

Alibaba Cloud DataWorks

User Guide

Issue: 20190117

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.









1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectu

al property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	It is used for commands.	Run the <code>cd /d C:/windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is a optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	It indicates that it is a required value, and only one item can be selected.	<code>swich {stand / slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Console guide.....	1
1.1 Overview of console.....	1
1.2 Project list.....	2
1.3 Scheduling resource list.....	10
1.4 Calculation engine list.....	12
2 Data integration.....	13
2.1 Data integration introduction.....	13
2.1.1 Data Integration Overview.....	13
2.1.2 Basic Terms.....	16
2.1.3 Billing FAQ.....	16
2.2 Data source configuration.....	18
2.2.1 Supported data sources.....	18
2.2.2 Data source testing connectivity.....	21
2.2.3 Configuring SQL server data source.....	27
2.2.4 Configure MongoDB data source.....	33
2.2.5 DataHub data source.....	37
2.2.6 Configure the DM data source.....	40
2.2.7 Configure DRDS data sources.....	43
2.2.8 Configure the FTP data source.....	46
2.2.9 Configuring HDFS data source.....	49
2.2.10 Add LogHub data source.....	51
2.2.11 Configure MaxCompute data source.....	54
2.2.12 Configure Memcache data source.....	57
2.2.13 Configure MySQL data source.....	59
2.2.14 Configure Oracle data source.....	65
2.2.15 Configure OSS data source.....	69
2.2.16 Configure Table Store(OTS) data source.....	72
2.2.17 Configure PostgreSQL data source.....	75
2.2.18 Configure Redis data source.....	80
2.3 Task Configuration.....	84
2.3.1 Data Synchronization task configuration.....	84
2.3.2 Configure Reader plug-in.....	84
2.3.2.1 Script mode configuration.....	84
2.3.2.2 Wizard mode configuration.....	91
2.3.2.3 Configure DRDS Reader.....	96
2.3.2.4 Configure HBase Reader.....	103
2.3.2.5 Configuring HDFS Reader.....	110
2.3.2.6 Configure MaxCompute Reader.....	120

2.3.2.7 Configure MongoDB Reader.....	126
2.3.2.8 Configure DB2 Reader.....	129
2.3.2.9 Configure MySQL Reader.....	134
2.3.2.10 Configure Oracle Reader.....	141
2.3.2.11 Configure OSS Reader.....	149
2.3.2.12 Configuring FTP Reader.....	156
2.3.2.13 Configure Table Store(OTS) Reader.....	163
2.3.2.14 Configuring PostgreSQL Reader.....	169
2.3.2.15 Configuring SQL server Reader.....	177
2.3.2.16 Configure LogHub Reader.....	185
2.3.2.17 Configure OTSReader-Internal.....	191
2.3.2.18 Configure OTSStream Reader.....	199
2.3.2.19 Configure RDBMS Reader.....	205
2.3.2.20 Configure Stream Reader.....	212
2.3.3 Configure Writer plug-in.....	214
2.3.3.1 Configure DataHub Writer.....	214
2.3.3.2 Configure DB2 writer.....	217
2.3.3.3 Configure DRDS Writer.....	220
2.3.3.4 Configure FTP Writer.....	225
2.3.3.5 Configure HBase Writer.....	230
2.3.3.6 Configure HBase11xsql Writer.....	236
2.3.3.7 Configure HDFS Writer.....	239
2.3.3.8 Configure MaxCompute Writer.....	246
2.3.3.9 Configure Memcache (OCS) Writer.....	252
2.3.3.10 Configure MongoDB Writer.....	256
2.3.3.11 Configure MySQL Writer.....	259
2.3.3.12 Configuring Oracle Writer.....	264
2.3.3.13 Configure OSS Writer.....	268
2.3.3.14 Configure PostgreSQL Writer.....	274
2.3.3.15 Configure Redis Writer.....	278
2.3.3.16 Configure SQL Server Writer.....	286
2.3.3.17 Configure ElasticSearch Writer.....	291
2.3.3.18 Configure LogHub Writer.....	295
2.3.3.19 Configure OpenSearch Writer.....	297
2.3.3.20 Configure Table Store (OTS) Writer.....	301
2.3.3.21 Configure RDBMS Writer.....	305
2.3.3.22 Configure Stream Writer.....	309
2.3.4 Optimizing configuration.....	311
2.4 Common configuration.....	316
2.4.1 Add security group.....	316
2.4.2 Add whitelist.....	317
2.4.3 Add scheduling resources.....	320
2.5 Metadata Collection.....	324
2.5.1 Overview of metadata collection.....	324
2.5.2 Metadata Collection.....	324

2.6 Full-database migration.....	327
2.6.1 Full-database migration overview.....	327
2.6.2 Configure MySQL full-database migration.....	329
2.6.3 Configure Oracle full-database migration.....	331
2.7 Bulk Sync.....	333
2.7.1 Bulk Sync.....	333
2.7.2 Add data sources in Bulk Mode.....	336
2.8 Best practice.....	337
2.8.1 Data integration when the network of data source (one side only) is disconnected.....	337
2.8.2 Data sync when the network of data source (both sides) is disconnected.....	342
2.8.3 Data increase synchronization.....	348
2.8.4 Import data into Elasticsearch using Data Integration.....	352
2.8.5 Use Data Integration to ship log data collected by LogHub.....	356
2.8.6 Import data into DataHub using Data Integration.....	363
2.8.7 Configure OTSStream data synchronization tasks.....	366
2.9 FAQ.....	371
2.9.1 How to troubleshoot data integration problems?.....	371
2.9.2 Synchronous task waiting for slots.....	387
2.9.3 RDS synchronization failure converted to JDBC format.....	388
2.9.4 Synchronous table column name is a key and task fails.....	389
2.9.5 How does the data synchronization task customize the table name?.....	389
2.9.6 Encoding formatting issues.....	390
2.9.7 Full-database migration data type.....	392
2.9.8 An error occurred when using username root to add MongoDB data source..	392
3 Data development.....	393
3.1 Solution.....	393
3.2 Encoding principles and standards for the SQL code.....	395
3.3 Console functions.....	400
3.3.1 Introduction to console.....	400
3.3.2 Version.....	402
3.3.3 Structure.....	403
3.3.4 Relationship.....	406
3.4 Business flow.....	407
3.4.1 Business flow.....	407
3.4.2 Resource.....	412
3.4.3 Register the UDFs.....	415
3.5 Node type.....	417
3.5.1 Node type overview.....	417
3.5.2 Data integration node.....	419
3.5.3 ODPS SQL node.....	419
3.5.4 ODPS MR node.....	422
3.5.5 PyODPS node.....	427
3.5.6 SHELL node.....	431
3.5.7 SQL Component node.....	433

3.5.8 Virtual node.....	438
3.5.9 Assignment node.....	440
3.5.10 Branch node.....	445
3.5.11 Merge node.....	450
3.6 Scheduling Configuration.....	454
3.6.1 Basic attributes.....	454
3.6.2 Parameter configuration.....	455
3.6.3 Time attributes.....	463
3.6.4 Dependencies.....	471
3.6.5 Resource type.....	487
3.6.6 Node Context.....	487
3.7 Configuration management.....	492
3.7.1 Overview of configuration management	492
3.7.2 Configuration center.....	493
3.7.3 Project configuration.....	497
3.7.4 Templates.....	498
3.7.5 Theme management.....	498
3.7.6 Table Levels.....	499
3.8 Publish management.....	499
3.8.1 Publish a task.....	499
3.8.2 Cross-project cloning.....	502
3.9 Manual business flow.....	503
3.9.1 Manual Business Flow Introduction.....	503
3.9.2 Resource.....	504
3.9.3 Function.....	508
3.9.4 Table.....	510
3.10 Manual task node type.....	515
3.10.1 ODPS SQL node.....	515
3.10.2 PyODPS node.....	517
3.10.3 Manual data intergration node.....	520
3.10.4 ODPS MR node.....	524
3.10.5 SQL component node.....	529
3.10.6 Virtual node.....	534
3.10.7 SHELL Node.....	536
3.11 Manual task parameter settings.....	538
3.11.1 Basic Attributes.....	538
3.11.2 Configure manual node parameters.....	540
3.12 Component management.....	546
3.12.1 Create components.....	546
3.12.2 Use components.....	553
3.13 Queries.....	554
3.14 Running log.....	557
3.15 Public Tables.....	558
3.16 Table Management.....	560
3.17 Functions.....	565

3.18 Recycle Bin.....	566
3.19 Editor shortcut list.....	567
4 Operation center.....	570
4.1 Operation center overview.....	570
4.2 O&M overview.....	571
4.3 Task list.....	573
4.3.1 Cyclic task.....	573
4.3.2 Manual task.....	576
4.4 Task O&M.....	578
4.4.1 Cycle instance.....	578
4.4.2 Manual instance.....	582
4.4.3 PatchData.....	583
4.4.4 Testing instances.....	586
4.5 Alarm.....	590
4.5.1 Alarm overview.....	590
4.5.2 Function introduction.....	591
4.5.2.1 Baseline alarm and Event warning.....	591
4.5.2.2 Custom notifications.....	594
4.5.2.3 Other functions.....	595
4.5.3 User guide.....	596
4.5.3.1 Baseline management and baseline instance.....	596
4.5.3.2 Event Management.....	599
4.5.3.3 Rule Management.....	600
4.5.3.4 Alarm info.....	601
4.5.4 Intelligent monitor FAQ.....	602
4.5.4.1 Why did my alarm report to someone else?.....	602
4.5.4.2 Task is not important and I do not want to receive alarm. What should I do?.....	603
4.5.4.3 Baseline is broken. Why not call the alarm?.....	603
4.5.4.4 My task is slowing down but I don't want to receive an alarm.....	603
4.5.4.5 Why is the task wrong but I didn't receive an alarm?.....	603
4.5.4.6 What should I do when receiving an alarm at night?.....	603
5 Project management.....	604
5.1 Project configuration.....	604
5.2 User management.....	605
5.3 Permission list.....	607
5.4 Project mode upgrade.....	613
6 Data quality.....	617
6.1 Data quality overview.....	617
6.2 Prerequisites.....	618
6.2.1 Prepare your data.....	618
6.2.2 Establish DQC.....	619
6.3 Overview.....	619
6.4 My subscription.....	620
6.5 Rule Configuration.....	621

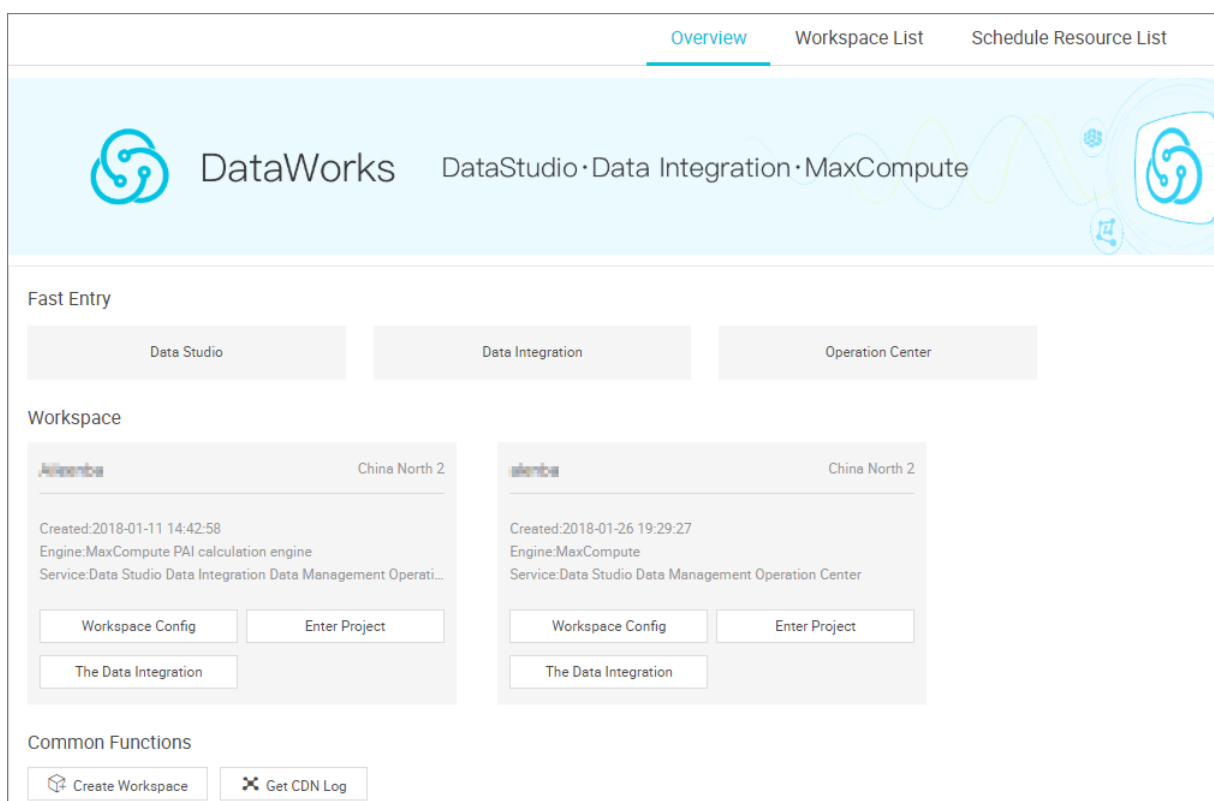
6.5.1 Rules configuration for DataHub data source.....	621
6.5.2 Rules Configuration for ODPS data source.....	622
6.6 Mission Inquiries.....	630
6.6.1 Viewing DataHub data source tasks.....	630
6.6.2 View ODPS data source tasks.....	631
6.7 Template rule.....	632
7 Data management.....	638
7.1 Introduction.....	638
7.2 Overview.....	638
7.3 All data.....	640
7.4 Table detail page.....	640
7.5 Apply for data permissions.....	646
7.6 Table management.....	649
7.7 Create a table.....	654
7.8 Permission management.....	659
7.9 Manage config.....	660
8 DataService studio.....	662
8.1 DataService studio overview.....	662
8.2 Glossary.....	663
8.3 Generate API.....	664
8.3.1 Configure the Data Source.....	664
8.3.2 Overview of generating API.....	664
8.3.3 Generate API in Wizard Mode.....	665
8.3.4 Generate API in Script Mode.....	670
8.4 Register API.....	676
8.5 API service test.....	679
8.6 Publish an API.....	680
8.7 Delete API.....	681
8.8 Call an API.....	682
8.9 FAQ.....	683
9 Data security guard.....	685
9.1 Enter Data Security Guard.....	685
9.2 Data distribution.....	686
9.3 Access analysis.....	687
9.4 Data risks.....	689
9.5 Audit.....	689
9.6 Rule setting.....	690
9.7 Classification management.....	691
9.8 Manual ajust.....	692
9.9 Risk Mgmt.....	693
10 MaxCompute manager.....	695
10.1 MaxCompute Manager.....	695

1 Console guide

1.1 Overview of console

You can view the recently used projects on the Overview page, and enter the work zone to configure a project, create a project, and one-click to import CDNs.

Log on to the [DataWorks console](#) page as an organization administrator (main account).



Note:

The overview page updates the display based on your usage and creation time, displays only the three projects that you recently used or created.

Page description:

- **Project**

It displays the three projects opened most recently. You can work with the project by clicking **Config** or **Data Studio**. Alternatively, you can also access the **Project List** to do so. For more information, see [Project list](#).

- **Common functions**

- You can [Create a project](#) on this page.
- You can also One-click to import CDNs on this page.



Note:

- If the sub-account is logged in without creating the corresponding project, you are prompted to contact your administrator, open project permissions.
- The sub-account displays up to two projects, and you can go to the **Project List** page to view all the projects.
- If the sub-account only has deployment privilege, you cannot enter the workspace.
- You can update your AK info [here](#).

1.2 Project list

In the Alibaba Cloud DTplus console, you can view all the projects under the current account of the **Project List** page. Enter project to configure projects, change the calculation services, create, activate, disable, and delete projects.

Procedure

1. Log on to the DTplus console and go to [DataWorks](#) product details page as an organization administrator (primary account).
2. Click **DataWorks console** to enter the console overview page.
3. Navigate to the **Project List** page to view all the projects under the current account.

Overview Workspace List Schedule Resource List						
<div> <div>China North 2</div> <div>China East 1</div> <div>China East 2</div> <div>China South 1</div> <div>Hong Kong</div> <div>US West 1</div> <div>Asia Pacific SE 1</div> <div>US East 1</div> <div>EU Central 1</div> <div>Asia Pacific SE 2</div> <div>Asia Pacific SE 3</div> <div>Asia Pacific NE 1</div> <div>Middle East 1</div> <div>Asia Pacific SOU 1</div> <div>Asia Pacific SE 5</div> <div>UK</div> </div> <div>Create Workspace Refresh</div>						
<div>Search</div>						
Workspace Name/Display Name	Workspace Mode	Create Time	Administrator	Status	Subscribed Service	Operation
alibaba	Simple Mode (Single Environment)	Jan 26, 2018, 19:29:27	longgalle@alibaba	Normal		Workspace Config Enter Project Modify Service More
alibaba	Simple Mode (Single Environment)	Jan 11, 2018, 14:42:58	longgalle@alibaba	Normal		Workspace Config Enter Project Modify Service The Data Integration More

Create Project

1. Click **Create Project**, select a region and a calculation engine service.

The new project is created under the current region. You may need to purchase related services for the region. The data development, O&M center, and data management are selected by default.

Create Workspace

Select region

China North 2

China East 1

China East 2

China South 1

Hong Kong

US West 1

Asia Pacific SE 1

US East 1

EU Central 1

Asia Pacific SE 2

Asia Pacific SE 3

Asia Pacific NE 1

Middle East 1

Asia Pacific SOU 1

Asia Pacific SE 5

UK

Choose Calculation Engine Services

☐

MaxCompute

☐ Pay-As-You-Go
 ☐ Subscription
 [Go Buy](#)

After opening, you can develop MaxCompute SQL, MaxCompute MR tasks in DataWorks.

☐

Machine learning

☐ Pay-As-You-Go
 [Go Buy](#)

After opening, you can use machine learning algorithms, deep learning frameworks, and online forecasting services. PAI using machine learning, you need to use MaxCompute

Choose DataWorks Service

☐

Data Integration

☐ Pay-As-You-Go
 [Go Buy](#)

After opening, you can develop data integration tasks in DataWorks and quickly implement data synchronization among more than 20 data sources.

☒

Data Development, O&M Center, Data Management

You can schedule workflows, schedule tasks, query information and permissions for all

Cancel

Next Step

- Choose Calculation Engine Services
 - MaxCompute: MaxCompute is a big data processing platform developed by Alibaba independently. It is mainly used for batch structural data storage and processing, which can provide massive data warehouse solution and big data modeling service. For more information, see [MaxCompute documentation](#).
 - Machine learning PAI: Machine learning refers to a machine that uses statistical algorithms to learn a large amount of historical data to generate empirical models, and use empirical models to guide businesses.
- Choose DataWorks services

Issue: 20190117

3

- Data integration: A data synchronization platform that provides stable, efficient, and elastically scalable services. The Data Integration is designed to implement fast and stable data movement and synchronization between multiple heterogeneous data sources in complex network environments. For more information, see [Data Integration Overview](#).
 - Data development: The data development helps you to design data computing processes according to your business demands and make mutually dependent tasks be automatically run in the scheduling system. For more information, see [Data Development Overview](#).
 - O&M center: The O&M center is a place where tasks and instances are displayed and operated. You can view all your tasks in Task List and perform such operations on the displayed tasks. For more information, see [Operation center overview](#).
 - Data management: The data management module of the Alibaba Cloud DTplus platform displays the global data view and metadata details of an organization, and enables operations such as divided permission management, data lifecycle management, and approval and management of data table/resource/function permissions. For more information, see [data management overview](#).
2. Configure the basic information and advanced settings for the new project.

Create Workspace

Basic Information

* Workspace Name :

Display Name :

* Workspace Mode :

Simple Mode (Single Environment)

Workspace Description :

Advanced Settings

* Enable Scheduling Frequency :

on

* Download Select Result :

on

For MaxCompute

* MaxCompute Project Name :

ailin

* MaxCompute Access Identity:

Workspace Owner

* Quota Group:

Pay per view default resource group

Previous

Create Workspace

- Basic configuration
 - project name: The length of the project name is between 3 and 27 characters.
 - display name: The length of the display name is not more than 27 characters.
- Advanced configuration
 - Enable scheduling frequency: Control the current project whether to enable or disable the scheduling system, and if it is disabled, it can not periodically schedule tasks.
 - Enable select result downloads in this project: Whether data results from select statement can be downloaded in this project, and if it is disabled, it cannot download the data query results from select statement.
- MaxCompute configuration

- Development Environment Maxcompute Project name: the default is the project name + "_dev" suffix, which can be modified.
- Development Environment Maxcompute access identity: default is a personal account.
- Production Environment Maxcompute Project name: the default name is the same as the Dataworks project.
- Development Environment Maxcompute access identity: the default is the production account, it is recommended not to change.
- Quota group: Quota is used to implement disk quotas.

