cloud Stream Service (CS). You can go to the CS console by choosing **Data Integration** from the console drop-down list in the upper left corner of the page, and selecting **CS** in the navigation tree. Alternatively, you can directly access the CS console to perform operations.

For details about how to use CS, see the *Cloud Stream Service User Guide*.

# 5 Data Development

## 5.1 Script Development

### 5.1.1 Creating a Script

DLF allows you to edit, debug, and run scripts online. You must add a script before developing it.

#### (Optional) Creating a Directory

If a directory exists, you do not need to create one.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.

**Step 3** In the directory list, right-click a directory and choose **Create Directory** from the shortcut menu.

**Step 4** In the displayed dialog box, configure directory parameters. [Table 5-1](#) describes the directory parameters.

**Table 5-1** Script directory parameters

Parameter	Description
Directory Name	Name of the script directory. Must consist of 1 to 32 characters and contain only letters, digits, underscores (_), and hyphens (-).
Select Directory	Parent directory of the script directory. The parent directory is the root directory by default.

**Step 5** Click **OK**.

----End

## Creating a Script

Currently, you can create the following types of scripts in DLF:

- DLI SQL
- Hive SQL
- DWS SQL
- Spark SQL
- Flink SQL
- RDS SQL
- PRESTO SQL script, which is supported only in the AP-Singapore region. After you use the PRESTO SQL script to run the **select** query statement, the query result is automatically dumped to the **s3a://dlf-log-*{project\_id}*/temp** directory of the OBS bucket.
- Shell

## Prerequisites

The quantity of scripts is less than the maximum quota (1,000).

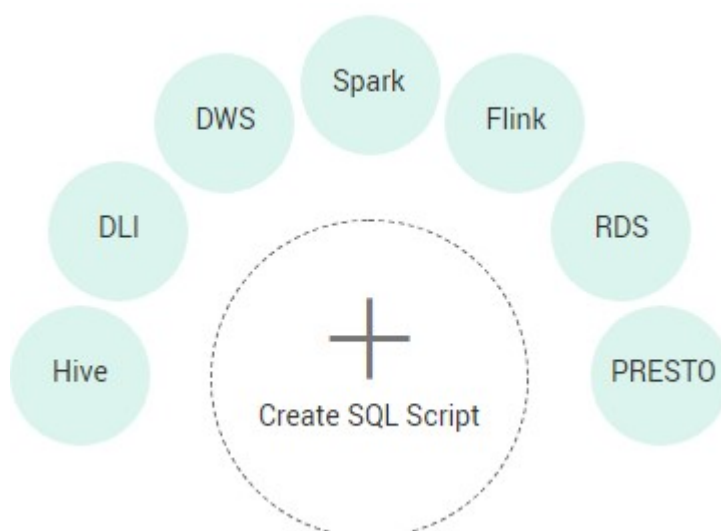
## Procedure

**Step 1** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.

**Step 2** Create a script using either of the following methods:

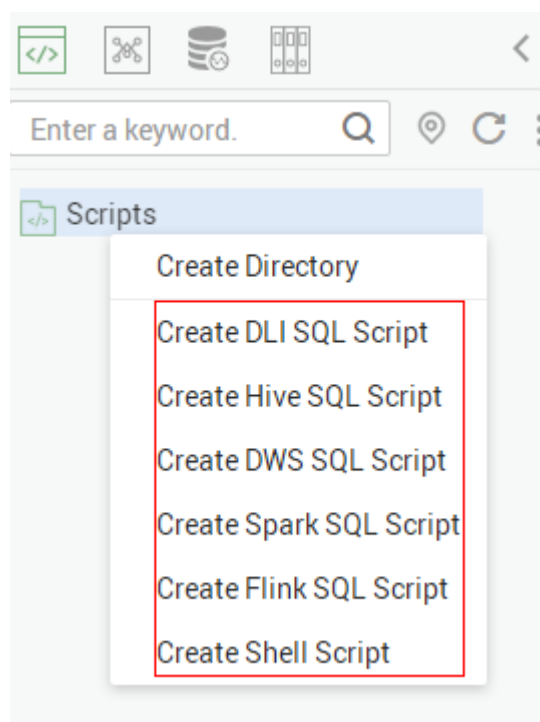
Method 1: In the area on the right, click **Create SQL Script/Create Shell Script**.

**Figure 5-1** Creating an SQL script (method 1)



**Figure 5-2** Creating a shell script (method 1)

Method 2: In the directory list, right-click a directory and choose **Create Script** from the shortcut menu.

**Figure 5-3** Creating a script (method 2)

**Step 3** Go to the script development page. For details, see [Developing an SQL Script](#) and [Developing a Shell Script](#).

----End

## 5.1.2 Developing an SQL Script

You can develop, debug, and run SQL scripts online. The developed scripts can be run in jobs. For details, see [Developing a Job](#).

### Prerequisites

- A corresponding cloud service has been enabled and a database has been created in the cloud service. For example, before developing a DLI script, DLI

has been enabled and a database has been created in DLI. This prerequisite is not applied to Flink SQL scripts, so you do not need to create a database before developing a Flink SQL script.




- A data connection that matches the data connection type of the script has been created in Data Development. For details, see [Creating a Data Connection](#). Flink SQL scripts are not involved.
- An SQL script has been added. For details, see [Creating a Script](#).

## Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
- Step 3** In the script directory list, double-click a script that you want to develop. The script development page is displayed.
- Step 4** In the upper part of the editor, select script properties. [Table 5-2](#) describes the script properties. When developing a Flink SQL script, skip this step.

**Table 5-2** SQL script properties

Property	Description
Data Connection	Selects a data connection.

Property	Description
Resource Queue	<p>Selects a resource queue for executing a DLI job. Set this parameter when a DLI or SQL script is created. You can create a resource queue using either of the following methods:</p> <ul style="list-style-type: none"><li>Click  to go to the <b>Queue Management</b> page.</li><li>Go to the DLI console.</li></ul> <p>To set properties for submitting SQL jobs in the form of <b>key/value</b>, click . A maximum of 10 properties can be set. The properties are described as follows:</p> <ul style="list-style-type: none"><li><b>dli.sql.autoBroadcastJoinThreshold</b>: specifies the data volume threshold to use BroadcastJoin. If the data volume exceeds the threshold, BroadcastJoin will be automatically enabled.</li><li><b>dli.sql.shuffle.partitions</b>: specifies the number of partitions during shuffling.</li><li><b>dli.sql.cbo.enabled</b>: specifies whether to enable the CBO optimization policy.</li><li><b>dli.sql.cbo.joinReorder.enabled</b>: specifies whether join reordering is allowed when CBO optimization is enabled.</li><li><b>dli.sql.multiLevelDir.enabled</b>: specifies whether to query the content in subdirectories if there are subdirectories in the specified directory of an OBS table or in the partition directory of an OBS partition table. By default, the content in subdirectories is not queried.</li><li><b>dli.sql.dynamicPartitionOverwrite.enabled</b>: specifies that only partitions used during data query are overwritten and other partitions are not deleted.</li></ul>
Database	Name of the database.
Data Table	<p>Name of the data table that exists in the database. You can also search for an existing table by entering the database name and clicking .</p>

**Step 5** Enter an SQL statement in the editor. You can enter multiple SQL statements. To facilitate script development, DLF provides system functions and script parameters (Flink SQL and RDS scripts are excluded).

 **NOTE**

SQL statements are separated by semicolons (;). If semicolons are used in other places but not used to separate SQL statements, escape them with backslashes (\). For example:

```
select 1;  
select * from a where b="dsfa\";
```

- System Functions

To view the functions supported by this type of data connection, click **System Function** on the right of the editor. You can double-click a function to the editor to use it.

- Script Parameters

You can directly write script parameters in SQL statements. When debugging scripts, you can enter parameter values in the script editor. If the script is referenced by a job, you can set parameter values on the job development page. The parameter values can use EL expressions (see [Expression Overview](#)).

An example is as follows:

```
select ${str1} from data;
```

In the preceding command, *str1* indicates the parameter name. It can contain only letters, digits, hyphens (-), underscores (\_), greater-than signs (>), and less-than signs (<), and can contain a maximum of 16 characters. The parameter name must be unique.

**Step 6** (Optional) In the upper part of the editor, click **Format** to format the SQL statement. When developing a Flink SQL script, skip this step.

**Step 7** In the upper part of the editor, click **Execute**. If you need to run some SQL statements separately, select the SQL statements first. After the SQL statement is executed, you can view the execution history and execution result of the script in the lower part of the editor. When developing a Flink SQL script, skip this step.


You can download or dump execution results as required. Only users with the DAYU Administrator or Tenant Administrator permissions can download or dump execution results. The following is an example:

- Download result: Download the CSV result files to the local host.
- Dump result: Dump the CSV result files to OBS. For details, see [Table 5-3](#).

**Table 5-3** Dump result

Parameter	Mandatory	Description
Data Format	Yes	Format of the data to be exported. Only CSV result files can be exported.
Resource Queue	No	DLI queue where the export operation is to be performed. This parameter is available only when the DLI SQL script is used.
Compression Format	No	Format of compression. Set this parameter when a DLI or SQL script is created. <ul style="list-style-type: none"><li>- none</li><li>- bzip2</li><li>- deflate</li><li>- gzip</li></ul>

Parameter	Mandatory	Description
Storage Path	Yes	OBS path where the result file is stored. After selecting an OBS path, customize a folder. Then, the system will create it automatically for storing the result file.
Cover Type	No	If a folder that has the same name as your customized folder exists in the storage path, select a cover type. This parameter is available only when a DLI SQL script is created. <ul style="list-style-type: none"><li>– <b>Overwrite:</b> The existing folder will be overwritten by the customized folder.</li><li>– <b>Report:</b> The system reports an error and suspends the export operation.</li></ul>

**Step 8** Above the editor, click  to save the script.

If the script is created but not saved, set the parameters listed in [Table 5-4](#).

**Table 5-4** Script parameters

Parameter	Mandatory	Description
Script Name	Yes	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.
Description	No	Descriptive information about the script.
Select Directory	Yes	Directory to which the script belongs. The root directory is selected by default.

----End

### 5.1.3 Developing a Shell Script

You can develop, debug, and run shell scripts online. The developed scripts can be run in jobs. For details, see [Developing a Job](#).

#### Prerequisites

- A shell script has been added. For details, see [Creating a Script](#).
- A host connection has been created. The host is used to execute shell scripts. For details, see [Managing Host Connections](#).

## Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
- Step 3** In the script directory list, double-click a script that you want to develop. The script development page is displayed.
- Step 4** In the upper part of the editor, select script properties. [Table 5-5](#) describes the script properties.

**Table 5-5** Shell script properties

Parameter	Description	Example
Host Connection	Selects the host where a shell script is to be executed.	-
Parameter	Parameter transferred to the script when the shell script is executed. Parameters are separated by spaces. For example: <b>a b c</b> . The parameter must be referenced by the shell script. Otherwise, the parameter is invalid.	For example, if you enter the following shell interactive script and the interactive parameters <b>1</b> , <b>2</b> , and <b>3</b> correspond to <b>begin</b> , <b>end</b> , and <b>exit</b> , you need to enter parameters <b>1</b> , <b>2</b> , and <b>3</b> . <pre>#!/bin/bash select ch in "begin" "end" "exit"; do case \$ch in "begin") echo "start something" ;; "end") echo "stop something" ;; "exit") echo "exit" break; ;; *) echo "Ignorant" ;; esac</pre>
Interactive Input	Interactive information (passwords for example) provided during shell script execution. Interactive parameters are separated by carriage return characters. The shell script reads parameter values in sequence according to the interaction situation.	-

- Step 5** Edit shell statements in the editor.

To facilitate script development, the DLF provides the script parameter function. The usage method is as follows:


Write the script parameter name and parameter value in the shell statement. When the shell script is referenced by a job, if the parameter name configured for the job is the same as the parameter name of the shell script, the parameter value of the shell script is replaced by the parameter value of the job.

An example is as follows:

```
a=1  
echo ${a}
```

In the preceding command, *a* indicates the parameter name. It can contain only letters, digits, hyphens (-), underscores (\_), greater-than signs (>), and less-than signs (<), and can contain a maximum of 16 characters. The parameter name must be unique.

**Step 6** In the upper part of the editor, click **Execute**. After executing the shell statement, view the execution history and result of the script in the lower part of the editor.

**Step 7** Above the editor, click  to save the script.

If the script is created but not saved, set the parameters listed in [Table 5-6](#).

**Table 5-6** Script parameters

Parameter	Mandatory	Description
Script Name	Yes	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.
Description	No	Descriptive information about the script.
Select Directory	Yes	Directory to which the script belongs. The root directory is selected by default.

----End

## 5.1.4 Renaming a Script

You can change the name of a script.

This section describes how to rename a script.

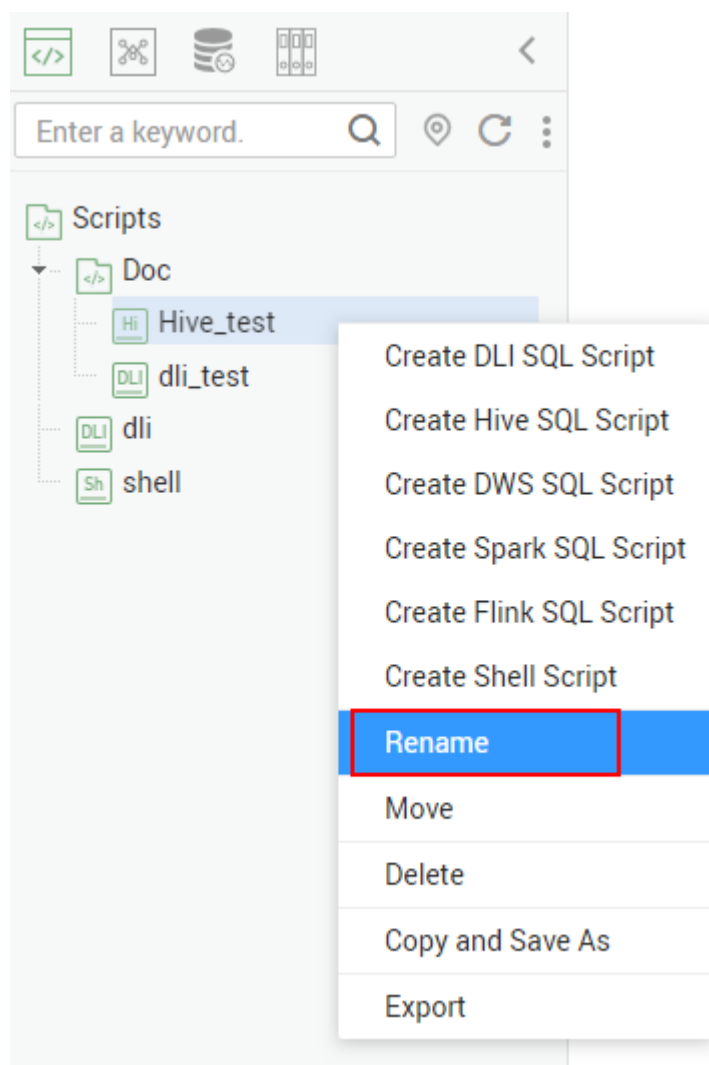
### Prerequisites

- You have developed a script. The script to be renamed exists in the script directory.
- For details about how to develop scripts, see [Developing an SQL Script](#) and [Developing a Shell Script](#).

## Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
- Step 3** In the script directory, select the script to be renamed. Right-click the script name and choose **Rename** from the shortcut menu.

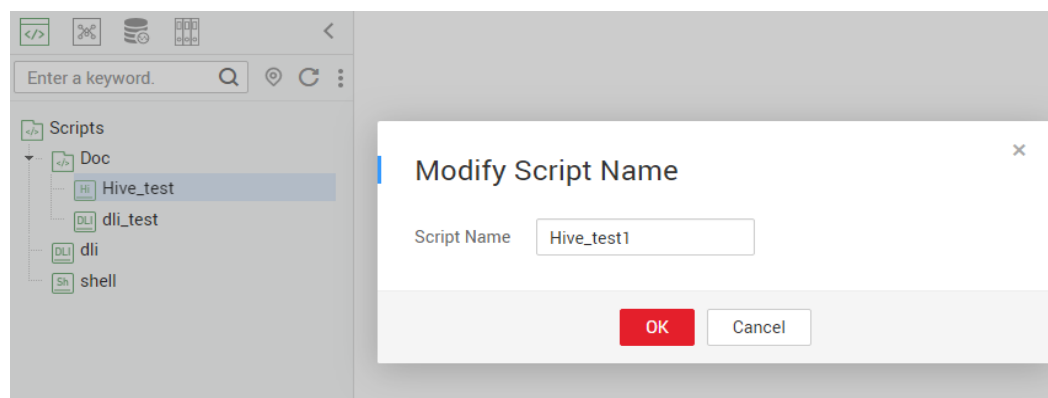
**Figure 5-4** Choosing Rename



### NOTE

An opened script file cannot be renamed.

- Step 4** On the page that is displayed, configure related parameters. [Table 5-7](#) describes the parameters.

**Figure 5-5** Renaming a script**Table 5-7** Script renaming parameters

Parameter	Description
Script Name	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

**Step 5** Click **OK**.

----End

## 5.1.5 Moving a Script

You can move a script from the current directory to another directory.

This section describes how to move a script.

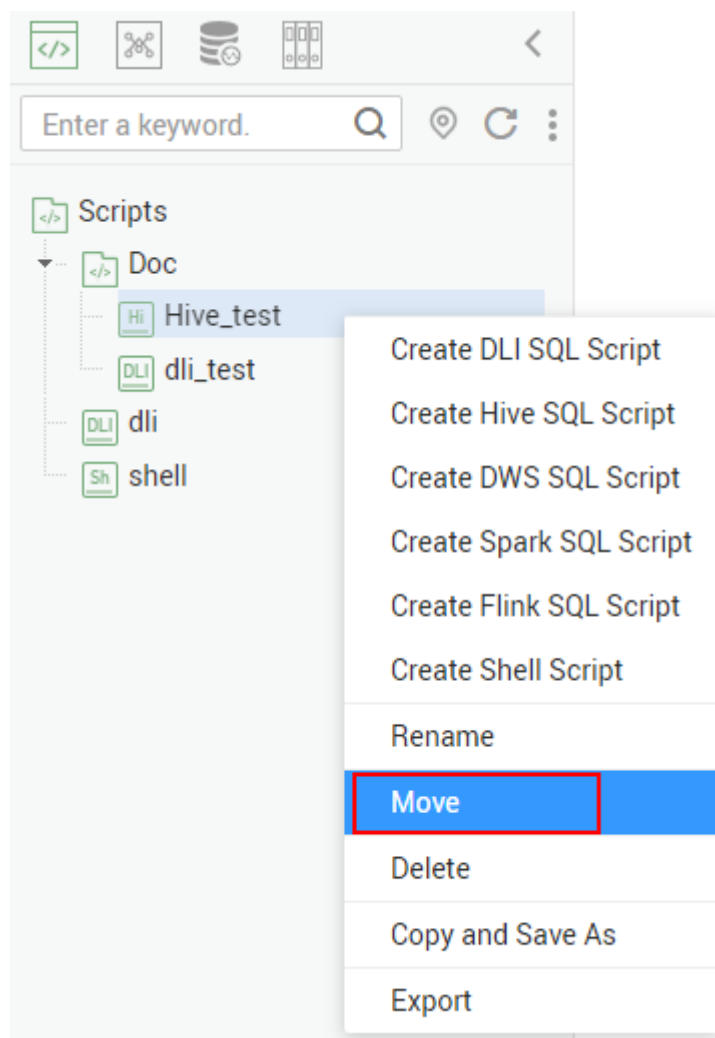
### Prerequisites

- You have developed a script. The script to be moved exists in the script directory.

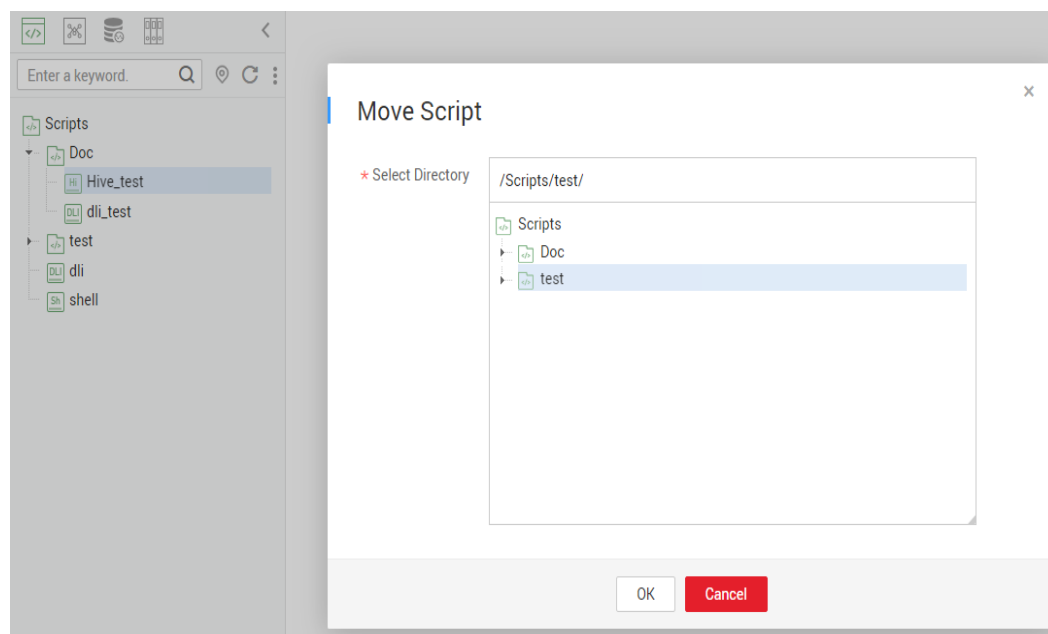
For details about how to develop scripts, see [Developing an SQL Script](#) and [Developing a Shell Script](#).

### Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
- Step 3** In the script directory, select the script to be moved. Right-click the script name and choose **Move** from the shortcut menu.

**Figure 5-6** Choosing Move

**Step 4** In the displayed dialog box, configure related parameters. [Table 5-8](#) describes the parameters.

**Figure 5-7** Moving a script**Table 5-8** Script moving parameters

Parameter	Description
Select Directory	Directory to which the script is to be moved. The parent directory is the <b>root</b> directory by default.

**Step 5** Click **OK**.

----End

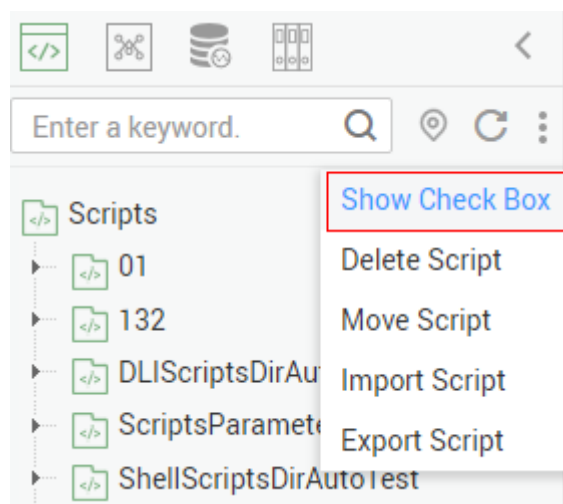
## 5.1.6 Exporting and Importing a Script

### Exporting a Script

You can export one or more script files from the script directory.

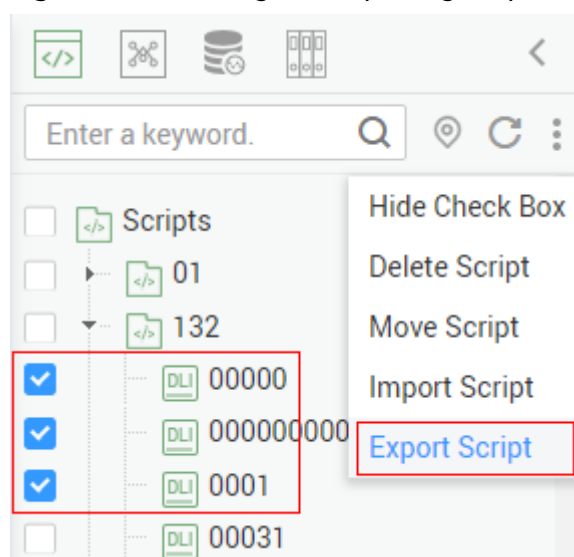
**Step 1** Click  in the script directory and select **Show Check Box**.

### Figure 5-8 Clicking Show Check Box



**Step 2** Select the scripts to be exported, click , and choose **Export Script**.


**Figure 5-9** Selecting and exporting scripts

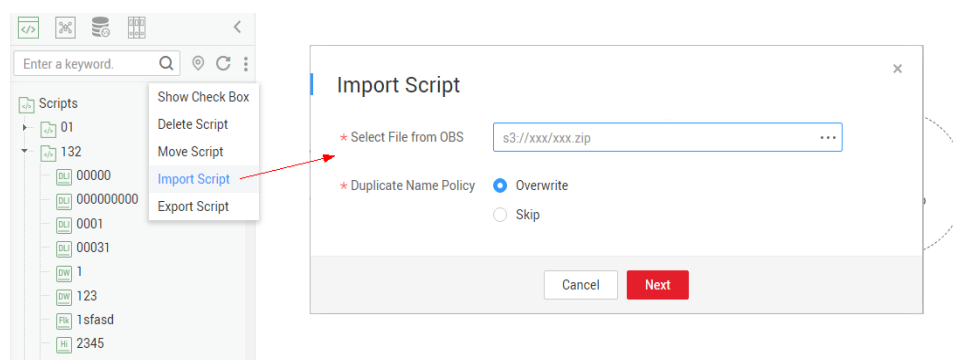


----End

## Importing a Script

You can import one or more script files in the script directory.

**Step 1** Click  and choose **Import Script** in the script directory, select the script file that has been uploaded to OBS, and set **Duplicate Name Policy**.

**Figure 5-10** Importing a Script

**Step 2** Click **Next**.

----End

## 5.1.7 Deleting a Script

If you do not need to use a script any more, perform the following operations to delete it.

### NOTICE



If you forcibly delete a script that is being associated with a job, ensure that services are not affected by going to the job development page and reassociating an available script with the job.

## Deleting a Script

- Step 1** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
- Step 2** In the script directory, right-click the script that you want to delete and choose **Delete** from the shortcut menu.
- Step 3** In the displayed dialog box, click **OK**.

----End

## Batch Deleting Scripts

- Step 1** In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
- Step 2** On the top of the script directory, click  and select **Show Check Box**.
- Step 3** Select the scripts to be deleted, click , and select **Batch Delete**.
- Step 4** In the displayed dialog box, click **OK** to delete scripts in batches.

----End

## 5.1.8 Copying a Script

This section describes how to copy a script.

### Prerequisites

The script file to be copied exists in the script directory.

### Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the DLF console, choose **Development > Develop Script**.
- Step 3** In the script directory, select the script to be copied, right-click the script name, and choose **Copy Save As**.
- Step 4** In the displayed dialog box, configure related parameters. [Table 5-9](#) describes the parameters.

**Table 5-9** Script directory parameters

Parameter	Description
Script Name	Name of the script. It contains a maximum of 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. <b>NOTE</b> The name of the copied script cannot be the same as the name of the original script.
Select Directory	Parent directory of the script directory. The parent directory is the root directory by default.

- Step 5** Click **OK**.

----End

## 5.2 Job Development

### 5.2.1 Creating a Job

A job is composed of one or more nodes that are performed collaboratively to complete data operations. Before developing a job, create a new one.

#### (Optional) Creating a Directory

If a directory exists, you do not need to create one.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.

- Step 3** In the directory list, right-click a directory and choose **Create Directory** from the shortcut menu.
- Step 4** In the displayed dialog box, configure directory parameters. [Table 5-10](#) describes the directory parameters.

**Table 5-10** Job directory parameters

Parameter	Description
Directory Name	Name of the job directory. Must consist of 1 to 32 characters and contain only letters, digits, underscores (_), and hyphens (-).
Select Directory	Parent directory of the job directory. The parent directory is the root directory by default.

- Step 5** Click **OK**.

----End

## Creating a Job

The quantity of jobs is less than the maximum quota (10,000).

- Step 1** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.

- Step 2** Create a job using either of the following methods:

Method 1: In the area on the right, click **Create Job**.

Method 2: In the directory list, right-click a directory and choose **Create Job** from the shortcut menu.

- Step 3** In the displayed dialog box, configure job parameters. [Table 5-11](#) describes the job parameters.