When the project is created successfully, the project list displays the corresponding content.

Overview <u>Workspace List</u> Schedule Resource List						
<div> <div>China North 2</div> <div>China East 1</div> <div>China East 2</div> <div>China South 1</div> <div>Hong Kong</div> <div>US West 1</div> <div>Asia Pacific SE 1</div> <div>US East 1</div> <div>EU Central 1</div> <div>Asia Pacific SE 2</div> <div>Asia Pacific SE 3</div> </div> <div> <div>Asia Pacific NE 1</div> <div>Middle East 1</div> <div>Asia Pacific SOU 1</div> <div>Asia Pacific SE 5</div> <div>UK</div> </div> <div>Create Workspace Refresh</div>						
<input type="text"/> <input type="button" value="Search"/>						
Workspace Name/Display Name	Workspace Mode	Create Time	Administrator	Status	Subscribed Service	Operation
workspace	Simple Mode (Single Environment)	Jan 26, 2018, 19:29:27	longgeline388	Normal		Workspace Config Enter Project Modify Service More
workspace	Simple Mode (Single Environment)	Jan 11, 2018, 14:42:58	longgeline359	Normal		Workspace Config Enter Project Modify Service The Data Integration More

- Project Status: the project is generally divided into normal, initialization, initialization failure, deleting and deleted five states. Creating a project is initially display initialized state, and then generally show the results of initialization failure or normal.

After the project is created successfully, you can perform the disable and delete operations. After the project is disabled, you can also activate and delete the project, and the project is normal after it is activated.

- Subscribe to a service: Your mouse moves on to the service, and all the services you have opened are displayed. Generally, the normal service icon displays blue, the outstanding payment service icon is red and there is corresponding outstanding payment sign, if the outstanding payment service have been deleted, it is displayed in gray, and the outstanding payment service is deleted automatically after 7 days.



Note:

- Once you become a project owner, it means that everything in the project is yours, and no one has permission to access your project before the authentication.
- For general users, it is not necessary to create a project. If you are added to a project, you can use the MaxCompute.

Configure a project

You can configure some basic and advanced attributes of the current project by configuring project operations, mainly manages and configures space, scheduling, and so on.

Click **Configuration** for the project to be configured.

The screenshot shows the 'Workspace Config' dialog box with a close button (X) in the top right corner. It is divided into two main sections: 'Basic Information' and 'Advanced Settings'.

Basic Information

- * Workspace Name :** alenba
- Display Name :** alenba (with an edit icon)
- * Workspace Mode :** Simple Mode (Single Environment)
- Workspace Description :** alenba (with an edit icon)

Advanced Settings

[More Settings](#)

- * Enable Scheduling Frequency :** on (with a help icon ?)
- * Download Select Result :** on (with a help icon ?)

For MaxCompute

- * MaxCompute Project Name: :** alenba (with a help icon ?)
- * MaxCompute Access Identity:** Personal (with an edit icon and a help icon ?)
- * Quota Group:** Pay per view default resource group (with an edit icon)

Enter Project

Click **Enter Project** to configure a project, go to the Data Development page for specific operations.

Overview <u>Workspace List</u> Schedule Resource List						
<div> <div>China North 2</div> <div>China East 1</div> <div>China East 2</div> <div>China South 1</div> <div>Hong Kong</div> <div>US West 1</div> <div>Asia Pacific SE 1</div> <div>US East 1</div> <div>EU Central 1</div> <div>Asia Pacific SE 2</div> <div>Asia Pacific SE 3</div> </div> <div> <div>Asia Pacific NE 1</div> <div>Middle East 1</div> <div>Asia Pacific SOU 1</div> <div>Asia Pacific SE 5</div> <div>UK</div> </div> <div>Create Workspace</div> <div>Refresh</div>						
<div> <div></div> <div>Search</div> </div>						
Workspace Name/Display Name	Workspace Mode	Create Time	Administrator	Status	Subscribed Service	Operation
	Simple Mode (Single Environment)	Jan 26, 2018, 19:29:27	longgalin388	Normal		Workspace Config Enter Project Modify Service More
	Simple Mode (Single Environment)	Jan 11, 2018, 14:42:58	longgalin359	Normal		Workspace Config Enter Project Modify Service The Data Integration More

Change the calculation services

Changing services is generally the operation of calculation services and DataWorks services. First, you must purchase a service, and then you can choose a corresponding service to modify it. The mode of payment is automatically displayed based on your purchase. You can recharge, upgrade, downgrade, and renew your MaxCompute.

Modify service

Choose Calculation Engine Services

☒ MaxCompute
 ☒ Pay-As-You-Go
 ☐ Subscription

After opening, you can develop MaxCompute SQL, MaxCompute MR tasks in DataWorks.

Recharge

Renew

upgrade

re-allocation

☐ Machine Learning Platform For AI
 ☐ Pay-As-You-Go [Go buy](#)

Machine Learning Platform For AI requires MaxCompute. This enables machine learning algorithms, deep learning frameworks, and online prediction services.

Choose DataWorks service

☒ Data Integration
 ☒ Pay-As-You-Go

After opening, you can develop data integration tasks in DataWorks and quickly implement data synchronization among more than 20 data sources.

☒ Data Development, O&M Center, Data Management

You can schedule workflows, schedule tasks, query information and permissions for all tables, and services are currently in open beta.

Cancel

submit

8

Issue: 20190117

- **Recharge:** You can recharge your services when the services receive an overdue warning.
- **Upgrade/Downgrade:** If your Pay-As-You-Go resource of MaxCompute is unable to meet your business demand, you can upgrade the resource by purchasing more services. to upgrade the resource.
- **Renew:** You can renew your package when the package expired, or the system freezes the corresponding instances that contained in this package. For more information, see [Renewal Management](#).

**Note:**

- **Subscription:** Only display the Recharge button.
- **Pay-As-You-Go:** All buttons are displayed.

Delete or disable a project

Click **More** after the corresponding item name to delete and disable the item.

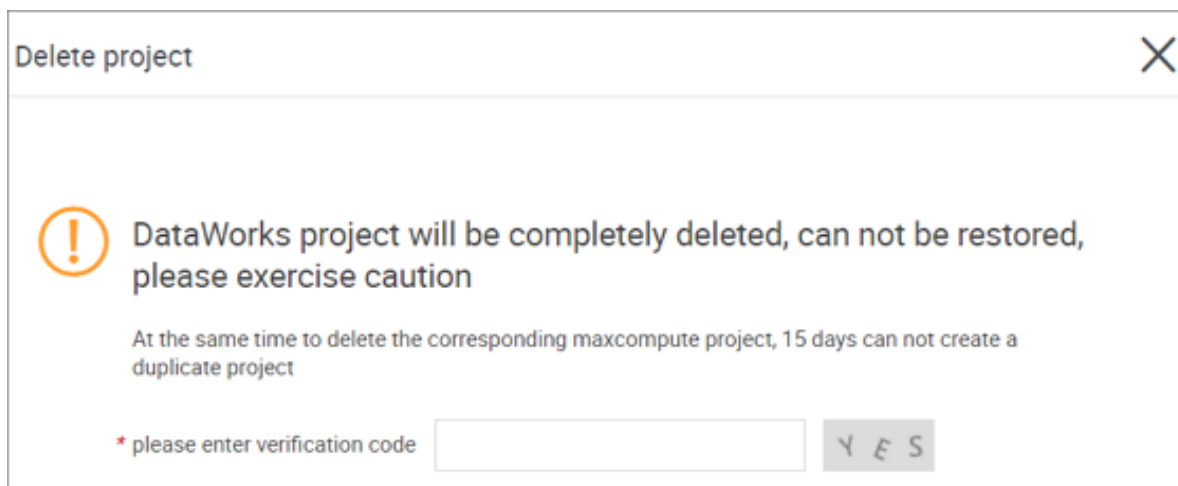
Overview Project List Schedule Resource List						
<div> <div>China East 2</div> <div>China North 2</div> <div>China East 1</div> <div>China South 1</div> <div>Hong Kong</div> <div>US West 1</div> <div>Asia Pacific SE 1</div> <div>US East 1</div> <div>EU Central 1</div> <div>Asia Pacific SE 2</div> <div>Asia Pacific SE 3</div> <div>Asia Pacific NE 1</div> </div> <div> <div>Middle East 1</div> <div>Asia Pacific SOU 1</div> <div>Asia Pacific SE 5</div> </div> <div>Create Project</div> <div>Refresh</div>						
<div> <div></div> <div>Search</div> </div>						
Project / display name	Project mode	Create time	administrator	status	Subscribed service	operation
DataWorks_DOC DataWorks_DOC	In simple mode (single environment)	2018-09-27 13:32:17		normal	Co- V	Config Data Studio Modify service Data Integration More
workshop_0820 DataWorks	Standard mode (development with the production isolation)	2018-09-20 19:27:18		normal	Co- V	Config Data Studio Data Integration <div>Delete project Disable project</div>

- **Delete a project**

After selecting **Delete Project**, fill in the verification code in the dialog box and click **confirm**.

**Note:**

- The verification code is not changed.
- The delete project operation is irreversible, use it carefully.

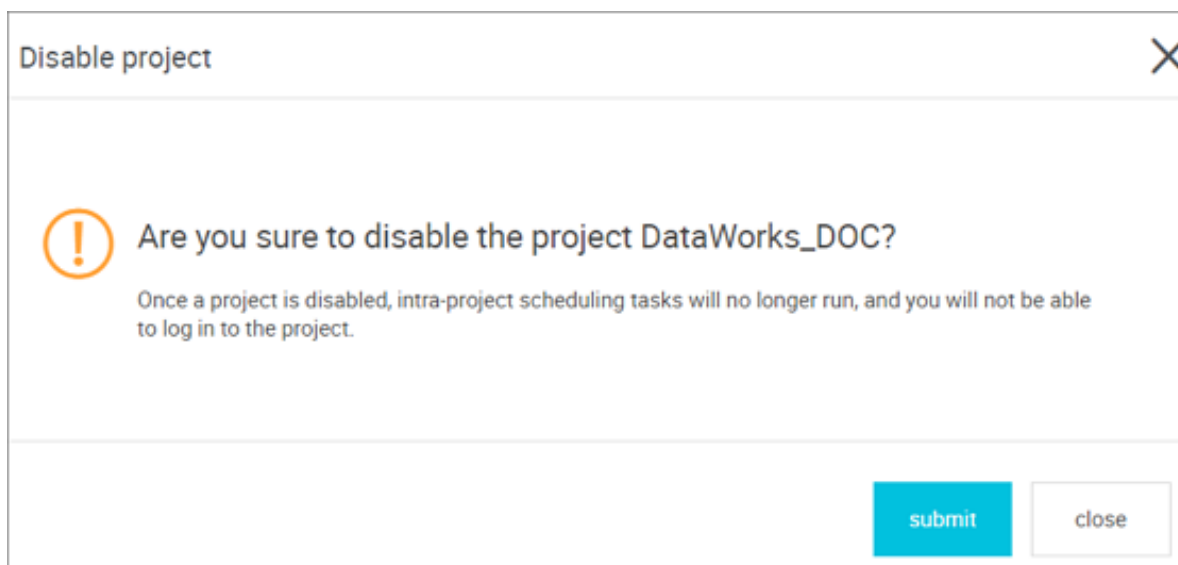


- Disable a project

Once a project is disabled, the cycle scheduling task in the project stops generating instances.

The instances which are generated before the status changes to disabled, run normally.

However you cannot log on to the project to view their corresponding status.



1.3 Scheduling resource list

On the DataWorks console, you can view all the scheduling resources under the current account on the **Scheduling Resource List** page. On this page, you can create scheduling resources, search for a resource by entering its name, and can also perform operations on the expected resource.

Procedure

1. Log on to the [DataWorks](#) product details page as an organization administrator (main account).
2. Click **DataWorks console** to enter the console overview page.

3. Navigate to the **Scheduling Resource List** page.

- Description of the items listed on this page is as follows:
 - Resource name: The name of the scheduling resource group, which consists of letters (a-z), underscores(_), and numbers (0-9) with a length no greater than 60 characters. Once created, the name cannot be changed.
 - Network type: The network type used by the ECS server is added as a scheduling resource. The types include VPC and classic networks.
 - Classic network: IP addresses are centrally allocated by Alibaba Cloud. Classic networks are easy to configure and use. This network type is suitable for users who demand quick accessibility to ECS and emphasize on easy and convenient operations.
 - VPC: A VPC is a logically isolated private network. Network topologies and IP addresses can be customized. VPC supports private line connection and is suitable for users who are familiar with network management.
 - Server: The name of the server contained in the current scheduling resource.
 - Operation type:
 - Initialize the server: Enter a machine initialization statement as prompted.
 - Modify the server: Modify server configurations of the current scheduling resource, such as adding or deleting a server and changing the maximum number of concurrent server tasks.
 - Modify owner project: You can allocate the current scheduling resource to a specific project. This operation can only be performed by the main account that activated the service. After creating the project, you can use an existing ECS by modifying the owner project.
- Add scheduling resources: For more information, see [Add scheduling resources](#).

What are scheduling resources?

Scheduling resources are used to perform or distribute the tasks from the scheduling system. The scheduling resources of DataWorks are divided into the following two types.

- Default scheduling resource.
- Custom scheduling resource.

Custom scheduling resources are the user-purchased ECSs, which can be configured as scheduling servers for performing distributed tasks. The organization administrator (main

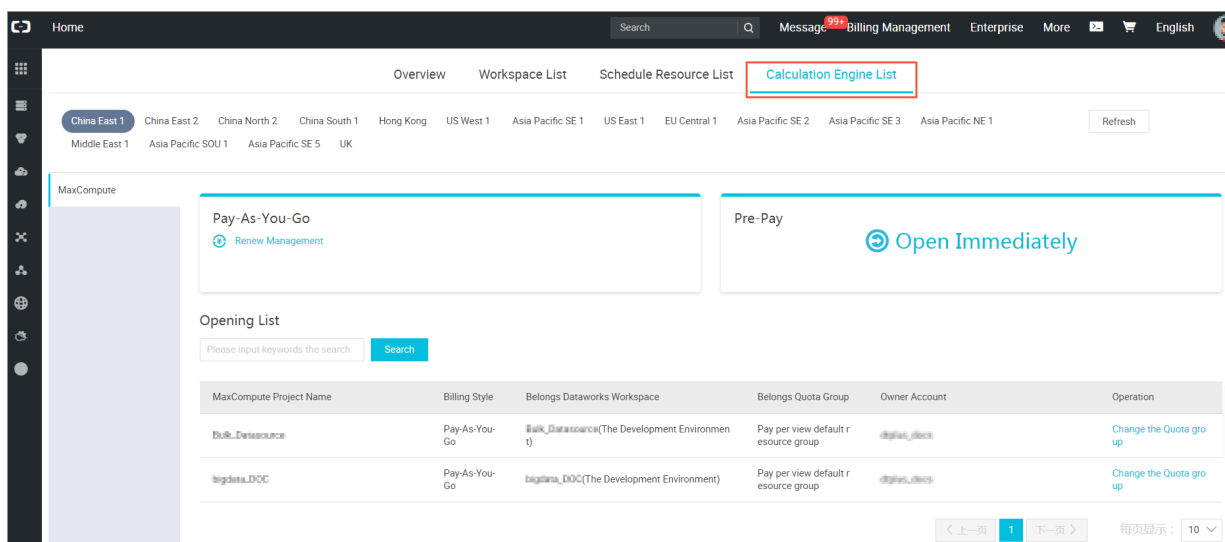
account) can create custom scheduling resources, which contain several physical machines or ECSs to perform data synchronization, SHELL, ODPS_SQL, and OPEN_MR tasks.

Usages of scheduling resource list

- Add resource groups and resource group servers.
- Manage the relationship between resource groups and projects so that a resource group can be shared by multiple projects.
- You can purchase ECSs and configure them as scheduling resources when a number of tasks are waiting for resources, which improves the efficiency in running scheduling tasks.

1.4 Calculation engine list

You can view the billing style and opening list for the MaxCompute project space through **Calculation Engine List** page in the Management Console.



- MaxCompute currently supports two billing styles: **Pay-As-You-Go** and **Pre-Pay**. **Renew Management** will be displayed under the opened billing style, while **Open Immediately** will be shown under the unopened billing style.
- **Opening List**: You can search through your project space name, the project space list displays basic information about the project space.

You can **Change the Quota group**. But only for pre-paid project space, click **Change the Quota group**, will jump to the MaxCompute Manager page; if you do not open pre-pay, you will be prompted that **You have no subscribed resources**.

2 Data integration

2.1 Data integration introduction

2.1.1 Data Integration Overview

The Alibaba Group offers Data Integration - a data synchronization platform that provides stable, efficient, and elastically scalable services. The Data Integration is designed to implement fast and stable data movement and synchronization between multiple heterogeneous data sources in complex network environments.

Offline (batch) data synchronization

The offline (batch) data channel provides a set of abstract data extraction plug-ins (Readers) and data writing plug-ins (Writers) by defining the source and target databases and data sets. Also, it designs a set of simplified intermediate data transmission formats based on the framework to transfer the data between any structured and semi-structured data sources.

Supported data source types

Data Integration provides extensive options for data sources shown as follows:

- Text storage (FTP/SFTP/OSS/Multimedia files),
- Database (RDS/DRDS/MySQL/PostgreSQL),
- NoSQL (Memcache/Redis/MongoDB/HBase),
- Big data (MaxCompute/AnalyticDB/HDFS),
- MPP database (HybridDB for MySQL).

For more information, see [Supported data sources](#).

**Note:**

The configuration information of different data sources varies dramatically from each other, and the parameter configuration information needs to be queried in detail based on the actual use case. For this reason, detailed description of parameters is available on the data source configuration and job configuration pages, which can be queried and used as needed.

Description of synchronization development

Synchronous development provides both the wizard mode and the script mode.

- **Wizard:** Provides wizard-like visualized development guidance and comprehensive details about configuration of data sync tasks. This mode is cost-effective but lacks certain advanced functions.
- **Script:** Allows you to directly write a data sync JSON script to complete data sync development. It is suitable for the advanced users but incurs a high learning cost. It also provides a rich set of flexible functions for refined configuration management.

**Note:**

- The code generated in the wizard mode can be converted to the script mode code. The conversion is unidirectional, and the code cannot be converted back to the wizard mode code. This is because that the capabilities of the script mode are a superset of those of the wizard mode.
- Always configure the data source and create the target table before writing codes.

Description of network types

Networks can be classified as the classic network, VPC, and local IDC network (planning).

- **Classic network:** A network that is centrally deployed on the Alibaba Cloud public infrastructure network and planned and managed by Alibaba Cloud. This type of network suits for customers with demanding ease-of-use requirements.
- **VPC:** An isolated network environment created based on Alibaba Cloud. For this type of network, you have full control over your virtual network, including customizing the IP address range, partitioning network segments, and configuring routing tables and gateways.
- **Local IDC network:** The network environment of your server room, which is isolated from the Alibaba Cloud network.

See classic network and VPC Frequently Asked Questions for questions related to [classic and proprietary networks](#).

Note:

- Public network access is supported - only select the classic network as the network type. Note the speed of the public network bandwidth and relevant network traffic charges when using this type of network. It is not recommended except for special cases.
- Network connections are planned for data synchronization, you can use the locally added resource + Script Mode scheme for synchronous data transfer, you can also use the shell + datax scheme.

- The Virtual Private Cloud (VPC) creates an isolated network environment and allows you to customize the IP address range, network segments, and gateways. VPC applications expand as the VPC security improves, and thus Data Integration provides RDS for MySQL, RDS for SQL Server, and RDS for PostgreSQL and eliminates the need to purchase extra ECSs that reside on the same network as the VPC. Instead, the system guarantees interconnectivity by detecting devices automatically through the reverse proxy. The support for other Alibaba Cloud databases including PPAS, OceanBase, Redis, MongoDB, Memcache, TableStore, and HBase is also be available in the future. For any non-RDS data sources, an ECS on the same network is required for configuring data integration synchronization tasks on the VPC network and ensuring interconnectivity.

Limits

- Only structured (such as RDS and DRDS), semi-structured, and non-structured (such as OSS and TXT, but the specific synchronization data must be abstracted as structured data) data synchronization are supported. That is, Data Integration supports data synchronization that is capable of transmitting data that can be abstract to a logical two-dimensional table, other fully unstructured data, such as a section of MP3 stored in Oss, the data integration does not yet support synchronizing it to maxcompute, which is implemented later.
- Data synchronization and exchange between a single and certain cross-region data storage are supported.

For certain regions, cross-region data transmission is supported but not guaranteed by the classic network. If you do need to use this function while the classic network is tested disconnected, consider using the public network connection instead.

- Only data synchronization (transmission) is performed and no consumption plans of data stream is provided.

References

- For a detailed description of the data synchronization task configuration, see [creating a data synchronization task](#).
- For a detailed introduction to processing unstructured data such as OSS, see [accessing OSS unstructured data](#).

2.1.2 Basic Terms

DMU

DMU is used to measure the amount of resources, including CPU, memory, and network used for data integration. One DMU represents the minimum amount of resources used for a data synchronization task.

Slot

Default resource group provide you 50 slots and each DMU takes 2 slots, which means default resource group supports 25 DMUs in the same time. You can open a ticket for more slots in your default resource group .

Number of concurrencies

Concurrency indicates the maximum number of threads used to concurrently read data from or write data in the data storage end in a data synchronization task.

Speed Limit

The speed limit indicates the maximum speed for synchronization tasks.

Dirty data

Dirty data indicates invalid data or incorrectly formatted data. For example, if the source has varchar type data but is written to a destination column having int type data, a data conversion exception occurs and the data cannot be written to the destination column.

Data sources

The source of data processed by DataWorks can be a database or a data warehouse. Dataworks supports various types of data sources, and supports transformation between data sources.

2.1.3 Billing FAQ

How does Alibaba Cloud Data Integration bill users?

The basic unit of measurement for data integration is DMU (Data Migration Unit), representing the ability of a single unit in data integration (including CPU, memory, network resource allocation).

A DMU represents the minimum amount of CPU, memory, and network resources used for data integration. A data integration task can run using single or multiple DMUs.

- If the system resource group is used when your synchronization task runs, the charge is calculated as follows:

Charge for one synchronization task = Number of DMUs configured for the task × Price of using a single DMU for one hour × Time consumed by task execution

The charge for using a single DMU for one hour is as follows:

Item	Price
DMU	\$0.35 per hour

- If a custom resource group is used when your synchronization task runs, the charge is calculated as follows:

Charge for one synchronization task = Price of running a synchronization task for one hour × Number of hours consumed by task execution

The charge for running a synchronization task for one hour is as follows:

Item	Price
Duration	\$0.14 per hour

**Note:**

The time consumed by task execution is measured in minutes. The number of minutes used is then rounded up to the nearest integer.

Data Integration service remains free for users in all regions. During this period, you can view your usage details and service usage records in **Billing Management** on the Alibaba Cloud console.

We will inform you before the charging starts.

Does Alibaba Cloud Data Integration incur other costs?

Data Integration is independent from the data source from which data is read, and the target data source to which data is written. You need to pay for upstream and downstream services related to the data source and target data source. For example, if you write data to the object storage service (OSS), you need to pay for the storage used. Check billing details for the storage product you are using. In addition, Internet traffic costs may result due to data transmission. These costs are not included in data integration bills.

2.2 Data source configuration

2.2.1 Supported data sources

Data Integration is a stable, efficient, and elastically scalable data synchronization platform provided by the Alibaba Group to external users. It provides offline (batch) data access channels for Alibaba Cloud's big data computing engines (including MaxCompute, AnalyticDB, and OSS).

The following table lists the data source types supported by Data Synchronization:

Data Source category	Data source type	Extraction (Reader)	Import (Writer)	Support Methods	Supported types:
Relational Databases	MySQL	Yes, see Configure MySQL Reader for more information.	Yes, see Configure MySQL Writer for more information.	Wizard/script	Alibaba Cloud/self-built
Relational Databases	SQL Server	Yes, see Configuring SQL server Reader for more information.	Yes, see Configure SQL Server Writer for more information.	Wizard/script	Alibaba Cloud/self-built
Relational Database	PostgreSQL	Yes, see Configuring PostgreSQL Reader for more information.	Yes, see Configure PostgreSQL Writer for more information.	Wizard/script	Alibaba Cloud/self-built
Relational Databases	Oracle	Yes, see Configure Oracle Reader for more information.	Yes, see Configuring Oracle Writer for more information.	Wizard/script	Self-developed
Relational Databases	DRDS	Yes, see Configure DRDS Reader for more information.	Yes, see Configure DRDS Writer for more information.	Wizard/script	Alibaba Cloud

Data Source category	Data source type	Extraction (Reader)	Import (Writer)	Support Methods	Supported types:
Relational Databases-	DB2	Yes, see Configure DB2 Reader for more information.	Yes, see Configure DB2 writer for more information.	script	Self-developed-
Relational Databases	DM	Yes	Yes	script	Self-developed
Relational Databases	RDS for PPAS	Yes	Yes	script	Alibaba Cloud
MPP	HybridDB for MySQL	Yes	Yes	Wizard/script	Alibaba Cloud
MPP	HybridDB for PostgreSQL released	Yes	Yes	Wizard/script	Alibaba Cloud
Big data storage	Maxcompute (corresponding data source name: ODPS)	Yes, see Configure MaxCompute Reader for more information.	Yes, see Configure MaxCompute Writer for more information.	Wizard/script	Alibaba Cloud
Big data storage	DataHub	No	Yes, see Configure DataHub Writer for more information.	script	Alibaba Cloud
Big data storage	ElasticSearch	No	Yes, see Configure ElasticSearch Writer for more information.	script	Alibaba Cloud
Big data storage	Analyticdb (corresponding data source name: ADS)	No	Yes, see Configure AnalyticDB Writer for more information.	Wizard/script	Alibaba Cloud

Data Source category	Data source type	Extraction (Reader)	Import (Writer)	Support Methods	Supported types:
Unstructured storage	OSS	Yes, see Configure OSS Reader for more information.	Yes, see Configure OSS Writer for more information.	Wizard/script	Alibaba Cloud
Unstructured storage	HDFS	Yes, see Configuring HDFS Reader for more information.	Yes, see Configure HDFS Writer for more information.	script	Self-developed
Unstructured storage	FTP	Yes, see Configuring FTP Reader for more information.	Yes, see Configure FTP Writer for more information.	Wizard/script	Self-developed
Message Queue	LogHub	Yes, see Configure LogHub Reader for more information.	Yes, see Configure LogHub Writer for more information.	Wizard/script	Alibaba Cloud
NoSQL	HBase	Yes, see Configure HBase Reader for more information.	Yes, see Configure HBase Writer for more information.	script	Alibaba Cloud/ self-built
NoSQL	MongoDB	Yes, see Configure MongoDB Reader for more information.	Yes, see Configure MongoDB Writer for more information.	script	Alibaba Cloud/ self-built
NoSQL	Memcache	No	Yes, see Configure Memcache (OCS) Writer for more information.	script	Alibaba Cloud /self-built memcached

Data Source category	Data source type	Extraction (Reader)	Import (Writer)	Support Methods	Supported types:
NoSQL	Table store (corresponding data source name: OTS)	Yes, see Configure Table Store(OTS) Reader for details.	Yes, see Configure Table Store (OTS) Writer for more information.	script	Alibaba Cloud
NoSQL	OpenSearch	No	Yes, see Configure OpenSearch Writer for more information	script	Alibaba Cloud
NoSQL	Redis	No	Yes, see Configure Redis Writer for more information.	script	Alibaba Cloud/ self-built
Performance Testing	Stream	Yes, see Configure Stream Reader for more information.	Yes, see Configure Stream Writer for more information.	script	-

2.2.2 Data source testing connectivity

Data Source	Data Source Type	Network Type	Do you support test connectivity ?	Add custom resource group
MySQL	ApsaraDB	Classic network	Yes	-
		VPC network	Yes	-
	With public IP address		Yes	-
	Without public IP address		No	Yes
	Self-built ECS	Classic network	Yes	-
		VPC network	No	Yes
SQL Server	ApsaraDB	Classic network	Yes	-

Data Source	Data Source Type	Network Type	Do you support test connectivity ?	Add custom resource group
		VPC network	Yes	-
	With public IP address		Yes	-
	Without public IP address		No	Yes
	Self-built ECS	Classic network	Yes	-
		VPC network	No	Yes
PostgreSQL	ApsaraDB	Classic network	Yes	-
		VPC network	Yes	-
	With public IP address		Yes	-
	Without public IP address		No	Yes
	Self-built ECS	Classic network	Yes	-
		VPC network	No	Yes
Oracle	With public IP address		Yes	-
	Without public IP address-		No	Yes
	Self-built ECS	Classic network	Yes	-
		VPC network	No	Yes
DRDS	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
HybridDB for MySQL	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
HybridDB for PostgreSQL released	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
MaxCompute (for ODPS data sources)	ApsaraDB	Classic network	Yes	-
AnalyticDB (for ADS data sources)	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
OSS	ApsaraDB	Classic network	Yes	-
		VPC network	Yes	-

Data Source	Data Source Type	Network Type	Do you support test connectivity ?	Add custom resource group
HDFS	With public IP address		Yes	-
	Self-built ECS	Classic network	Yes	-
		VPC network	No	-
FTP	With public IP address		Yes	-
	Without public IP address		No	-
	Self-built ECS	Classic network	Yes	-
		VPC network	No	-
MongoDB	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
	With public IP address		Yes	-
	Self-built ECS	Classic network	Yes	-
		VPC network	No	Yes
Memcache	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
Redis	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
	With public IP address		Yes	-
	Self-built ECS	Classic network	Yes	-
		VPC network	No	Yes
Table Store (for OTS data sources)	ApsaraDB	Classic network	Yes	-
		VPC network	In scheduling	Yes
DataHub	ApsaraDB	Classic network	Yes	-
		VPC network	No	-

**Note:**

Whether to add a Custom Resource Group, see [Add scheduling resources](#).

Description

In the preceding table, "-" means that this item is unavailable, "No" means that the connectivity test fails and a custom resource group must be added but the synchronization tasks can be configured.

- Data sources in VPC environment:
 - Connectivity test for RDS data sources in VPC environment is supported.
 - Other data sources in VPC environment are under planning.
 - Connectivity tests are not supported for Financial Cloud networks.
- User-created ECS data sources:
 - The classic network supports JDBC-based connectivity tests normally on the public network.
 - The VPC does not support connectivity tests for now.
 - Connectivity tests for cross-region sources are not supported for now.
 - Connectivity tests are not supported for Financial Cloud networks.

Currently, data synchronization is implemented solely by adding a custom resource group. For details, see [Data Synchronization Configuration for the VPC \(Financial Cloud\)](#).

For user-created ECS data sources, ensure to add the IP address of the scheduling cluster to the security group for both inbound and outbound traffic (which is true for both the public network and the classic network). If the security group is not added, disconnection can occur during synchronization. For more information, see [Add security group](#).

You cannot add an extensive range of ports on the ECS security group page. To add them, use the security group API of ECS. For details, see [AuthorizeSecurityGroup](#).

- Data sources created in local IDCs or on the ECS server without public IP addresses:
 - Connectivity tests are not supported.
 - A custom resource group must be added for configuring synchronization tasks.
- Data sources created in local IDCs or on the ECS server with public IP addresses:

For such data sources, public-network-based JDBC is applied to connectivity tests. If the connectivity test fails, check the constraints of the local network or relevant databases.

**Note:**

For connectivity tests of data sources, you may concern the charge for public-network-based synchronization the most. The following example describes the charge for synchronizing data from RDS to MaxCompute:

Currently, Data Integration is free of charge but still may involve certain charged products. For configuring MaxCompute data synchronization in DataWorks, it is free of charge. Charge is applied only when you want to manually add a parameter in the script mode to set a public IP address for the MaxCompute tunnel (however, this parameter is unavailable in the template generated in the script mode.)

Conclusion

When test connectivity fails, you need to verify that the data source area, the network type, the RDS whitelist. Add the full instance id, the database name, and the user name is correct. Examples of common errors are as follows:

- The Database Password is incorrect, as shown below.



- The network doesn't have a diagram, as shown below.

"com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure"

- During synchronization, there is a network disconnect and so on.

First look at the full log to determine which scheduled resource it is and whether it is a custom resource.

If so, check whether the IP address of the custom resource group has been added to the whitelist of the data source such as the RDS (this also applies to the MongoDB).

Check whether the connectivity test between both data sources is succeeded and their whitelists are complete (if the whitelists are incomplete, the test result varies randomly; specifically, the test is succeeded if the task is assigned to the added scheduling server and is failed if no scheduling server has been added.)

- For the condition where the task is displayed as succeeded but the disconnection error 8000 can be found in the log:

This condition occurs when the custom scheduling resource group is used and the IP address 10.116.134.123 and port 8000 are not set as permitted in the security group for inbound traffic. In this condition, add the IP address and the port, and run the task again.

Connectivity test failure examples

Example 1

- Problem

Test connection failed. Connectivity test of data source failed. An error occurred while connecting to the database. The database connection string is "jdbc:mysql://xx.xx.xx.x:xxxx/t_uoer_brade"; the user name is "xxxx_test"; and the exception message is "Access denied for user "xxxx_test"@"%" to database "yyyy_demo".

- Troubleshooting

1. Check whether the entered information is correct.
2. Check whether the password, whitelist, or your account has the permission to access the database. You can add the required permissions in the RDS console.

- Example 2

- Problem

Test connection failed. Connectivity test of data source failed. The error message is as follows:

```
error message: Timed out after 5000 ms while waiting for a server
that matches ReadPreferenceServerSelector{readPreference=primary
}. Client view of cluster state is {type=UNKNOWN, servers=[(
xxxxxxxxxxx), type=UNKNOWN, state=CONNECTING, exception={com.
```

```
mongodb.MongoSocketReadException: Prematurely reached end of stream}}]
```

- Troubleshooting

For non-VPC MongoDB, you must add a whitelist for the connectivity test of the MongoDB data source. For details, see [Add a Whitelist](#) *Add whitelist*.

2.2.3 Configuring SQL server data source

The SQL Server data source allows you to read data from and write data to the SQL Server instances, and supports configuring synchronization tasks in wizard mode and script mode.



Note:

If SQL Server is in a VPC environment, you need to be aware of the following issues.

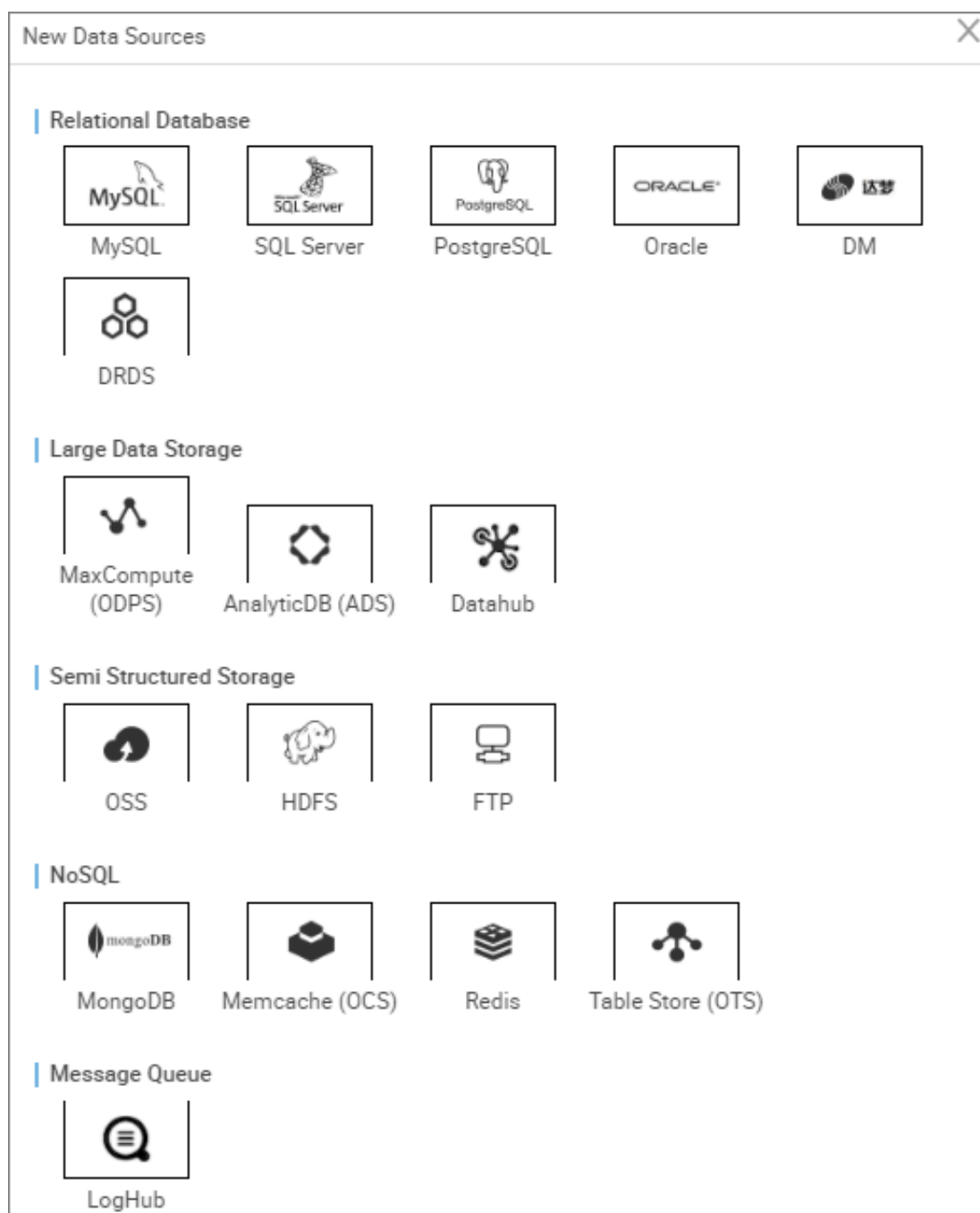
- Add an SQL Server data source
 - Test connectivity is not supported, but the configuration synchronization task is still supported, and you can click **confirm** when creating the data source.
 - You must use a custom scheduled Resource Group to run the corresponding synchronization tasks, make sure that the Custom Resource Group can connect to your self-built database. For more information, see [Data integration when the network of data source \(one side only\) is disconnected](#) and [Data sync when the network of data source \(both sides\) is disconnected](#).

- SQL Server data sources created with RDS

You do not need to select a network environment, and the system automatically determines based on the information you fill in for the RDS instance.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** source to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **SQL Server**.
5. Configure individual information items for the SQL Server data source.

SQL Server data source types are divided into **Ali cloud database (RDS)**, **Public Network IP**, and **non-public network IP**, you can choose according to your situation.

Consider a data source of the new **SQL Server > Alibaba cloud database (RDS)** type.

New SQL Server Data Sources

* Type

ali cloud database (rds)

* Name

sqlserver_source_ali

Description

sqlserver

* Instance ID of RDS

ali-rds-xxxxxx

?

* Main Buyer of RDS

ali-rds-xxxxxx

?

* Database Name

ali-rds-xxxxxx

* Username

ali-rds-xxxxxx

* Password

.....

Test Connectivity

Test Connectivity

ⓘ

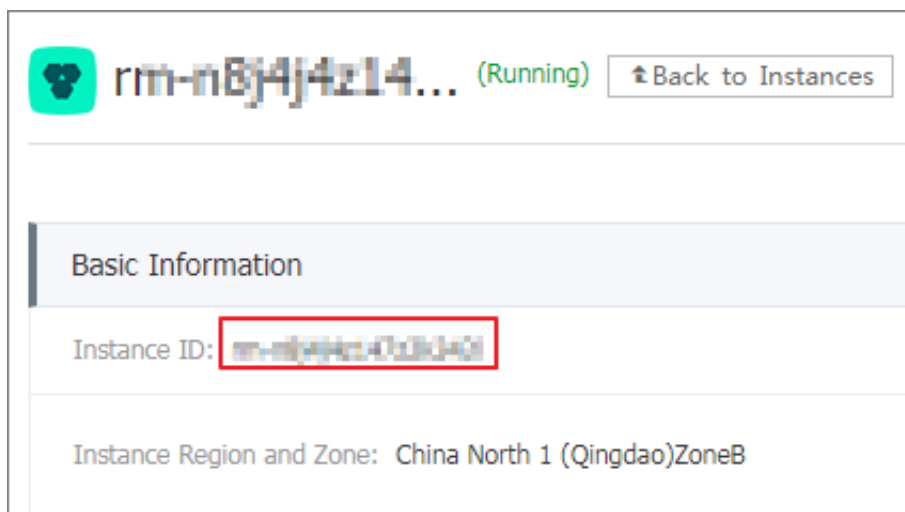
Will need to add rds white list can connect successfully, [point i checked to see how to add the white list](#) .
Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous

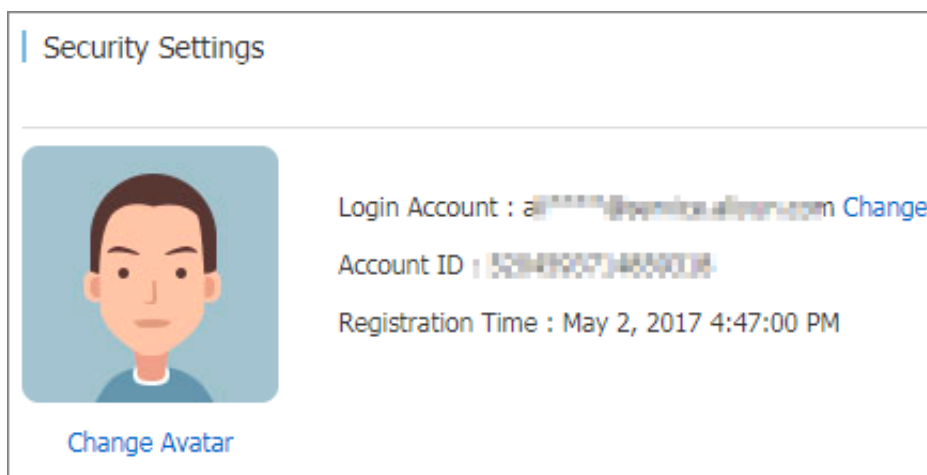
Complete

Configurations:

- Type: Alibaba cloud database (RDS).
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Instance ID of RDS: You can view the instance ID of the RDS in the control desk of the RDS.