**Table 5-11** Job parameters

Parameter	Description
Job Name	Name of the job. Must consist of 1 to 128 characters and contain only letters, digits, hyphens (-), underscores (_), and periods (.).
Processing Mode	Type of the job. <ul style="list-style-type: none"><li>• <b>Batch</b>: Data is processed periodically in batches based on the scheduling plan, which is used in scenarios with low real-time requirements.</li><li>• <b>Real-Time</b>: Data is processed in real time, which is used in scenarios with high real-time performance.</li></ul>

Parameter	Description
Creation Method	Selects a job creation mode. <ul style="list-style-type: none"><li>• Create Empty Job: Create an empty job.</li><li>• Create Based on Template: Create a job using a template.</li></ul>
Select Directory	Directory to which the job belongs. The root directory is selected by default.
Job Owner	Owner of the job.
Job Priority	Priority of the job. The value can be <b>High</b> , <b>Medium</b> , or <b>Low</b> .
Log Path	Selects the OBS path to save job logs. By default, logs are stored in a bucket named <b>dlf-log-{ProjectId}</b> . <b>NOTE</b> If you want to customize a storage path, select the bucket that you have created on OBS by referring to the instructions in <a href="#">Configuring a Log Storage Path</a> .

**Step 4** Click **OK**.

----End

## 5.2.2 Developing a Job

DLF allows you to develop existing jobs.

### Prerequisites

You have created a job. For details about how to create a job, see [Creating a Job](#).

### Compiling Job Nodes

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.

**Step 3** In the job directory, double-click a job that you want to develop. The job development page is displayed.

**Step 4** Drag the desired node to the canvas, move the mouse over the node, and select the icon and drag it to connect to another node.

#### NOTE

Each job can contain a maximum of 200 nodes.

----End

## Configuring Basic Job Information

After you configure the owner and priority for a job, you can search for the job by the owner and priority. The procedure is as follows:

Select a job. On the job development page, click the **Basic Job Information** tab. On the displayed page, configure parameters. [Table 5-12](#) describes the parameters.

**Table 5-12** Basic job information

Parameter	Description
Owner	An owner configured during job creation is automatically matched. This parameter value can be modified.
Executor	User that executes the job. When you enter an executor, the job is executed by the executor. If the executor is left unspecified, the job is executed by the user who submitted the job for startup.
Priority	Priority configured during job creation is automatically matched. This parameter value can be modified.
Execution Timeout	Timeout of the job instance. If this parameter is set to 0 or is not set, this parameter does not take effect. If the notification function is enabled for the job and the execution time of the job instance exceeds the preset value, the system sends a specified notification.
Custom Parameter	Set the name and value of the parameter.



## Configuring Job Parameters



Job parameters can be globally used in any node in jobs. The procedure is as follows:

Select a job. On the job development page, click the **Job Parameter Setup** tab. On the displayed page, configure parameters. [Table 5-13](#) describes the parameters.

**Table 5-13** Job parameter setup

Function	Description
Variable Parameter	

Function	Description
Add	<p>Click <b>Add</b> and enter the variable parameter name and parameter value in the text boxes.</p> <ul style="list-style-type: none"><li>• Parameter name The parameter name must be unique, consist of 1 to 64 characters, and contain only letters, digits, underscores (_), hyphens (-), less-than signs (&lt;), and greater-than signs (&gt;).</li><li>• Parameter Value<ul style="list-style-type: none"><li>– The function type of parameter value starts with a dollar sign (\$). For example: <b>\$getCurrentTime(@@yyyyMMdd@@,0)</b></li><li>– The string type of parameter value is a character string. For example: <b>str1</b> When a character string and function are used together, use @@ to enclose the character string and use + to connect the character string and function. For example: <b>@@str1@@+\$getCurrentTime(@@yyyyMMdd@@,0)</b></li><li>– The numeric type of parameter value is a number or operation expression.</li></ul></li></ul> <p>After the parameter is configured, it is referenced in the format of <b><i>\${parameter name}</i></b> in the job.</p>
Modify	Modify the parameter name and parameter value in text boxes and save the modifications.
Save	Click <b>Save</b> to save the settings.
Delete	 Click  next to the parameter value text box to delete the job parameter.
<b>Constant Parameter</b>	

Function	Description
Add	<p>Click <b>Add</b> and enter the constant parameter name and parameter value in the text boxes.</p> <ul style="list-style-type: none"><li>• Parameter name The parameter name must be unique, consist of 1 to 64 characters, and contain only letters, digits, underscores (_), hyphens (-), less-than signs (&lt;), and greater-than signs (&gt;).</li><li>• Parameter Value<ul style="list-style-type: none"><li>– The function type of parameter value starts with a dollar sign (\$). For example: <b>\$getCurrentTime(@@yyyyMMdd@@,0)</b></li><li>– The string type of parameter value is a character string. For example: <b>str1</b> When a character string and function are used together, use @@ to enclose the character string and use + to connect the character string and function. For example: <b>@@str1@@+\$getCurrentTime(@@yyyyMMdd@@,0)</b></li><li>– The numeric type of parameter value is a number or operation expression.</li></ul></li></ul> <p>After the parameter is configured, it is referenced in the format of <i>\${parameter name}</i> in the job.</p>
Modify	Modify the parameter name and parameter value in text boxes and save the modifications.
Save	Click <b>Save</b> to save the settings.
Delete	<p> Click  next to the parameter value text box to delete the job constant.</p>

## Configuring Job Scheduling Tasks

You can configure job scheduling tasks for batch jobs. There are three scheduling types available: **Run once**, **Run periodically**, and **Event-driven**. The procedure is as follows:

Select a job. On the job development page, click the **Scheduling Parameter Setup** tab. On the displayed page, configure parameters. [Table 5-14](#) describes the parameters.

**Table 5-14** Scheduling parameter setup

Parameter	Description
Schedule Type	Job schedule type. Possible values: <ul style="list-style-type: none"><li>• <b>Run once</b>: The job will be run only once.</li><li>• <b>Run periodically</b>: The job will be run periodically.</li><li>• Event-driven: The job will be run when certain external conditions are met.</li></ul>
Parameters for <b>Run periodically</b>	
Effective Time	Period during which a job runs.
Schedule Cycle	Frequency at which a job is run. The job can be run once every: <ul style="list-style-type: none"><li>• <b>Minute</b></li><li>• <b>Hour</b></li><li>• <b>Day</b></li><li>• <b>Week</b></li><li>• <b>Month</b></li></ul>
Dependency Job	Job that is depended on. The constraints are as follows: <ul style="list-style-type: none"><li>• A short-cycle job cannot depend on a long-cycle job.</li><li>• A job whose schedule cycle is <b>Week</b> cannot depend on a job whose schedule cycle is <b>Minute</b>.</li><li>• A job whose schedule cycle is <b>Week</b> cannot depend on or be depended on by another job.</li><li>• A job whose schedule cycle is <b>Month</b> can depend only on the job whose schedule cycle is <b>Day</b>.</li></ul>

Parameter	Description
Action After Dependency Job Failure	Action that will be performed on the current job if the dependency job fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>Suspend</b> The current job will be suspended. A suspended job will block the execution of the subsequent jobs. You can manually force the dependency job to succeed to solve the blocking problem. For details, see <a href="#">Monitoring a Batch Job</a>.</li><li>• <b>Continue</b> The current job will continue to be performed.</li><li>• <b>Terminate</b> The current job will be terminated. Then the status of the current job will become <b>Canceled</b>.</li></ul>
Run upon completion of the dependency job's last schedule cycle.	Specify whether, before executing the current job, the dependency job's last schedule cycle must end.
Cross-Cycle Dependency	Dependency between instances in the job. Possible values: <ul style="list-style-type: none"><li>• <b>Independent on the previous schedule cycle</b></li><li>• <b>Self-dependent</b> (The current job can continue to run only after the previous schedule cycle ends.)</li></ul>
Parameters for <b>Event-driven</b>	
Triggering Event Type	Type of the event that triggers job running.
DIS Stream Name	Name of the DIS stream. When a new message is sent to the specified DIS stream, Data Development transfers the new message to the job to trigger the job running.
Concurrent Events	Number of jobs that can be concurrently processed. The maximum number of concurrent events is 128.
Event Detection Interval	Interval at which the system detects the DIS stream or OBS path for new messages. The unit of the interval can be set to second or minute.

Parameter	Description
Access Policy	Select the location where data is to be read: <ul style="list-style-type: none"><li>• Access from the previous location: For the first access, data is accessed from the most recently recorded location. For the subsequent access, data is accessed from the previously recorded location.</li><li>• Access from the latest location: Data is accessed from the most recently recorded location each time.</li></ul>
Failure Policy	Select a policy to be performed after scheduling fails. <ul style="list-style-type: none"><li>• Stop scheduling</li><li>• Ignore failure and proceed</li></ul>

## Configuring Node Properties

Click a node in the canvas. On the displayed **Node Properties** page, configure node properties. For details, see [Node Overview](#).

## Configuring Node Scheduling Tasks

You can configure node scheduling tasks for real-time jobs. There are three scheduling types available: **Run once**, **Run periodically**, and **Event-driven**. The procedure is as follows:

Select a node. On the node development page, click the **Scheduling Parameter Setup** tab. On the displayed page, configure parameters. [Table 5-15](#) describes the parameters.

**Table 5-15** Node scheduling setup

Parameter	Description
Schedule Type	Job schedule type. Possible values: <ul style="list-style-type: none"><li>• <b>Run once</b>: The job will be run only once.</li><li>• <b>Run periodically</b>: The job will be run periodically.</li><li>• <b>Event-driven</b>: The job will be run when certain external conditions are met.</li></ul>
Parameters for <b>Run periodically</b>	
Effective Time	Period during which a job runs.

Parameter	Description
Schedule Cycle	Frequency at which a job is run. The job can be run once every: <ul style="list-style-type: none"><li>• Minute</li><li>• Hour</li><li>• Day</li><li>• Week</li><li>• Month</li></ul>
Cross-Cycle Dependency	Dependency between instances in the job. Possible values: <ul style="list-style-type: none"><li>• <b>Independent on the previous schedule cycle</b></li><li>• <b>Self-dependent</b> (The current job can continue to run only after the previous schedule cycle ends.)</li></ul>
Parameters for <b>Event-driven</b>	
Triggering Event Type	Type of the event that triggers job running.
DIS Stream Name	Name of the DIS stream. When a new message is sent to the specified DIS stream, Data Development transfers the new message to the job to trigger the job running.
Concurrent Events	Number of jobs that can be concurrently processed. The maximum number of concurrent events is 10.
Event Detection Interval	Interval at which the system detects the DIS stream or OBS path for new messages. The unit of the interval can be set to second or minute.
Failure Policy	Select a policy to be performed after scheduling fails. <ul style="list-style-type: none"><li>• Stop scheduling</li><li>• Ignore failure and proceed</li></ul>

## More Node Functions

To obtain more node functions, right-click the node icon in the canvas and choose a function as required. [Table 5-16](#) describes the node functions.

**Table 5-16** More mode functions

Function	Description
Configure	Goes to the <b>Node Property</b> page of the node.

Function	Description
Delete	<p>Deletes one or more nodes at the same time.</p> <ul style="list-style-type: none"><li>Deleting one node: Right-click the node icon in the canvas and choose <b>Delete</b> or press the <b>Delete</b> shortcut key.</li><li>Deleting multiple nodes: Click the icons of the nodes to be deleted in the canvas while holding on <b>Ctrl</b>, right-click the blank area of the current job canvas, and choose <b>Delete</b> or press the <b>Delete</b> shortcut key.</li></ul>
Copy	<p>Copies one or more nodes to any job.</p> <ul style="list-style-type: none"><li>Single-node copy: You can either right-click the node icon in the canvas, choose <b>Copy</b>, and paste the node to a target location, or click the node icon in the canvas and press <b>Ctrl+C</b> and <b>Ctrl+V</b> to paste the node to a target location. The copied node carries the configuration information of the original node.</li><li>Multi-node copy: Click the icons of the nodes to be copied in the canvas while holding on <b>Ctrl</b>. Then you can either right-click the blank area of the canvas, choose <b>Copy</b>, and paste the nodes to a target location, or press <b>Ctrl+C</b> and <b>Ctrl+V</b> to paste the nodes to a target location. The copied node carries the configuration information of the original node, but does not contain the connection relationship between nodes.</li></ul>
Test Run	Runs the node for a test.
Upload Simulation Data	<p>Uploads the simulation data to the DIS Stream node. The following types of simulation data are provided:</p> <ul style="list-style-type: none"><li>Simulation data from track analysis</li><li>Simulation data from boiler abnormality monitoring</li></ul>
Add Note	Adds a note to the node. Each node can have multiple notes.
Edit Script	Goes to the script editing page and edits the associated script. This option is available only for the node associated with a script.

## Saving and Starting Jobs

After a job is configured, complete the following operations:

### Batch processing job

**Step 1** Click  to test the job.


**Step 2** After the test is completed, click  to save the job configuration information.

**Step 3** Click  to start the job.

----End

### Processing jobs in real time

**Step 1** Click  to save the job configuration.

**Step 2** Click  to submit and start the job.

#### NOTE

DLF allows you to modify a running real-time processing job. After the job is modified, click **Save**. On the page that is displayed, click **OK** to save the job.

----End

## 5.2.3 Renaming a Job

You can change the name of a job.

This section describes how to rename a job.

### Prerequisites

You have created a job. The job to be renamed exists in the job directory. For details about how to create a job, see [Creating a Job](#).

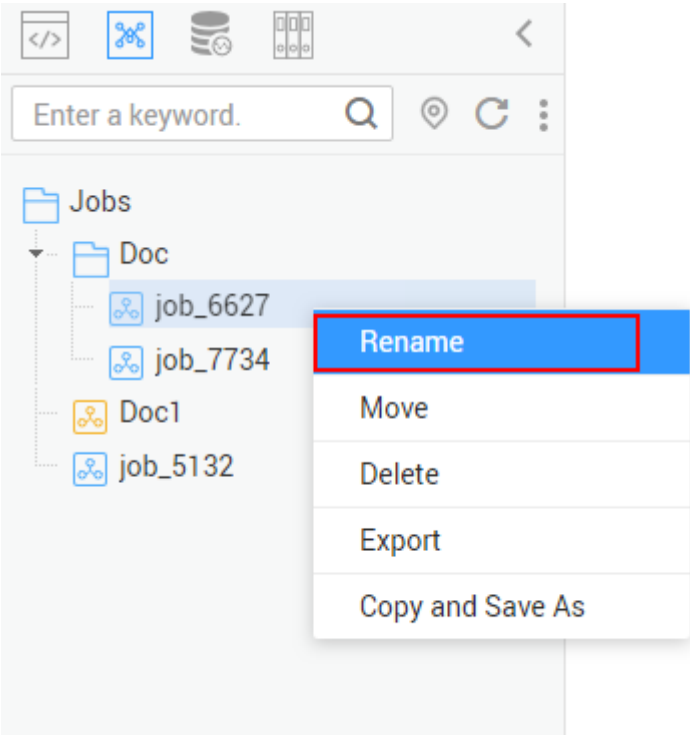
### Procedure

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.

**Step 3** In the job directory, select the job to be renamed. Right-click the job name and choose **Rename** from the shortcut menu.

Figure 5-11 Choosing Rename



**Step 4** On the page that is displayed, configure related parameters. [Table 5-17](#) describes the parameters.

Figure 5-12 Renaming a job

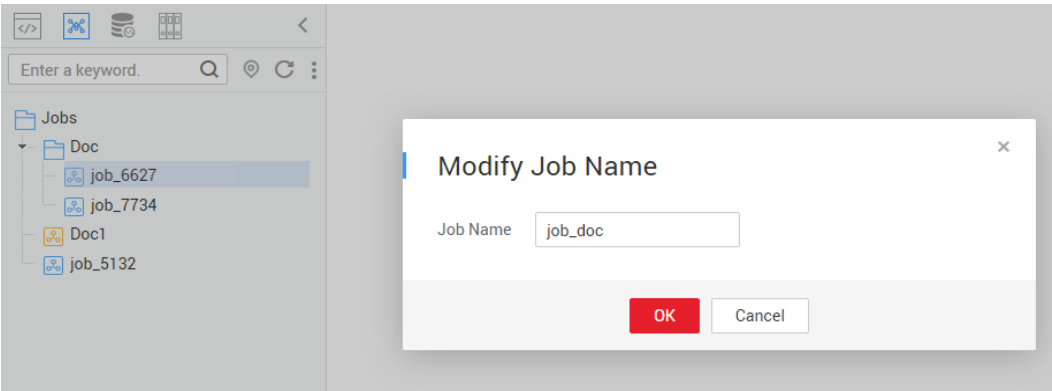


Table 5-17 Job renaming parameters

Parameter	Description
Job Name	Name of the job. Must consist of 1 to 128 characters and contain only letters, digits, hyphens (-), underscores (_), and periods (.).

**Step 5** Click **OK**.

----End

## 5.2.4 Moving a Job

You can move a job from the current directory to another directory.

This section describes how to move a job.

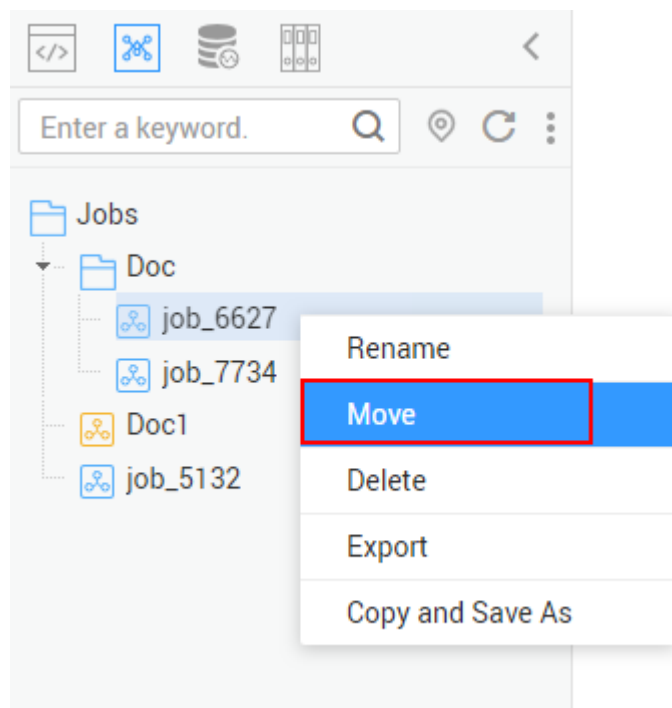
### Prerequisites

You have created a job. The job to be moved exists in the job directory. For details about how to create a job, see [Creating a Job](#).

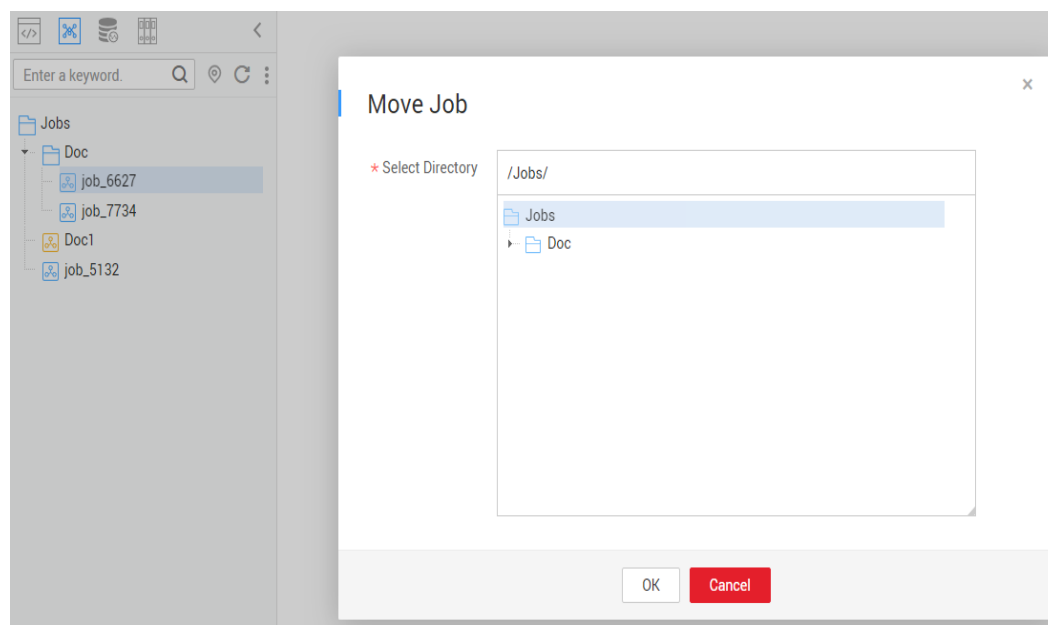
### Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.
- Step 3** In the job directory, select the job to be moved. Right-click the job name and choose **Move** from the shortcut menu.

**Figure 5-13** Choosing Move



- Step 4** In the displayed dialog box, configure related parameters. [Table 5-18](#) describes the parameters.

**Figure 5-14** Moving a job**Table 5-18** Job moving parameters

Parameter	Description
Select Directory	Directory to which the job is to be moved. The parent directory is the <b>root</b> directory by default.


**Step 5** Click **OK**.

-----End

## 5.2.5 Exporting and Importing a Job

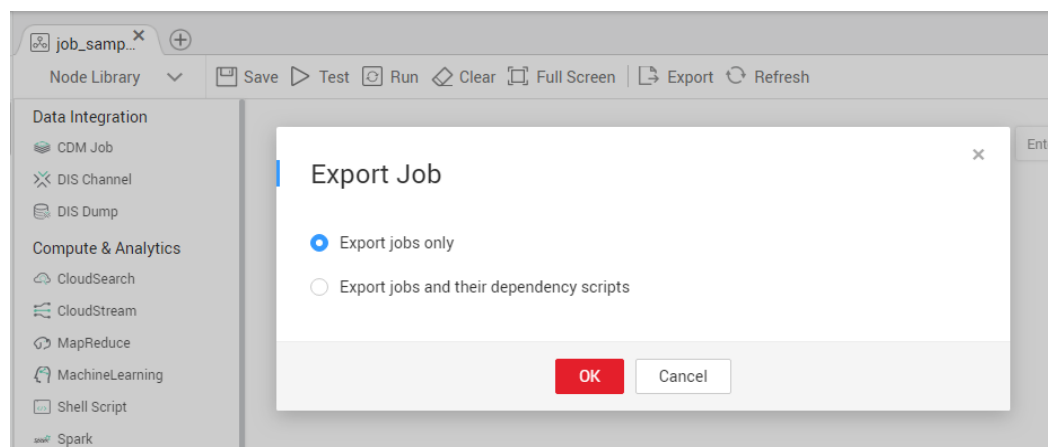
### Exporting Jobs

Method 1: Export a job on the job development page.

**Step 1** Go to the development page of a job, click , and select the type of the job to be exported.

- **Export jobs only:** Export the connection relationships and property configurations of nodes, excluding sensitive information such as passwords. After the export is completed, the user obtains a .zip file.
- **Export jobs and their dependency scripts:** Export the node connection relationships, node property configurations, job scheduling configurations, parameter configurations, and dependency scripts, excluding sensitive information such as passwords. After the export is completed, the user obtains a .zip file.

**Figure 5-15** Exporting a job (method 1)



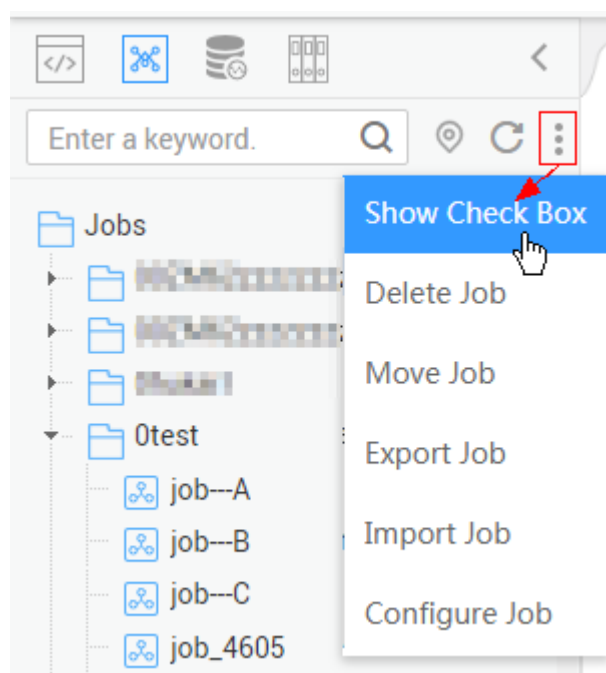
**Step 2** Click **OK** to export the required job file.


----End

Method 2: Export one or more jobs from the job directory.

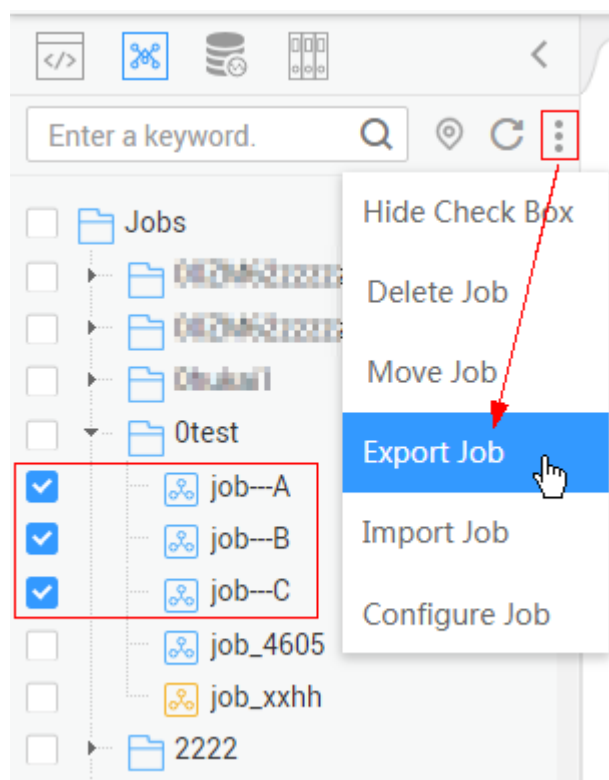
**Step 1** Click  in the job directory and select **Show Check Box**.

**Figure 5-16** Show Check Box



**Step 2** Select the job to be exported and choose  > **Export Job** to export the job and its dependencies.

**Figure 5-17** Selecting and exporting a job



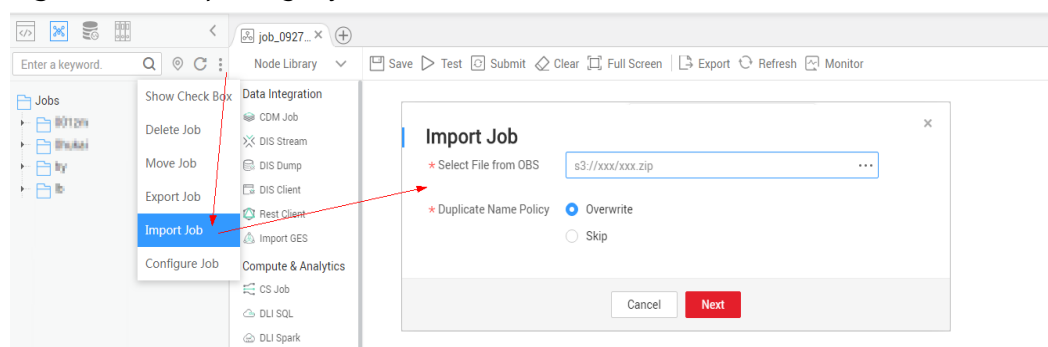
----End

## Importing a Job

Export one or more jobs from the job directory.

- Step 1** Click > **Import Job** in the job directory, select the job file that has been uploaded to OBS, and rename the policy.

**Figure 5-18** Importing a job



- Step 2** Click **Next** to import the job as instructed.

### NOTE

During the import, if the data connection, DIS stream, DLI queue, or GES graph associated with the job does not exist in DLF, the system prompts you to select one again.

----End

## Example

Context:


- A DWS data connection **doctest** is created in DLF.
- A real-time job **doc1** is created in the job directory. Node **DWS SQL** is added to the job. The **Data Connection** of the node is set to **doctest**. **SQL Script** and **Database** are both configured.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the DLF console, choose **Development > Develop Script**.