- RDS instance buyer ID: You can view the information in the RDS console security settings.



- User name/Password: The user name and password used to connect to the database.

**Note:**

Before you can connect successfully, you need to add an RDS white list.

Consider a data source with a new **SQL Server > public network IP** type.

New SQL Server Data Sources

* Type: there are public ip

* Name: sqlserver_source_ip

Description: sqlserver

* JDBC URL: jdbc:sqlserver://ServerIP:Port;DatabaseName=Database

* Username: sa

* Password:

Test Connectivity: [Test Connectivity](#)

ⓘ Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

[Previous](#) [Complete](#)

Configurations:

- Type: With a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: JDBC connection information in the form of jdbc:sqlserver://ServerIP:Port; DatabaseName=Database.
- Username/Password: The user name and password used to connect to the database.

Consider a data source with a new **SQL Server > public network IP** type.

New SQL Server Data Sources

* Type: no public ip
this type of data sources need to use custom scheduling
resources group can be carried out simultaneously, click here for [help manual](#)

* Name: sqlserver_source

Description: sqlserver

* select resources: Default resource group
group [additional resources group](#)

* JDBC URL: jdbc:sqlserver://ServerIP:Port;DatabaseName=DatabaseName

* Username: example

* Password:

Test Connectivity: [Test Connectivity](#) No public IP data source does not support testing connectivity.

[Previous](#) [Complete](#)

Configurations:

- Type: data source without a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Resource Group: It is used to run synchronization tasks, and generally multiple machines can be bound when you add a resource group. For more information, see [Add scheduling resources](#).
- JDBC URL: JDBC connection information in the form of jdbc:sqlserver://ServerIP:Port; DatabaseName=Database.
- User name/Password: The user name and password used to connect to the database.

6. Click **Test Connectivity**.

7. When the connectivity test is passed, click **Complete**.

Connectivity test description

- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- Private Network and no public network IP, data source connectivity test is currently not supported, click **OK**.

Next step

Now you have learned how to configure the SqlServer data source. The document explains how to configure the SQL Server Writer plug-in later. For more information, see [Configure SQL Server Writer](#).

2.2.4 Configure MongoDB data source

MongoDB, as a NoSQL database, is one of the world's most popular document-based databases, next only to Oracle and MySQL. The MongoDB data source allows you to read data from and write data to MongoDB, and supports configuring synchronization tasks in script mode.



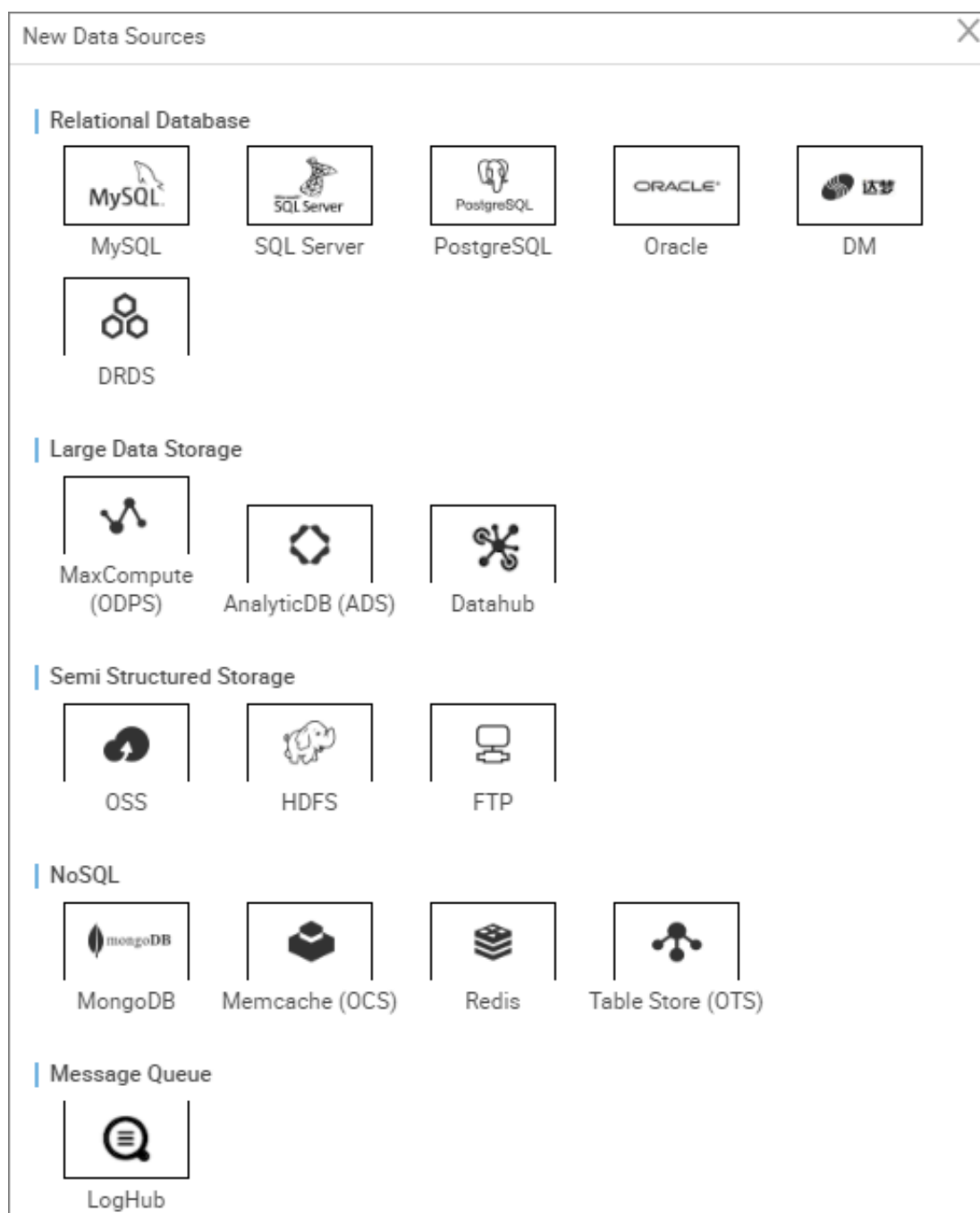
Note:

To add a MongoDB data source, please set up a white list in the MongoDB administration console, IP White List fill in the address as follows (addresses are separated by a comma in English):

```
11.192.97.82,11.192.98.76,10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8
```

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **MongoDB**.
5. Complete the configuration items for the MongoDB data source.

MongoDB data source types are divided into **Alibaba cloud database** and **public network IP self-built database**.

- Alibaba Cloud databases: These databases generally use classic networks. Classic networks within the same region can connect to each other, but those in different regions may not.

- User-created databases with public IPs: These databases generally use public networks, which may cause a certain cost.

Consider a data source with a new **MongoDB > Ali cloud database** type.

New MongoDB Data Sources

* Type: ali cloud database

* Name: MongoDB_source_ali

Description: MongoDB

* area: 华北-1

* Instance ID: [redacted] ?

* database name: AliyunMongoDB

* username: AliyunMongoDB

* password: [masked]

Test Connectivity: [Test Connectivity](#)

Note: if you are using a cloud database For MongoDB edition for reasons of security policy considerations, only support the use of data integration mongodb database corresponding accounts in connection please avoid using root as a visit to the account

[Previous](#) [Complete](#)

Configurations:

- Data Source Type: The selected data source type "MongoDB: Alibaba Cloud database".



Note:

If you have not already authorized the default role of the data integration system, you need the master account to go to ram for the Role authorization, then refresh the page.

- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Region: refers to the region selected when purchasing MongoDB.

- Instance ID: You can view the MongoDB instance ID in the MongoDB console.
- Database Name: you can create a new database in the MongoDB console, set the appropriate data name, user name, and password.
- Username/Password: The user name and password used to connect to the database.

Consider a data source with a new **MongDB > self-built database with public network IP** as an example.

The screenshot shows a configuration window titled "New MongoDB Data Sources". It includes the following fields and elements:

- * Type:** A dropdown menu with the selected value "there are public ip".
- * Name:** A text input field containing "MongoDB_ip".
- Description:** A text input field containing "MongoDB".
- * visit the address:** A text input field containing "MongoDB". Below it is a blue button labeled "add visit address".
- * database name:** A text input field containing "database".
- * username:** A text input field containing "username".
- * password:** A text input field with masked characters (dots).
- Test Connectivity:** A blue button labeled "Test Connectivity".
- Warning Message:** A red circular icon with an exclamation mark followed by the text: "if you are using a cloud database For Mongoddb edition for reasons of security policy considerations, only support the use of data integration mongoddb database corresponding accounts in connection please avoid using root as a visit to the account".
- Navigation:** At the bottom right, there are two buttons: "Previous" and "Complete".

Configurations:

- Type: The selected data source type "MongoDB: User-created database with a public IP".
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Visit the address: The format is host:port.
- Add visit address: Add an access address in the format of host:port.
- Database name: It is the name of the database mapped to the data source.

- Username/Password: The user name and password used to connect to the database.

6. Click **Test Connectivity**

7. When the connectivity test is passed, click **Complete**.



Note:

- A MongoDB cloud database in a VPC environment that is added with a public network IP data source type and saved.
- The VPC does not support connectivity tests for now.

Next step

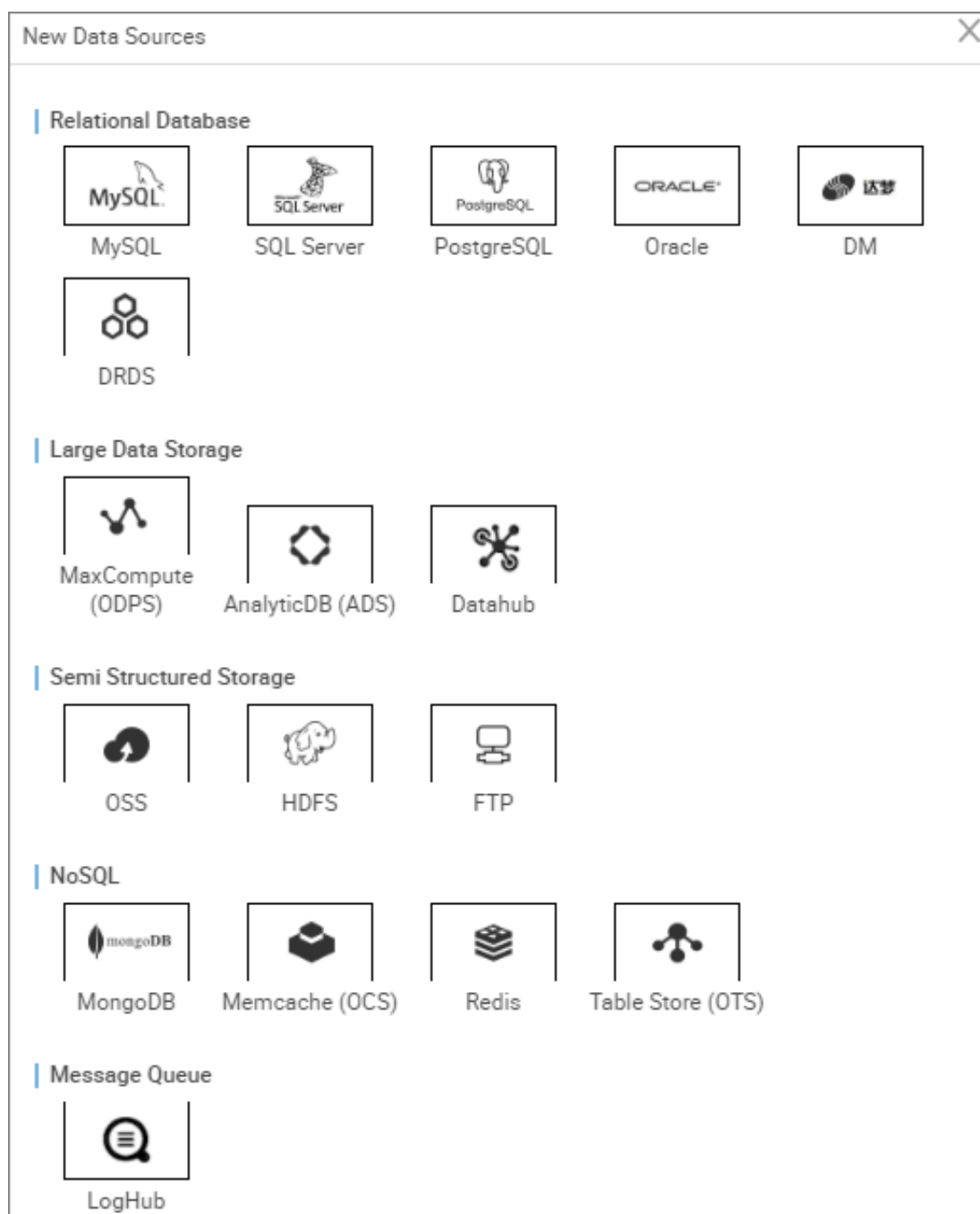
Now you have learned how to configure the MongoDB data source. The document explains how to configure the MongoDB Writer plug-in later. For more information, see [Configure MongoDB Writer](#).

2.2.5 DataHub data source

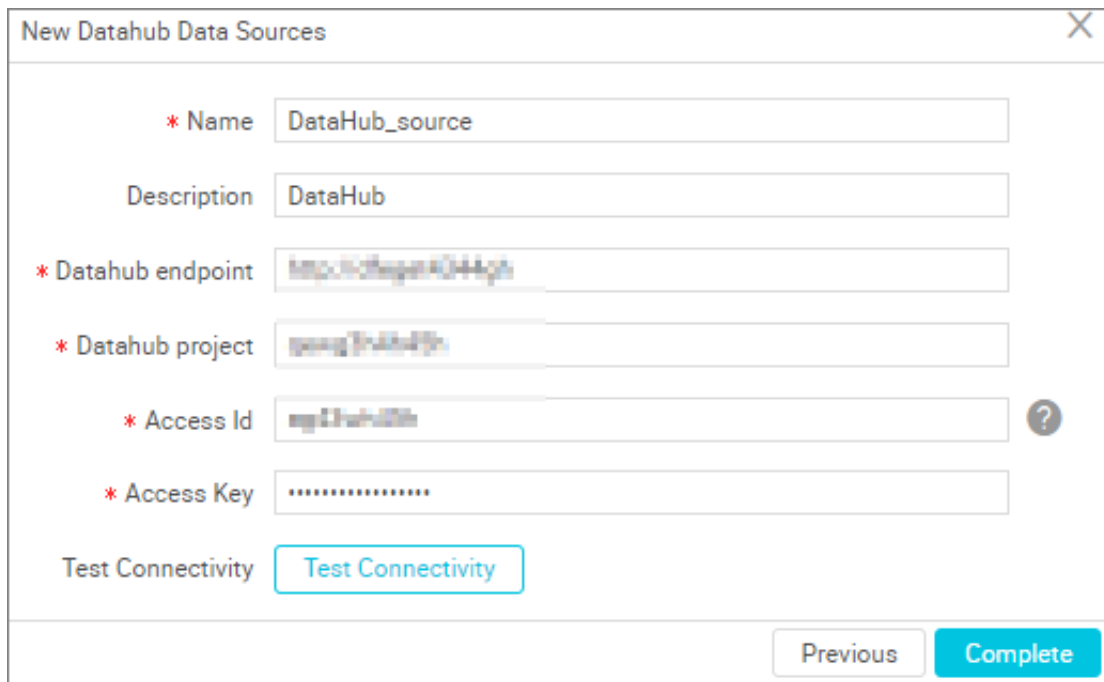
DataHub provides a comprehensive data import solution that allows quicker massive data computing. The DataHub data source, as the data pivot, allows other data sources to write data to DataHub and supports the Writer plug-in.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as datahub.
5. Configure individual information items for the datahub data source.



New Datahub Data Sources

* Name

Description

* Datahub endpoint

* Datahub project

* Access Id ?

* Access Key

Test Connectivity

Configurations:

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Datahub endpoint: This parameter is read-only by default and is automatically read from the system configuration.
- Datahub project: ID of the DataHub Project.
- AccessID/AceessKey: [the access key](#) (AccessKeyId and AccessKeySecret) is equivalent to the logon password.

6. Click **Test Connectivity**.

7. When the connectivity test is passed, click **Complete**.

Provides the ability to test connectivity to determine if the information entered is correct.

Next step

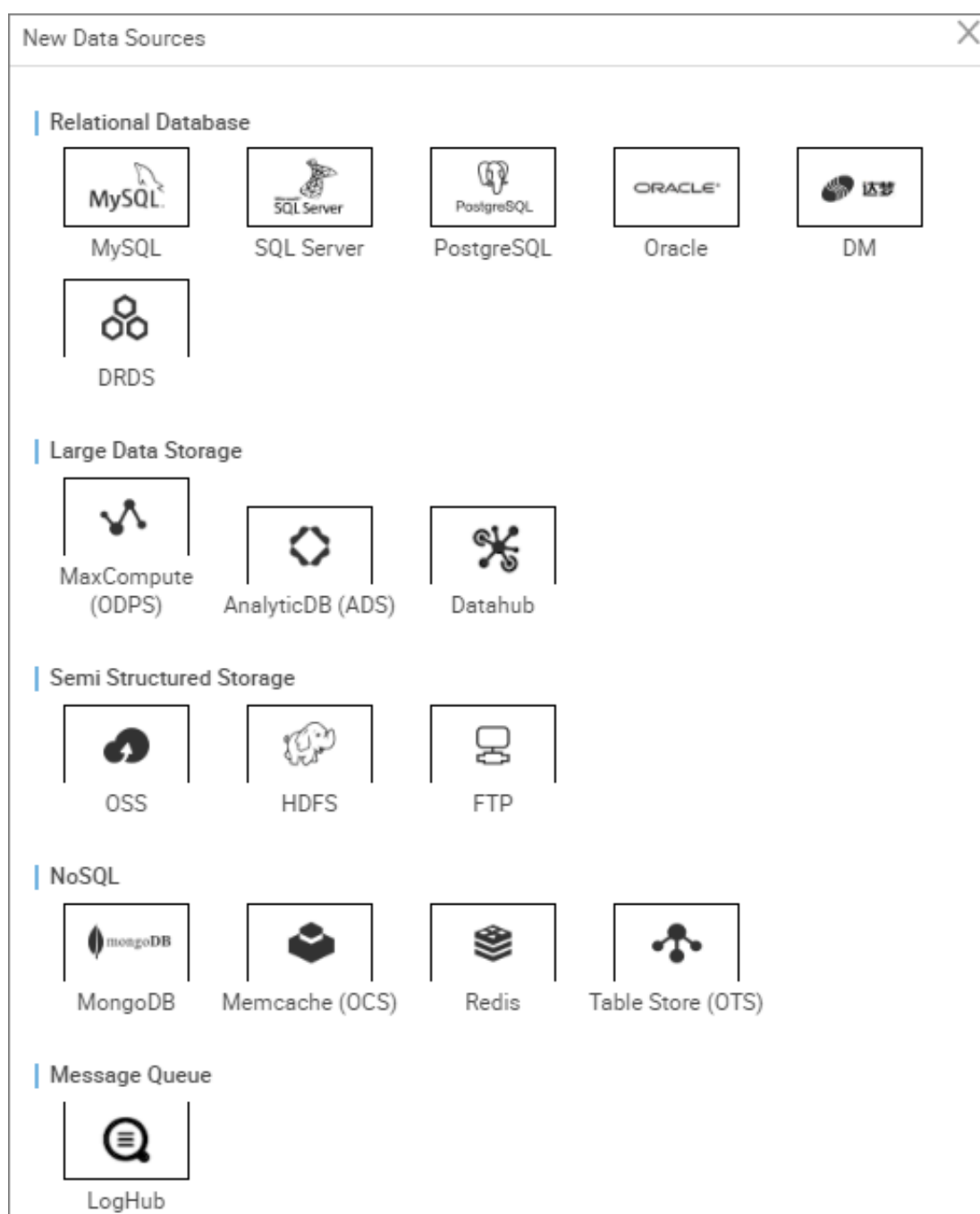
Now you have learned how to configure the DataHub data source. The document explains how to configure the Oracle Writer plug-in later. For more information, see [Configure DataHub Writer](#).