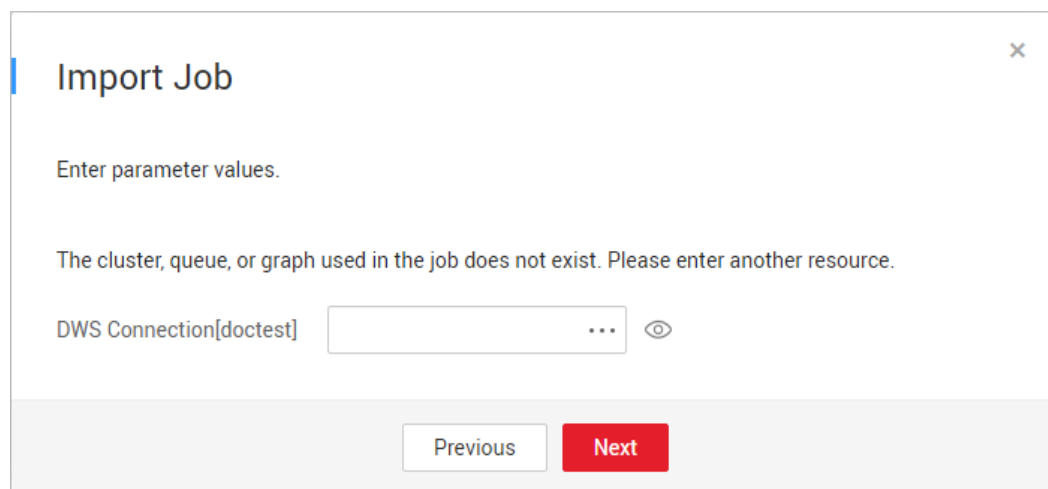
**Step 3** Search for **doc1** in the job search box, export the job to the local host, and then upload it to the OBS folder.

**Step 4** Delete the **doctest** data connection associated with the job in DLF.

**Step 5** Click  > **Import Job** in the job directory, select the job file that has been uploaded to OBS, and set the duplicate name policy.

**Step 6** Click **Next** and select another data connection as prompted.

**Figure 5-19** Importing a job



**Step 7** Click **Next** and then **Close**.

----End

## 5.2.6 Deleting a Job

If you do not need to use a job any more, perform the following operations to delete it to reduce the quota usage of the job.



### NOTE

Deleted jobs cannot be recovered. Exercise caution when performing this operation.

## Deleting a Script

- Step 1** Log in to the DLF console.
  - Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.
  - Step 3** In the job directory, right-click the job that you want to delete and choose **Delete** from the shortcut menu.
  - Step 4** In the displayed dialog box, click **OK**.
- End

## Batch Deleting Scripts

- Step 1** Log in to the DLF console.
  - Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.
  - Step 3** On the top of the job directory, click  and select **Show Check Box**.
  - Step 4** Select the jobs to be deleted, click , and select **Batch Delete**.
  - Step 5** In the displayed dialog box, click **OK**.
- End

## 5.2.7 Copying a Job

This section describes how to copy a job.

### Prerequisites

The job file to be copied exists in the job directory.

### Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.
- Step 3** In the job directory, select the job to be copied, right-click the job name, and choose **Copy Save As**.
- Step 4** In the displayed dialog box, configure related parameters. [Table 5-19](#) describes the parameters.

**Table 5-19** Job and directory parameters

Parameter	Description
Job Name	Name of the job. Must consist of 1 to 128 characters and contain only letters, digits, hyphens (-), underscores (_), and periods (.).
Select Directory	Parent directory of the job directory. The parent directory is the root directory by default.

**Step 5** Click **OK**.

----End

# 6 Solution

## Context

The solution aims to provide users with convenient and systematic management operations and better meet service requirements and objectives. Each solution can contain one or more business-related jobs, and one job can be reused by multiple solutions.

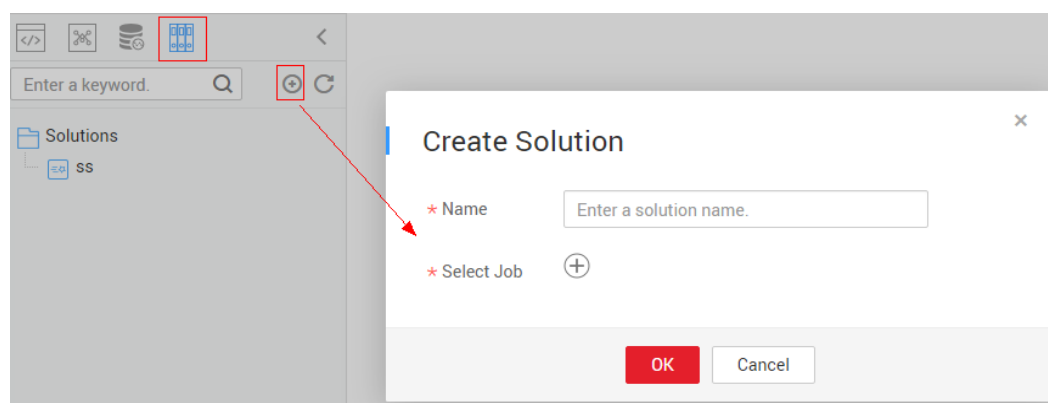
You can perform the following operations on a solution:



- [Creating a Solution](#)
- [Editing a Solution](#)
- [Editing a Solution](#)
- [Importing a Solution](#)
- [Upgrading a Solution](#)
- [Deleting a Solution](#)

## Creating a Solution

On the development page of DLF, create a solution, set the solution name, and select business-related jobs.

**Figure 6-1** Creating a solution



- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Development > Develop Script/Data Development > Develop Job**.
- Step 3** In the upper part of the directory on the left, click . The solution directory is displayed.
- Step 4** Click  in the upper part of the solution directory. The **Create Solution** page is displayed. [Table 6-1](#) describes the solution parameters.

**Table 6-1** Solution Parameters

Parameter	Description
Name	Name of the solution.
Select Job	Select the jobs contained in the solution.

- Step 5** Click **OK**. The new solution is displayed in the directory on the left.

----End

## Editing a Solution

In the solution directory, right-click the solution name and choose **Edit** from the shortcut menu.

## Exporting a Solution

In the solution directory, right-click the solution name and choose **Export** from the shortcut menu to export the solution file in ZIP format to the local host.

## Importing a Solution

In the solution directory, right-click a solution and choose **Import Solution** from the shortcut menu to import the solution file that has been uploaded to OBS.

## Upgrading a Solution

In the solution directory, right-click the solution name and choose **Upgrade** from the shortcut menu to import the solution file that has been uploaded to OBS. During the solution upgrade, the running jobs are stopped. The system determines whether to restart the jobs after the upgrade based on the configured upgrade restart policy.

## Deleting a Solution

In the solution directory, right-click the solution name and choose **Delete** from the shortcut menu. A deleted solution cannot be restored. Exercise caution when performing this operation.

# 7 O&M and Scheduling

---

## 7.1 Overview

Log in to the DLF console and choose **Monitoring > Overview**. On the **Overview** page, you can view the statistics of job instances in charts. Currently, you can view four types of statistics:

- Today's Job Instance Scheduling
- Latest 7 Days' Job Instance Scheduling
- Top 10 Job Instances with the Longest Execution Time in the Last 30 Days  
Click a job name to go to the **Monitor Instance** page and view the detailed running records of the job instance with a long execution time.
- Top 10 Job Instances with the Largest Number of Running Exceptions in the Last 30 Days

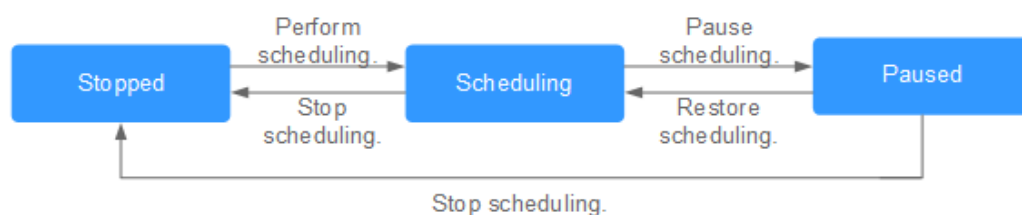
Click the value in the **Exception Count** column. On the displayed **Monitor Instance** page, view the detailed running records of the job instance that is running abnormally.

## 7.2 Job Monitoring

### 7.2.1 Monitoring a Batch Job

#### Batch Processing: Scheduling Jobs

After developing a job, you can manage job scheduling tasks on the **Monitor Job** page. Specific operations include to run, pause, restore, or stop scheduling.

**Figure 7-1** Scheduling a job

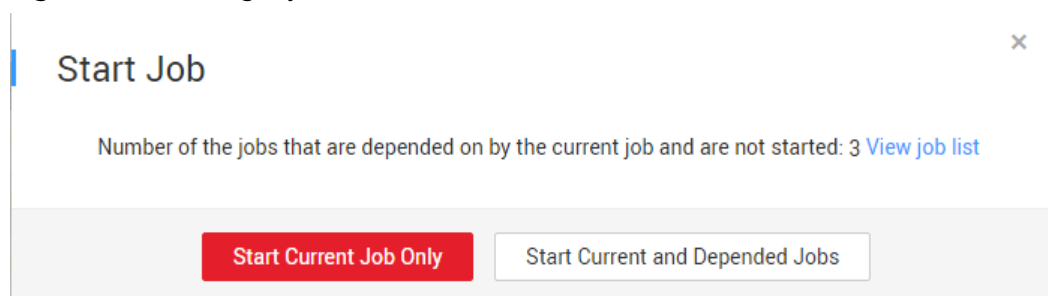
- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab.
- Step 4** In the **Operation** column of the job, click **Run/Pause/Restore/Stop**.
- End

## Batch Processing: Scheduling the Depended Jobs

You can configure whether to start the depended jobs when scheduling a batch job on the job monitoring page. For details about how to configure depended jobs, see [Configuring Job Scheduling Tasks](#).

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab and select a job that has depended jobs.
- Step 4** In the **Operation** column of the job, click **Schedule**.

You can start only the current job or start the depended jobs at the same time when scheduling the job.

**Figure 7-2** Starting a job

----End

## Batch Processing: Notification Settings

You can configure DLF to notify you of job success or failure. The following provides the method for configuring a notification task. You can also configure a

notification task on the **Monitoring** page. For details, see [Managing a Notification](#).

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab.
- Step 4** In the **Operation** column of the job, choose **More > Set Notification**. In the displayed dialog box, configure notification parameters. [Table 7-9](#) describes the notification parameters.
- Step 5** Click **OK**.
- End

## Batch Processing: Instance Monitoring

You can view the running records of all instances of a job on the **Monitor Instance** page.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab.
- Step 4** In the **Operation** column of a job, choose **More > Monitor Instance** to view the running records of all instances of the job.
- For details about the **Operation** column of the instance, see [Table 7-2](#).
  - For details about the **Operation** column of the node, see [Table 7-3](#).
- End

## Batch Processing: Scheduling Configuration

You can perform the following steps to go to the development page of a specific job.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab.
- Step 4** In the **Operation** column of a job, choose **More > Configure Scheduling**.
- End

## Batch Processing: Job Dependency View

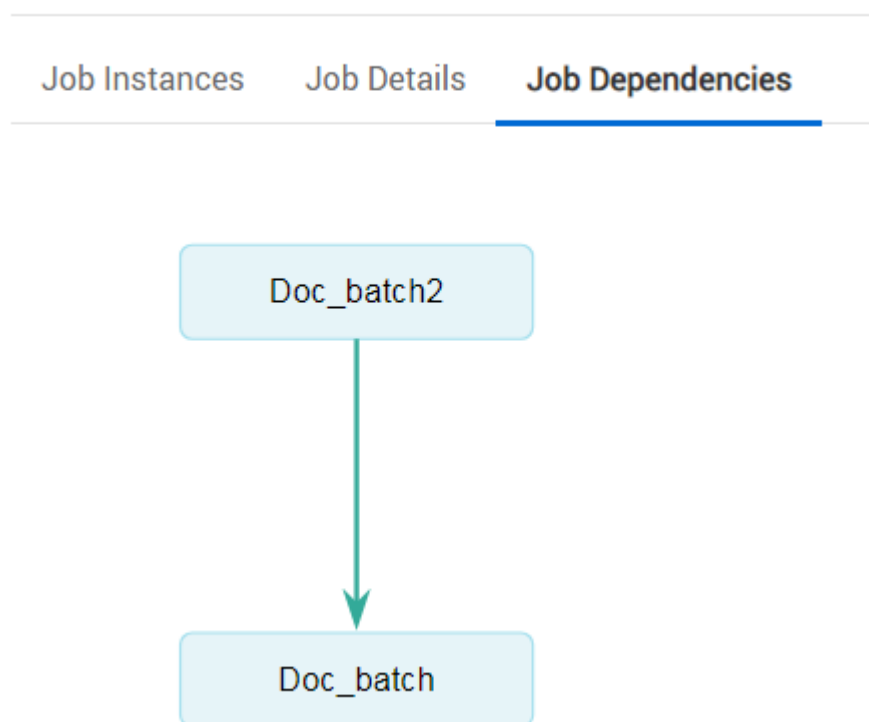
You can view the dependencies between jobs.

- Step 1** Log in to the DLF console.

- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab.
- Step 4** Click the job name and click the **Job Dependencies** tab. View the dependencies between jobs.

**Figure 7-3** Job dependencies view

Job Monitoring > Doc\_batch



Click a job in the view. The development page of the job will be displayed.

----End

## Batch Processing: PatchData

A job executes a scheduling task to generate a series of instances in a certain period of time. This series of instances are called PatchData. PatchData can be used to fix the job instances that have data errors in the historical records or to build job records for debugging programs.

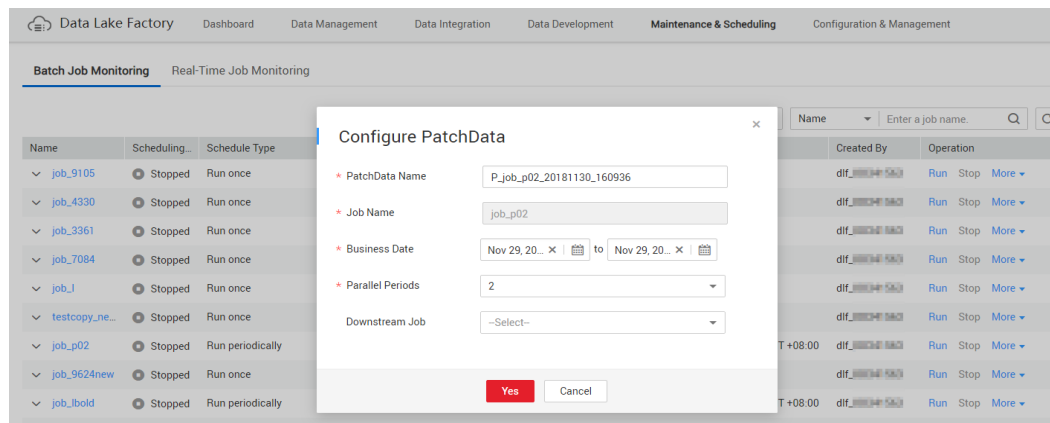
Only the periodically scheduled jobs support PatchData. For details about the execution records of PatchData, see [PatchData Monitoring](#).

### NOTE

Do not modify the job configuration when PatchData is being performed. Otherwise, job instances generated during PatchData will be affected.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.
- Step 3** Click the **Batch Job Monitor** tab.
- Step 4** In the **Operation** column of the job, choose **More > PatchData**.
- Step 5** Configure PatchData parameters. [Table 7-1](#) shows the PatchData parameters.

**Figure 7-4** PatchData parameters



**Table 7-1** Parameter description

Parameter	Description
PatchData Name	Name of the automatically generated PatchData task. The value can be modified.
Job Name	Name of the job that requires PatchData.
Date	Period of time when PatchData is required.
Parallel Instances	Number of instances to be executed at the same time. A maximum of five instances can be executed at the same time.
Downstream Job Requiring PatchData	Downstream job (job that depends on the current job) that requires PatchData. You can select more than one downstream jobs.

- Step 6** Click **OK**. The system starts to perform PatchData and the **PatchData Monitoring** page is displayed.

-----End

## Batch Processing: Batch Processing Jobs

You can schedule and stop jobs and configure notification tasks in batches.

- Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.

**Step 3** Click the **Batch Job Monitor** tab.

**Step 4** Select the jobs and click **Schedule/Stop/Configure Notification** to process the jobs in batches.

----End


## Batch Processing: Viewing Latest Instances

This function enables you to view the information about five instances that are running in a job.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.

**Step 3** Click the **Batch Job Monitor** tab.

**Step 4** Click  in front of the job name. The page of the latest instance is displayed. You can view the details about the nodes contained in the latest instances.

----End

## Batch Processing: Viewing All Instances

You can view all running records of a job on the **Running History** page and perform more operations on instances or nodes based on site requirements.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Monitor Job**.

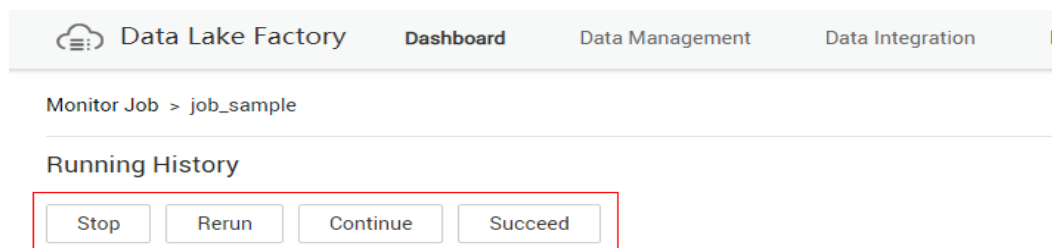
**Step 3** Click the **Batch Job Monitor** tab.

**Step 4** Click a job name. The **Running History** page is displayed.

You can stop, rerun, continue to run, or forcibly run jobs in batches. For details, see [Table 7-2](#).

When multiple instances are rerun in batches, the sequence is as follows:


- If a job does not depend on the previous schedule cycle, multiple instances run concurrently.
- If jobs are dependent on their own, multiple instances are executed in serial mode. The instance that first finishes running in the previous schedule cycle is the first one to rerun.

**Figure 7-5** Batch operations

**Step 5** View actions in the **Operation** column of an instance. [Table 7-2](#) describes the actions that can be performed on the instance.

**Table 7-2** Actions for an instance

Action	Description
Stop	Stops an instance that is in the <b>Waiting</b> , <b>Running</b> , or <b>Abnormal</b> state.
Rerun	Reruns an instance that is in the <b>Succeeded</b> or <b>Canceled</b> .
View Waiting Job Instance	When the instance is in the waiting state, you can view the waiting job instance.
Continue	If an instance is in the <b>Abnormal</b> state, you can click <b>Continue</b> to begin running the subsequent nodes in the instance. <b>NOTE</b> This operation can be performed only when <b>Failure Policy</b> is set to <b>Suspend the current job execution plan</b> . To view the current failure policy, click a node and then click <b>Advanced Settings</b> on the <b>Node Properties</b> page.
Succeed	Forcibly changes the status of an instance from <b>Abnormal</b> , <b>Canceled</b> , <b>Failed</b> to <b>Succeed</b> .
View	Goes to the job development page and view job information.

**Step 6** Click  in front of an instance. The running records of all nodes in the instance are displayed.

**Step 7** View actions in the **Operation** column of a node. [Table 7-3](#) describes the actions that can be performed on the node.

**Table 7-3** Actions for a node

Action	Description
View Log	View the log information of the node.

Action	Description
Manual Retry	To run a node again after it fails, click <b>Retry</b> . <b>NOTE</b> This operation can be performed only when <b>Failure Policy</b> is set to <b>Suspend the current job execution plan</b> . To view the current failure policy, click a node and then click <b>Advanced Settings</b> on the <b>Node Properties</b> page.
Succeed	To change the status of a failed node to <b>Succeed</b> , click <b>Succeed</b> . <b>NOTE</b> This operation can be performed only when <b>Failure Policy</b> is set to <b>Suspend the current job execution plan</b> . To view the current failure policy, click a node and then click <b>Advanced Settings</b> on the <b>Node Properties</b> page.
Skip	To skip a node that is to be run or that has been paused, click <b>Skip</b> .
Pause	To pause a node that is to be run, click <b>Pause</b> . Nodes queued after the paused node will be blocked.
Resume	To resume a paused node, click <b>Resume</b> .

----End

## 7.2.2 Monitoring a Real-Time Job

You can view information such as the running status, start time, and end time of the real-time processing job on the **Real-Time Job Monitoring** page, and start and stop the real-time processing job.

**Figure 7-6** Real-time job monitoring page

Batch Job Monitoring Real-Time Job Monitoring

Start Stop Enter a job or owner name: Q All (23) All

Name	Status	Priority	Start Time	End Time	Owner	Last Modified By	Created By	Operation
<input type="checkbox"/> job_zm3	Stopped	High	-	-		dif_j00341563	dif_j00341563	Start Stop
<input type="checkbox"/> job_zm	Stopped	High	-	-		dif_j00341563	dif_j00341563	Start Stop
<input type="checkbox"/> job_6294	Normal	High	Mar 28, 2019 20:23:52 GMT	-		dif_j00341563	dif_j00341563	Start Stop

- Batch processing jobs: Select jobs and click **Start/Stop**.
- Viewing node information of the job: Click a job name. On the page that is displayed, view the node connection relationship and monitoring information.
- Viewing status of the latest instances: Click the drop-down arrow next to the name of a job to view the latest instance status.

## Real-Time Jobs: Configuring Event-based Scheduling on a Node

If event-driven scheduling is configured for a node in a real-time job, right-click the node and choose **Configure Scheduling** from the shortcut menu to view and modify the scheduling information about the node.

**Step 1** Log in to the DLF console.

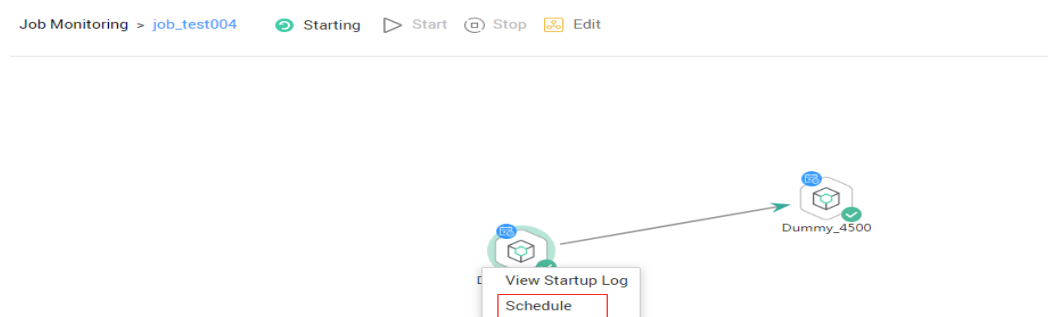
**Step 2** In the navigation tree of the DLF console, choose **Monitoring > Monitor Job**.

**Step 3** On the **Real-Time Job Monitoring** tab page, click a job name.

**Step 4** Click the node where event-driven scheduling is configured and choose **Configure Scheduling**. Set the parameters described in [Table 7-4](#).

When you click **View Startup Log**, you can view the job startup log information.

**Figure 7-7** Configuring scheduling



- When **Triggering Event Type** of **Event-Driven Scheduling** is set to **DIS**, configure a DIS scheduling policy as follows:

**Figure 7-8** Configuring a DIS scheduling policy

**Configure Scheduling Policy**

★ DIS Stream ?

★ Concurrent Events ?

★ Event Detection Interval ?

★ Failure Policy ? ☒ Suspend  
☐ Ignore the failure and proceed with the next event

**Table 7-4** Policy parameters

Parameter	Description
DIS Stream	Name of the DIS stream. When a new message is sent to the specified DIS stream, Data Development transfers the new message to the job to trigger the job running.
Concurrent Events	Number of jobs that can be concurrently processed. The maximum number of concurrent events is 10.
Event Detection Interval	Interval at which the system detects the DIS stream or OBS path for new messages. The unit of the interval can be set to second or minute.
Failure Policy	Select a policy to be performed after scheduling fails. <ul style="list-style-type: none"><li>• Stop scheduling</li><li>• Ignore failure and proceed</li></ul>

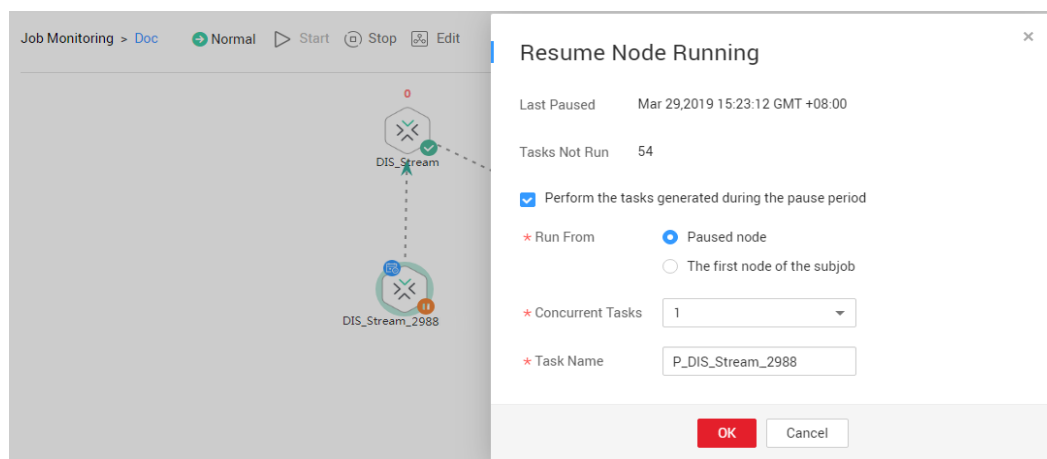
----End

## Rerunning a Job After the Job Is Paused

If event-based scheduling is configured for a node in a real-time job, you can disable the node and then restore it. When the node is restored, you can select where you can resume running.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the DLF console, choose **Monitoring > Monitor Job**.
- Step 3** On the **Real-Time Job Monitoring** tab page, click a job name.
- Step 4** On the job monitoring page, right-click a node with event-driven scheduling configured and choose **Disable** from the shortcut menu.
- Step 5** Right-click the node and choose **Resume** from the shortcut menu. The **Resume Node Running** dialog box is displayed, as shown in [Table 7-5](#).

**Figure 7-9** Resuming node running



**Table 7-5** Resumption parameters

Parameter	Description
Last Paused	Start time when a node is suspended.
Tasks Not Run	Number of tasks that are not running during node suspension.
Parameters for performing the tasks generated during the pause period	
Run From	Position from which running restarts. <ul style="list-style-type: none"><li>Paused node</li><li>The first node of the subjob</li></ul>
Concurrent Tasks	Number of tasks to be processed.
Task Name	Task to be resumed.

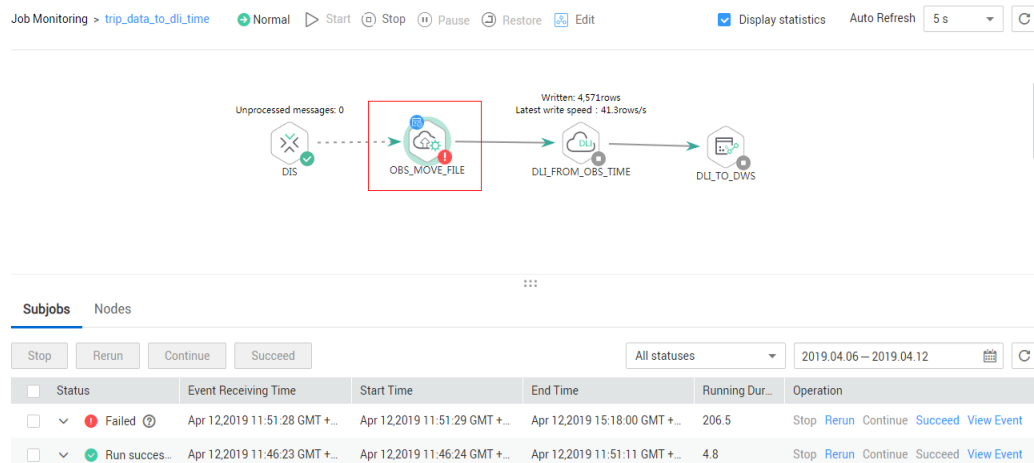
----End

## 7.2.3 Monitoring Real-Time Subjobs

When event-based scheduling is configured for a node in a job, you can click this node to query monitoring information of subjobs. On the **Subjob** page, you can stop, rerun, continue, and succeed subjobs as well as view subjob events.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the DLF console, choose **Monitoring > Monitor Job**.
- Step 3** On the **Real-Time Job Monitoring** tab page, click a job name.
- Step 4** Click a node with event-based scheduling configured, as shown in **Figure 7-10**.