2.2.6 Configure the DM data source

The DM relational database data source provides the ability to read data from and write data to DM databases, and supports configuring synchronization tasks in wizard and script modes.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** from the Actions column of the relevant project in the Project List.
2. Select **Data Integration** in the top navigation bar. Click **Data Source** from the left-side navigation pane.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select a data source type of **dream**.
5. Configure the information items of the DM data source.

Select either of the following data source types as needed when creating a DM data source:

- With public IP address

New DM Data Sources

* Type: there are public ip

* Name: DM_source_ip

Description: DM

* JDBC URL: jdbc:dm://ServerIP:Port/Database

* Username: username

* Password:

Test Connectivity: Test Connectivity

ⓘ Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous Complete

Parameters:

- Type: With a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: In the format of jdbc:mysql://ServerIP:Port/Database.
- Username/Password: The user name and password used to connect to the database.
- Without public IP address

New DM Data Sources

* Type

no public ip

this type of data sources need to use custom scheduling

resources group can be carried out simultaneously, click here for

[help manual](#)

* Name

DM_source

Description

DM

* select resources

please select a resource group

group [additional resources group](#)

* JDBC URL

jdbc:dm://ServerIP:Port/Database

* Username

* Password

Test Connectivity

Test Connectivity

No public IP data source does not support testing connectivity.

ⓘ

Ensure that the database can be network access

Ensure that the database is not a firewall prohibits

Ensure that the database can be parsed by the domain name

Ensure that the database has been launched

Previous

Complete

Parameters:

- Type: No public network IP, selecting a data source of this type requires the use of custom scheduling resources for synchronization, you can click **Help manual** for details.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Resource Group: It is used to run synchronization tasks, and generally multiple machines can be bound when you add a resource group. For more information, see [Add scheduling resources](#).
- JDBC URL: In the format of jdbc:mysql://ServerIP:Port/Database.
- Username/Password: The user name and password used to connect to the database.

42

Issue: 20190117

6. (Optional). Click **Test Connectivity** to test the connectivity after entering all the required information in the relevant fields.
7. When the connectivity test is passed, click **Complete**.

Provides the ability to test connectivity to determine if the information entered is correct.

Connectivity test description

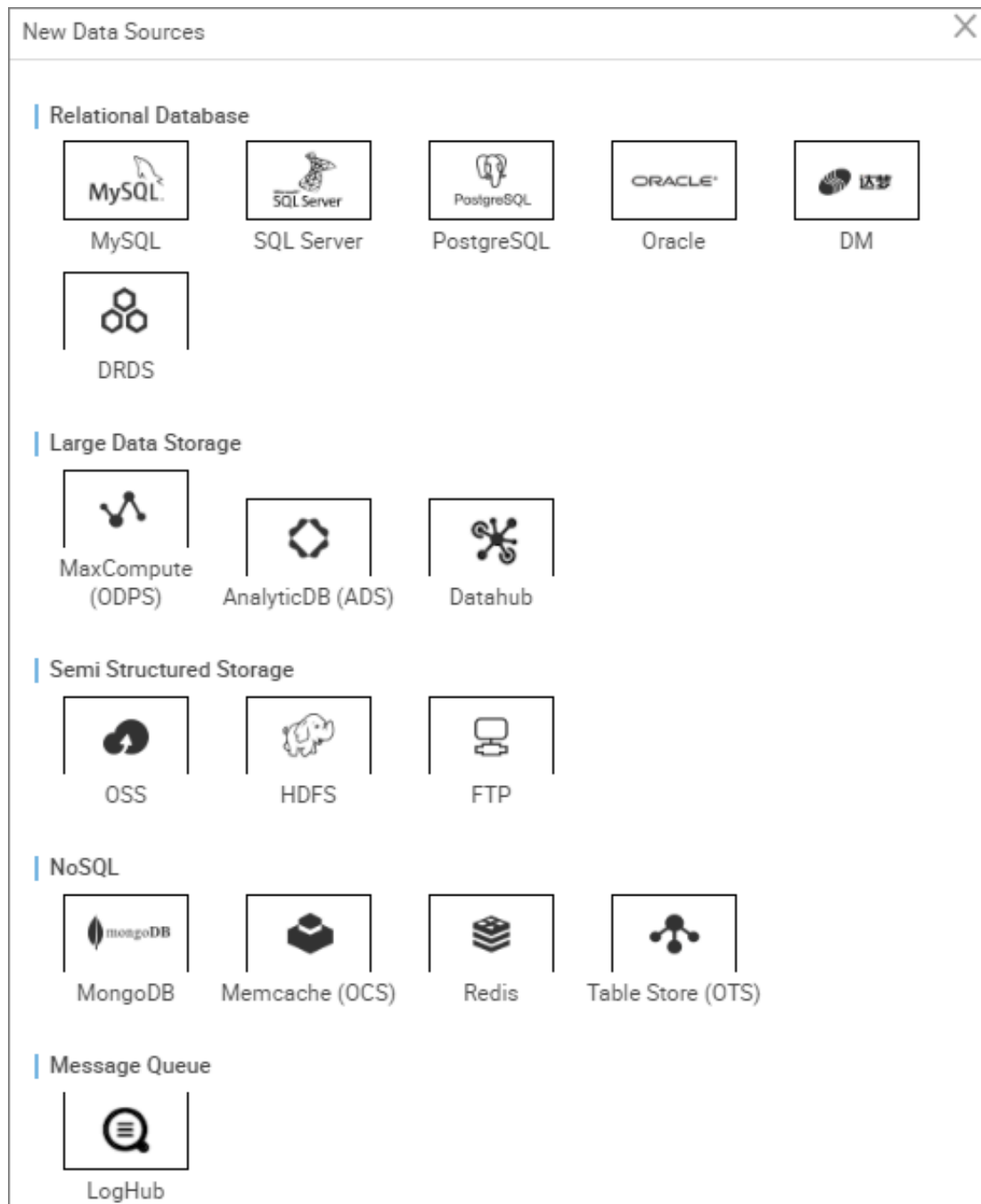
- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- Currently, connectivity test is not supported for the VPC and without-public-IP-address data source types. Thus, click **Confirm** directly.

2.2.7 Configure DRDS data sources

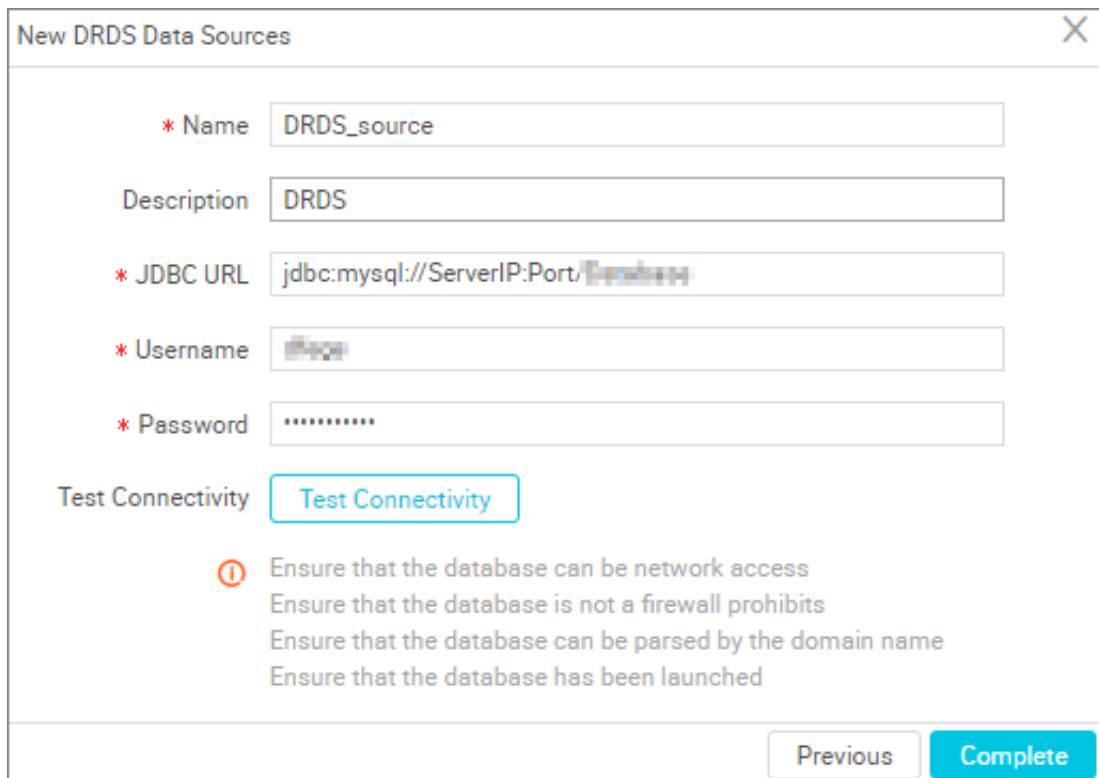
The DRDS data source allows you to read data from and write data to DRDS, and supports configuring synchronization tasks in wizard mode and script mode.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **DRDS**.
5. Fill in configuration items for the DRDS data source to be created.



New DRDS Data Sources

* Name

Description

* JDBC URL

* Username

* Password

Test Connectivity

ⓘ Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Configurations:

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: JDBC URL, in the format of jdbc:mysql://serverIP:Port/database.
- Username/Password: The user name and password used to connect to the database.

6. Click Test Connectivity

7. When the connectivity test is passed, click Complete.

Provides the ability to test connectivity to determine if the information entered is correct.

Connectivity test description

- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- Private Network and no public network IP, data source connectivity test is currently not supported, click **confirm**.

Next step

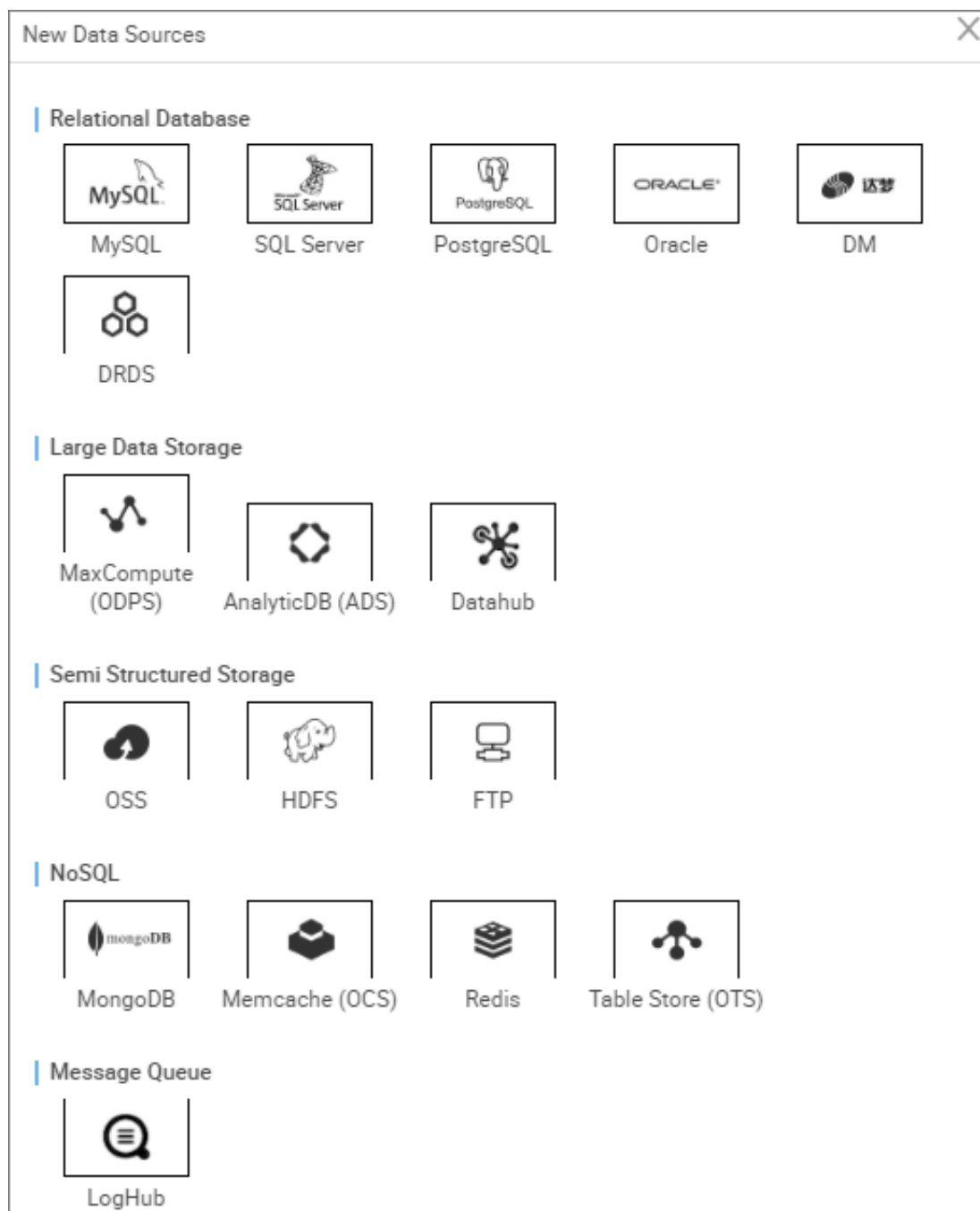
Now you have learned how to configure the DRDS data source. The document explains how to configure the DRDS Writer plug-in later. For more information, see [Configure DRDS Writer](#).

2.2.8 Configure the FTP data source

The FTP data source allows you to read data from and write data to FTP, and supports configuring synchronization tasks in wizard mode and script mode.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **FTP**.
5. Configure the information items of the FTP data source.

You can create either of the following two FTP data sources as required:

- With public IP address

New FTP Data Sources

* Type: there are public ip

* Name: FTP_source_ip

Description: FTP

* Protocol: ☒ ftp ☐ sftp

* Host: 111.111.111.111

* Port: 21

* username: username

* password:

Test Connectivity: Test Connectivity

Previous Complete

Configurations:

- Type: With a public IP address.
 - Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
 - Description: It is a brief description of the data source with no more than 80 characters.
 - Protocol: Currently only FTP and SFTP are supported.
 - Host: The FTP host IP address.
 - Port: If you select the FTP protocol, the port defaults to 21. If SFTP is selected, the port 22 is used by default.
 - Username/Password: The account and password for accessing the FTP service.
- Without public IP address

New FTP Data Sources

* Type

no public ip

this type of data sources need to use custom scheduling resources group can be carried out simultaneously, click here for [help manual](#)

* Name

FTP_source

Description

FTP

* select resources

please select a resource group

group [additional resources group](#)

* Protocol

☒ ftp ☐ sftp

* Host

* Port

21

* username

* password

Test Connectivity

Test Connectivity

No public IP data source does not support testing connectivity.

Previous

Complete

Configurations:

- Data source type: data source without a public IP address. The data source of this type must use custom scheduling resources so that it can synchronize data. For details, click **Help Manual**.
- Data source name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- Resource Group: It is used to run synchronization tasks, and generally multiple machines can be bound when you add a resource group. For details, see [Add scheduling resources](#).
- Protocol: Currently only FTP and SFTP are supported.
- Host: The FTP host IP address.

- Port: If you select the FTP protocol, the port defaults to 21. If SFTP is selected, the port 22 is used by default.
- User Name/Password: The account and password for accessing the FTP service.

6. Click **Test Connectivity**

7. When the connectivity test is passed, click **Complete**.

Provides the ability to test connectivity to determine if the information entered is correct.

Connectivity test description

- The connectivity test is available in the classic network to identify whether the input host, port, user name, and password information is correct.
- The data source connectivity test is currently not supported by the proprietary network, and you can click **confirm**.

Next step

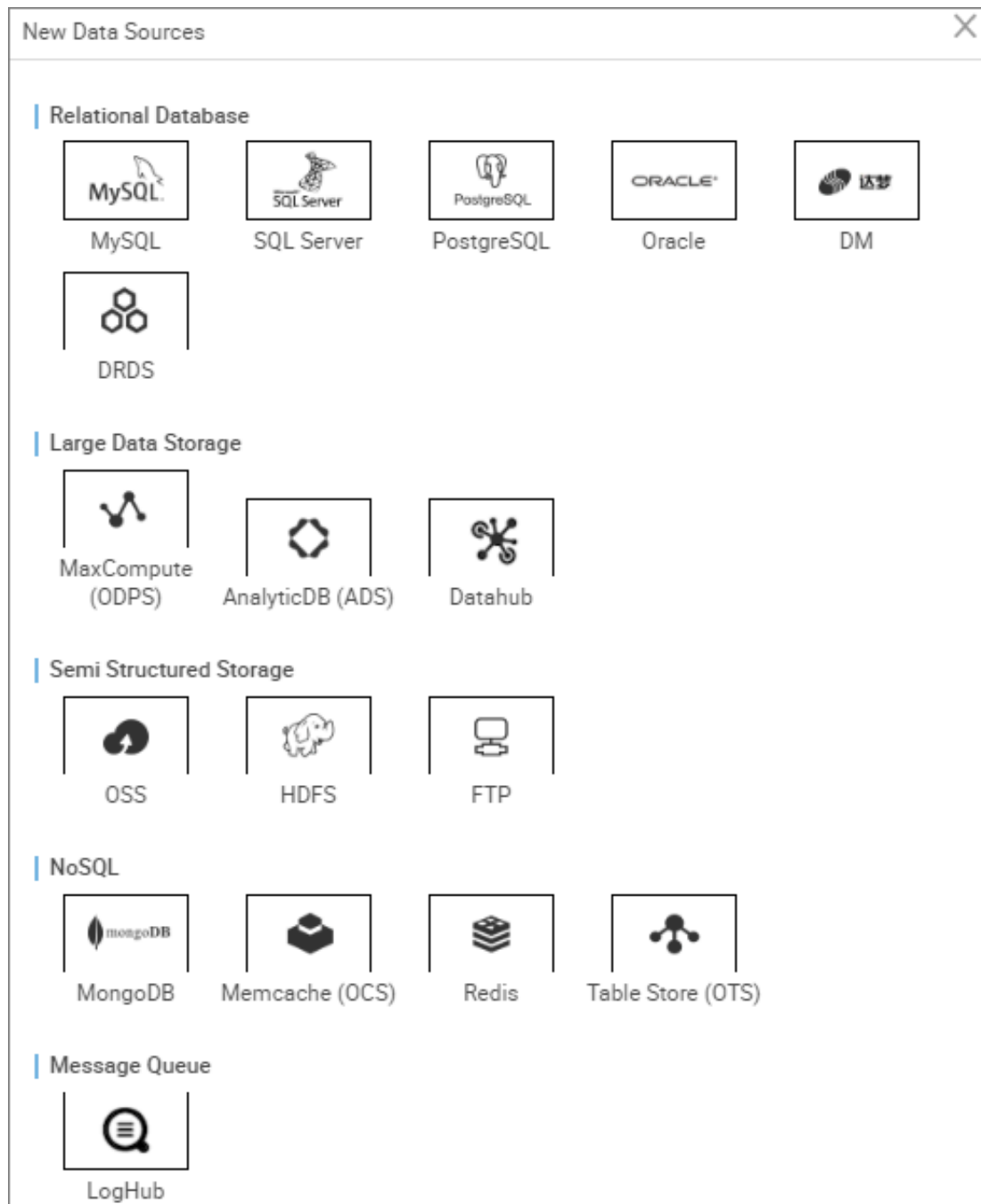
Now you have learned how to configure the FTP data source. The document explains how to configure the FTP Writer plug-in later. For more information, see [Configure FTP Writer](#).