**Figure 7-10** Subjob monitoring page



**Table 7-6** describes the actions listed in the **Operation** column of each subjob.

**Table 7-6** Actions in the **Operation** column

Action	Description
Stop	Stops a subjob instance that is in the <b>Running</b> state.
Rerun	Reruns a subjob instance that is in the <b>Succeed</b> or <b>Failed</b> state.
Continue	If a subjob instance is in the <b>Abnormal</b> state, you can click <b>Continue</b> to begin running the subsequent nodes in the subjob instance. <b>NOTE</b> This operation can be performed only when <b>Failure Policy</b> is set to <b>Suspend the current job execution plan</b> . To view the current failure policy, click a node and then click <b>Advanced Settings</b> on the <b>Node Properties</b> page.
Succeed	Forcibly changes the status of a subjob instance from <b>Failed</b> to <b>Succeed</b> .
View Event	Displays the event content of a subjob.

- Step 5** Click  in the **Status** column. The running records of the subjob node are displayed.

**Table 7-7** describes the actions that can be performed on the node.

**Table 7-7** Actions for a node

Action	Description
View Log	View the log information of the node.
Manual Retry	To run a node again after it fails, click <b>Retry</b> . <b>NOTE</b> This operation can be performed only when <b>Failure Policy</b> is set to <b>Suspend the current job execution plan</b> . To view the current failure policy, click a node and then click <b>Advanced Settings</b> on the <b>Node Properties</b> page.
Succeed	To change the status of a failed node to <b>Succeed</b> , click <b>Succeed</b> . <b>NOTE</b> This operation can be performed only when <b>Failure Policy</b> is set to <b>Suspend the current job execution plan</b> . To view the current failure policy, click a node and then click <b>Advanced Settings</b> on the <b>Node Properties</b> page.
Skip	To skip a node that is to be run or that has been paused, click <b>Skip</b> .
Pause	To pause a node that is to be run, click <b>Pause</b> . Nodes queued after the paused node will be blocked.
Resume	To resume a paused node, click <b>Resume</b> .

----End

## 7.3 Instance Monitoring

In the navigation tree of the Data Development console, choose **Monitoring**. On the Monitor Instance page, you can view the job instance information and perform more operations on instances and nodes as required. For details, see [Batch Processing: Viewing All Instances](#).

If the job instance is in the **Failed** state, you can mouse over the question mark and view the pop-up message to understand the failure cause, as shown in [Figure 7-11](#).

**Figure 7-11** Job instance failure prompt

Monitor Instance				
<div>Stop Rerun Continue Succeed</div>				
<input type="checkbox"/>	Job Name	Status	Running Type	Event Receiving Time
<input type="checkbox"/>	test	<span>Failed</span> <span>?</span>	Normal Scheduling	Mar 26,2019 18:01:06 GMT ...
<input type="checkbox"/>	job_9993	<span>Run su...</span>	Normal Scheduling	Mar 26,2019 17:55:57 GMT ...

## Rerunning Job Instances

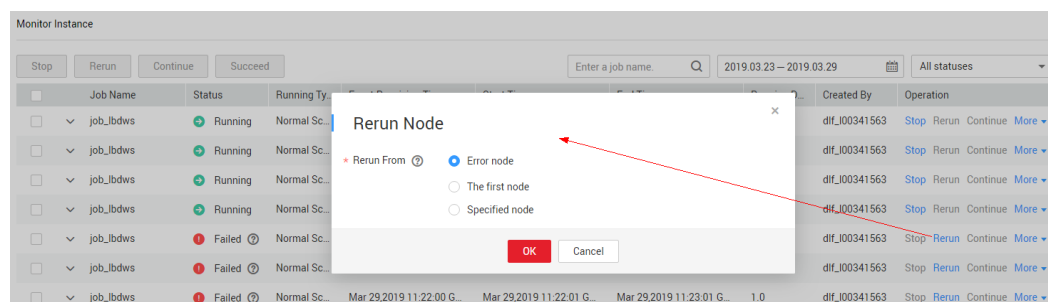
You can rerun a job instance that is successfully executed or fails to be executed by setting its rerun position.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the DLF console, choose **Monitoring > Monitor Instance**.

**Step 3** In the **Operation** column of a job, click **Rerun** to rerun the job instance. Alternatively, click the check box on the left of a job, and then click the **Rerun** button to rerun the job.

**Figure 7-12** Setting the rerunning position



**Table 7-8** Parameter description

Parameter	Description
Rerun From	<p>Select the start position from which the job instance reruns.</p> <ul style="list-style-type: none"><li>• <b>Error node:</b> When a job instance fails to be run, it reruns since the error node of the job instance.</li><li>• <b>The first node:</b> When a job instance fails to be run, it reruns since the first node of the job instance.</li><li>• <b>Specified node:</b> When a job instance fails to be run, it reruns since the node specified in the job instance.</li></ul> <p><b>NOTE</b> A job instance reruns from its first node if either of the following cases occurs:</p> <ul style="list-style-type: none"><li>• The quantity or name of a node in the job changes.</li><li>• The job instance has been successfully run.</li></ul>

----End

## 7.4 PatchData Monitoring

In the navigation tree of the Data Development console, choose **Monitoring > Monitor PatchData**.

On the , you can view the task status, service date, number of parallel periods, and PatchData job names, and stop a running task.

**Figure 7-13** PatchData Monitoring page

PatchData Monitoring

<input type="text" value="Enter a PatchData name"/> <input type="button" value="Q"/> <input type="button" value="All"/> <input type="button" value="C"/>					
PatchData Name	Status	Date	Parallel Periods	PatchData Job Name	Operation
P_wangwei_20190328_205650	Running	Mar 27 2019 00:00:00 GMT +08:00 - Mar ...	1	wangwei	Stopped

On the page shown in [Figure 7-13](#), click PatchData name. On the displayed page, you can view the PatchData execution status. For more information, see [Batch Processing: Viewing All Instances](#).

**Figure 7-14** PatchData Monitoring details

Monitor PatchData > P_job_5433_20190409_172351					
Date : 2019/04/08 - 2019/04/08					
Task Name	Status	Plan Time	Start Time	End Date	Operation
^ 2019/04/08	Successfully				
^ job_5433	Run successfully	2019/Apr/08 23:58:00 GMT +08:00	2019/Apr/09 21:39:04 GMT +08:00	2019/Apr/09 21:39:56 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:56:00 GMT +08:00	2019/Apr/09 21:38:34 GMT +08:00	2019/Apr/09 21:39:10 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:54:00 GMT +08:00	2019/Apr/09 21:38:13 GMT +08:00	2019/Apr/09 21:39:01 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:52:00 GMT +08:00	2019/Apr/09 21:37:43 GMT +08:00	2019/Apr/09 21:38:31 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:50:00 GMT +08:00	2019/Apr/09 21:37:37 GMT +08:00	2019/Apr/09 21:38:10 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:48:00 GMT +08:00	2019/Apr/09 21:37:04 GMT +08:00	2019/Apr/09 21:37:40 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:46:00 GMT +08:00	2019/Apr/09 21:37:04 GMT +08:00	2019/Apr/09 21:37:35 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:44:00 GMT +08:00	2019/Apr/09 21:36:19 GMT +08:00	2019/Apr/09 21:37:01 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:42:00 GMT +08:00	2019/Apr/09 21:36:19 GMT +08:00	2019/Apr/09 21:37:01 GMT +08:00	Stop Rerun View Wai... More
^ job_5433	Run successfully	2019/Apr/08 23:40:00 GMT +08:00	2019/Apr/09 21:35:19 GMT +08:00	2019/Apr/09 21:36:16 GMT +08:00	Stop Rerun View Wai... More

## 7.5 Notification Management

### 7.5.1 Managing a Notification

You can configure DLF to notify you of job success after it is performed.

#### Configuring a Notification

Before configuring a notification, ensure that Simple Message Notification (SMN) has been enabled, a topic has been configured in SMN, and a job has been submitted and is not in the **Not Started** state.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Manage Notification**.

**Step 3** On the **Notification Management** tab page, click **Configure Notification**. In the displayed dialog box, configure parameters. [Table 7-9](#) describes the parameters.

**Table 7-9** Notification parameters

Parameter	Mandatory	Description
Job Name	Yes	Name of the job.
Notification Mode	Yes	Select a notification mode: <ul style="list-style-type: none"><li>• Voice</li><li>• Topic Message</li></ul>
Notification Type	Yes	Type of the notification. <ul style="list-style-type: none"><li>• <b>Run abnormally/Fail:</b> When a job cannot run normally or fail to run, a notification is sent to notify the user of the abnormality.</li><li>• <b>Run successfully:</b> When a job runs successfully, a notification is sent to notify the user of the success.</li><li>• <b>Uncompleted:</b> This function supports only the jobs scheduled by day. If the job execution time is later than the configured time by which the job has not finished, a notification is sent.</li></ul> <b>NOTE</b> For a real-time job, a notification is allowed to be sent only when the real-time job is in the <b>Run abnormally</b> or <b>Failed</b> state. For a batch job, a notification can be sent no matter when the batch job is in the <b>Run normally</b> , <b>Run abnormally</b> , or <b>Failed</b> state.
Topic for Abnormal/Failed Running	Yes	Select a topic for abnormal job running or job failure.
Topic for Successful Running	Yes	Select a topic for successful job running.
Notification	Yes	Specifies whether to enable the notification function. The function is enabled by default.

**Step 4** Click **OK**.

 **NOTE**

DLF sends notifications through the Simple Message Notification (SMN). Using SMN may incur fees. For price details, contact the SMN support personnel.

----End



## Editing a Notification

After a notification is created, you can modify the notification parameters as required.

- Step 1** In the navigation tree of the Data Development console, choose **Monitoring > Manage Notification**.
- Step 2** Click the **Notification Management** tab.
- Step 3** In the **Operation** column of a notification, click **Edit**. In the displayed dialog box, edit notification parameters. [Table 7-9](#) describes the notification parameters.
- Step 4** Click **Yes**.
- End

## Disabling the Notification Function

You can disable the notification function on the **Edit Notification** page or in the notification list.

- Step 1** In the navigation tree of the Data Development console, choose **Monitoring > Manage Notification**.
- Step 2** Click the **Notification Management** tab.
- Step 3** In the **Notification Function** column, click . When it changes to , the notification function is disabled.
- End

## Viewing a Notification

You can view all notification information on the **Notification Records** tab page.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Manage Notification**.
- Step 3** Click the **Notification Records** tab.
- End

## Deleting a Notification

If you do not need to use a notification any more, perform the following operations to delete it:

- Step 1** In the navigation tree of the Data Development console, choose **Monitoring > Manage Notification**.
- Step 2** Click the **Notification Management** tab.
- Step 3** In the **Operation** column of the notification, click **Delete**. The **Delete Notification** dialog box is displayed.
- Step 4** Click **OK**.
- End

## 7.5.2 Cycle Overview

### Scenarios

Notifications can be set to specified personnel by day, week, or month, allowing related personnel to regularly understand job scheduling information about the quantity of successfully/unsuccessfully scheduled jobs and failure details.

### Prerequisites

- Simple Message Notification (SMN) has been enabled, topics have been configured, and subscriptions have been added to the topics.
- Jobs are not in **Not started** status and have been submitted.
- OBS has been enabled and a folder has been created in OBS.

### Creating a Notification

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Monitoring > Manage Notification**.
- Step 3** On the **Cycles** tab page, click **Create Notification**. In the displayed dialog box, configure parameters. [Table 7-10](#) describes the notification parameters.

**Figure 7-15** Create a notification

**Configure Notification**

\* Notification Name: notification

\* Schedule Cycle: Every week

\* Select Time: Monday x Tuesday x

\* Start Time: 00 h 59 min

\* Topic: doc [View Topic](#)

SMN required for the topic may incur charges. For details, click [View Pricing Rule](#)

\* Select OBS Bucket: s3a://dlf-test/data/ OBS

\* Notification: ☒

OK Cancel

**Table 7-10** Notification parameters

Parameter	Mandatory	Description
Notification Name	Yes	Name of the notification to be sent.
Cycle	Yes	Interval for sending notifications, which can be set to <b>Daily</b> , <b>Weekly</b> , or <b>Monthly</b> . <b>NOTE</b> When <b>Cycle</b> is set to <b>Daily</b> , <b>Weekly</b> , or <b>Monthly</b> , a notification is sent every day, week, or month, and the notification content comes from the data generated from the last 24 hours, seven days, or 30 days.
Select Time	Yes	Time when the notification is sent. <ul style="list-style-type: none"><li>• If <b>Cycle</b> is set to <b>Weekly</b>, the value can be any day or any several days from Monday to Sunday in a week.</li><li>• If <b>Cycle</b> is set to <b>Monthly</b>, the value can be any day or any several days from 1st to 31st in a month.</li></ul>
Start Time	Yes	Point in time when the notification is sent. The value can be accurate to hour or minute.
Topic	Yes	Select a notification topic from the drop-down list box.
OBS Bucket	Yes	Enter an OBS bucket in the text box or click <b>OBS</b> and select one from the displayed dialog box.
Notification	Yes	Specifies whether to enable the notification function. The function is enabled by default.

**Step 4** Click **OK**. **NOTE**

DLF sends notifications through SMN. Using SMN may incur fees. For price details, contact the SMN support personnel.

**Step 5** After the notification is created, you can perform the following operations on the notification:

- Click **Edit**. In the **Create Notification** dialog box, edit the notification again.
- Click **View Record**. In the **View Record** dialog box, view the job scheduling details.
- Click **Delete**. In the **Delete Notification** dialog box, click **OK** to delete the notification.

----End

## 7.6 Backing Up and Restoring Assets

You can back up all jobs, scripts, resources, and environment variables on a daily basis.

You can also restore assets that have been backed up, including jobs, scripts, resources, and environment variables.

### Prerequisites

OBS has been enabled and a bucket has been created in OBS.

### Backing Up Assets

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Manage Backup**.

**Step 3** Click **Start Daily Backup**. In the **Browse OBS File** dialog box, select an OBS folder.

**Figure 7-16** Managing backup

Manage Backup			
<div>Start Daily Backup Stop Daily Backup ?</div>			
Date	Path	Progress	Status
2019-05-13	s3a://001zmwulanchabu3/22...	100%	finished
2019-05-12	s3a://aaaaa1111	100%	finished
2019-05-11	s3a://aaaaa1111	100%	finished
2019-05-10	s3a://aaaaa1111	100%	finished
2019-05-09	s3a://aaaaa1111	100%	finished

 **NOTE**

- Daily Backup starts at 00:00 every day to back up all jobs, scripts, resources, and environment variables of the previous day. The jobs, scripts, resources, and environment variables of the previous day are not backed up on the current day.
- If you select only the bucket name as the OBS storage path, the backup object is automatically stored in the folder named after the backup date. Environment variables, resources, scripts, and jobs are stored in the **1\_env**, **2\_resources**, **3\_scripts**, and **4\_jobs** folders, respectively.
- After the backup is successful, the **backup.json** file is automatically generated in the folder named after the backup date. The file stores job information based on the node type and can be modified before job restoration.
- To stop daily backup, click **Stop Daily Backup**.

----End

## Restoring Assets

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Manage Backup**.

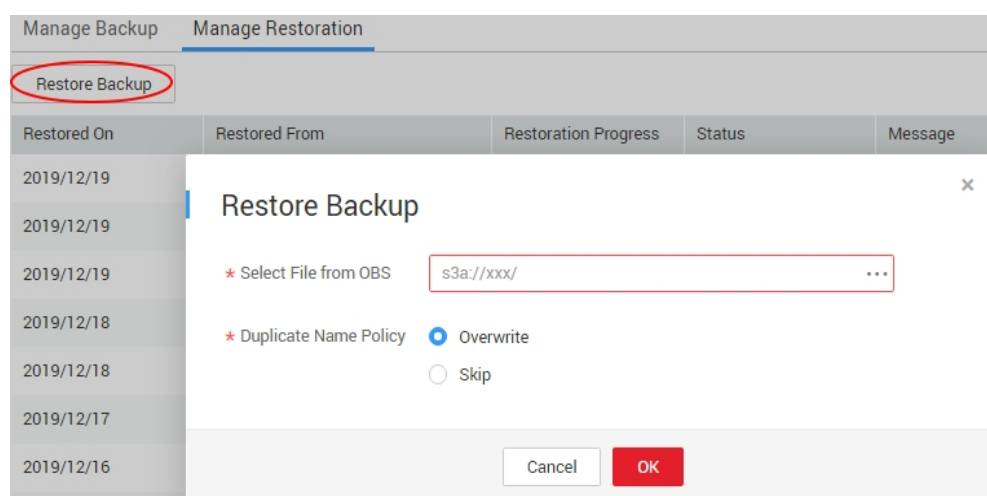
**Step 3** On the **Manage Restoration** tab, click **Restore Backup**.

In the **Restore Backup** dialog box, select the storage path of the asset to be restored from the OBS bucket and set the duplicate name policy.

 **NOTE**

- The storage path is the file path generated in [Backing Up Assets](#).
- Before restoring assets, you can modify the **backup.json** file in the backup path. You can change the connection name (connectionName), database name (database), and cluster name (clusterName).

**Figure 7-17** Restoring assets



**Step 4** Click **OK**.

----End

# 8 Configuration and Management

## 8.1 Managing Host Connections

DLF can save the host connection information required for scripts and jobs. If the host connection information changes, you only need to edit it on the **Host Connection Management** page, but do not need to edit it in scripts or jobs one by one.

### Creating a Host Connection

Ensure that the quantity of host connections is less than the maximum connection quota (20).

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Host Connection**.
- Step 3** Click **Create Host Connection**. In the displayed dialog box, set the host connection parameters.

**Table 8-1** Host connection parameters

Parameter	Mandatory	Description
Host Connection Name	Yes	Name of the host connection. Must consist of 1 to 100 characters, contain only letters, digits, and underscores (_), and be unique.
Host IP Address	Yes	IP address of the host. For details, see "Viewing Details About an ECS" in the <i>Elastic Cloud Server User Guide</i> .
Port	Yes	Port number of the host.
Username	Yes	Username of the host.

Parameter	Mandatory	Description
Login Mode	Yes	Selects the login mode of the host. <ul style="list-style-type: none"><li>• Key pair</li><li>• Password</li></ul>
Key Pair	Yes	If Key Pair is the login mode of the host, the user needs to obtain the private key file, upload it to OBS, and select an OBS path. <b>NOTE</b> The uploaded private key file must be in PEM format, and the uploaded private key file and the public key configured on the host must be in the same key pair.
Key Pair Password	No	This parameter does not need to be configured if the key pair does not have password.
Password	Yes	If the login mode of the host is to use a password, enter a login password.
CDM Cluster	Yes	Name of the Cloud Data Migration (CDM) cluster.
KMS Key	Yes	Key created on Key Management Service (KMS) and used for encrypting and decrypting user passwords and key pairs. You can select a created key from KMS.
Host Connection Description	No	Descriptive information about the host connection.

**Step 4** Click **Test** to test connectivity to the host. If the host can be connected, the host connection has been successfully created.

**Step 5** Click **OK**.

----End

## Editing a Host Connection

After a host connection is created, you can modify host connection parameters as required.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Host Connection**.

**Step 3** In the **Operation** column of the host connection, click **Edit**. In the displayed dialog box, modify the host connection parameters as required.

 **NOTE**

For security purposes, you need to enter the password of the host again when modifying the host connection parameters.

**Step 4** Click **Test** to test connectivity to the host. If the host can be connected, the host connection has been successfully created.

**Step 5** Click **OK**.

----End

## Deleting a Host Connection

If you do not need to use a host connection any more, perform the following operations to delete the host connection:

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Host Connection**.

**Step 3** In the **Operation** column of the host connection, click **Delete**. The **Delete Host Connection** dialog box is displayed.

**Step 4** Click **OK**.

---

**NOTICE**

If you forcibly delete a host connection that is being associated with a script or job, ensure that services are not affected by going to the script or job development page and reassociating an available host connection with the script or job.

---

----End

## 8.2 Managing Resources

You can upload custom code or text files as resources on Manage Resource and schedule them when running nodes. Nodes that can invoke resources include DLI Spark, MRS Spark, MRS MapReduce, and DLI Flink Job.


After creating a resource, configure the file associated with the resource. Resources can be directly referenced in jobs. When the resource file is changed, you only need to change the resource reference location. You do not need to modify the job configuration. For details about resource usage examples, see [Developing a Spark Job](#).

### (Optional) Creating a Directory

If a directory exists, you do not need to create one.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Resource**.

- Step 3** In the directory list, click . In the displayed dialog box, configure directory parameters. [Table 8-2](#) describes the directory parameters.

**Table 8-2** Resource directory parameters

Parameter	Description
Directory Name	Name of the resource directory. Must consist of 1 to 32 characters and contain only letters, digits, underscores (_), and hyphens (-).
Select Directory	Parent directory of the resource directory. The parent directory is the root directory by default.

- Step 4** Click **OK**.

----End

## Creating a Resource

You have enabled OBS before creating a resource.

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Resource**.
- Step 3** Click **Create Resource**. In the displayed dialog box, configure resource parameters. [Table 8-3](#) describes the resource parameters.

**Table 8-3** Resource management parameters

Parameter	Mandator y	Description
Name	Yes	Name of the resource. Must consist of 1 to 32 characters and contain only letters, digits, underscores (_), and hyphens (-).
Type	Yes	File type of the resource. Possible values: <ul style="list-style-type: none"><li>• jar</li><li>• file</li><li>• archive</li></ul>
Resource Location	Yes	Location of the resource. Possible values: <ul style="list-style-type: none"><li>• Local</li><li>• OBS</li></ul>
Main JAR package	Yes	Main JAR package that has been uploaded to OBS. This parameter is required when <b>Type</b> is set to <b>jar</b> and <b>Resource Location</b> is set to <b>OBS</b> .

Parameter	Mandatory	Description
Depended JAR Package	No	Depended JAR package that has been uploaded to OBS. This parameter is required when <b>Type</b> is set to <b>jar</b> and <b>Resource Location</b> is set to <b>OBS</b> .
Select Resource	Yes	Specific resource file.
Storage Path	Yes	Path to a directory where the resource is stored. This parameter is required only when <b>Resource Location</b> is set to <b>Local</b> .
Description	No	Descriptive information about the resource.
Select Directory	Yes	Directory to which the resource belongs. The root directory is selected by default.

**Step 4** Click **Yes**.

----End

## Editing a Resource

After a resource is created, you can modify resource parameters.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Resource**.

**Step 3** In the **Operation** column of the resource, click **Edit**. In the displayed dialog box, modify the resource parameters as required. For details, see [Table 8-3](#).

**Step 4** Click **OK**.

----End

## Deleting a Resource

If you do not need to use a resource any more, perform the following operations to delete it. Before deleting a resource, ensure that it is not used by any jobs.

**Step 1** Log in to the DLF console.

**Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Resource**.


**Step 3** In the **Operation** column of the resource, click **Delete**. The **Delete Resource** dialog box is displayed.

**Step 4** Click **Yes**.

----End

## Importing a Resource

To import a resource, perform the following operations:

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the Data Development console, choose **Configuration > Manage Resource**.
- Step 3** In the resource directory, click  and select **Import Resource**. The **Import Resource** dialog box is displayed.
- Step 4** Import a resource file uploaded to OBS, click **Next** and then **Close**.

----End

## Exporting a Resource

To export a resource, perform the following operations:

- Step 1** In the navigation tree of the Data Development console, choose **Configuration > Manage Resource**.
- Step 2** In the **Operation** column of the resource, click **Export**. The **Export Resource** dialog box is displayed.
- Step 3** Select **Export resource definitions only** or **Export resource definitions and files**, and click **OK**.

----End

# 9 Specifications

---

You can set environment variables, workspaces, and log storage path on the **Specifications** page. The workspace area contains project-level variables and the environment variable area contains workspace-level variables. Environment variables configured in a workspace can be used by jobs and scripts in the workspace.

## 9.1 Workspace

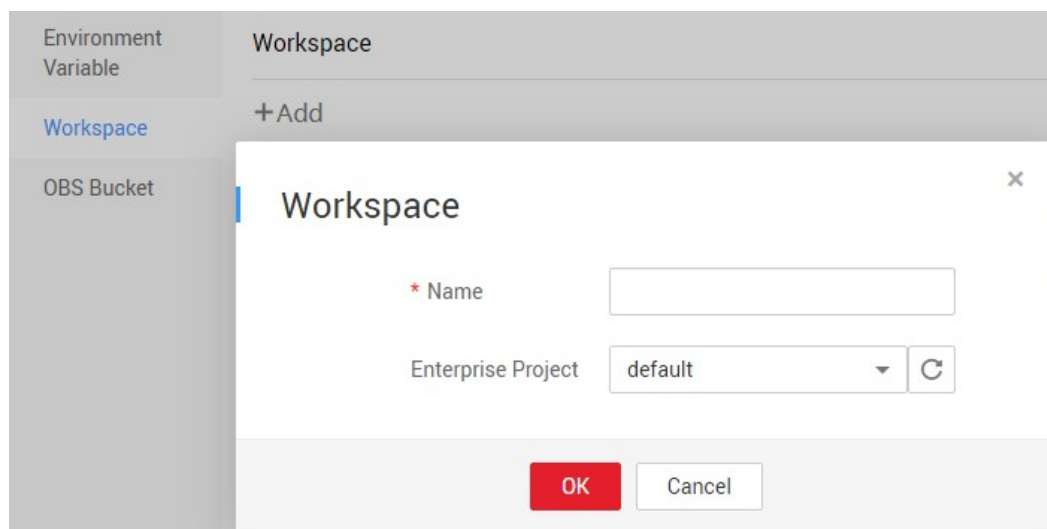
You can customize a workspace to isolate resources.

### Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the DLF console, choose **Configure Specifications**.
- Step 3** Click **Workspace**. On the **Workspace** configuration page, click **Add** and enter the workspace name, and configure the enterprise project to which the workspace belongs.

#### NOTE

- You can configure this parameter only when the Enterprise Project Management service is enabled. The default value is **default**.
- An enterprise project facilitates project-level management and grouping of cloud resources and users.
- You can select the default enterprise project (**default**) or other existing enterprise projects. To create an enterprise project, log in to the Enterprise Management console. For details, see the *Enterprise Management User Guide*.

**Figure 9-1** Creating a workspace

**Step 4** Click **OK**.

After creating a workspace, you can add, edit, or delete it.

- **Add:** Click **Add** to create a new workspace.
- **Edit:** Click **Edit** to change the workspace name.
- **Delete:** Click **Delete** to delete the workspace.

----End

## Using a Workflow

In the **Workspace** drop-down list on the upper menu of the service page, view all created workspaces.

- Select a workspace and create environment variables in the workspace. The created environment variables can be used by jobs and scripts in the workspace. For details about how to use environment variables, see [Environment Variables](#).
- Jobs and scripts can be created in the workspace. For details, see [Script Development](#) and [Job Development](#).
- Log storage paths can be configured in the workspace. For details, see [Configuring a Log Storage Path](#).

## 9.2 Managing Enterprise Projects

An enterprise project is a cloud resource management mode. Enterprise Management provides users with comprehensive management of cloud-based resources, personnel, permissions, and finances. Common management consoles are oriented to the control and configuration of individual cloud products. The Enterprise Management console, in contrast, is more focused on resource management. It is designed to help enterprises manage cloud-based resources, personnel, permissions, and finances, in a hierarchical management manner, such as management of companies, departments, and projects.

Users who have enabled Enterprise Project Management can use it to manage cloud service resources on .

## Binding an Enterprise Project

When creating a workspace, you can select an enterprise project for the workspace to associate the DLF workspace with the enterprise project. For details, see [Workspace](#). The **Enterprise Project** drop-down list displays the projects you created. The system also has an internally built enterprise project **default**. If you do not select an enterprise project for the workspace, the default project is used.

During workspace creation, if the workspace is successfully bound to an enterprise project, the workspace has been created. If the binding fails, the system sends an alarm and the workspace fails to be created.

When the DLF workspace is deleted, the association between the DLF workspace and the enterprise project is automatically deleted.

## Viewing Enterprise Projects

After a workspace is created, you can view the enterprise projects associated with the workspace on the workspace configuration page. You can query only the workspace resources of the project on which you have the access permission.

In the list on the workspace configuration page, view the enterprise project to which the workspace belongs.

**Figure 9-2** Viewing the enterprise project

Environment Variable	Workspace			
Workspace	+ Add			
OBS Bucket	Name	ID	Enterprise Project	Operation
	workspace	5FDBA9E72716423AACD49CA0FD30F5D9	default	<a href="#">Edit</a> <a href="#">Delete</a>
	default	62099355b894428e8916573ae635f1f9	default	<a href="#">Edit</a> <a href="#">Delete</a>

When querying the resource list of a specified project on the Enterprise Management console, you can also query the DLF resources.