2.2.9 Configuring HDFS data source

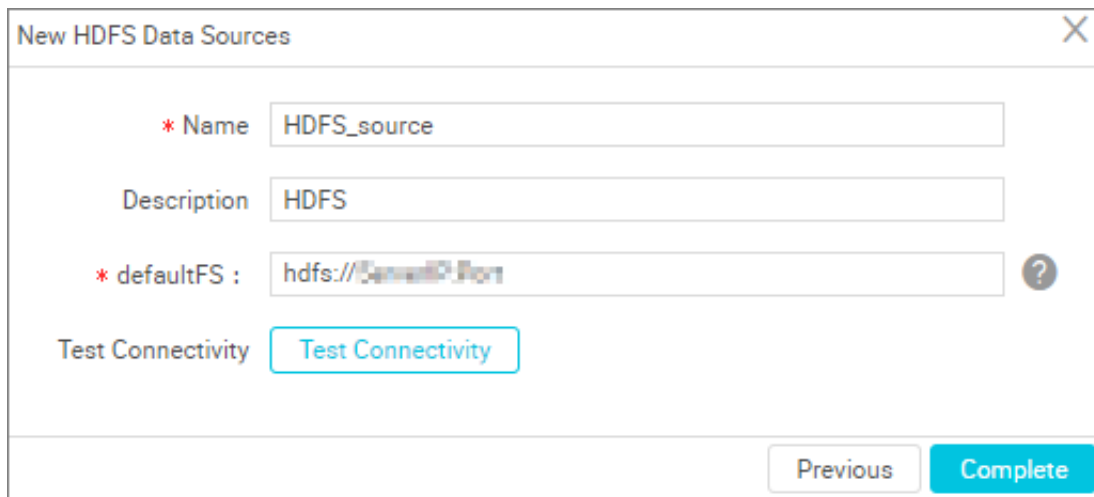
HDFS, as a distributed file system, allows you to read data from and write data to HDFS, and supports configuring synchronization tasks in script mode.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** from the Actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type **HDFS**.
5. Configure individual information items for HDFS data sources.



New HDFS Data Sources

* Name

Description

* defaultFS : ?

Test Connectivity

Configurations:

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- defaultFS: The node address of nameNode in the format of hdfs://ServerIP:Port.

6. Click Test Connectivity

7. When the connectivity test is passed, click Complete.

Provides the ability to test connectivity to determine if the information entered is correct.

Connectivity test description

- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- The data source connectivity test is currently not supported by the proprietary network, and you can click **confirm**.

Next step

Now you have learned how to configure the HDFS data source. The document explains how to configure the HDFS Writer plug-in later. For more information, see [Configure HDFS Writer](#).

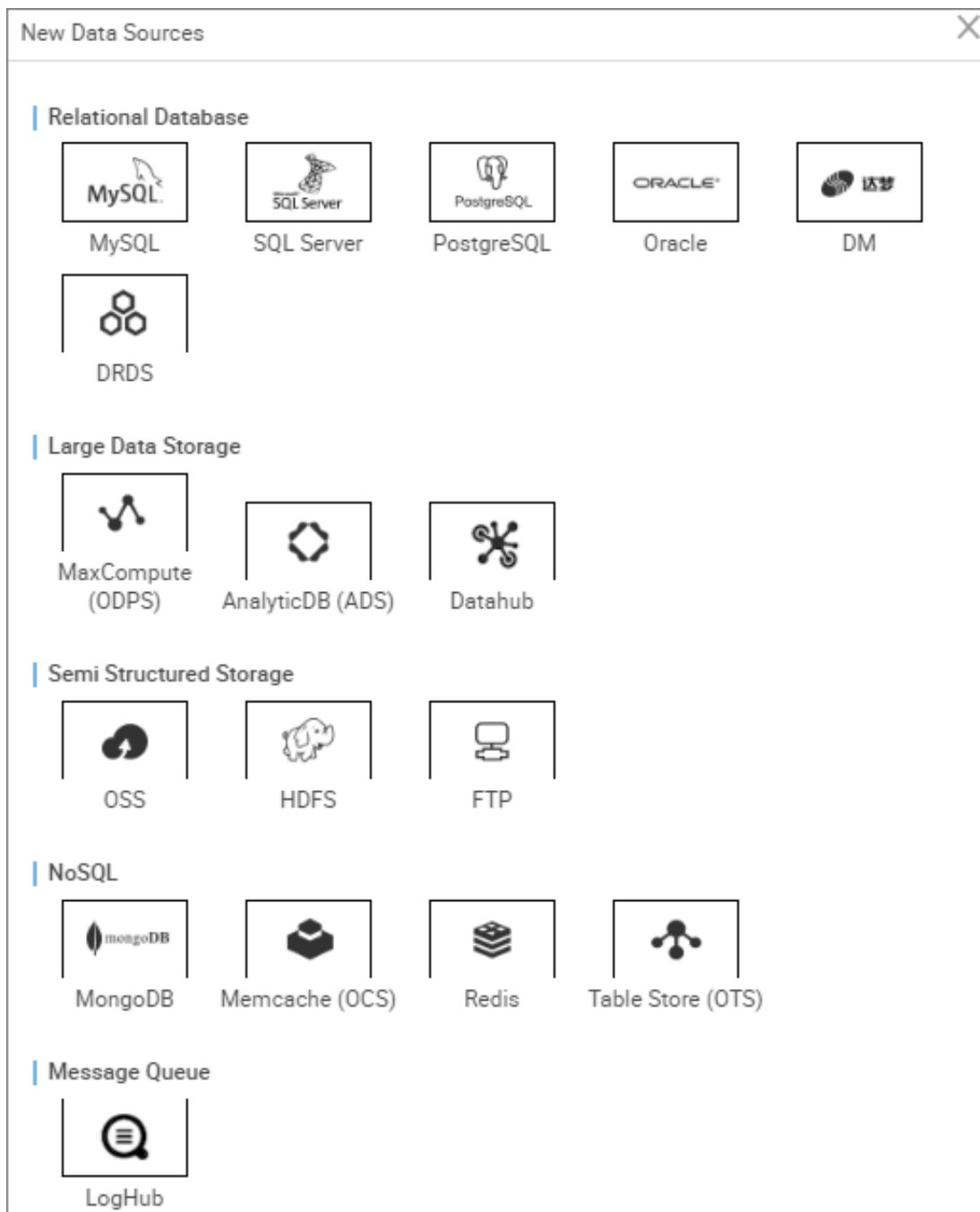
2.2.10 Add LogHub data source

As a data hub, the LogHub data source allows you to read data from and write data to LogHub, and supports the Reader and Writer plug-ins.

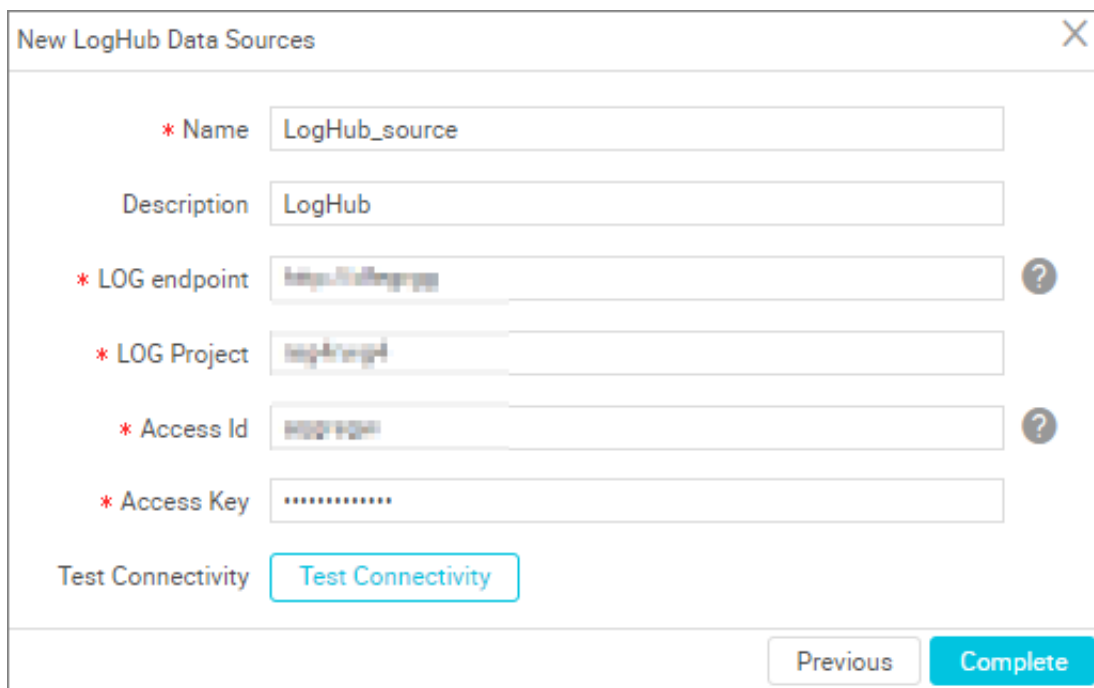
Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.

2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type **LogHub**.
5. Configure individual information items for the loghub data source.



New LogHub Data Sources

* Name

Description

* LOG endpoint ?

* LOG Project

* Access Id ?

* Access Key

Test Connectivity

Configurations:

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- LogHub Endpoint: Generally in the format of <http://cn-shanghai.log.aliyun.com>. Please refer to the service portal for details. [service entrance](#).
- Project: Name of the project.
- AccessID/AceessKey: the [access key](#) (AccessKeyID and AccessKeySecret) is equivalent to the logon password.

6. Click **Test Connectivity**.

7. When the connectivity test is passed, click **Complete**.

The connectivity test is provided to identify whether the input project/AK information is correct.

Next step

Now you have learned how to configure the LogHub data source. In this tutorial you will learn how to [Configure LogHub Reader](#) and [Configure LogHub Writer](#).

2.2.11 Configure MaxCompute data source

MaxCompute (formerly known as ODPS) provides a comprehensive data import solution that allows quicker massive data computing. As a data hub, the MaxCompute data source allows you to read data from and write data to MaxCompute, and supports the Reader and Writer plug-ins.



Note:

A default data source (odps_first) is generated for each project, and the MaxCompute project name is the same as the one for the computing engine of the current project.

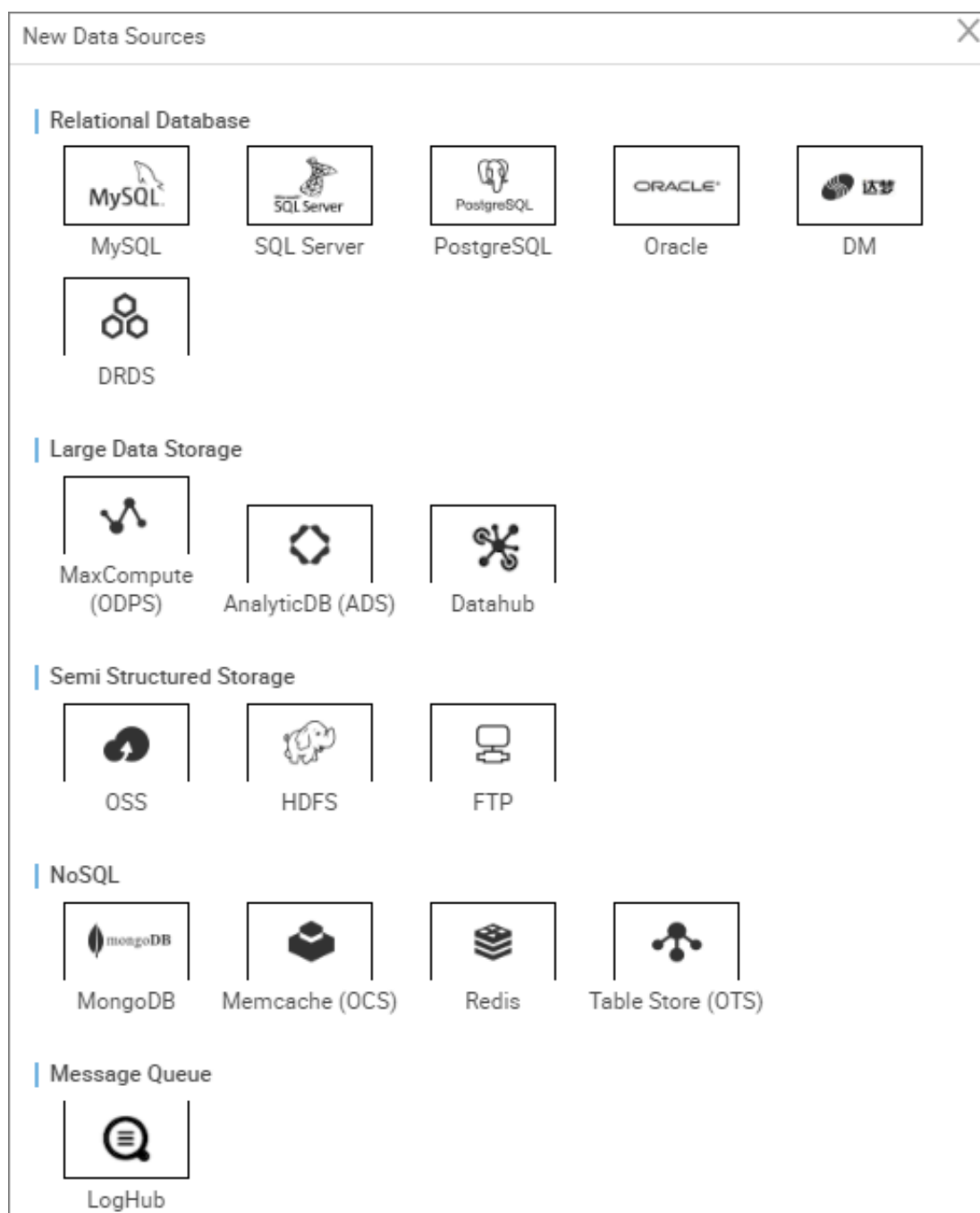
The AK of the default data source can click on the user information in the upper right and switch at the modification of AccessKey information, but it should be noted that:

1. You can only switch from the main account AK to the main account AK.
2. When switching, there must be no tasks currently in operation (data integration or data development and all other tasks related to DataWorks).

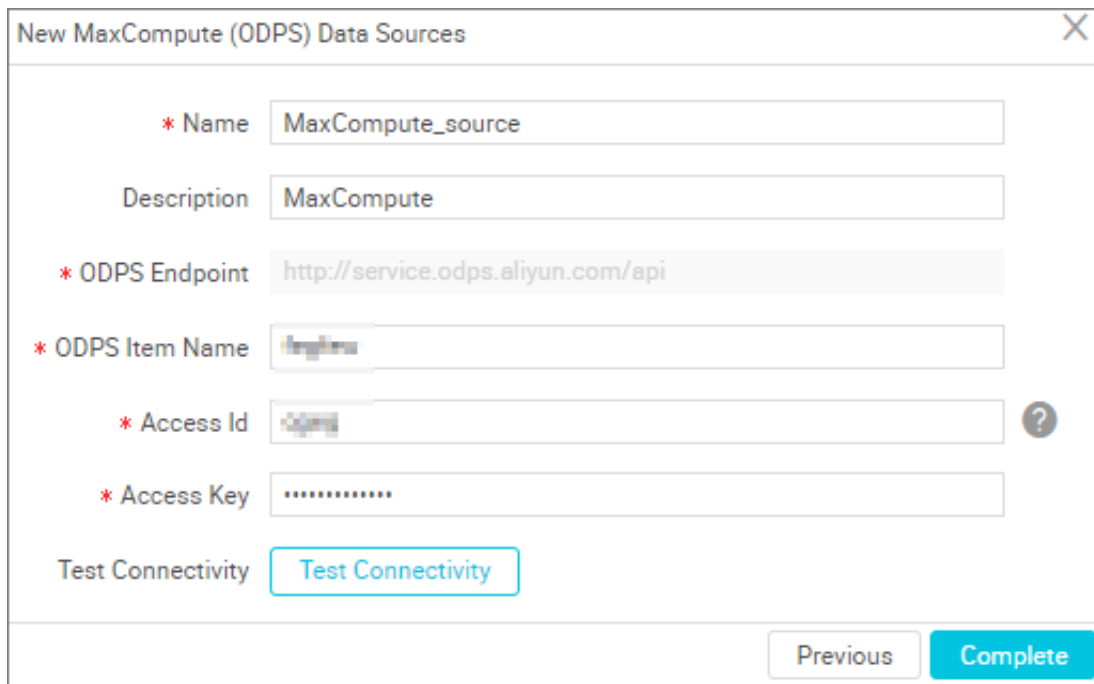
The MaxCompute data source you add yourself can use the subaccount AK.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source window box, select the data source type as **MaxCompute(ODPS)**.
5. Complete the configuration items of the MaxCompute data source.



The screenshot shows a configuration window titled "New MaxCompute (ODPS) Data Sources". It contains the following fields and controls:

- Name:** A text input field with the value "MaxCompute_source".
- Description:** A text input field with the value "MaxCompute".
- ODPS Endpoint:** A text input field with the value "http://service.odps.aliyun.com/api".
- ODPS Item Name:** A text input field with a dropdown menu showing "MaxCompute".
- Access Id:** A text input field with a dropdown menu showing "AccessId".
- Access Key:** A text input field with masked characters "*****".
- Test Connectivity:** A button labeled "Test Connectivity".
- Navigation:** "Previous" and "Complete" buttons at the bottom right.

Configurations:

- Data source name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- ODPS endpoint: defaults to read-only. The value is automatically read from the system configuration.
- ODPS project name: the corresponding MaxCompute project indicator.
- AccessID/AceessKey: the [access key](#) (AccessKeyID and AccessKeySecret) is equivalent to the logon password.

6. Click **Test Connectivity**.

7. When the connectivity test is passed, click **Complete**.

The connectivity test is provided to identify whether the input project/AK information is correct.

Next step

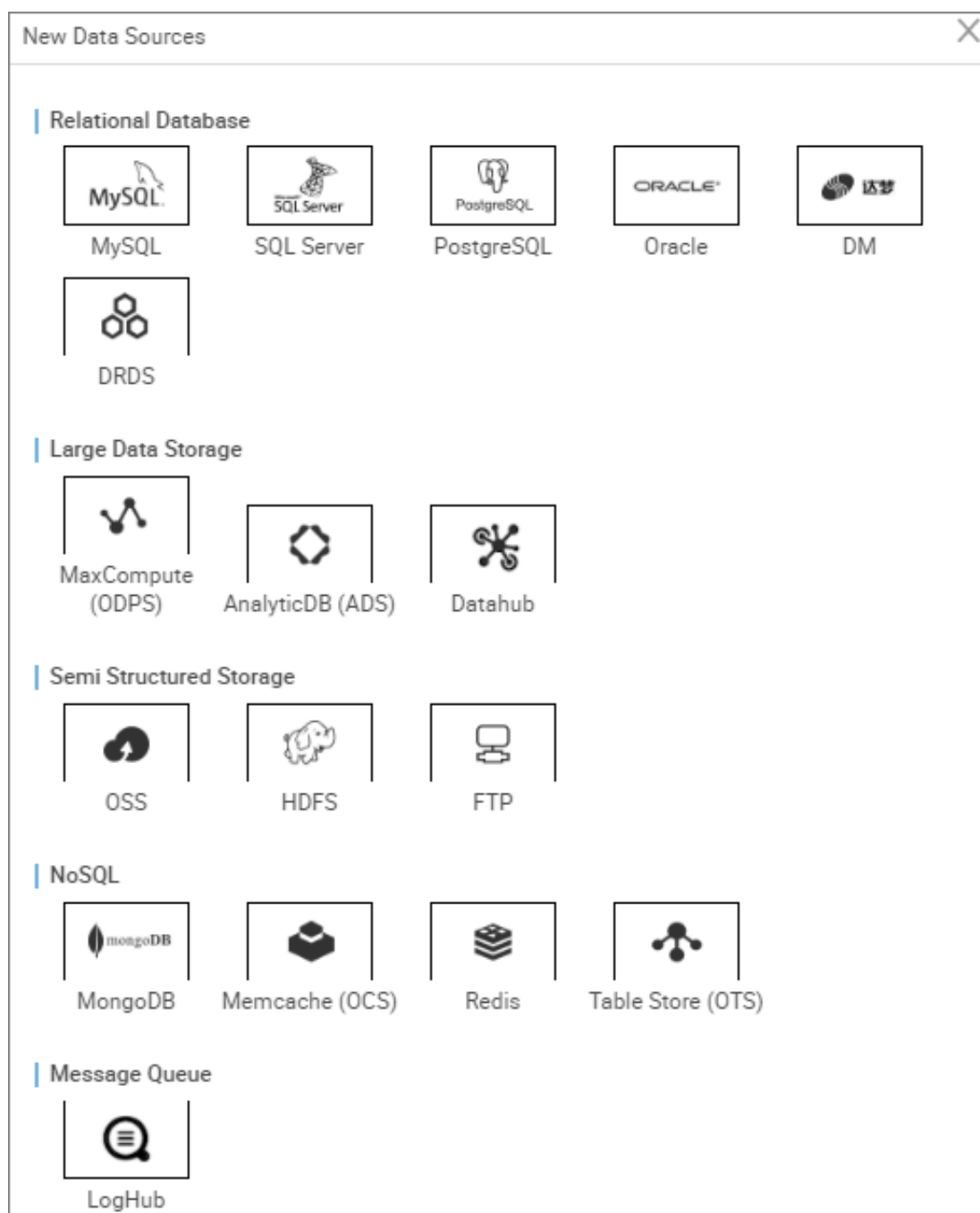
Now you have learned how to configure the MaxCompute data source. The document explains how to configure the MaxCompute Writer plug-in later. For more information, see [Configure MaxCompute Writer](#).