## Migrating a Workspace to or Out of an Enterprise Project

One DLF workspace can be associated with only one enterprise project. After a workspace is created, you can migrate it from its current enterprise project to another one on the Enterprise Management console, or migrate a DLF workspace from another enterprise project to a specified enterprise project. After the migration, the workspace is associated with the new enterprise project. The association between the workspace and the original enterprise project is automatically released. For details, see **Enterprise Project Management > Managing Resources** in the *Enterprise Management User Guide*.

## 9.3 Environment Variables

This topic describes how to configure and use environment variables.

## Procedure

- Step 1** Log in to the DLF console.
- Step 2** In the navigation tree of the DLF console, choose **Configure Specifications**.
- Step 3** On the **Environment Variable** page, set the parameters described in [Table 9-1](#) and click **Save**.

**Figure 9-3** Configuring environment variables



Configure Environment Variable



**Table 9-1** Configuring environment variables

Parameter	Mandatory	Description
Parameter	Yes	The parameter name must be unique, consist of 1 to 64 characters, and contain only letters, digits, underscores (_), and hyphens (-).
Value	Yes	Parameter values support constants and EL expressions but do not support system functions. For example, <b>123</b> and <b>abc</b> are supported. For details about how to use EL expressions, see <a href="#">Expression Overview</a> .

After configuring an environment variable, you can add, edit, or delete it.

- **Add:** Click **Add** to add an environment variable.
- **Edit:** If the parameter value is a constant, change the parameter value in the text box. If the parameter value is an EL expression, click  next to the text box to edit the EL expression. Click **Save**.
- **Delete:** Click  next to the parameter value text box to delete the environment variable.

----End

## How-Tos

The configured environment variables can be used in either of the following ways:

1. `${Environment variable}`
2. `#{Evn.get("environment variable")}`

## Example

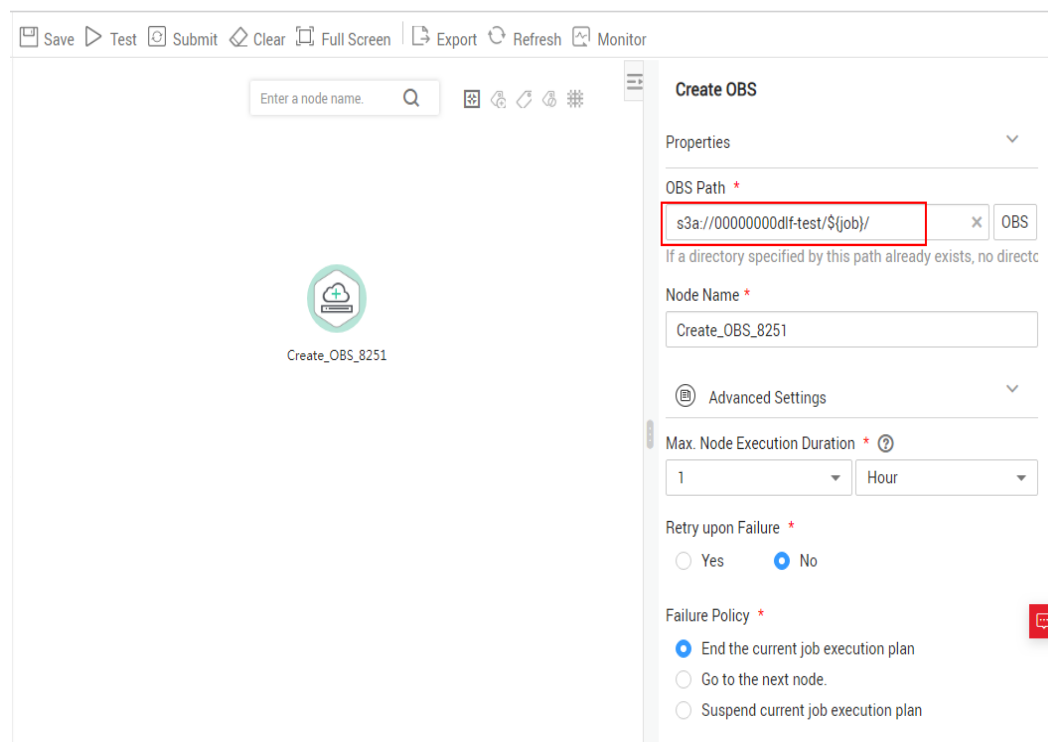
Context:

- A job named **test** has been created on DLF.
- An environment variable has been added. The parameter name is **job** and the parameter value is **123**.

**Step 1** Open **test** and drag a **Create OBS** node from the node library.

**Step 2** On the **Node Properties** tab page, configure the node properties.

**Figure 9-4** Create OBS



**Step 3** Click **Save** and then **Monitor** to monitor the running status of the job.

-----End

## 9.4 Configuring a Log Storage Path

By default, job logs and DLI dirty data are stored in an OBS bucket named **dlf-log-*{project ID}***. You can customize a log storage path. DLF allows you to configure an OBS bucket globally based on the workspace.

**Step 1** Log in to the DLF console.

**Step 2** Select a workspace at the top of the console.

- Step 3** In the navigation tree of the DLF console, choose **Specifications**.
- Step 4** Click **OBS Bucket**. The OBS bucket configuration page is displayed.
- Step 5** Click **OBS** next to **Bucket for Job Logs** and **Bucket for DLI Dirty Data**, select an OBS bucket name, and confirm the selection.

----End

## 9.5 Configuring Agencies

The following problems may occur during job execution in Data Development:

- The job execution mechanism of the Data Development module is to execute the job as the user who starts the job. For a job that is executed in periodic scheduling mode, if the IAM account used to start the job is deleted during the scheduling period, the system cannot obtain the user identity authentication information. As a result, the job fails to be executed.
- If a job is started by a low-privilege user, the job fails to be executed due to insufficient permissions.

To solve the preceding problems, configure an agency. When an agency is configured, the job interacts with other services as an agency during job execution to prevent job execution failures in the preceding scenarios.

### Role of an Agency

Cloud services interwork with each other, and some cloud services are dependent on other services. You can create an agency to delegate a cloud service to access other services and perform resource O&M on your behalf.

### Agency Classification

Agencies are classified into workspace-level agencies and job-level agencies.

- Workspace-level agencies can be globally applied to all jobs in the workspace.
- Job-level agencies can only be applied to a single job.

The job-level agency has a higher priority than the workspace-level agency. If neither of them is configured, execute the job as the user who starts the job.

### Constraints

- To create or modify an agency, you must have the **Security Administrator** permissions.
- To configure a workspace-level agency, you must have the **DAYUDGC Administrator** or **Tenant Administrator** policy.
- All users who have the permissions to view the list of agencies can configure a job-level agency.

### Creating an Agency

1. Log in to the IAM console.

2. Choose **Agencies**. On the displayed page, click **Create Agency**.
3. Enter an agency name, Example: DAYUDGC \_agency.
4. Set **Agency Type** to **Cloud service**. Select **DAYUDGC** for **Cloud Service**. DAYUDGC then can perform resource O&M operations on behalf of you. See [Figure 9-5](#).
5. Set **Validity Period** to **Unlimited**.

**Figure 9-5** Creating an agency

\* Agency Name

\* Agency Type ☐ Account ☒ Cloud service

\* Cloud Service DAYU

\* Validity Period

Description

0/255

6. Click **Assign Permissions** in the **Permissions** area.
7. On the displayed page, search for the **Tenant Administrator** policy, select it, and click **OK**. See [Figure 9-6](#).
  - Users assigned the **Tenant Administrator** policy have all permissions on all services except on IAM. Therefore, delegate the **Tenant Administrator** policy to DAYUDGC so that DAYUDGC can access all related services.
  - If you want to meet the security control requirements for fewer permissions, you only need to configure the **OBS OperateAccess** permissions (During job execution, execution log information needs to be written to OBS. Therefore, you need to add the **OBS OperateAccess** permissions.) . Then, configure different agency permissions based on the node type in the job. For example, if a job contains only the **Import GES** node, you can configure the **GES Administrator** and **OBS OperateAccess** permissions. For details, see [Permission Assignment](#).

**Figure 9-6** Assigning permissions

Assign Permissions

Multiple policies can be selected. You can also [modify](#) or [create](#) policies.

View Selected (1)

<input checked="" type="checkbox"/> Policy/Role Name	Description	Project (Region)
<input checked="" type="checkbox"/> Tenant Administrator	Tenant Administrator (Exclude IAM)	<input type="text" value="All project..."/>

8. Click **OK**.

## Permission Assignment

After the operation permissions of an account are delegated to DAYUDGC, you need to configure the permissions of the agency identity so that DAYUDGC can interact with other services.

To meet the security control requirements for fewer permissions, you can configure the **Admin** permissions for services based on the node types in jobs. For details, see [Table 9-2](#).

The **Admin** permissions can also be configured based on the operations, resources, and request conditions for a specific service. Based on the node types in jobs, permissions are defined by service APIs to allow for more fine-grained, secure access control of cloud resources. Configure the permissions according to [Table 9-3](#). For example, for a job containing the **Import GES** node, you only need to create a custom policy and select **ges:graph:getDetail** (viewing graph details), **ges:jobs:getDetail** (querying task status), and **ges:graph:access** (using graphs).

#### NOTICE

- MRS-related nodes (MRS Presto SQL, MRS Spark, MRS Spark Python, MRS Flink Job, and MRS MapReduce) and directly connected nodes (MRS Spark SQL and MRS Hive SQL) do not support job submission in agency mode, therefore, jobs of these types cannot be configured with agencies.
  - MRS clusters that support job submission in agency mode are as follows:
    - Non-security cluster
    - Security cluster whose version is later than 2.1.0, and that has MRS 2.1.0.1 or later installed
- 
- Configure the service-level **Admin** permissions.  
During job execution, execution log information needs to be written to OBS. Therefore, the **OBS OperateAccess** permissions must be added for all jobs during coarse-grained authorization.

**Table 9-2** The **admin** permissions for related nodes

Node Name	System Permission	Description
CDM Job, DIS Stream, DIS Dump, and DIS Client	DAYUDGC Administrator	All DAYUDDataLake Governance Center permissions
Import GES	GES Administrator	Permissions required to perform all operations on GES. This role depends on the <b>Tenant Guest</b> and <b>Server Administrator</b> roles in the same project.

Node Name	System Permission	Description
<ul style="list-style-type: none"><li>MRS Presto SQL, MRS Spark, MRS Spark Python, MRS Flink Job, and MRS MapReduce</li><li>MRS Spark SQL and MRS Hive SQL (connecting to MRS clusters through MRS APIs)</li></ul>	MRS Administrator KMS Administrator	Users assigned the <b>MRS Administrator</b> role can perform all operations on MRS. This role depends on the <b>Tenant Guest</b> and <b>Server Administrator</b> roles in the same project.  Users assigned the <b>KMS Administrator</b> role have the administrator permissions for encryption keys in DEW.
MRS Spark SQL, MRS Hive SQL, MRS Kafka, and Kafka Client (connecting to the clusters in proxy mode)	DAYUDGC Administrator KMS Administrator	DAYUDGC Administrator has all permissions required for DAYUDGC.  Users assigned the <b>KMS Administrator</b> role have the administrator permissions for encryption keys in DEW.
DLI Flink Job, DLI SQL, and DLI Spark	DLI Service Admin	All operation permissions for DLI.
DWS SQL, Shell, and RDS SQL (connecting to data sources in proxy mode)	DAYUDGC Administrator KMS Administrator	DAYUDGC Administrator has all permissions required for DAYUDGC.  Users assigned the <b>KMS Administrator</b> role have the administrator permissions for encryption keys in DEW.
CSS	DAYUDGC Administrator Elasticsearch Administrator	DAYUDGC Administrator has all permissions required for DAYUDGC.  Users assigned the <b>Elasticsearch Administrator</b> role have all permissions for CSS. This role depends on the <b>Tenant Guest</b> and <b>Server Administrator</b> roles in the same project.
Create OBS, Delete OBS, and OBS Manager	OBS OperateAccess	Basic object operation permissions, such as viewing buckets, uploading objects, obtaining objects, deleting objects, and obtaining object ACLs.
SMN	SMN Administrator	All operation permissions for SMN.

- Configure fine-grained permissions. (Create custom policies based on the actions supported by each service.)

For details on how to create a custom policy, see [Creating Custom Policies](#).

 **NOTE**

- During job execution, you need to write execution logs to OBS. When the fine-grained authorization mode is used, the following OBS permissions need to be added for all types of jobs:
  - obs:bucket:GetBucketLocation
  - obs:object:GetObject
  - obs:bucket>CreateBucket
  - obs:object:PutObject
  - obs:bucket:ListAllMyBuckets
  - obs:bucket:ListBucket
- CDM Job, DIS Stream, DIS Dump and DIS Client nodes belong to the DAYUDGC module. DAYUDGC does not support fine-grained authorization. Therefore, only the **DAYUDGC Administrator** policy can be configured for jobs containing these types of nodes.
- CSS does not support fine-grained authorization and requires a proxy. Therefore, the **DAYUDGC Administrator** and **Elasticsearch Administrator** policies can be configured for jobs containing these nodes.
- SMN does not support fine-grained authorization. Therefore, jobs containing these nodes require the **SMN Administrator** permissions.

**Table 9-3** Creating a custom policy

Node Name	Action
Import GES	<ul style="list-style-type: none"><li>• ges:graph:access</li><li>• ges:graph:getDetail</li><li>• ges:jobs:getDetail</li></ul>
<ul style="list-style-type: none"><li>• MRS Presto SQL, MRS Spark, MRS Spark Python, MRS Flink Job, and MRS MapReduce</li><li>• MRS Spark SQL and MRS Hive SQL (connecting to MRS clusters through MRS APIs)</li></ul>	<ul style="list-style-type: none"><li>• mrs:job:delete</li><li>• mrs:job:stop</li><li>• mrs:job:submit</li><li>• mrs:cluster:get</li><li>• mrs:cluster:list</li><li>• mrs:job:get</li><li>• mrs:job:list</li><li>• kms:dek:crypto</li><li>• kms:cmk:get</li></ul>
MRS Spark SQL, MRS Hive SQL, MRS Kafka, and Kafka Client (connecting to the clusters in proxy mode)	<ul style="list-style-type: none"><li>• kms:dek:crypto</li><li>• kms:cmk:get</li><li>• DAYUDGC Administrator (role)</li></ul>

Node Name	Action
DLI Flink Job, DLI SQL, and DLI Spark	<ul style="list-style-type: none"><li>• dli:jobs:get</li><li>• dli:jobs:update</li><li>• dli:jobs:create</li><li>• dli:queue:submit_job</li><li>• dli:jobs:list</li><li>• dli:jobs:list_all</li></ul>
DWS SQL, Shell, and RDS SQL (connecting to data sources in proxy mode)	<ul style="list-style-type: none"><li>• kms:dek:crypto</li><li>• kms:cmk:get</li><li>• DAYUDGC Administrator (role)</li></ul>
Create OBS, Delete OBS, and OBS Manager	<ul style="list-style-type: none"><li>• obs:bucket:GetBucketLocation</li><li>• obs:bucket:ListBucketVersions</li><li>• obs:object:GetObject</li><li>• obs:bucket&gt;CreateBucket</li><li>• obs:bucket&gt;DeleteBucket</li><li>• obs:object&gt;DeleteObject</li><li>• obs:object:PutObject</li><li>• obs:bucket:ListAllMyBuckets</li><li>• obs:bucket:ListBucket</li></ul>

## Configuring a Workspace-Level Agency

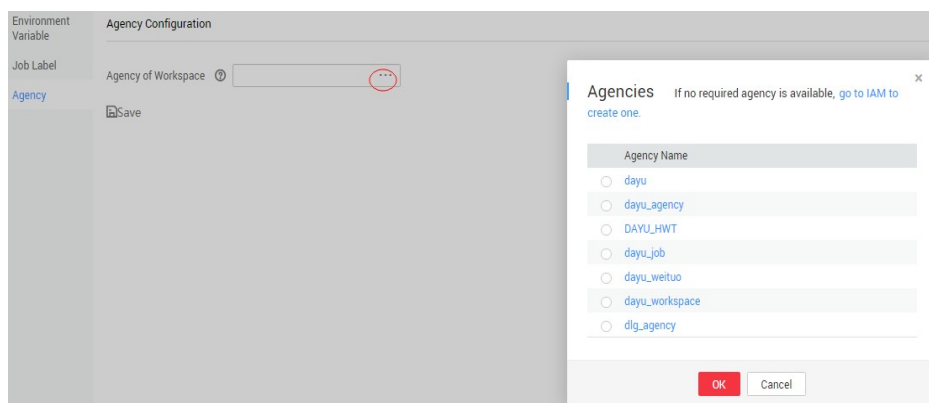
---

### CAUTION

A workspace-level agency impacts on all jobs. Some jobs contain nodes related to MRS. Exercise caution when performing this operation.

---

1. In the navigation tree on the left, choose **Specifications**.
2. Click **Agency**. On the displayed page, configure an agency.
3. You can select an agency from the agency list or create a new one. For details on how to create an agency and configure permissions, see [Creating an Agency](#).

**Figure 9-7** Configuring a workspace-level agency

4. Click **OK** to return to the **Agency Configuration** page. Then, click  to save the settings.

## Configuring a Job-level Agency

### NOTE

You can create a job-level agency when creating a job. You can also modify the agency of an existing job.

### Configuring an agency when creating a job

1. In the navigation pane of the Data Development homepage, choose **Development** > **Develop Job**.
2. Right-click the job directory and choose **Create Job** from the shortcut menu. The **Create Job** dialog box is displayed. If a workspace-level agency has been configured, it is used for the job by default. You can also select another agency from the agency list.

**Figure 9-8** Configuring a Job Agency

### Create Job

A maximum of 10000 jobs can be created. You can create 9989 more jobs.

* Job Name	<input type="text" value="job_2969"/>
* Processing Mode	<input checked="" type="radio"/> Batch processing <input type="radio"/> Real-time processing
* Creation Method	<input checked="" type="button" value="Create Empty Job"/> <input type="button" value="Create Based on Template"/>
* Select Directory	<input type="text" value="/Jobs/"/> ...
Owner	<input type="text" value="Owner"/>
Priority	<input checked="" type="radio"/> High <input type="radio"/> Medium <input type="radio"/> Low
Agency ?	<input type="text" value="dayu_agency"/> x ...
* Log Path	<input type="text" value="obs://dlf-log-62099355b894428e8916573ae635f1f9/"/> <a href="#">To change the log path, go to the DAYU space management page.</a> <a href="#">For details, see the documentation.</a>

### Modifying the agency of an existing job

1. In the navigation pane of the Data Development homepage, choose **Development > Develop Job**.
2. In the job directory, double-click an existing job. On the far right of the displayed page, click **Basic Info**. The dialog box of the job's basic settings is displayed. If a workspace-level agency has been configured, it is used by default. You can also select another agency from the agency list.

# 10 Usage Tutorials

---

## 10.1 Developing a Spark Job

This section introduces how to develop a Spark job on Data Development.

### Scenario Description

In most cases, SQL is used to analyze and process data when using Data Lake Insight (DLI). However, SQL is usually unable to deal with complex processing logic. In this case, Spark jobs can help. This section uses an example to demonstrate how to submit a Spark job on Data Development.

The general submission procedure is as follows:

1. Create a DLI cluster and run a Spark job using physical resources of the DLI cluster.
2. Obtain a demo JAR package of the Spark job and associate with the JAR package on Data Development.
3. Create a Data Development job and submit it using the DLI Spark node.

### Preparations

- Object Storage Service (OBS) has been enabled and a bucket, for example, **s3a://dlfexample**, has been created for storing the JAR package of the Spark job.
- DLI has been enabled, and the Spark cluster **spark\_cluster** has been created for providing physical resources required for the Spark job.

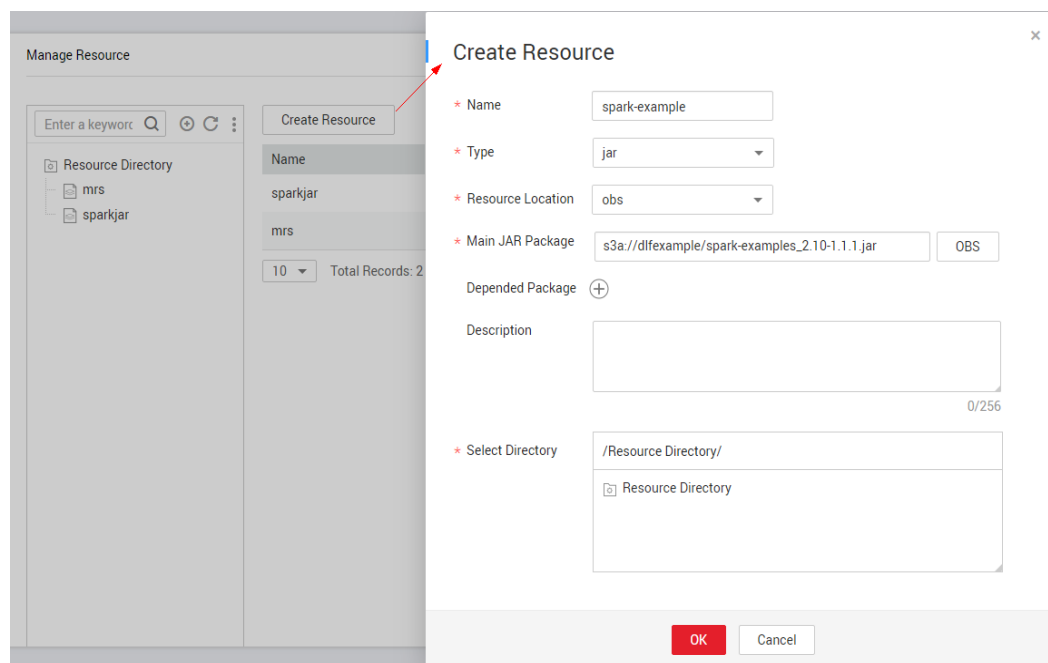
### Obtaining Spark Job Codes

The Spark job code used in this example comes from the maven repository (download address: [Spark job code](#)). Download **spark-examples\_2.10-1.1.1.jar**. The Spark job is used to calculate the approximate value of  $\pi$ .

**Step 1** After obtaining the JAR package of the Spark job codes, upload it to the OBS bucket. The save path is **s3a://dlfexample/spark-examples\_2.10-1.1.1.jar**.

- Step 2** In the navigation tree of the console, choose **Configuration > Manage Resource**. Create resource **spark-example** on Data Development and associate it with the JAR package obtained in [Step 1](#).

**Figure 10-1** Creating a resource



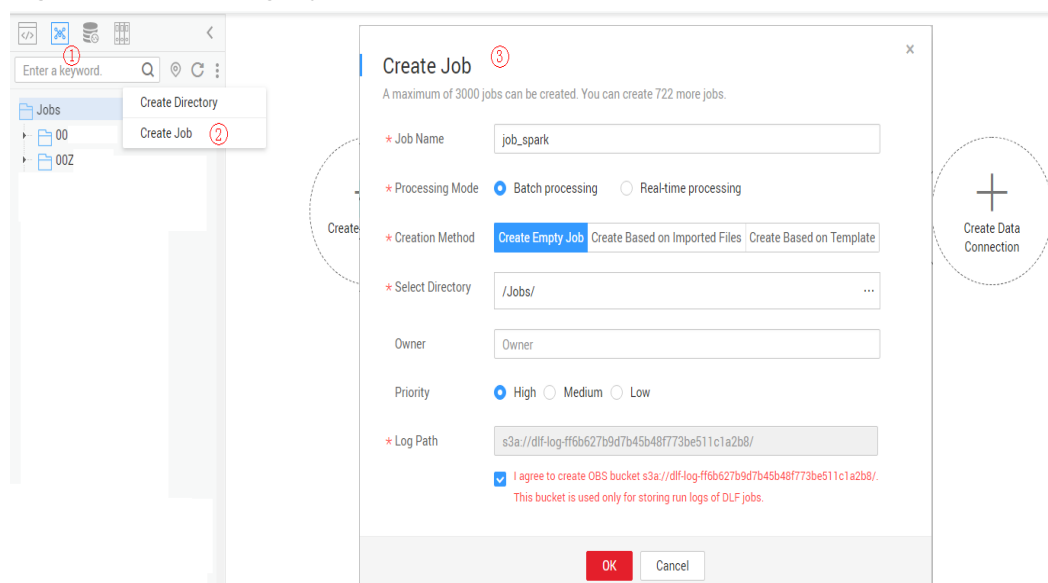
----End

## Submitting a Spark Job

You need to create a job on Data Development and submit the Spark job using the DLI Spark node of the job.

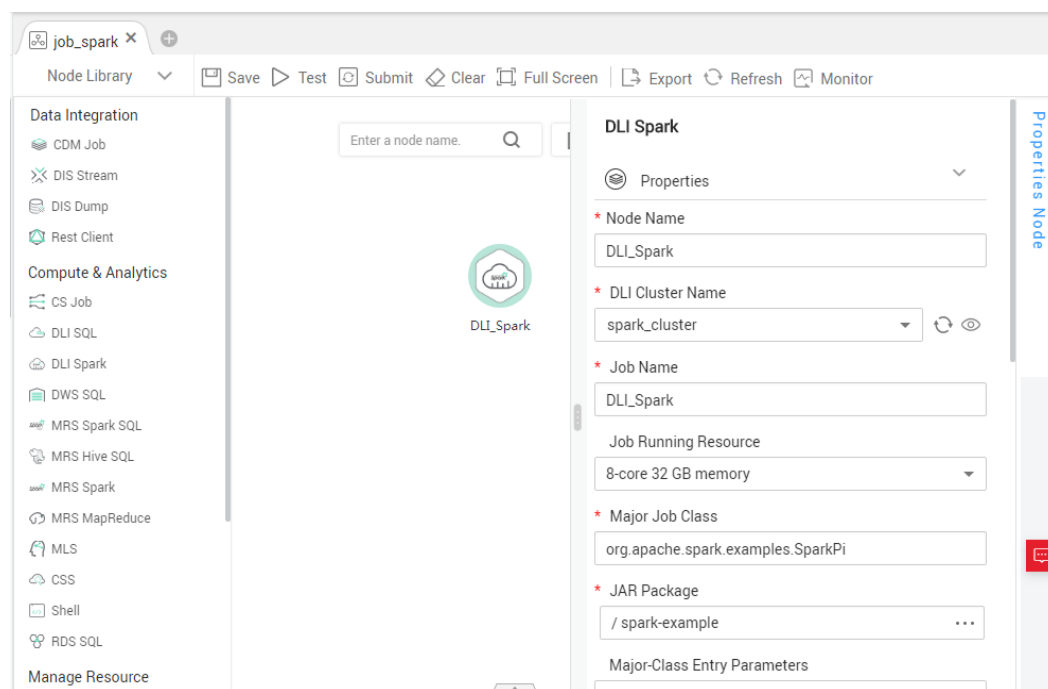
- Step 1** Create an empty DLF job named **job\_spark**.

**Figure 10-2** Creating a job




**Step 2** Go to the job development page, drag the DLI Spark node to the canvas, and click the node to configure node properties.

**Figure 10-3** Configuring node properties

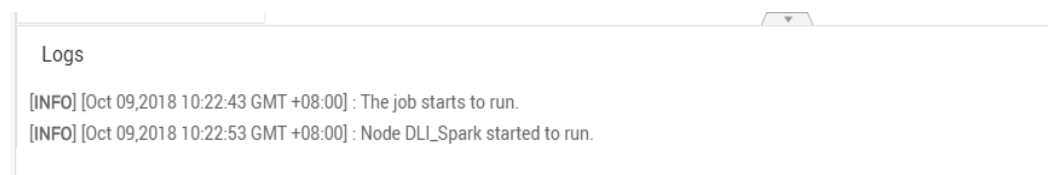


Description of key properties:

- DLI Cluster Name: Name of the Spark cluster created in [Preparations](#).
- Job Running Resource: Maximum CPU and memory resources that can be used when a DLI Spark node is running.
- Major Job Class: Main class of a DLI Spark node. In this example, the main class is **org.apache.spark.examples.SparkPi**.
- JAR Package: Resource created in [Step 2](#).

**Step 3** After the job orchestration is complete, click  to test the job.

**Figure 10-4** Job logs (for reference only)



**Step 4** If no problems are recorded in logs, click  to save the job.

----End

## 10.2 Developing a Hive SQL Script

This section introduces how to develop a Hive SQL script on Data Development.

### Scenario Description

As a one-stop big data development platform, Data Development supports development of multiple big data tools. Hive is a data warehouse tool running on Hadoop. It can map structured data files to a database table and provides a simple SQL search function that converts SQL statements into MapReduce tasks.

### Preparations

- MapReduce Service (MRS) has been enabled and the MRS cluster **MRS\_1009** has been created for providing an operating environment for Hive SQL.  
When creating an MRS cluster, note the following:
  - Kerberos authentication is disabled for the cluster.
  - Hive is available.
- Cloud Data Migration (CDM) has been enabled and the CDM cluster **cdm-7357** has been created for providing a proxy for communication between Data Development and MRS.

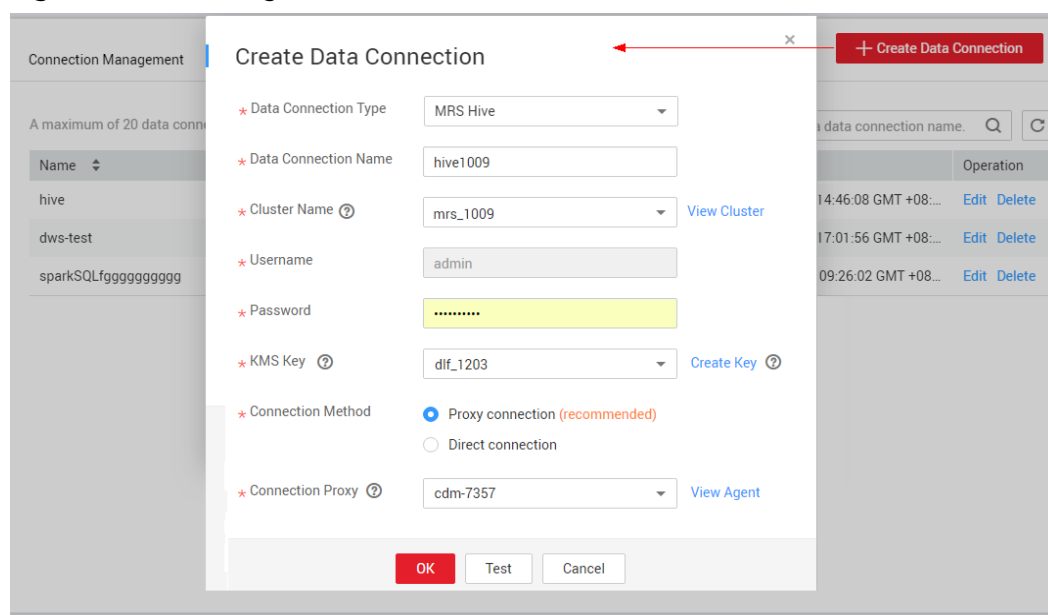
When creating a CDM cluster, note the following:

- The virtual private cloud (VPC), subnet, and security group can communicate with the MRS cluster **MRS\_1009**.

### Creating a Hive Data Connection

Before developing a Hive SQL script, create a connection to MRS Hive on Data Development. In this example, the data connection is named **hive1009**.

**Figure 10-5** Creating a data connection



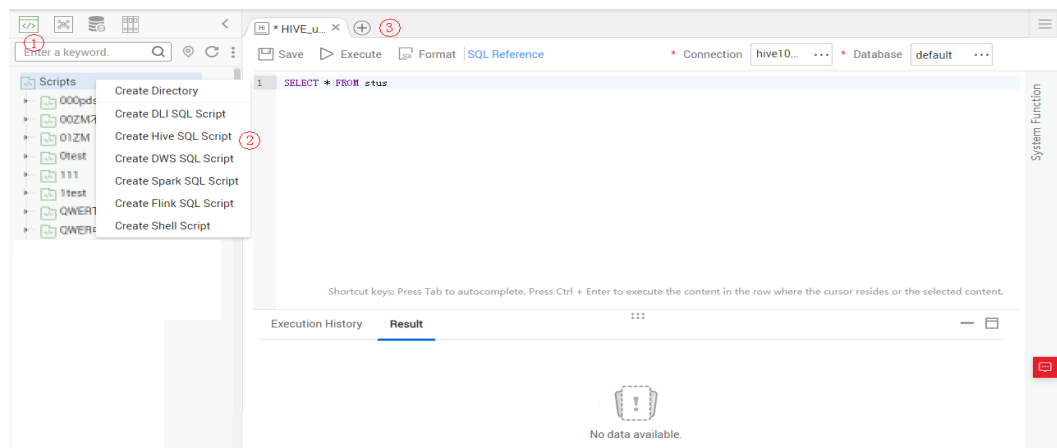
Description of key parameters:

- Cluster Name: Name of the MRS cluster created in [MapReduce Preparations](#).
- CDM Cluster: Name of the CDM cluster created in [CDM Preparations](#).


## Developing a Hive SQL Script

Create a Hive SQL script named **hive\_sql** on Data Development. Then enter SQL statements in the editor to fulfill business requirements.

**Figure 10-6** Developing a script



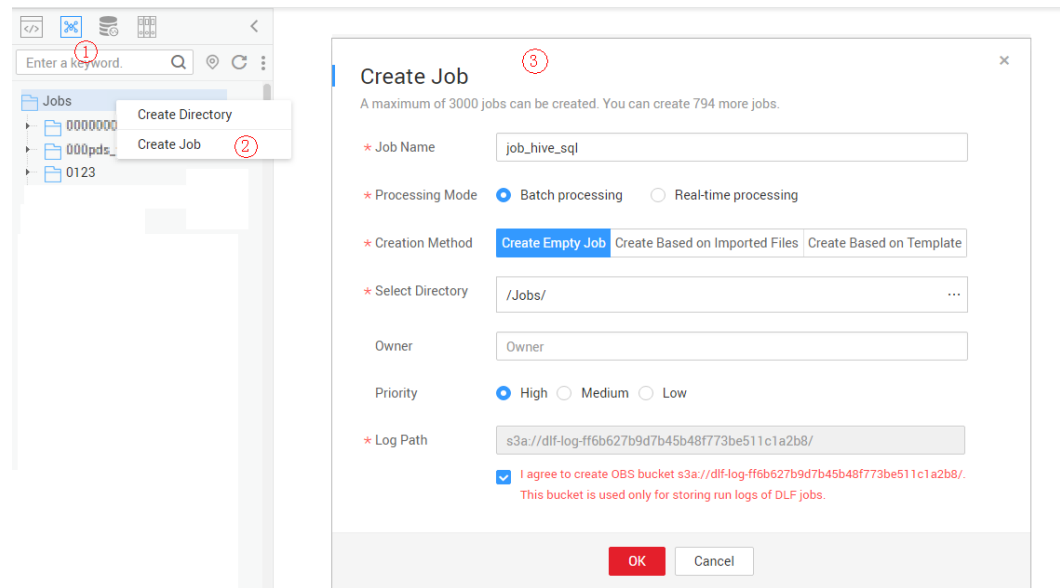
Notes:

- The script development area in [Figure 10-6](#) is a temporary debugging area. After you close the tab page, the development area will be cleared. Click  to save the script to a specified directory.
- Data Connection: Connection created in [Creating a Hive Data Connection](#).

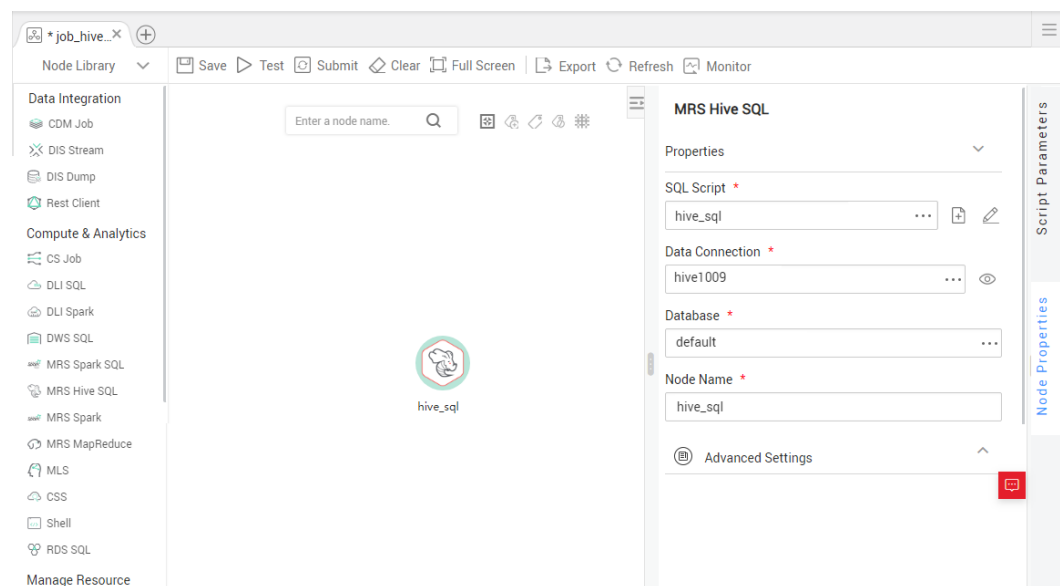
## Developing a Hive SQL Job

After the Hive SQL script is developed, build a periodically deducted job for the Hive SQL script so that the script can be executed periodically.

**Step 1** Create an empty Data Development job named **job\_hive\_sql**.


**Figure 10-7** Creating a job named **job\_hive\_sql**

**Step 2** Go to the job development page, drag the MRS Hive SQL node to the canvas, and click the node to configure node properties.

**Figure 10-8** Configuring properties for an MRS Hive SQL node

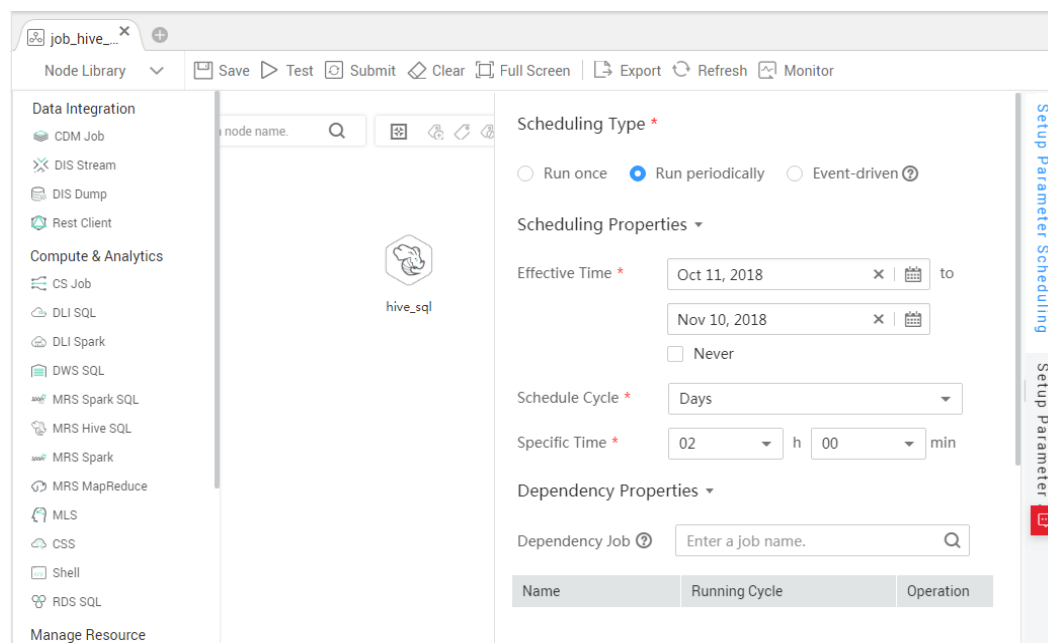
Description of key properties:

- **SQL Script:** Hive SQL script **hive\_sql** that is developed in [Developing a Hive SQL Script](#).
- **Data Connection:** Data connection that is configured in the SQL script **hive\_sql** is selected by default. The value can be changed.
- **Database:** Database that is configured in the SQL script **hive\_sql** and is selected by default. The value can be changed.
- **Node Name:** Name of the SQL script **hive\_sql** by default. The value can be changed.

**Step 3** After the job orchestration is complete, click  to test the job.



**Step 4** If logs do not record any errors, click the blank area on the canvas and configure the job scheduling policy on the scheduling configuration page on the right.

**Figure 10-9** Configuring the scheduling mode



**Note:**

- 2018/10/11 to 2018/11/10: The job is executed at 02:00 a.m. every day.

**Step 5** Click  to save the job and click  to schedule the job to enable the job to run automatically every day.

----End

# 11 References

## 11.1 Nodes

### 11.1.1 Node Overview

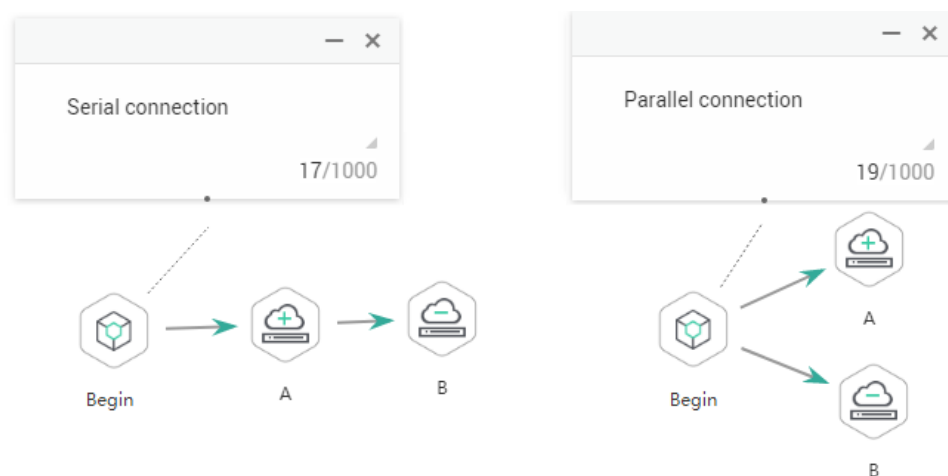
A node defines operations performed on data. Data Development provides types of node used for data integration, computing and analysis, database operations, and resource management. Users can select required nodes based on service models.

- Node parameters can be presented using Expression Language (EL). For details about how to use EL, see [Expression Overview](#).
- Nodes cannot be connected in serial or parallel mode.

Serial connection: Nodes are run one by one. Specifically, run node B only after node A is finished running.

Parallel connection: Nodes are run at the same time.

**Figure 11-1** Connection diagram



## 11.1.2 CDM Job

### Functions

The CDM Job node is used to execute a predefined CDM job for data migration.

### Parameters

[Table 11-1](#), [Table 11-2](#), and [Table 11-3](#) describe the parameters of the CDM Job node. Configure the lineage to identify the data flow direction, which can be viewed in the data asset module.

**Table 11-1** Parameters of CDM Job nodes

Parameter	Mandator y	Description
CDM Cluster Name	Yes	Name of the CDM cluster to which the CDM job to be executed belongs.
CDM Job Name	Yes	Name of the CDM job to be executed.
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).




**Table 11-2** Advanced parameters




Parameter	Mandator y	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

**Table 11-3** Lineage

Parameter	Description
Input	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: name of the CSS index.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
View Details	Click  to view details about the table created based on the input lineage.
<b>Output</b>	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: Enter a CSS index name.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
Table Details	Click  to view details about the table created based on the output lineage.

## 11.1.3 DIS Stream

### Functions


The DIS Stream node is used to query the status of a DIS stream. If the DIS stream is normal, you can perform other nodes. If the DIS stream is abnormal, the DIS

Stream node will send an error message and exit. If you want to perform other nodes, you must set **Failure policy** to **Proceed to the next node**. For details about how to set **Failure policy**, see [Table 11-5](#).

## Parameters

[Table 11-4](#) and [Table 11-5](#) describe the parameters of the DIS Stream node.

**Table 11-4** Parameters of DIS Stream nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Stream Name	Yes	Select or enter the DIS stream to be queried. When entering the stream name, you can reference job parameters and use the EL expression (for details, see <a href="#">Expression Overview</a> ).  To create a DIS stream, you can use either of the following methods: <ul style="list-style-type: none"><li>Click . On the <b>Stream Management</b> page of DLF, create a DIS stream.</li><li>Go to the DIS console to create a DIS stream.</li></ul>

**Table 11-5** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.4 DIS Dump

### Functions



The DIS Dump node is used to configure data dump tasks in DIS.

### Parameters

[Table 11-6](#) and [Table 11-7](#) describe the parameters of the DIS Dump node.

**Table 11-6** Parameters of DIS Dump nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

Parameter	Mandatory	Description
Stream Name	Yes	<p>Select or enter the DIS stream to be queried. When entering the stream name, you can reference job parameters and use the EL expression (for details, see <a href="#">Expression Overview</a>).</p> <p>To create a DIS stream, you can use either of the following methods:</p> <ul style="list-style-type: none"><li>Click . On the <b>Stream Management</b> page of DLF, create a DIS stream.</li><li>Go to the DIS console to create a DIS stream.</li></ul>
Duplicate Name Policy	Yes	<p>Select a duplicate name policy. If the name of a dump task already exists, you can adopt either of the following policies based on site requirements:</p> <ul style="list-style-type: none"><li>Ignore: Give up adding the dump task and exit DIS Dump. The status of DIS Dump is <b>Succeeded</b>.</li><li>Overwrite: Continue to add the dump task by overwriting the one with the same name.</li></ul>
Dump Destination	Yes	<ol style="list-style-type: none"><li>Destination to which data is dumped. Possible values:<ul style="list-style-type: none"><li><b>OBS</b>: After the streaming data is stored to DIS, it is then periodically imported to OBS. After the real-time file data is stored to DIS, it is imported to OBS immediately.</li></ul></li><li>Click . In the dialog box that is displayed, set dump parameters. For details, see <i>Data Ingestion Service User Guide</i>.</li></ol>

**Table 11-7** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.5 DIS Client

### Functions


The DIS Client node is used to send messages to a DIS stream.

### Parameters

[Table 11-8](#) describes the parameters of the DIS Client node.

**Table 11-8** Parameters of DIS Client nodes

Parameter	Mandatory	Description
Using DIS Connection	No	If this function is enabled, messages can be sent to the DIS streams of another account. If this function is not enabled, messages can be sent only to the DIS streams in all regions of the current account.

Parameter	Mandatory	Description
DIS Connection	No	Select a created DIS connection.
Region	No	Region that the target DIS stream belongs to. The DIS Client node is used to send messages to the target DIS stream.
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Stream Name	Yes	DIS stream to which messages will be sent. You can select a DIS stream by entering the DIS stream address or clicking ***.
Sent Data	Yes	Text sent to the DIS stream. You can directly enter text or click  to use the EL expression.
Related Job	No	<p>Click *** to select a related job. You can select a batch job or a real-time job.</p> <p><b>NOTE</b> A maximum of 10 jobs can be selected.</p> <p>After selecting a job, click <b>Monitor</b>. On the <b>Monitory Job</b> page, select the DIS Client node and click <b>View Related Job</b> on the lower part of the page.</p> <p>In the <b>View Related Job</b> dialog box, click <b>View</b> in the <b>Operation</b> column of a job to view the details about the job.</p>

**Table 11-9** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.6 Rest Client


### Functions

The Rest Client node is used to respond to RESTful requests in HUAWEI CLOUD. Only the RESTful requests that have been authenticated by IAM Token are supported.

### Parameters

[Table 11-10](#), [Table 11-11](#), and [Table 11-12](#) describe the parameters of the Rest Client node.

**Table 11-10** Parameters of RestAPI nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
URL Address	Yes	IP address or domain name and port number of the request host. For example: https://192.160.10.10:8080
HTTP Method	Yes	Type of the request. Possible values: <ul style="list-style-type: none"><li>• <b>GET</b></li><li>• <b>POST</b></li><li>• <b>PUT</b></li><li>• <b>DELETE</b></li></ul>
Request Header	No	Click  to add a request header. The parameters are described as follows: <ul style="list-style-type: none"><li>• <b>Parameter Name</b> Name of a parameter. The options are <b>Content-Type</b>, <b>Accept-Language</b>, and <b>X-Auth-Token</b>.</li><li>• <b>Parameter Value</b> The value can contain a maximum of 64 characters and consists only of letters, digits, hyphens (-), underscores (_), slashes (/), and semicolons (;).</li></ul>
URL Parameter	No	Enters a URL parameter. The value is a character string in <b>key=value</b> format. Character strings are separated by newlines. This parameter is available only when <b>HTTP Method</b> is set to <b>GET</b> . Set these parameters as follows: <ul style="list-style-type: none"><li>• <b>Parameter</b> The value can contain a maximum of 32 characters and consists only of letters, digits, hyphens (-), and underscores (_).</li><li>• <b>Value</b> The value can contain a maximum of 64 characters and consists only of letters, digits, hyphens (-), underscores (_), dollar signs (\$), open braces { }, and close braces }.</li></ul>

Parameter	Mandatory	Description
Request Body	Yes	The request body is in JSON format. This parameter is available only when <b>HTTP Method</b> is set to <b>POST</b> or <b>PUT</b> .
Check Return Value	No	Checks whether the value of the returned message is the same as the expected value. This parameter is available only when <b>HTTP Method</b> is set to <b>GET</b> . Possible values: <ul style="list-style-type: none"><li>• <b>YES</b>: Check whether the return value is the same as the expected one.</li><li>• <b>NO</b>: No need to check whether the return value is the same as the expected one. A 200 response code is returned (indicating that the node is successfully performed).</li></ul>
Property Path	Yes	<p>Path of the property in the JSON response message. Each Rest Client node can have only one property path. This parameter is available only when <b>Check Returned Value</b> is set to <b>YES</b>.</p> <p>For example, the returned result is as follows:</p> <pre>{   "param1": "aaaa",   "inner": {     "inner": {       "param4": 2014247437     },     "param3": "cccc"   },   "status": 200,   "param2": "bbbb" }</pre> <p>The <b>param4</b> path is <b>inner.inner.param4</b>.</p>
Request Success Flag	Yes	<p>Enter the request success flag. If the returned value of the response matches one of request success flags, the node is successfully performed. This parameter is available only when <b>Check Returned Value</b> is set to <b>YES</b>.</p> <p>The request success flag can contain only letters, digits, hyphens (-), underscores (_), dollar signs (\$), open braces ({), and close braces (}). Separate values with semicolons (;).</p>




Parameter	Mandatory	Description
Request Failure Flag	No	<p>Enter the request failure flag. If the returned value of the response matches one of request failure flags, the node is successfully performed. This parameter is available only when <b>Check Returned Value</b> is set to <b>YES</b>.</p> <p>The request success flag can contain only letters, digits, hyphens (-), underscores (_), dollar signs (\$), open braces ({), and close braces (}). Separate values with semicolons (;).</p>
Retry Interval (seconds)	Yes	<p>If the return value of the response message does not match the request success flag, the node keeps querying the matching status at a specified interval until the return value of the response message is the same as the request success flag. By default, the timeout interval of the node is one hour. If the return value of the response message does not match the request success flag within this period, the node status changes to <b>Failed</b>. This parameter is available only when <b>Check Returned Value</b> is set to <b>YES</b>.</p>
The response message body parses the transfer parameter.	No	<p>Specify the mapping between the job variable and JSON property path. Separate parameters by newline characters.</p> <p>For example: var4=inner.inner.param4</p> <p><b>var4</b> is a job variable. The job variable must consist of 1 to 16 characters and contain only letters and digits. <b>inner.inner.param4</b> is the JSON property path.</p> <p>This parameter takes effect only when it is referenced by the subsequent node. When this parameter is referenced, the format is <b><code>\${var4}</code></b></p>




**Table 11-11** Advanced parameters

Parameter	Mandator y	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

**Table 11-12** Lineage

Parameter	Description
Input	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: name of the CSS index.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
View Details	Click  to view details about the table created based on the input lineage.
<b>Output</b>	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: Enter a CSS index name.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
Table Details	Click  to view details about the table created based on the output lineage.

## 11.1.7 Import GES

### Function

The Import GES node is used to import files from an OBS bucket to a GES graph.

## Parameters

[Table 11-13](#) and [Table 11-14](#) describe the parameters of the Import GES node.

**Table 11-13** Parameters of Import GES nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Graph Name	Yes	Select the graph to be imported. To create a GES graph, go to the GES console.
Metadata	Yes	Select the corresponding metadata.
Edge Dataset	Yes	Select the corresponding edge dataset.
Vertex Dataset	No	Select the corresponding vertex dataset. If it is not selected, the vertices in the edge dataset are used as the source of the vertex dataset.
Edge Processing	Yes	The edge processing supports the following modes: <ul style="list-style-type: none"><li>• Allow repetitive edges</li><li>• Ignore subsequent repetitive edges</li><li>• Overwrite previous repetitive edges</li></ul>
Offline	No	Indicates whether offline import is selected. The value is <b>true</b> or <b>false</b> , and the default value is <b>false</b> . <ul style="list-style-type: none"><li>• <b>true</b>: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import.</li><li>• <b>false</b>: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.</li></ul>

Parameter	Mandatory	Description
Ignore Labels on Repetitive Edges	No	Indicates whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b> , and the default value is <b>true</b> . <ul style="list-style-type: none"><li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li><li>• <b>false</b>: Indicates that the repetitive edge definition contains the label. That is, the &lt;source vertex, target vertex, label&gt; indicates an edge.</li></ul>
Log Storage Path	No	Stores vertex and edge datasets that do not comply with the metadata definition, as well as detailed logs generated during graph import.

**Table 11-14** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes</b>: The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No</b>: The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.

Parameter	Mandator y	Description
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.8 MRS Kafka

### Functions

The MRS Kafka node is used to query the number of messages that are not consumed by a topic.

### Parameters

[Table 11-15](#) and [Table 11-16](#) describe the parameters of the MRS Kafka node.

**Table 11-15** Parameters of MRS Kafka nodes

Parameter	Mandator y	Description
Data Connection	Yes	Select the MRS Kafka connection created in the management center.
Topic Name	Yes	Select a topic that has been created in MRS Kafka. The SDK or command line can be used to create a topic.
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

**Table 11-16** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.9 Kafka Client


### Functions

The Kafka Client node is used to send data to Kafka topics.

### Parameters

[Table 11-17](#) describes the parameters of the Kafka Client node.

**Table 11-17** Parameters of Kafka Client nodes

Parameter	Mandatory	Description
Data Connection	Yes	Select the MRS Kafka connection created in the management center.
Topic Name	Yes	Select the topic to which data is to be uploaded. If there are multiple partitions, data is sent to partition 0 by default.
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Text	Yes	Text content sent to Kafka. You can directly enter text or click  to use the EL expression.

**Table 11-18** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>

Parameter	Mandator y	Description
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.10 CS Job

### Functions

The CS Job node is used to execute a predefined Cloud Stream Service (CS) job for real-time analysis of streaming data.

### Context


This node enables you to start a CS job or query whether a CS job is running. If you do not select an existing real-time job, DLF creates and starts the job based on the job status configured on the node. You can customize jobs and use DLF job parameters.

### Parameters

[Table 11-19](#) and [Table 11-20](#) describe the parameters of the CS Job node.

**Table 11-19** Parameters of CS Job nodes

Parameter	Mandato ry	Description
Job Type	Yes	Select a job type for CS. <ul style="list-style-type: none"><li>• Existing CS job</li><li>• Flink SQL job</li><li>• User-defined Flink job</li><li>• User-defined Spark job</li></ul>
<b>Existing CS job</b>		

Parameter	Mandatory	Description
Streaming Job Name	Yes	Name of the CS job to be executed. To create a CS job, you can use either of the following methods: <ul style="list-style-type: none"><li>Click . On the <b>Data Integration</b> page of DLF, create a CS job.</li><li>Go to the CS console to create a CS job.</li></ul>
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
<b>Flink SQL job</b>		
SQL Script	Yes	Path to a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a> .
Script Parameter	Yes	If the associated SQL script uses a parameter, the parameter name is displayed. Set the parameter value in the text box next to the parameter name. The parameter value can be a built-in function and EL expression. For details about built-in functions and EL expressions, see <a href="#">Expression Overview</a> .
CloudStream Cluster	Yes	Name of the CS cluster. To create a CS cluster, go to the CS console.
SPUs	Yes	1 SPU = 1 core and 4 GB memory
Parallelism	Yes	Number of tasks that run CS jobs at the same time. You are advised to set this parameter to 1 or 2 times of the SPU.
UDF JAR	No	After the JAR package is imported, you can use SQL statements to call the custom functions in the package. You need to upload the JAR package to the OBS bucket.
Auto Restart upon Exception	No	If you enable this function, the system automatically restarts and restores abnormal CS jobs upon job exceptions.
Streaming Job Name	Yes	Name of the Flink SQL job. The name is 1 to 57 characters long and consists only of letters, digits, hyphens (-), and underlines (_).