2.2.12 Configure Memcache data source

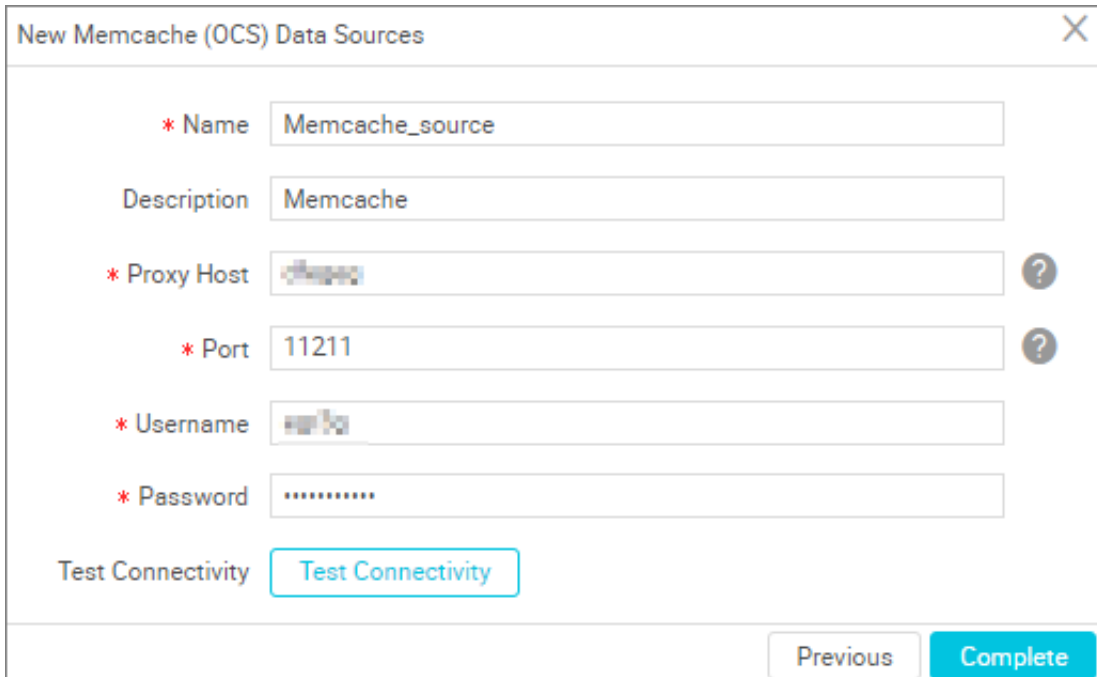
The Memcache (formerly known as OCS) data source provides the ability to write data from other data sources to Memcache, and supports configuring synchronization tasks only in script mode.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **Memcached**.
5. Complete the configuration items for the Memcache data source.



Configurations:

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Type: The selected data source type Memcache.
- Proxy Host: the appropriate Memcache Proxy.
- Port: the appropriate memcache port, with a default of 11211.
- Username/Password: The username and password of the database.

6. Click **Test Connectivity**
7. When the connectivity test is passed, click **Complete**.

Provides the ability to test connectivity to determine if the information entered is correct.

Next step

Now you have learned how to configure the Memcache data source. The document explains how to configure the Memcache Writer plug-in later. For more information, see [Configure Memcache \(OCS\) Writer](#).

2.2.13 Configure MySQL data source

The MySQL data source allows you to read data from and write data to MySQL, and supports configuring synchronization tasks in wizard mode and script mode.



Note:

If you are using MySQL in a VPC environment, you need to be aware of the following issues.

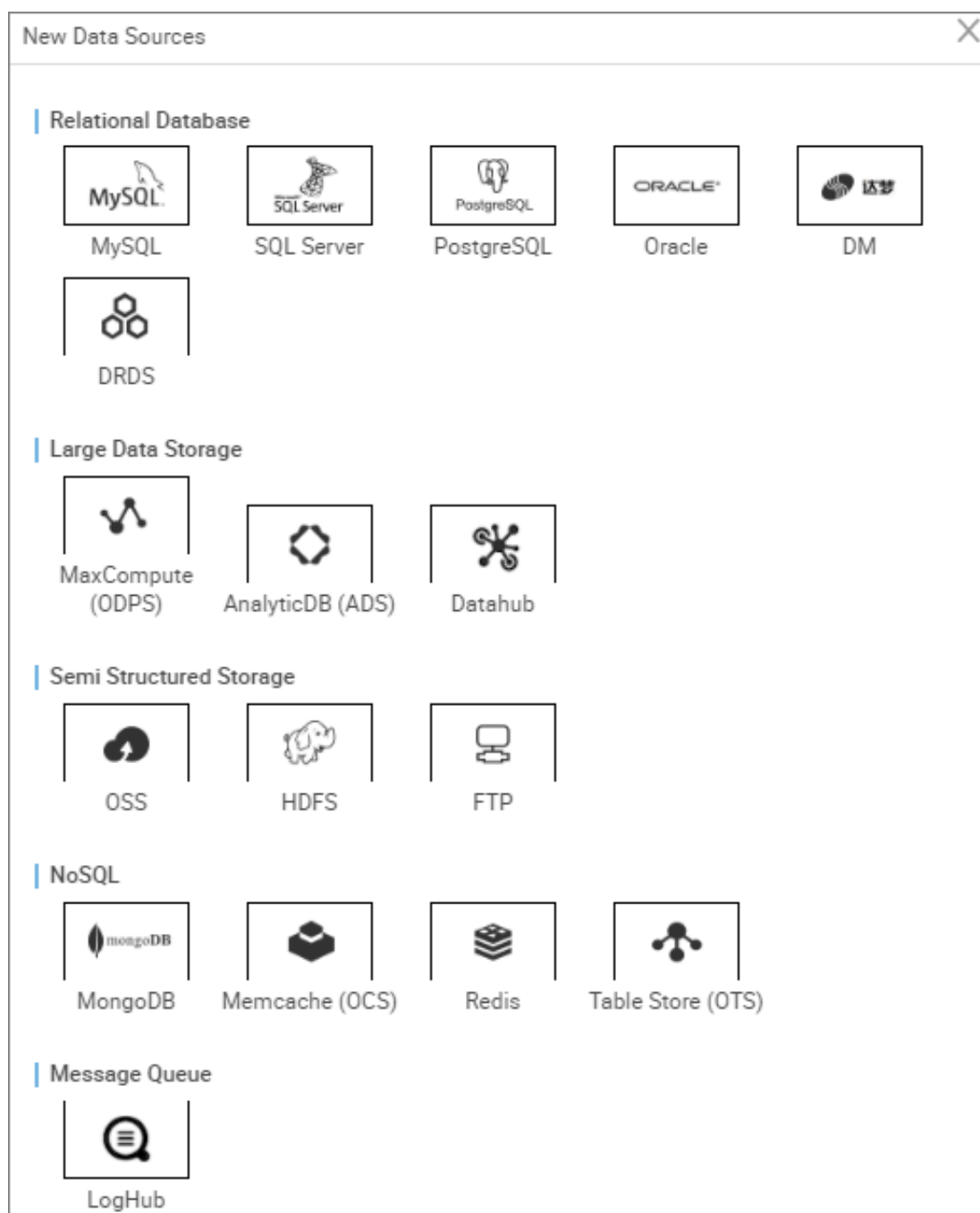
- Self-built MySQL Data Source
 - Test connectivity is not supported, but the configuration synchronization task is still supported, and you can click **confirm** when creating the data source.
 - You must use a custom scheduled Resource Group to run the corresponding synchronization tasks, make sure that the Custom Resource Group can connect to your self-built database. For more information, see [Data integration when the network of data source \(one side only\) is disconnected](#) and [Data sync when the network of data source \(both sides\) is disconnected](#).

- MySQL data sources created with RDS

You do not need to select a network environment, and the system automatically determines based on the information you fill in for the RDS instance.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
3. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
4. Click **Add new data source** to pop up the supported data source.



5. In the new data source dialog box, select the data source type as **MySQL**.
6. Configure individual information items for the MySQL data source.

MySQL Data source types are divided into the **Ali cloud database (RDS)**, the **public network IP** and the **non-public network IP**.

Consider a data source of the new **MySQL > Ali cloud database (RDS)** type.

New MySQL Data Sources

* Type

ali cloud database (rds)

* Name

rds_source

Description

rds

* Instance ID of RDS

* Main Buyer of RDS

* Database Name

* Username

* Password

.....

Test Connectivity

Test Connectivity

①

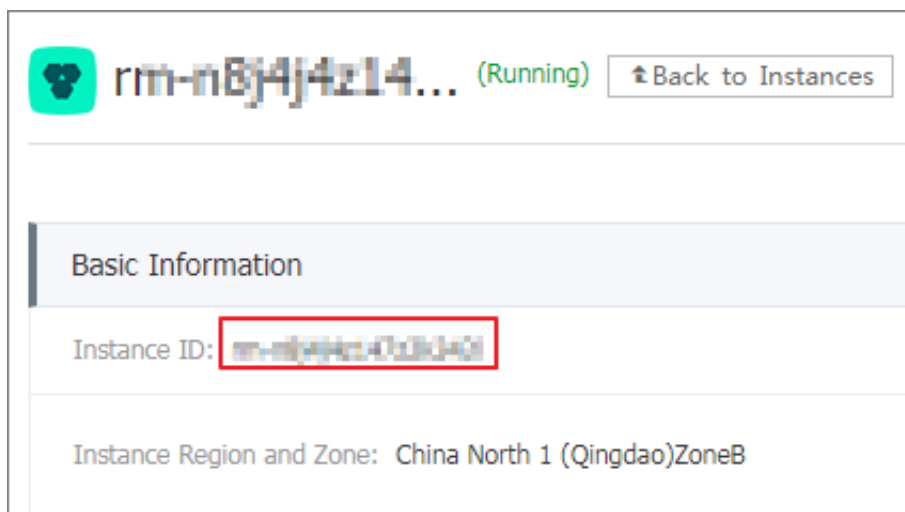
Will need to add rds white list can connect successfully, [point i checked to see how to add the white list](#) .
Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous

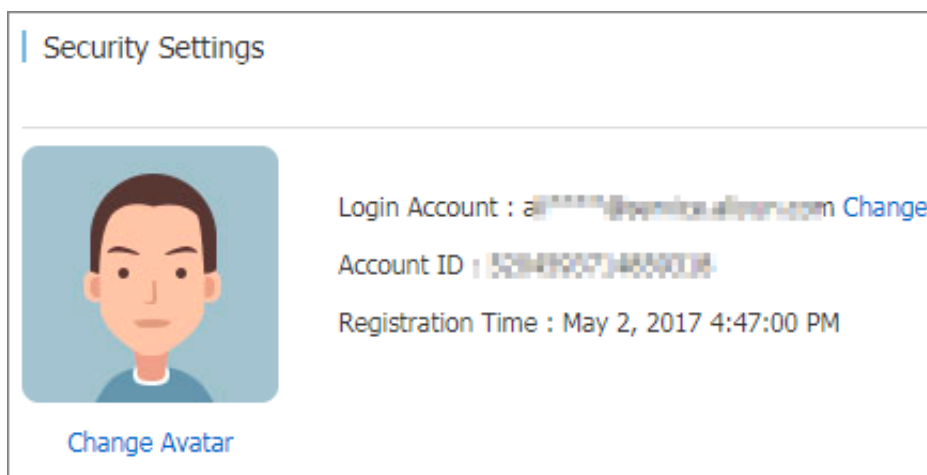
Complete

Configurations:

- Type: the currently selected data source type mysql > Ali cloud database (RDS).
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Instance ID of RDS: You can go to the RDS console to view the instance ID of the RDS.



- RDS instance buyer ID: You can view the information in the RDS console security settings.



- User name/Password: The user name and password used to connect to the database.

**Note:**

Before you can connect successfully, you need to add an RDS white list. For more information, see [Add whitelist](#).

Consider a data source of the new **MySQL > public network IP** type as an example.

New MySQL Data Sources

* Type: there are public ip

* Name: mysql_source_ip

Description: mysql

* JDBC URL: jdbc:mysql://serverIP:Port/Database

* Username: username

* Password:

Test Connectivity: Test Connectivity

ⓘ Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous Complete

Configurations:

- =Type: With a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: In the format of jdbc://mysql://serverIP:Port/database.
- User =name/Password: The user name and password used to connect to the database.

Consider a data source with a new **MySQL > non-public network IP** type.

New MySQL Data Sources

* Type: no public ip
this type of data sources need to use custom scheduling
resources group can be carried out simultaneously, click here for [help manual](#)

* Name: mysql_source

Description: mysql

* select resources: Default resource group
group: [additional resources group](#)

* JDBC URL: jdbc:mysql://ServerIP:Port/Database

* Username: defwg

* Password:

Test Connectivity: [Test Connectivity](#) No public IP data source does not support testing connectivity.

[Previous](#) [Complete](#)

Configurations:

- Data source type: data source without a public IP address.
- Data source name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- Resource Group: It is used to run synchronization tasks, and generally multiple machines can be bound when you add a resource group. For more information, see [Add scheduling resources](#).
- JDBC URL: JDBC URL, in the format of jdbc://mysql://serverIP:Port/Database.
- User name/Password: The user name and password used to connect to the database.

7. Click **Test Connectivity**.

8. When the connectivity test is passed, click **OK**.

Connectivity test description

- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- Private Network and no public network IP, data source connectivity test is currently not supported, click **confirm**.

Next step

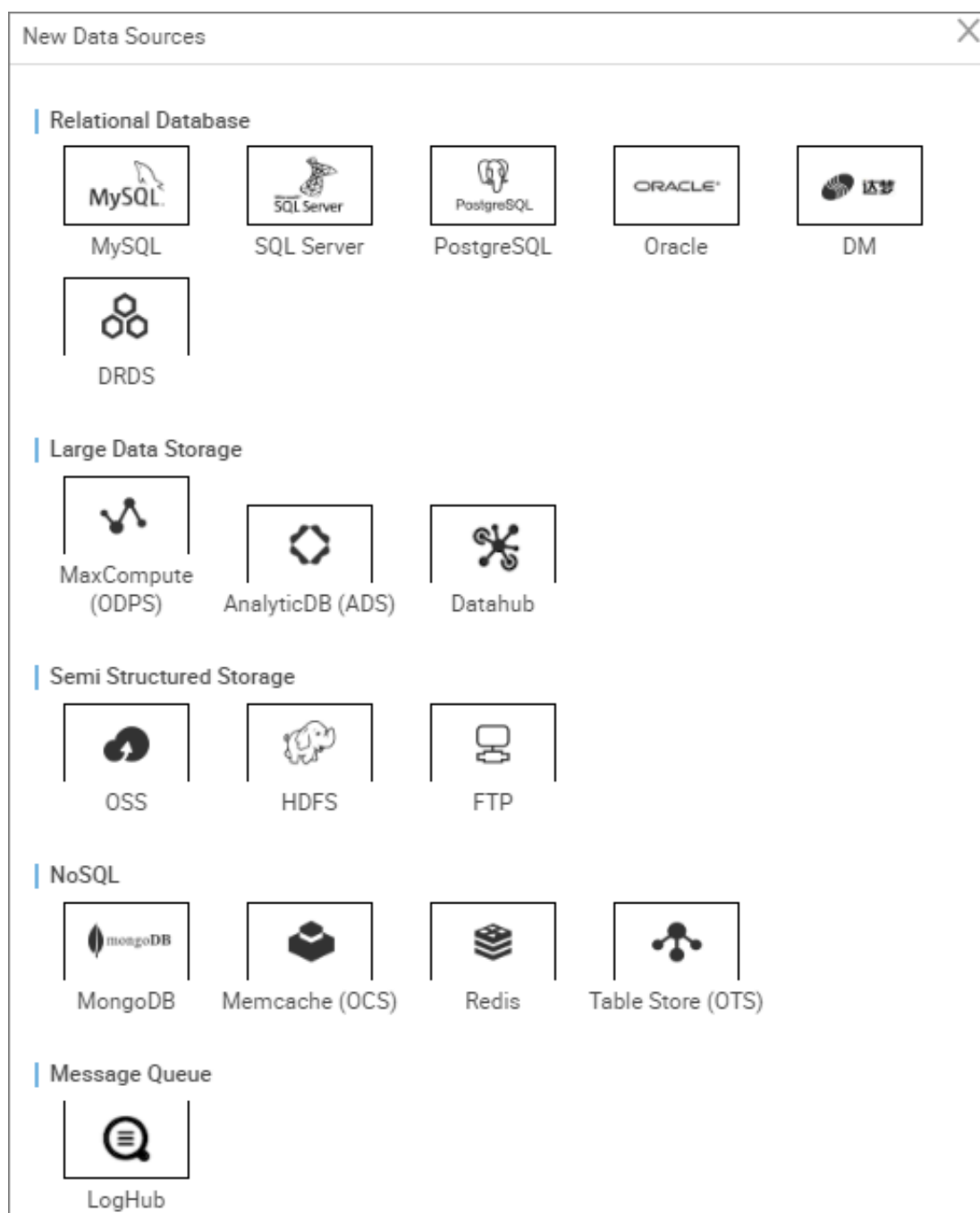
Now you have learned how to configure the MySQL data source. The document explains how to configure the MySQL Writer plug-in later. For more information, see [Configure MySQL Writer](#).

2.2.14 Configure Oracle data source

The Oracle data source allows you to read data from and write data to Oracle, and supports configuring synchronization tasks in wizard mode and script mode.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **Oracle**.
5. Configure each item of information for the Oracle data source.

Oracle Data source types are divided into **public network IP** and **non-public network IP**, and you can choose according to your own situation.

Consider a data source that adds a new **Oracle > network IP** type.

New Oracle Data Sources

* Type

there are public ip

* Name

Oracle_source_ip

Description

Oracle

* JDBC URL

jdbc:oracle:thin:@ServerIP:Port:Database

* Username

username

* Password

.....

Test Connectivity

Test Connectivity

Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous

Complete

Configurations:

- Type: With a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: Format: jdbc:oracle:thin:@serverIP:Port:Database.
- Username/Password: The user name and password used to connect to the database.

Consider a data source that adds a new **Oracle > network IP** type.

New Oracle Data Sources

* Type

no public ip

this type of data sources need to use custom scheduling
resources group can be carried out simultaneously, click here for
[help manual](#)

* Name

Oracle_source

Description

Oracle

* select resources

Default resource group

group [additional resources group](#)

* JDBC URL

jdbc:oracle:thin:@ServerIP:Port:Database

* Username

username


* Password

password

Test Connectivity

Test Connectivity

No public IP data source does not support
testing connectivity.



Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous

Complete

Configurations:

- Data Source Type: No public network IP, this type of data source requires custom scheduling resources for synchronization, you can click the **help manual** to view it.
- Data source name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: Format: jdbc:oracle:thin:@serverIP:Port:Database.
- Username/Password: The user name and password used to connect to the database.

6. Click **Test Connectivity**

7. When the connectivity test is passed, click **Complete**.

Connectivity test description

- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- Private Network and no public network IP, data source connectivity test is currently not supported, click **confirm**.

Next step

Now you have learned how to configure the Oracle data source. The document explains how to configure the Oracle Writer plug-in later. For more information, see [Configuring Oracle Writer](#).

2.2.15 Configure OSS data source

Object Storage Service (OSS) is a massive, secure, and highly reliable cloud storage service offered by Alibaba Cloud.