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
<b>User-defined Flink job</b>		
JAR Package Path	Yes	The JAR package path can be selected only after you have uploaded the custom JAR package to the OBS bucket.
Main Class	No	Name of the main class in the JAR file to be uploaded, for example, <b>KafkaMessageStreaming</b> . If this parameter is not specified, the main class name is determined based on the <b>Manifest</b> file in the JAR file.
Main Class Parameter	No	List of parameters for the main class. Parameters are separated by spaces, for example, test tmp/result.txt.
CloudStream Cluster	Yes	Name of the CS cluster. To create a CS cluster, go to the CS console.
SPUs	Yes	1 SPU = 1 core and 4 GB memory
Driver SPU	Yes	Number of SPUs used for each driver node.
Parallelism	Yes	Number of tasks that run CS jobs at the same time. You are advised to set this parameter to 1 or 2 times of the SPU.
Auto Restart upon Exception	No	If you enable this function, the system automatically restarts and restores abnormal CS jobs upon job exceptions.
Streaming Job Name	Yes	Name of the user-defined Flink job. The name is 1 to 57 characters long and consists only of letters, digits, hyphens (-), and underlines (_).
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
<b>User-defined Spark job</b>		

Parameter	Mandatory	Description
JAR Package Path	Yes	The JAR package path can be selected only after you have uploaded the custom JAR package to the OBS bucket.
Main Class	No	Name of the main class in the JAR file to be uploaded, for example, <b>KafkaMessageStreaming</b> . If this parameter is not specified, the main class name is determined based on the <b>Manifest</b> file in the JAR file.
Main Class Parameter	No	List of parameters for the main class. Parameters are separated by spaces, for example, test tmp/result.txt.
CloudStream Cluster	Yes	Name of the CS cluster. To create a CS cluster, go to the CS console.
SPUs	Yes	1 SPU = 1 core and 4 GB memory
Driver SPU	Yes	Number of SPUs used for each driver node.
Executors	Yes	Number of the executor nodes.
Executor SPUs	Yes	Number of SPUs used for each executor node.
Auto Restart upon Exception	No	If you enable this function, the system automatically restarts and restores abnormal CS jobs upon job exceptions.
Streaming Job Name	Yes	Name of the user-defined Spark job. The name is 1 to 57 characters long and consists only of letters, digits, hyphens (-), and underlines (_).
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

**Table 11-20** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.11 DLI SQL

### Functions

The DLI SQL node is used to transfer SQL statements to DLI for data source analysis and exploration.



### Working Principles

This node enables you to execute DLI statements during periodical or real-time job scheduling. You can use parameter variables to perform incremental import and process partitions for your data warehouses.

## Parameters

[Table 11-21](#), [Table 11-22](#), and [Table 11-23](#) describe the parameters of the DLI SQL node.

**Table 11-21** Parameters of DLI SQL nodes

Parameter	Mandatory	Description
SQL Statement or Script	Yes	<p>You can select <b>SQL statement</b> or <b>SQL script</b>.</p> <ul style="list-style-type: none"><li>SQL Statement In the <b>SQL statement</b> text box, enter the SQL statement to be executed.</li><li>SQL Script Select a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a>.</li></ul> <p><b>NOTE</b> If you select the SQL statement mode, the DLF cannot parse the parameters contained in the SQL statement.</p>
Database Name	Yes	Database that is configured in the SQL script. The value can be changed.
Queue Name	Yes	<p>Name of the DLI queue configured in the SQL script. The value can be changed.</p> <p>You can create a resource queue using either of the following methods:</p> <ul style="list-style-type: none"><li>Click . On the <b>Queue Management</b> page of DLF, create a resource queue.</li><li>Go to the DLI console to create a resource queue.</li></ul>
Script Parameter	No	<p>If the associated SQL script uses a parameter, the parameter name is displayed. Set the parameter value in the text box next to the parameter name. The parameter value can be a built-in function and EL expression. For details about built-in functions and EL expressions, see <a href="#">Expression Overview</a>.</p> <p>If the parameters of the associated SQL script are changed, click  to refresh the parameters.</p>

Parameter	Mandator y	Description
Node Name	Yes	Name of the SQL script. The value can be changed. The rules are as follows: Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Record Dirty Data	Yes	Click <input type="radio"/> to specify whether to record dirty data. <ul style="list-style-type: none"><li>• If you select <input type="radio"/>, dirty data will be recorded.</li><li>• If you do not select <input type="radio"/>, dirty data will not be recorded.</li></ul>





**Table 11-22** Advanced parameters



Parameter	Mandator y	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.

Parameter	Mandatory	Description
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

Table 11-23 Lineage

Parameter	Description
<b>Input</b>	
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: name of the CSS index.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>

Parameter	Description
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
View Details	Click  to view details about the table created based on the input lineage.
<b>Output</b>	
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: Enter a CSS index name.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.

Parameter	Description
Delete	Click  to delete the parameter settings.
Table Details	Click  to view details about the table created based on the output lineage.

## 11.1.12 DLI Spark

### Functions

The DLI Spark node is used to execute a predefined Spark job.

### Parameters

[Table 11-24](#) and [Table 11-25](#) describe the parameters of the DLI Spark node.

**Table 11-24** Parameters of DLI Spark nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
DLI Queue	Yes	Select a queue from the drop-down list box.
Job Name	Yes	Name of the DLI Spark job. It must consist of 1 to 64 characters and contain only letters, digits, and underscores (_). The default value is the same as the node name.
Job Running Resources	No	Select the running resource specifications of the job. <ul style="list-style-type: none"><li>• 8-core, 32 GB memory</li><li>• 16-core, 64 GB memory</li><li>• 32-core, 128 GB memory</li></ul>
Major Job Class	Yes	Main class of the DLI Spark job, that is, the main class of the JAR package.

Parameter	Mandatory	Description
Spark program resource package	Yes	JAR package of the user-defined Spark application. Before selecting a resource package, you need to upload the JAR package and its dependency packages to the OBS bucket and create resources on the resource management page. For details, see <a href="#">Creating a Resource</a> .
Major-Class Entry Parameters	No	Enter the entry parameters of the program. Press <b>Enter</b> to separate the parameters.
Spark Job Running Parameters	No	Enter a parameter in the format of <b>key/value</b> . Press Enter to separate multiple key-value pairs. For details about the parameters, see <a href="#">Spark Configuration</a> .
Module Name	No	Dependency modules provided by DLI for executing datasource connection jobs. To access different services, you need to select different modules. <ul style="list-style-type: none"><li>• CloudTable/MRS HBase: sys.datasource.hbase</li><li>• CloudTable/MRS OpenTSDB: sys.datasource.opentsdb</li><li>• RDS MySQL: sys.datasource.rds</li><li>• RDS PostGre: sys.datasource.rds</li><li>• DWS: sys.datasource.dws</li><li>• CSS: sys.datasource.css</li></ul>

**Table 11-25** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.13 DWS SQL

### Functions

The DWS SQL node is used to transfer an SQL statement to DWS.


### Context

This node enables you to execute DWS statements during batch or real-time job processing. You can use parameter variables to perform incremental import and process partitions for your data warehouses.

### Parameters

[Table 11-26](#), [Table 11-27](#), and [Table 11-28](#) describe the parameters of the DWS SQL node.

**Table 11-26** Parameters of DWS SQL nodes




Parameter	Mandator y	Description
SQL Statement or Script	Yes	<p>You can select <b>SQL statement</b> or <b>SQL script</b>.</p> <ul style="list-style-type: none"><li>• <b>SQL Statement</b> In the <b>SQL statement</b> text box, enter the SQL statement to be executed.</li><li>• <b>SQL Script</b> Select a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a>.</li></ul> <p><b>NOTE</b> If you select the SQL statement mode, the DLF cannot parse the parameters contained in the SQL statement.</p>
Data Connection	Yes	Data connection that is configured in the SQL script. The value can be changed.
Database	Yes	Database that is configured in the SQL script. The value can be changed.
Script Parameter	No	<p>If the associated SQL script uses a parameter, the parameter name is displayed. Set the parameter value in the text box next to the parameter name. The parameter value can be a built-in function and EL expression. For details about built-in functions and EL expressions, see <a href="#">Expression Overview</a>.</p> <p>If the parameters of the associated SQL script are changed, click  to refresh the parameters.</p>
Dirty Data Table	No	Enter the name of the dirty data table defined in the SQL script.
Node Name	Yes	<p>Name of the SQL script. The value can be changed. The rules are as follows:</p> <p>Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (&lt;), and greater-than signs (&gt;).</p>




**Table 11-27** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

**Table 11-28** Lineage

Parameter	Description
Input	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: name of the CSS index.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
View Details	Click  to view details about the table created based on the input lineage.
<b>Output</b>	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: Enter a CSS index name.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
Table Details	Click  to view details about the table created based on the output lineage.

## 11.1.14 MRS SparkSQL


### Functions

The MRS Spark SQL node is used to execute a predefined SparkSQL statement on MRS.

### Parameters

[Table 11-29](#) and [Table 11-30](#) describe the parameters of the MRS Spark SQL node.

**Table 11-29** Parameters of MRS Spark SQL nodes

Parameter	Mandatory	Description
SQL Script	Yes	Path to a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a> .
Data Connection	Yes	Data connection that is configured in the SQL script. The value can be changed.
Database	Yes	Database that is configured in the SQL script. The value can be changed.
Script Parameter	No	<p>If the associated SQL script uses a parameter, the parameter name is displayed. Set the parameter value in the text box next to the parameter name. The parameter value can be a built-in function and EL expression. For details about built-in functions and EL expressions, see <a href="#">Expression Overview</a>.</p> <p>If the parameters of the associated SQL script are changed, click  to refresh the parameters.</p>
Program Parameter	No	<p>Used to configure optimization parameters such as threads, memory, and vCPUs for the job to optimize resource usage and improve job execution performance.</p> <p><b>NOTE</b> This parameter is mandatory if the cluster version is MRS 1.8.7 or later than MRS 2.0.1.</p> <p>For details about the program parameters of MRS SparkSQL jobs, see <a href="#">Running a SparkSql Job</a> &gt; <b>Table 2 Program Parameter parameters</b> in the <i>MapReduce User Guide</i>.</p>

Parameter	Mandator y	Description
Node Name	Yes	Name of the SQL script. The value can be changed. The rules are as follows: Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

**Table 11-30** Advanced parameters

Parameter	Mandator y	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.

Parameter	Mandator y	Description
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.15 MRS Hive SQL

### Functions


The MRS Hive SQL node is used to execute a predefined Hive SQL script on DLF.

### Parameters

[Table 11-31](#) and [Table 11-32](#) describe the parameters of the MRS Hive SQL node.

**Table 11-31** Parameters of MRS Hive SQL nodes

Parameter	Mandator y	Description
SQL Script	Yes	Path to a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a> .
Data Connection	Yes	Data connection that is configured in the SQL script. The value can be changed.
Database	Yes	Database that is configured in the SQL script. The value can be changed.

Parameter	Mandatory	Description
Script Parameter	No	<p>If the associated SQL script uses a parameter, the parameter name is displayed. Set the parameter value in the text box next to the parameter name. The parameter value can be a built-in function and EL expression. For details about built-in functions and EL expressions, see <a href="#">Expression Overview</a>.</p> <p>If the parameters of the associated SQL script are changed, click  to refresh the parameters.</p>
Program Parameter	No	<p>Used to configure optimization parameters such as threads, memory, and vCPUs for the job to optimize resource usage and improve job execution performance.</p> <p><b>NOTE</b> This parameter is mandatory if the cluster version is MRS 1.8.7 or later than MRS 2.0.1.</p> <p>For details about the program parameters of MRS SparkSQL jobs, see <a href="#">Running a HiveSql Job</a> &gt; <b>Table 2 Program Parameter parameters</b> in the <i>MapReduce User Guide</i>.</p>
Node Name	Yes	<p>Name of the SQL script. The value can be changed. The rules are as follows:</p> <p>Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (&lt;), and greater-than signs (&gt;).</p>

**Table 11-32** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.16 MRS Presto SQL


### Functions

The MRS Presto SQL node is used to execute a predefined Presto SQL script on Data Development.

### Parameters

[Table 11-33](#) and [Table 11-34](#) describe the parameters of the MRS Presto SQL node.

**Table 11-33** Parameters of MRS Presto SQL nodes

Parameter	Mandatory	Description
SQL Script	Yes	Path to a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a> .
Data Connection	Yes	Data connection that is configured in the SQL script. The value can be changed.
Schema	Yes	Database that is configured in the SQL script. The value can be changed.
Script Parameter	No	If the associated SQL script uses a parameter, the parameter name is displayed. Set the parameter value in the text box next to the parameter name. The parameter value can be a built-in function and EL expression. For details about built-in functions and EL expressions, see <a href="#">Expression Overview</a> . If the parameters of the associated SQL script are changed, click  to refresh the parameters.
Node Name	Yes	Name of the SQL script. The value can be changed. The rules are as follows: Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

**Table 11-34** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.17 MRS Spark

### Functions


The MRS Spark node is used to execute a predefined Spark job on MRS.

### Parameters

[Table 11-35](#) and [Table 11-36](#) describe the parameters of the MRS Spark node.

**Table 11-35** Parameters of MRS Spark nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

Parameter	Mandatory	Description
MRS Cluster Name	Yes	Name of the MRS cluster. To create an MRS cluster, use either of the following methods: <ul style="list-style-type: none"><li>Click . On the <b>Clusters</b> page, create an MRS cluster.</li><li>Go to the MRS console to create an MRS cluster.</li></ul>
Spark Job Name	Yes	Name of the MRS job. Must consist of 1 to 64 characters and contain only letters, digits, and underscores (_).
JAR Package	Yes	Select a JAR package. Before selecting a JAR package, you need to upload the JAR package to the OBS bucket, create a resource on the <b>Resource Management</b> page, and add the JAR package to the resource management list. For details, see <a href="#">Creating a Resource</a> .
JAR File Parameters	No	Parameters of the JAR package.
Program Parameter	No	Used to configure optimization parameters such as threads, memory, and vCPUs for the job to optimize resource usage and improve job execution performance. <b>NOTE</b> This parameter is mandatory if the cluster version is MRS 1.8.7 or later than MRS 2.0.1. For details about the program parameters of MRS Spark jobs, see <a href="#">Running a Spark Job</a> in the <i>MapReduce User Guide</i> .
Input Data Path	No	Path where the input data resides.
Output Data Path	No	Path where the output data resides.

**Table 11-36** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.18 MRS Spark Python


### Functions

The MRS Spark Python node is used to execute a predefined Spark Python job on MRS.

### Parameters

[Table 11-37](#) and [Table 11-38](#) describe the parameters of the MRS Spark Python node.

**Table 11-37** Parameters of MRS Spark Python nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
MRS Cluster Name	Yes	Select an MRS cluster that supports Spark Python. Only a specific version of MRS supports Spark Python. Test the cluster first to ensure that it supports Spark Python. To create an MRS cluster, use either of the following methods: <ul style="list-style-type: none"><li>Click . On the <b>Clusters</b> page, create an MRS cluster.</li><li>Go to the MRS console to create an MRS cluster.</li></ul> For details about how to create an MRS cluster, see <a href="#">Custom Purchase of a Cluster</a> in <i>MapReduce User Guide</i> .
Job Name	Yes	Name of the MRS Spark Python job. It must consist of 1 to 64 characters and contain only letters, digits, and underscores (_).
Parameter	Yes	Enter the parameters of the executable program of MRS. Use <b>Enter</b> to separate multiple parameters.
Attribute	No	Enter parameters in the key=value format. Use <b>Enter</b> to separate multiple parameters.

**Table 11-38** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.19 MRS Flink Job

### Function


The MRS Flink node is used to execute predefined Flink jobs in MRS.

### Parameter

Configure the parameters of the MRS Flink node by referring to [Table 11-39](#) and [Table 11-40](#).

**Table 11-39** Property parameters

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

Parameter	Mandatory	Description
MRS Cluster Name	Yes	Select the MRS cluster. To create an MRS cluster, use either of the following methods: <ul style="list-style-type: none"><li>Click . On the <b>Clusters</b> page, create an MRS cluster.</li><li>Go to the MRS console to create an MRS cluster.</li></ul>
Flink Job Name	Yes	Name of the MRS Flink job. Must consist of 1 to 64 characters and contain only letters, digits, and underscores (_).
Flink Job Resource Package	Yes	Select a JAR package. Before selecting a JAR package, you need to upload the JAR package to the OBS bucket, create a resource on the <b>Resource Management</b> page, and add the JAR package to the resource management list. For details, see <a href="#">Creating a Resource</a> .
Flink Job Parameter	No	Key parameter of the program that executes the Flink job. This parameter is specified by a function in the user program. Separate multiple parameters with space.
Program Parameter	No	Used to configure optimization parameters such as threads, memory, and vCPUs for the job to optimize resource usage and improve job execution performance. <b>NOTE</b> This parameter is mandatory if the cluster version is MRS 1.8.7 or later than MRS 2.0.1. For details about the program parameters of MRS Flink jobs, see <a href="#">Running a Flink Job</a> in the <i>MapReduce User Guide</i> .
Input Data Path	No	Path where the input data resides.
Output Data Path	No	Path where the output data resides.

**Table 11-40** Advanced settings

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Maximum duration of executing a node. When <b>Retry upon Failure</b> is set to <b>Yes</b> for a node, the node can be re-executed for numerous times upon an execution failure within the maximum duration.
Retry upon Failure	Yes	<p>Specifies whether to re-execute a node after the node fails to be executed.</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node will be re-executed after it fails to be executed. The following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Policies to be performed after the node fails to be executed:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next node</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>End execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.20 MRS MapReduce


### Functions

The MRS MapReduce node is used to execute a predefined MapReduce program on MRS.

### Parameters

[Table 11-41](#) and [Table 11-42](#) describe the parameters of the MRS MapReduce node.

**Table 11-41** Parameters of MRS MapReduce nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
MRS Cluster Name	Yes	Name of the MRS cluster. To create an MRS cluster, use either of the following methods: <ul style="list-style-type: none"><li>Click . On the <b>Clusters</b> page, create an MRS cluster.</li><li>Go to the MRS console to create an MRS cluster.</li></ul>
MapReduce Job Name	Yes	Name of the MRS job. Must consist of 1 to 64 characters and contain only letters, digits, and underscores (_).
JAR Package	Yes	Select a JAR package. Before selecting a JAR package, you need to upload the JAR package to the OBS bucket, create a resource on the <b>Resource Management</b> page, and add the JAR package to the resource management list. For details, see <a href="#">Creating a Resource</a> .
JAR File Parameters	No	Parameters of the JAR package.
Input Data Path	No	Path where the input data resides.
Output Data Path	No	Path where the output data resides.

**Table 11-42** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandator y	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.21 CSS

### Functions

The CSS node is used to process CSS requests and enable online distributed searching.

### Parameters

[Table 11-43](#) and [Table 11-44](#) describe the parameters of the CSS node.

**Table 11-43** Parameters of CSS nodes

Parameter	Mandator y	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
CloudSearch Cluster	Yes	Connection to CloudSearch. A CloudSearch cluster has been created in CloudService. Currently, only clusters of version 5.5.1 is supported.
CDM Cluster Name	Yes	A CDM cluster. The Agent provided by the CDM cluster is the operating environment of CSS.  If there are no CDM clusters available in the drop-down list, create one on the CDM console.
Request Type	Yes	Possible values: <ul style="list-style-type: none"><li>• GET</li><li>• POST</li><li>• PUT</li><li>• HEAD</li><li>• DELETE</li></ul>
Request Parameter	No	Parameter of the request.  For example, to query the dlfddata mapping type in the dlf_search index, set this parameter to: <b>/dlf_search/dlfddata/_search</b>
Request Body	No	The request body is in JSON format.
CloudSearch Output Path	No	Path where output data is to be stored.

**Table 11-44** Advanced parameters

Parameter	Mandator y	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.22 Shell

### Functions

The Shell node is used to execute a shell script.

#### NOTE

With EL expression `#{Job.getNodeOutput()}`, you can obtain the desired content (4000 characters at most and counted backwards) in the output of the shell script run by the Shell node.

Example:

To obtain `<name>jack<name1>` from a shell script (script name: shell\_job1) output, enter the following EL expression:

```
#{StringUtil.substringBetween(Job.getNodeOutput("shell_job1"), "<name>", "<name1>")}
```

## Parameters

[Table 11-45](#) and [Table 11-46](#) describe the parameters of the Shell node.

**Table 11-45** Parameters of Shell nodes

Parameter	Mandator y	Description
SQL Statement or Script	Yes	<p>You can select <b>SQL statement</b> or <b>SQL script</b>.</p> <ul style="list-style-type: none"><li>SQL Statement In the <b>SQL statement</b> text box, enter the SQL statement to be executed.</li><li>SQL Script Select a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing a Shell Script</a>.</li></ul> <p><b>NOTE</b> If you select the SQL statement mode, the DLF cannot parse the parameters contained in the SQL statement.</p>
Host Connection	Yes	Selects the host where a shell script is to be executed.
Script Parameter	No	Parameter transferred to the script when the shell script is executed. Parameters are separated by spaces. For example: <b>a b c</b> . The parameter must be referenced by the shell script. Otherwise, the parameter is invalid.
Interactive Input	No	Interactive information (passwords for example) provided during shell script execution. Interactive parameters are separated by carriage return characters. The shell script reads parameter values in sequence according to the interaction situation.
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

**Table 11-46** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.23 RDS SQL

### Functions

The RDS SQL node is used to transfer SQL statements to RDS.

### Parameters

[Table 11-47](#) and [Table 11-48](#) describe the parameters of the RDS SQL node.

**Table 11-47** Parameters of RDS SQL nodes

Parameter	Mandator y	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Data Connection	Yes	Name of the data connection.
Database	Yes	Name of the database. The database has been created. You are advised not to use the default database.
SQL or Script	Yes	<p>You can select <b>SQL statement</b> or <b>SQL script</b>.</p> <ul style="list-style-type: none"><li>• <b>SQL statement</b> In the <b>SQL statement</b> text box, enter the SQL statement to be executed.</li><li>• <b>SQL script</b> Select a script to be executed. If the script is not created, create and develop the script by referring to <a href="#">Creating a Script</a> and <a href="#">Developing an SQL Script</a>.</li></ul> <p><b>NOTE</b> If you select <b>SQL statement</b>, DLF cannot parse the parameters contained in the SQL statement.</p>

**Table 11-48** Advanced parameters

Parameter	Mandator y	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.24 ETL Job


### Functions

The ETL Job node is used to extract data from a specified data source, preprocess the data, and import the data to the target data source.

### Parameters

[Table 11-49](#) and [Table 11-50](#) describe the parameters of the ETL Job node.

**Table 11-49** Parameters of Transform Load nodes

Parameter	Mandator y	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
ETL Configuration	Yes	<p>Click  to edit the source and destination data to be transformed.</p> <p>The supported source data types are DLI, OBS, and MySQL.</p> <ul style="list-style-type: none"><li>• When the source data type is DLI, the supported destination data types are DWS, GES, CSS, OBS, and DLI.</li><li>• When the source data type is MySQL, the supported destination data type is MySQL.</li><li>• When the source data type is OBS, the supported destination data types are DLI and DWS.</li></ul> <p><b>NOTICE</b></p> <ul style="list-style-type: none"><li>• Data transformation from DLI to DWS: Before importing data from DLI to DWS, ensure that a DWS data connection has been created on DLF and a DWS table have been created on DWS.</li><li>• Data transformation from DLI to CSS: Before importing data from DLI to CSS, ensure that a cross-source connection associated with CSS has been created on DLI. For details about how to create a cross-source connection on DLI, see <i>Data Lake Insight User Guide</i>.</li></ul>
Configure SQL Template	No	Click <b>Obtain Template</b> to obtain an SQL template.




**Table 11-50** Advanced parameters




Parameter	Mandator y	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

**Table 11-51** Lineage

Parameter	Description
Input	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: name of the CSS index.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
View Details	Click  to view details about the table created based on the input lineage.
<b>Output</b>	

Parameter	Description
Add	<p>Click <b>Add</b>. In the <b>Type</b> drop-down list, select the type to be created. The value can be DLI, DWS, OBS, or CSS.</p> <ul style="list-style-type: none"><li>• DWS<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DWS data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DWS database.</li><li>– <b>Schema</b>: Click ... In the displayed dialog box, select a DWS schema.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DWS table.</li></ul></li><li>• OBS<ul style="list-style-type: none"><li>– <b>Path</b>: Click ... In the displayed dialog box, select an OBS path.</li></ul></li><li>• CSS<ul style="list-style-type: none"><li>– <b>Cluster Name</b>: Click ... In the displayed dialog box, select a CSS cluster.</li><li>– <b>Index</b>: Enter a CSS index name.</li></ul></li><li>• DLI<ul style="list-style-type: none"><li>– <b>Connection Name</b>: Click ... In the displayed dialog box, select a DLI data connection.</li><li>– <b>Database</b>: Click ... In the displayed dialog box, select a DLI database.</li><li>– <b>Table Name</b>: Click ... In the displayed dialog box, select a DLI table.</li></ul></li></ul>
OK	Click <b>OK</b> to save the parameter settings.
Cancel	Click <b>Cancel</b> to cancel the parameter settings.
Edit	Click  to modify the parameter settings. After the modification, save the settings.
Delete	Click  to delete the parameter settings.
Table Details	Click  to view details about the table created based on the output lineage.

## 11.1.25 OCR

### Functions

The OCR node is used to recognize text on images and store the recognition result in a specified OBS directory.

#### NOTE

The OCR node can be used only in CN North-Beijing4.

### Parameters

[Table 11-52](#) and [Table 11-53](#) describe the parameters of the OCR node.

**Table 11-52** Parameters of OCR nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Image Path	Yes	OBS path of the image to be recognized.
Category	Yes	Image type that can be recognized. Currently, only General Text OCR is supported. It can be used in various scenarios, such as scanned files, electronic documents, books, documents, and forms.
Storage Directory	Yes	OBS directory for storing the recognition result.

**Table 11-53** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	Indicates whether to re-execute a node task if its execution fails. Possible values: <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.
Failure Policy	Yes	Operation that will be performed if the node task fails to be executed. Possible values: <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.26 Create OBS

### Functions

The Create OBS node is used to create buckets and directories in OBS.

### Parameters

[Table 11-54](#) and [Table 11-55](#) describe the parameters of the Create OBS node.

**Table 11-54** Parameters of Create OBS nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

Parameter	Mandatory	Description
OBS Path	Yes	<p>Path to the OBS bucket or directory.</p> <ul style="list-style-type: none"><li>• To create a bucket, enter <i>//OBS bucket name</i>. The OBS bucket name must be unique</li><li>• To create an OBS directory, select the path to the OBS directory to be created, and enter the <i>/Directory name</i> following the path. The directory name must be unique.</li></ul>

**Table 11-55** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	<p>Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.</p>
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.27 Delete OBS

### Functions

The Delete OBS node is used to delete a bucket or directory on OBS.

### Parameters

[Table 11-56](#) and [Table 11-57](#) describe the parameters of the Delete OBS node.

**Table 11-56** Parameters of Delete OBS nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
OBS Path	Yes	Path to the OBS bucket or directory. <b>NOTE</b> If you delete an OBS bucket or directory, files stored in it are also deleted and cannot be restored. Before you delete a bucket or directory, back up the files stored in it if they need to be retained.

**Table 11-57** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.28 OBS Manager

### Function

The OBS Manager node is used to move or copy files from an OBS bucket to a specified directory.

### Parameters

[Table 11-58](#) and [Table 11-59](#) describe the parameters of the OBS Manager node.

**Table 11-58** Parameters of OBS Manager nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Operation Type	Yes	Operations that can be performed on the node. <ul style="list-style-type: none"><li>• Move File</li><li>• Copy File</li></ul>
Source File	Yes	OBS file to be moved or copied from the OBS bucket.
Target Directory	Yes	Directory for storing OBS files to be moved or copied from the OBS bucket.
File Filter	No	Wildcard for file filtering. Only the files that meet the filtering condition can be moved or copied. If this parameter is not specified, all source files are moved or copied by default. For example, when you enter *.csv, files in this format will be moved or copied.

**Table 11-59** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.29 Open/Close Resource

### Functions

The Open/Close Resource node is used to enable or disable the HUAWEI CLOUD service.

### Parameters

[Table 11-60](#) and [Table 11-61](#) describe the parameters of the Open/Close Resource node.

**Table 11-60** Parameters of Open/Close Resource nodes

Parameter	Mandator y	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Service	Yes	Service to be opened or closed. <ul style="list-style-type: none"><li>• ECS</li><li>• CDM</li><li>• CS</li></ul>
Open/Close Resource	Yes	Possible values: <ul style="list-style-type: none"><li>• On</li><li>• Off</li></ul>
Instance	Yes	Object to be opened or closed, for example, to open a CDM cluster.

**Table 11-61** Advanced parameters

Parameter	Mandator y	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.30 Data Quality Monitor

### Functions

The Data Quality Monitor node is used to monitor the quality of running data.

### Parameters

[Table 11-62](#) and [Table 11-63](#) describe the parameters of the Data Quality Monitor node.

**Table 11-62** Parameters of Data Quality Monitor nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Quality Rule Name	Yes	Name of the rule for monitoring the running data. For details about how to create a rule, see the <i>DAYU User Guide</i> .

**Table 11-63** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.31 Subjob

### Function

The Subjob node is used to call the batch job that does not contain the subjob node.

### Parameter

[Table 11-64](#) and [Table 11-65](#) describe the parameters of the Subjob node.

**Table 11-64** Parameters of subjob nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Subjob Name	Yes	Select the name of the subjob to be called. <b>NOTE</b> You can only select the name of an existing batch job that does not contain the Subjob node.
Subjob Parameter	Yes/No	<ul style="list-style-type: none"><li>If the subjob parameters are left unspecified, the subjob is executed with its own parameter variables. The <b>Subjob Parameter Name</b> of the parent job is not displayed.</li><li>If the subjob parameters are specified, the subjob is executed with the configured parameter values. In this case, the <b>Subjob Parameter Name</b> of the parent job is displayed, and the data or EL expression configured for the subjob is accessed and replaced according to the environment variable of the parent job.</li></ul>

**Table 11-65** Advanced parameters

Parameter	Mandatory	Description
Node Status Polling Interval (s)	Yes	Specifies how often the system check completeness of the node task. The value ranges from 1 to 60 seconds.

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• Suspend execution plans of the current and subsequent nodes</li></ul>

## 11.1.32 SMN

### Functions

The SMN node is used to send notifications to users.

### Parameters

[Table 11-66](#) and [Table 11-67](#) describe the parameters of the SMN node.

**Table 11-66** Parameters of SMN nodes

Parameter	Mandator y	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Topic Name	Yes	Name of the topic. The topic has been created in SMN.
Message Title	No	Title of the message. The title cannot exceed 512 characters.
Message Type	Yes	<p>Format of the message.</p> <ul style="list-style-type: none"><li>• <b>Text:</b> The message is sent in text format.</li><li>• <b>JSON:</b> The message is sent in JSON format. You can send different messages to types of subscribers.<ul style="list-style-type: none"><li>– Manual: You can enter a message in <b>Message Content</b>.</li><li>– Automatic: Click <b>Generate JSON Message</b>. In the displayed dialog box, enter a message and select a protocol.</li></ul></li><li>• <b>Template:</b> The message is sent in template format, that is, in fixed format. The variables can be processed by tags.<ul style="list-style-type: none"><li>– Manual: You can enter a message in <b>Message Content</b>.</li><li>– Automatic: Click <b>Generate Template Message</b>. In the displayed dialog box, select a template name and set the value of <b>tag</b>.</li></ul></li></ul>

Parameter	Mandatory	Description
Message Content	Yes	<p>Message content to be provided. The requirements for entering different types of messages are as follows:</p> <ul style="list-style-type: none"><li>• <b>Text:</b> The size cannot exceed 10 KB.</li><li>• <b>JSON:</b> The JSON message must contain the Default protocol and the size cannot exceed 10 KB. Example:<pre>{  "default": "Dear Sir or Madam, this is a default message.",  "email": "Dear Sir or Madam, this is an email message.",  "http": "{ 'message': 'Dear Sir or Madam, this is an HTTP message.' }",  "https": "{ 'message': 'Dear Sir or Madam, this is an HTTPS message.' }",  "sms": "This is an SMS message."}</pre></li><li>• <b>Template:</b> The size cannot exceed 10 KB. Example:<pre>"message_template_name": "confirm_message",  "tags": {    "topic_urn": "urn:smn:regionId:xxxx:SMN_01"  }</pre><p>In the preceding information, <b>message_template_name</b> indicates the template name, and <b>tags</b> indicates all tags in the template.</p><p>For details about how to configure SMN, see section <i>Simple Message Notification User Guide</i>.</p></li></ul>

**Table 11-67** Advanced parameters

Parameter	Mandatory	Description
Max. Node Execution Duration	Yes	Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.

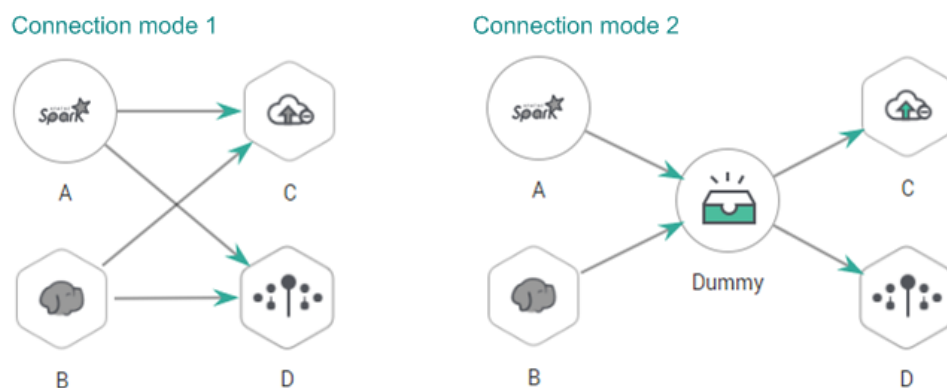
Parameter	Mandator y	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.1.33 Dummy

### Functions

The Dummy node is empty and does not perform any operations. It is used to simplify the complex connection relationships of nodes.

**Figure 11-2** Connection modes



## Parameters

[Table 11-68](#) describes the parameter of the Dummy node.

**Table 11-68** Parameters of Dummy nodes

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).

## 11.1.34 For Each

### Functions

Specifies a subjob to be executed cyclically and assigns values to variables in a subjob with a dataset.

### Parameters

[Table 11-69](#) describes the parameters of the For Each node.

**Table 11-69** Parameters of the For Each node

Parameter	Mandatory	Description
Node Name	Yes	Name of the node. Must consist of 1 to 128 characters and contain only letters, digits, underscores (_), hyphens (-), slashes (/), less-than signs (<), and greater-than signs (>).
Subjob in a Loop	Yes	Name of the subjob to be executed cyclically.

Parameter	Mandator y	Description
Dataset	Yes	<p>The For Loop operator needs to define a dataset, which is used to cyclically replace the variables in subjobs. The dataset comes from:</p> <ul style="list-style-type: none"><li>Upstream nodes For example, the SELECT statement of DLI SQL or the echo of the shell node. Currently, only the SELECT statements of Hive SQL, Spark SQL, and DLI SQL are supported.</li><li>Files in CSV format</li></ul>
Concurrent Subjobs	Yes	<p>Subjobs generated cyclically can be executed concurrently. You can set the number of concurrent subjobs.</p>
Subjob Suffix	No	<p>Name of the subjob generated by For Each: For Each node name + underscore (_) + suffix.</p> <p>The suffix is configurable. If the suffix is not configured, the suffix increases in ascending order based on the number.</p>
Job Running Parameter	No	<ul style="list-style-type: none"><li>If the subjob parameters are left unspecified, the subjob is executed with its own parameter variables.</li><li>If the subjob parameters are specified, the subjob is executed with the configured parameter values. The method or EL expression configured for the subjob parameter in the node attribute is read and replaced based on the environment variable of the parent job.</li></ul>

**Table 11-70** Advanced parameters

Parameter	Mandator y	Description
Max. Node Execution Duration	Yes	<p>Execution timeout interval for the node. If retry is configured and the execution is not complete within the timeout interval, the node will not be retried and is set to the failed state.</p>

Parameter	Mandatory	Description
Retry upon Failure	Yes	<p>Indicates whether to re-execute a node task if its execution fails. Possible values:</p> <ul style="list-style-type: none"><li>• <b>Yes:</b> The node task will be re-executed, and the following parameters must be configured:<ul style="list-style-type: none"><li>– <b>Maximum Retries</b></li><li>– <b>Retry Interval (seconds)</b></li></ul></li><li>• <b>No:</b> The node task will not be re-executed. This is the default setting.</li></ul> <p><b>NOTE</b> If <b>Timeout Interval</b> is configured for the node, the node will not be executed again after the execution times out. Instead, the node is set to the failure state.</p>
Failure Policy	Yes	<p>Operation that will be performed if the node task fails to be executed. Possible values:</p> <ul style="list-style-type: none"><li>• <b>End the current job execution plan</b></li><li>• <b>Go to the next job</b></li><li>• <b>Suspend the current job execution plan</b></li><li>• <b>Suspend execution plans of the current and subsequent nodes</b></li></ul>

## 11.2 EL

### 11.2.1 Expression Overview

Node parameter values in a Data Development job can be dynamically generated based on the running environment by using Expression Language (EL). EL uses simple arithmetic and logic to calculate and references embedded objects, including job objects and tool objects.

Job object: provides properties and methods of obtaining the output message, job scheduling plan time, and job execution time of the previous node in a job.

Tool job: Provides methods of operating character strings, time, and JSON. For example, truncating a substring from a string or formatting time.

### Syntax

Expression syntax:

```
#{expr}
```

In the preceding information, **expr** indicates an expression. **#** and **{}** are common operators used in EL, allowing you to access job properties using embedded objects.

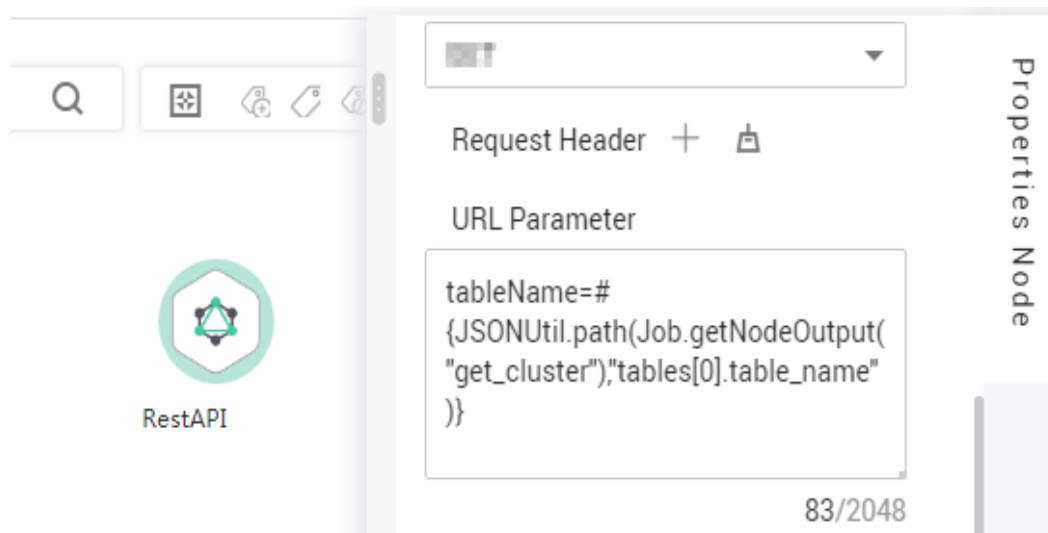
## Example

In the **URL** parameter of the Rest Client node, use expression **tableName=#{JSONUtil.path(Job.getNodeOutput("get\_cluster"),"tables[0].table\_name")}**, as shown in [Figure 11-3](#).

Expression description:

1. **Job.getNodeOutput("get\_cluster")** is used to obtain the execution result of the **get\_cluster** node in the job. The execution result is a JSON character string.
2. **tables[0].table\_name** is used to obtain the value of a field in the JSON character string.

**Figure 11-3** Expression example



## 11.2.2 Basic Operators

EL supports most of the arithmetic and logic operators provided by Java.

### Operator List

**Table 11-71** Basic operators

Operator	Description
.	Accesses a Bean property or a mapping entry.
[]	Accesses an array or linked list.
()	Organizes a subexpression to change priority.

Operator	Description
+	Plus sign
-	Minus or negative sign
*	Multiplication sign
/ or div	Division sign
% or mod	Modulo
== or eq	Test whether equal to.
!= or ne	Test whether unequal to.
< or lt	Test whether less than.
> or gt	Test whether greater than.
<= or le	Check whether less than or equal to.
>= or ge	Test whether greater than or equal to.
&& or and	Test logic and.
or or	Test logic or.
! or not	Test negation.
empty	Test whether empty.
?:	The expression is similar to if else. If the statement in front of ? is true, the value of the expression between ? and : is returned. Otherwise, the value following : is returned.

## Example

If variable a is empty, default is returned. If variable a is not empty, a itself is returned. The expression is as follows:

```
# {empty a?"default":a}
```

## 11.2.3 Date and Time Mode

The date and time in the EL expression can be displayed in the user-specified format. The date and time format is specified by the date and time mode character string. The date and time mode character string consists of letters from A to Z and from a to z, as shown in [Table 11-72](#).

**Table 11-72** Letter description

Letter	Description	Example
G	Epoch	AD

Letter	Description	Example
y	Year	2001
M	Month in a year	July or 07
d	Day in a month	10
h	Hour in the 12-hour clock	12
H	Hour in the 24-hour clock	22
m	Minute	30
s	Second	55
S	Millisecond	234
E	Day of a week	Tuesday
D	Date in the year	360
F	Day in a week of a month	2(second Wed. in July)
w	Week in a year	40
W	Week in a month	1
a	A.M. /P.M.	PM
k	Hour in the 24-hour clock	24
K	Hour in the 12-hour clock	10
z	Time zone	Eastern Standard Time
'	Text delimiter	No example
"	Single quotation mark	No example

## 11.2.4 Env Embedded Objects

The Env embedded object provides a method to obtain an environment variable value.

### Method

**Table 11-73** Method description

Method	Description
String get(String name)	Obtains the value of a specified environment variable.

## Example

The EL expression used to obtain the value of environment variable **test** is as follows:

```
#{Env.get("test")}
```

## 11.2.5 Job Embedded Objects

A job object provides properties and methods of obtaining the output message, job scheduling plan time, and job execution time of the previous node in a job.

### Properties and Methods

**Table 11-74** Property description

Property	Type	Description
name	String	Name of the job.
planTime	java.util.Date	Job scheduling plane time, that is, the time configured for periodic scheduling, for example, to schedule a job at 1:01 a.m. every day.
startTime	java.util.Date	Job execution time. It may be the same as or later than the planTime (because the job engine is busy).
eventData	String	Message obtained from the DIS stream when the event-driven scheduling is used.
projectId	String	ID of the project where DLF is located.

**Table 11-75** Method description

Method	Description
String getNodeStatus(String nodeName)	<p>Obtains the running status of a specified node. If the node runs properly, success is returned. If the node fails to run, fail is returned.</p> <p>For example, the following condition can be used to determine whether a node is successfully executed:</p> <pre><code>#{(Job.getNodeStatus("test")) == "success" }</code></pre>

Method	Description
String getNodeOutput(String nodeName)	Obtains the output of a specified node. This method can only obtain the output of the previous dependent node. Currently, only the output of the Rest Client node, that is, the returned message of the REST request, can be obtained.
String getParam(String key)	Obtains a job parameter.
String getPlanTime(String pattern)	Obtains the plan time character string in a specified pattern. Pattern indicates the date and time mode. For details, see <a href="#">Date and Time Mode</a> .
String getYesterday(String pattern)	Obtains the time character string of the day before the plan time. Pattern indicates the date and time mode. For details, see <a href="#">Date and Time Mode</a> .
String getLastHour(String pattern)	Obtains the time character string of last hour before the plan time. Pattern indicates the date and time mode. For details, see <a href="#">Date and Time Mode</a> .
String getRunningData(String nodeName)	Obtains the data recorded during the running of a specified node. This method can only obtain the output of the previous dependent node. Currently, only the DLI job ID recorded during the running of the DLI SQL node can be obtained. For example, to obtain the job ID in the third statement of DLI node <b>DLI_INSERT_DATA</b> , run the following command: <code>#{JSONUtil.path(Job.getRunningData("DLI_INSERT_DATA"),"jobIds[2]")}</code>
String getInsertJobId(String nodeName)	Returns the job ID in the first DLI Insert SQL statement of the specified DLI SQL or Transform Load node. If the <b>nodeName</b> parameter is not specified, the job ID in the first DLI Insert SQL statement of the DLI SQL node is obtained. If the job ID cannot be obtained, the <b>null</b> value is returned.

## Example

The expression used to obtain the output of node **test** in the job is as follows:

```
#{Job.getNodeOutput("test")}
```

## 11.2.6 StringUtil Embedded Objects

A StringUtil embedded object provides methods of operating character strings. For example, truncating a substring from a character string.

StringUtil is implemented through org.apache.commons.lang3.StringUtils. For details about how to use the object, see the apache commons document.

### Example

If variable a is character string No.0010, the substring after . is returned. The EL expression is as follows:

```
#${StringUtil.substringAfter(a,".")}
```

## 11.2.7 DateUtil Embedded Objects

A DateUtil embedded object provides methods of formatting time and calculating time.

### Methods

**Table 11-76** Method description

Method	Description
String format(Date date, String pattern)	Formats Date to character strings according to the specified pattern.
Date addMonths(Date date, int amount)	After the specified number of months is added to Date, the new Date object is returned. The amount can be a negative number.
Date addDays(Date date, int amount)	After the specified number of days is added to Date, the new Date object is returned. The amount can be a negative number.
Date addHours(Date date, int amount)	After the specified number of hours is added to Date, the new Date object is returned. The amount can be a negative number.
Date addMinutes(Date date, int amount)	After the specified number of minutes is added to Date, the new Date object is returned. The amount can be a negative number.
int getDay(Date date)	Obtains the day from the date. For example, if the date is 2018-09-14, 14 is returned.

Method	Description
int getMonth(Date date)	Obtains the month from the date. For example, if the date is 2018-09-14, 9 is returned.
int getYear(Date date)	Obtains the year from the date. For example, if the date is 2018-09-14, 2018 is returned.
Date now()	Returns the current time.
long getTime(Date date)	Converts the date type to the long type.
Date parseDate(String str, String pattern)	Converts the character string to the date by pattern. The pattern is the date and time mode. For details, see <a href="#">Date and Time Mode</a> .

## Example

The previous day of the job scheduling plan time is used as the subdirectory name to generate an OBS path. The EL expression is as follows:

```
#{"obs://test/"+DateUtil.format(DateUtil.addDays(Job.planTime,-1),"yyyy-MM-dd")}
```

## 11.2.8 JSONUtil Embedded Objects

A JSONUtil embedded object provides JSON object methods.

## Methods

**Table 11-77** Method description

Method	Description
Object parse(String jsonStr)	Converts a JSON character string into an object.
String toString(Object jsonObject)	Converts an object to a JSON character string.
Object path(String jsonStr,String jsonPath)	Returns the field value in a path specified by the JSON character string. This method is similar to XPath and can be used to retrieve or set JSON by path. You can use . or [] in the path to access members and values. For example, tables[0].table_name.

## Example

The content of variable str is as follows:

```
{
  "cities": [{
    "name": "Shenzhen",
    "areaCode": "0755"
  },
  {
    "name": "city2",
    "areaCode": "010"
  },
  {
    "name": "Shanghai",
    "areaCode": "021"
  }
]}
```

The expression for obtaining the area code of Shenzhen is as follows:

```
#JSONUtil.path(str,"cities[0].areaCode")
```

## 11.2.9 Loop Embedded Objects

You can use Loop embedded objects to obtain data from the For Each dataset.

### Property

**Table 11-78** Property description

Property	Type	Description
dataArray	String	Dataset input by the For Each node. It is a two-dimensional array.
current	String	Data row traversed by the For Each node. It is a one-dimensional array.
offset	Int	Current offset of the For Each node, starting from 0. Loop.dataArray[Loop.offset] = Loop.current.

## 11.2.10 Expression Use Example

With this example, you can understand how to use EL expressions in the following applications:

- Using variables in the SQL script of Data Development
- Transferring parameters to SQL script variables?
- Using EL expressions in parameters?

### Context

Use the job orchestration and job scheduling functions to generate daily transaction statistics reports according to transaction details tables.

The data tables involved in this example are as follows:

- **trade\_log**: Records data generated in each transaction.
- **trade\_report**: Is generated based on **trade\_log** and records the daily transaction summary.

## Prerequisites

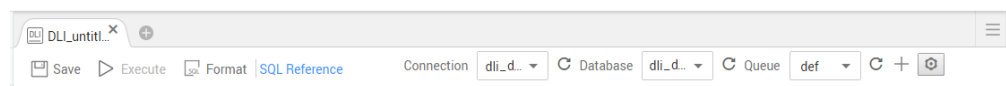
- A DLI data connection named **dli\_demo** has been created.  
If this data connection is not created, create one. For details, see [Creating a Data Connection](#).
- A database named **dli\_db** has been created in DLI.  
If this database is not created, create one. For details, see [Creating a Database](#).
- Data tables **trade\_log** and **trade\_report** have been created in the **dli\_db** database.  
If the data tables are not created, create them. For details, see [Creating a Data Table \(Visualized Mode\)](#) or [Creating a Data Table \(DDL Mode\)](#).

## Procedure

### Step 1 Create and develop a SQL script.


1. In the navigation tree of the Data Development console, choose **Data Development > Develop Script**.
2. Access the area on the right and choose **Create SQL Script > DLI**.
3. Go to the SQL script development page and set the data connection, database, and resource queue on the script property bar.

**Figure 11-4** Property bar



4. Enter the following SQL statements in the script editor:

```
INSERT OVERWRITE TABLE trade_report
SELECT
  sum(trade_count),
  '${yesterday}'
FROM
  trade_log
where
  date_format(trade_time, 'yyyy-MM-dd') = '${yesterday}'
```

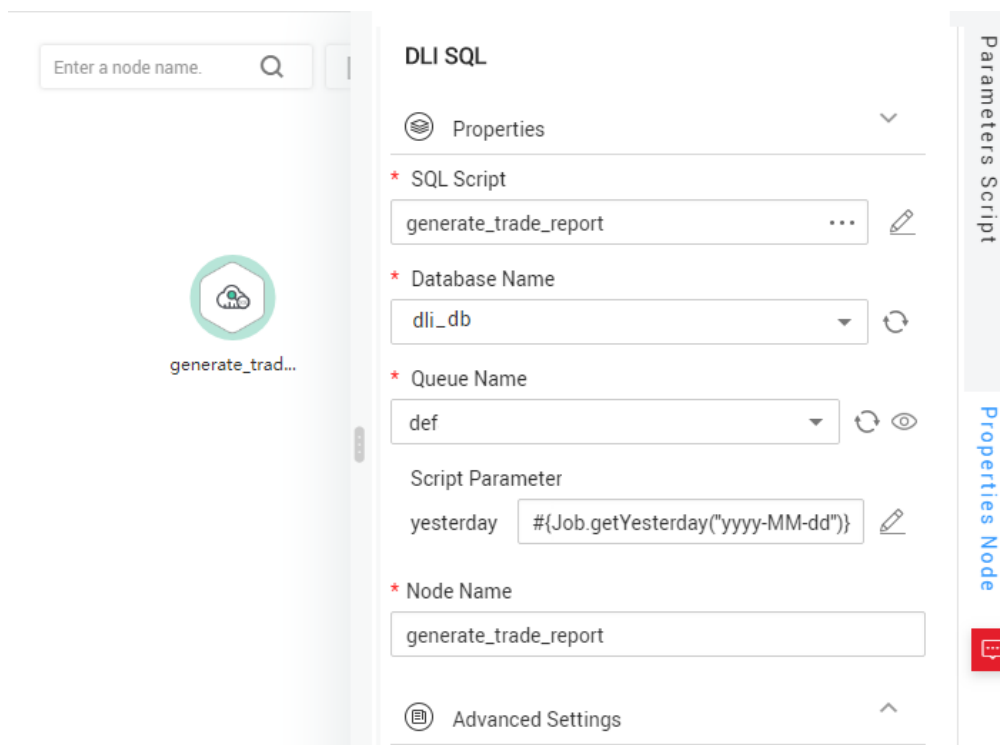
5. Click  and set the script name to **generate\_trade\_report**.

### Step 2 Create and develop a job.

1. In the navigation tree of the Data Development console, choose **Data Development > Develop Job**.
2. Access the area on the right and click **Create Job** to create an empty job named **job**.

**Figure 11-5** Creating a job

3. Go to the job development page, drag the DLI SQL node to the canvas, click the icon, and configure node properties.

**Figure 11-6** Node properties



Description of key properties:

- SQL Script: SQL script **generate\_trade\_report** that is developed in [Step 1](#).
- Database Name: Database configured in SQL script **generate\_trade\_report**.
- Queue Name: Resource queue configured in SQL script **generate\_trade\_report**.
- Script Parameter: Parameter **yesterday** configured in SQL script **generate\_trade\_report**. Enter the following EL expression as the parameter values:  
`#{Job.getYesterday('yyyy-MM-dd')}`

Expression Description: The job object uses the getYesterday method to obtain the time of the day before the job plan execution time. The time format is yyyy-MM-dd.

If the job plan time is 2018/9/26 01:00:00, the calculation result of this expression is 2018-09-25. The calculation result will replace the value of parameter \${yesterday} in the SQL script. The SQL statements after the replacement are as follows:

```
INSERT OVERWRITE TABLE trade_report
SELECT
  sum(trade_count),
  '2018-09-25'
FROM
  trade_log
where
  date_format(trade_time, 'yyyy-MM-dd') = '2018-09-25'
```

4. Click  to test the running job.
5. After the job test is complete, click  to save the job configuration.

----End

# A Change History

Release Date	What's New
2020-08-12	<p>This is the seventh official release.</p> <p>Added the following sections:</p> <ul style="list-style-type: none"><li>• Added how to configure agencies. After an agency is configured, a job can interact with other services as an agency during job execution. For details, see <a href="#">Configuring Agencies</a>.</li><li>• Added the description of the MRS Flink node. For details, see <a href="#">MRS Flink Job</a>.</li><li>• Hided the function description.</li></ul>
2019-09-30	<p>This is the sixth official release.</p> <p>Added the PRESTO SQL script. For details, see <a href="#">Creating a Script</a>.</p>
2019-07-08	<p>This is the fifth official release.</p> <ul style="list-style-type: none"><li>• Added the MRS Python Spark job. For details, see <a href="#">MRS Spark Python</a>.</li><li>• Added lineage settings. For details, see the following sections:<ul style="list-style-type: none"><li>– <a href="#">CDM Job</a></li><li>– <a href="#">Rest Client</a></li><li>– <a href="#">DLI SQL</a></li><li>– <a href="#">DWS SQL</a></li><li>– <a href="#">ETL Job</a></li></ul></li></ul>
2019-06-21	<p>This is the fourth official release.</p> <ul style="list-style-type: none"><li>• Allowed users to modify and save real-time jobs in the running state. For details, see <a href="#">Developing a Job</a>.</li></ul>

Release Date	What's New
2019-05-28	<p>This is the third official release.</p> <p>Added the following sections:</p> <ul style="list-style-type: none"><li>• <a href="#">Renaming a Script</a></li><li>• <a href="#">Renaming a Job</a></li><li>• <a href="#">Specifications</a></li><li>• <a href="#">Env Embedded Objects</a></li></ul> <p>Modified the following sections:</p> <ul style="list-style-type: none"><li>• <a href="#">DLI SQL</a></li><li>• <a href="#">DWS SQL</a></li><li>• <a href="#">Rest Client</a></li><li>• <a href="#">MRS SparkSQL</a></li><li>• <a href="#">MRS Hive SQL</a></li><li>• <a href="#">Shell</a></li><li>• <a href="#">ETL Job</a></li><li>• <a href="#">Job Embedded Objects</a></li></ul>
2019-05-24	<p>This is the second official release.</p> <p>Added the following section:</p> <ul style="list-style-type: none"><li>• <a href="#">Exporting a Data Connection</a></li><li>• <a href="#">Importing a Data Connection</a></li><li>• <a href="#">Cycle Overview</a></li><li>• <a href="#">Backing Up and Restoring Assets</a></li></ul> <p>Modified the following sections:</p> <ul style="list-style-type: none"><li>• <a href="#">Exporting and Importing a Job</a></li><li>• <a href="#">ETL Job</a></li><li>• <a href="#">DIS Client</a></li></ul>
2019-04-28	<p>This is the first official release.</p>