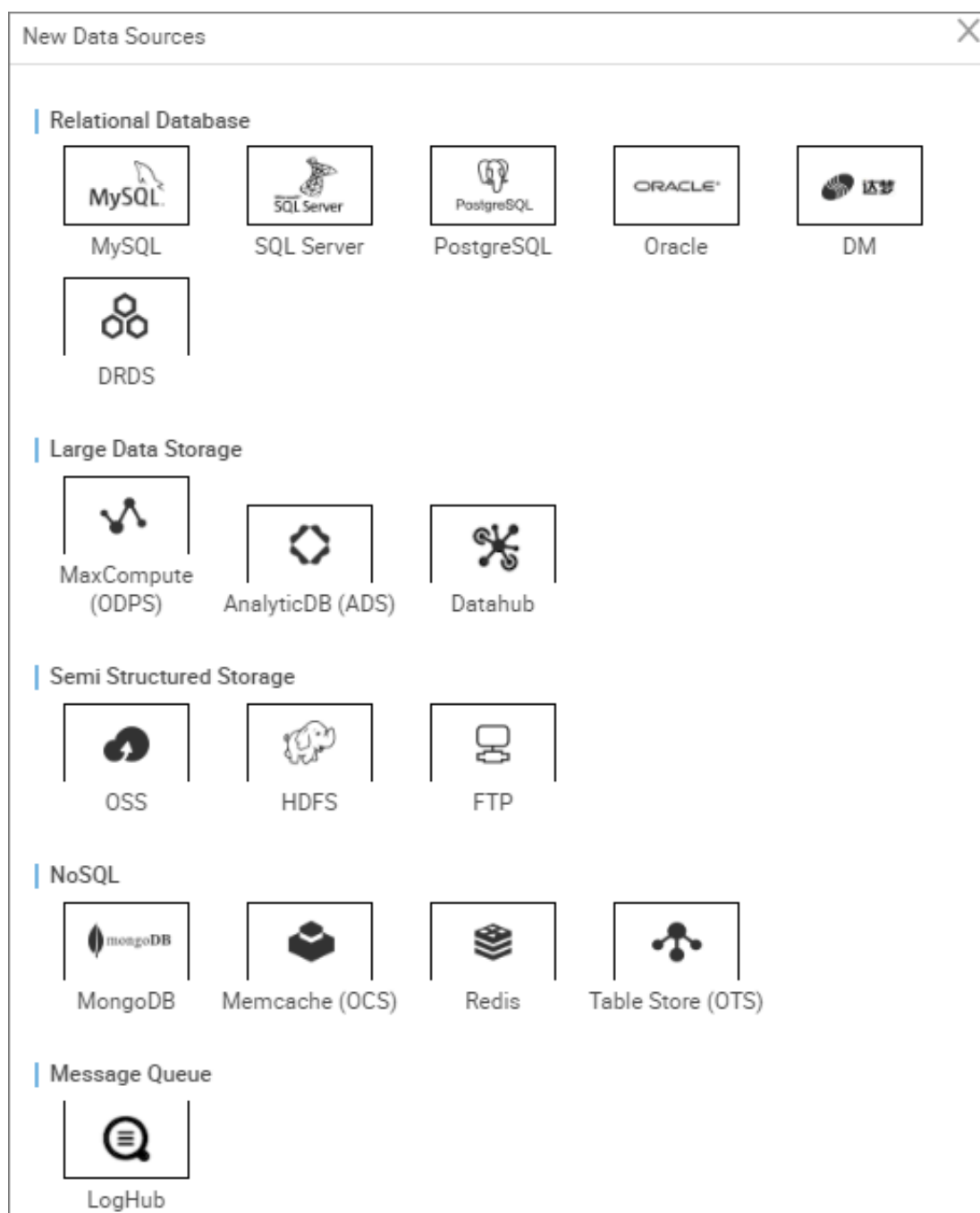


Note:

- If you want to learn more about [OSS products](#), see the OSS Product Overview.
- The OSS Java SDK can be found in the [Alibaba Cloud OSS Java SDK](#).

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New source** to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **OSS**.
5. Fill in configuration items for the OSS data source to be created.

Configurations:

- **Name:** It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- **Description:** It is a brief description of the data source with no more than 80 characters.
- **Endpoint:** OSS endpoint information, in the form of `http://oss.aliyuncs.com`, the Endpoint of the OSS service, and the Region, when you visit different Regions, you need to fill in different domain names.



Note:

The correct filling format for Endpoint is `http://oss.aliyuncs.com`, but add the bucket value before the OSS to connect `http://oss.aliyuncs.com` in the form of a point number, for example `http://xxx.oss.aliyuncs.com`, test connectivity can pass, but synchronization will report errors.

- **Bucket:** The bucket of the OSS instance. The bucket is a storage space and serves as the container for storing objects. You can create one or more buckets and add one or more files to each bucket. The bucket entered here searches for corresponding files in the data synchronization task, and file searching is unavailable for non-added buckets.
- **AccessID/AcessKey:** the [access key](#) (AccessKeyId and AccessKeySecret) is equivalent to the logon password.

6. Click **Test Connectivity**

7. When the connectivity test is passed, click **Complete**.

Connectivity test description

- The connectivity test is available in the classic network to identify whether the input Endpoint/AK information is correct.
- The data source connectivity test is currently not supported by the proprietary network, and you can click **confirm**.

Next step

Now you have learned how to configure the OSS data source. The document explains how to configure the OSS Writer plug-in later. For more information, see [Configure OSS Writer](#).

2.2.16 Configure Table Store(OTS) data source

Table Store is a NoSQL database service that built on Alibaba Cloud's Apsara distributed file system, enabling you to store and access massive volumes of structured data in real time.

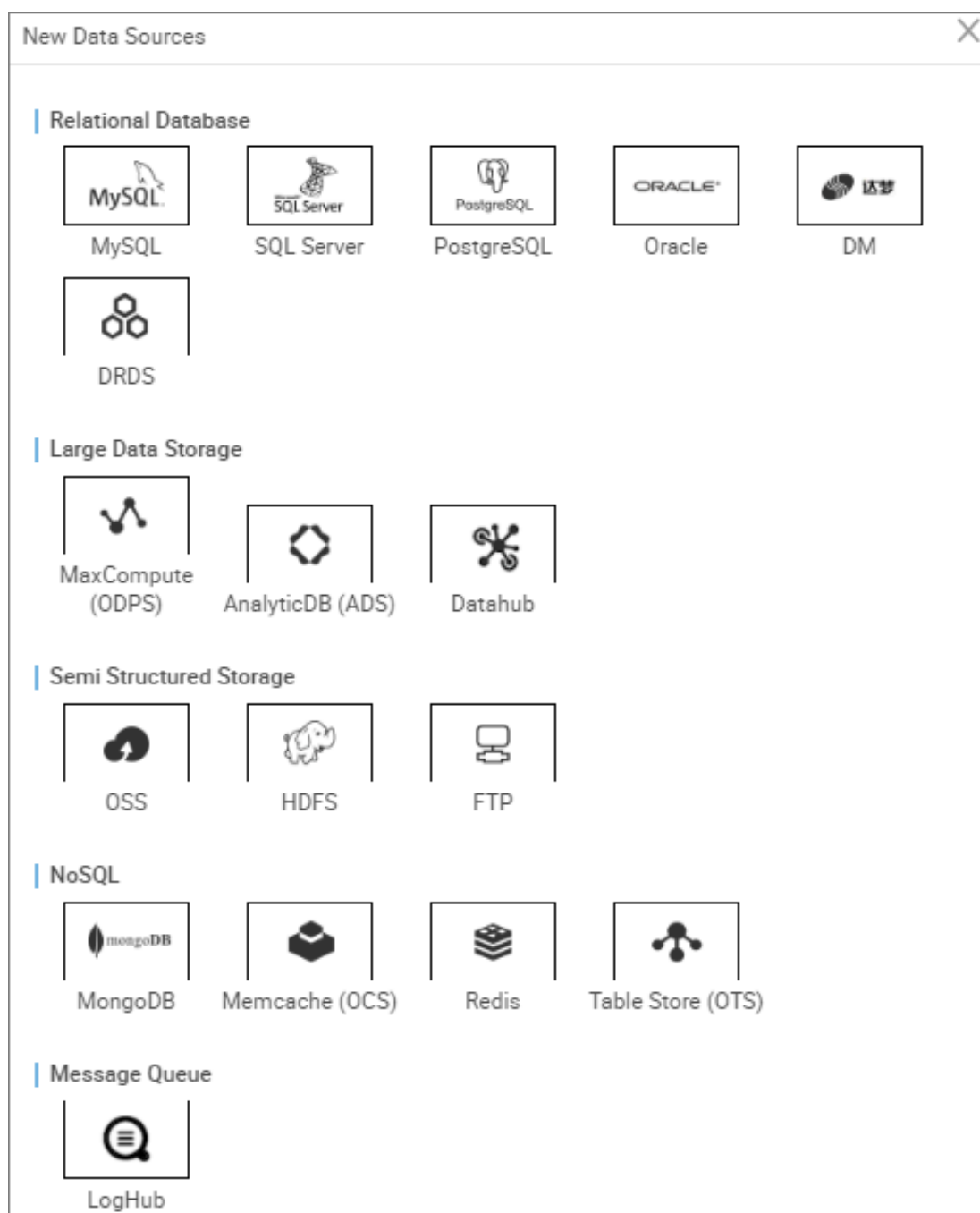


Note:

If you want to learn more about Table Store, see the [Table Store](#) Product Overview.

Procedure

1. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
2. Click **New source** to pop up the supported data source.



3. In the new data source dialog box, select the data source type as **Table Store(OTS)**.
4. Complete the configuration items for the Table Store data source.

New Table Store (OTS) Data Sources

* Name

Description

* Endpoint ?

* Table store ?

Instance ID

* Access Id ?

* Access Key

Test Connectivity

Configurations:

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Endpoint: the endpoint of the table store server in the format `http://yyy.com`. For more information, see [Endpoint](#).
- Table Store Instance ID: Instance ID corresponding to the Table Store service.
- AccessID/AceessKey: the [access key](#) (AccessKeyID and AccessKeySecret) is equivalent to the logon password.

5. Click **Test Connectivity**

6. When the connectivity test is passed, click **Complete**.

Connectivity test description

- The connectivity test is available in the classic network to identify whether the input endpoint/AK information is correct.
- The data source connectivity test is currently not supported by the proprietary network, and you can click **confirm**.

2.2.17 Configure PostgreSQL data source

The PostgreSQL data source allows you to read data from and write data to PostgreSQL, and supports configuring synchronization tasks in wizard mode and script mode.



Note:

If PostgreSQL is in a VPC environment, you need to be aware of the following issues.

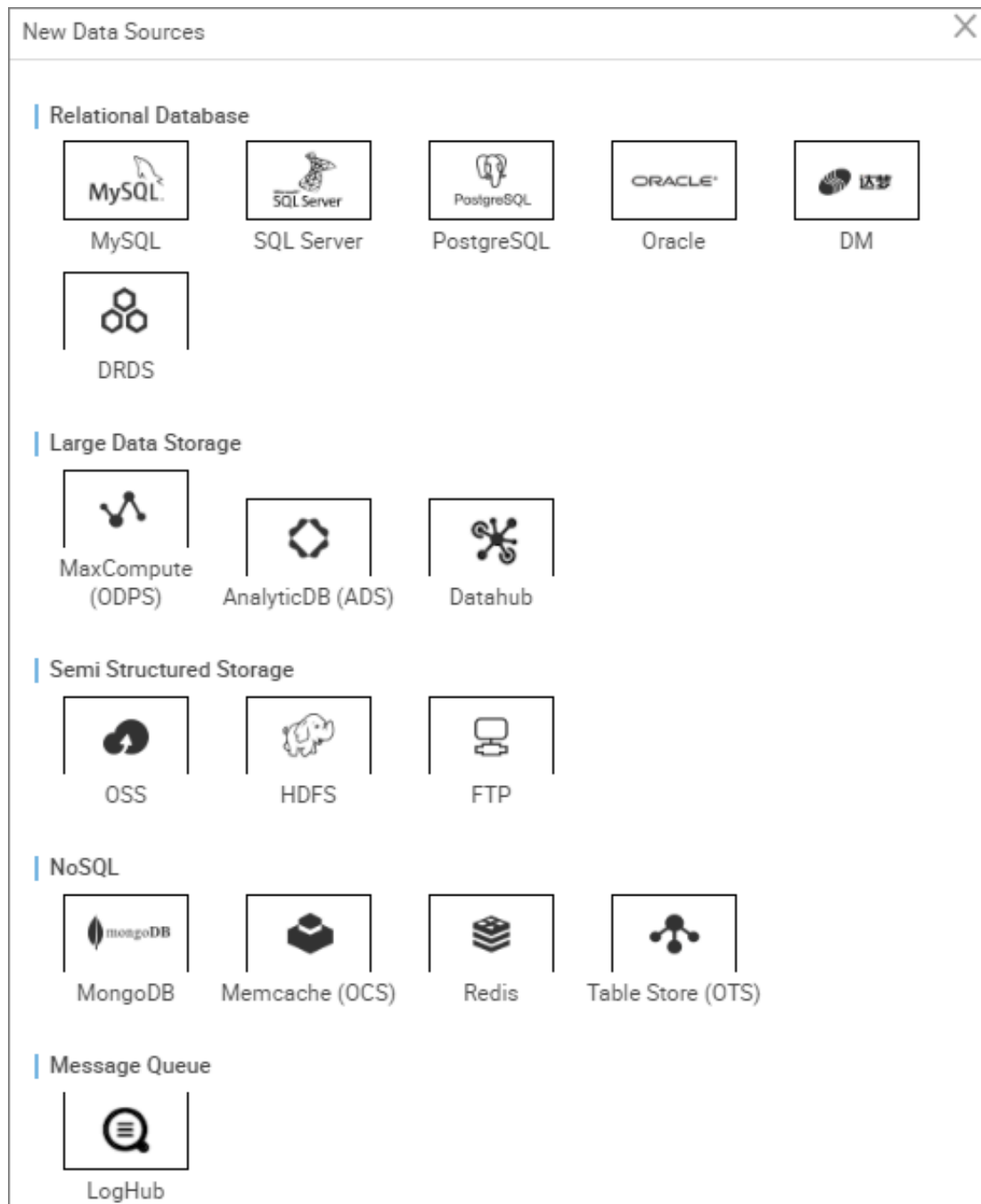
- Self-built PostgreSQL Data Source
 - Test connectivity is not supported, but the configuration synchronization task is still supported, and you can click **confirm** when creating the data source.
 - You must use a custom scheduled Resource Group to run the corresponding synchronization tasks, make sure that the Custom Resource Group can connect to your self-built database. For more information, see [Data integration when the network of data source \(one side only\) is disconnected](#) and [Data sync when the network of data source \(both sides\) is disconnected](#).

- PostgreSQL data sources created with RDS

You do not need to select a network environment, and the system automatically determines based on the information you fill in for the RDS instance.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New Source** source to pop up the supported data source.



4. In the new data source dialog box, select the data source type as **PostgreSQL**.
5. Configure individual information items for the PostgreSQL data source.

PostgreSQL data source types are divided into **Alibaba cloud database (RDS)**, **Public Network IP**, and **non-public network IP**, you can choose according to your situation.

Consider a data source of the new **PostgreSQL > Ali cloud database (RDS)** type.

New PostgreSQL Data Sources

* Type

ali cloud database (rds)

* Name

PostgreSQL_source_rds

Description

PostgreSQL

* Instance ID of RDS

PostgreSQL

* Main Buyer of RDS

PostgreSQL

* Database Name

PostgreSQL

* Username

PostgreSQL

* Password

.....

Test Connectivity

Test Connectivity

ⓘ

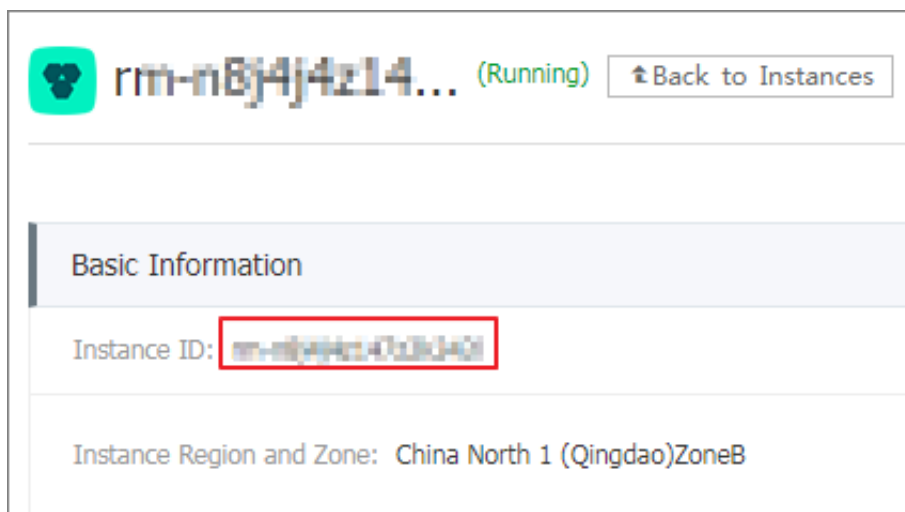
Will need to add rds white list can connect successfully, point i checked to see how to add the white list .
Ensure that the database can be network access
Ensure that the database is not a firewall prohibits
Ensure that the database can be parsed by the domain name
Ensure that the database has been launched

Previous

Complete

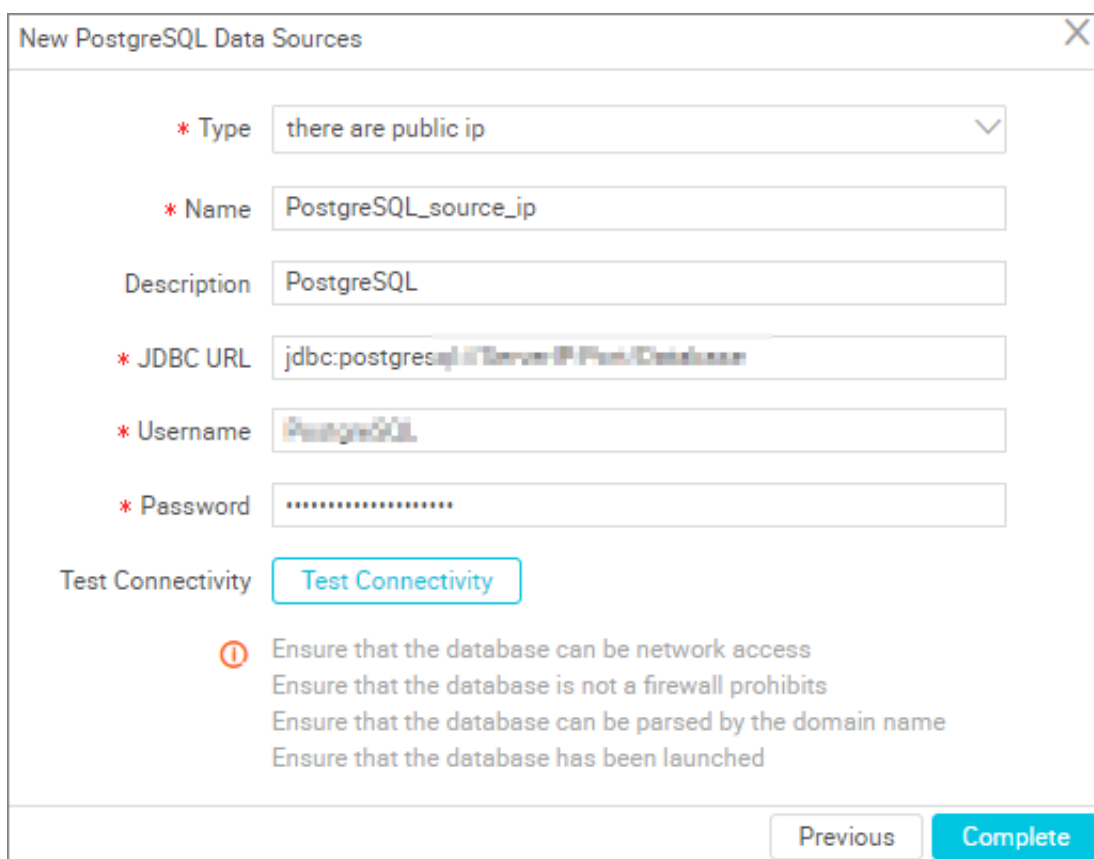
Configurations:

- Type: Alibaba cloud database (RDS).
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- RDS instance ID: You can view the instance id of the RDS in the control desk of the RDS.



The screenshot shows a DataWorks instance page. At the top, there is a green icon with three dots, followed by the instance name 'rm-n8j4j4z14...' and the status '(Running)'. A button labeled 'Back to Instances' is to the right. Below this is a section titled 'Basic Information'. Under 'Instance ID:', the value 'rm-n8j4j4z14...' is displayed and highlighted with a red rectangle. Below that, 'Instance Region and Zone:' is shown with the value 'China North 1 (Qingdao)ZoneB'.

Consider a data source that adds a **PostgreSQL > with a common network IP** type.



The screenshot shows a 'New PostgreSQL Data Sources' configuration window. It contains the following fields and options:

- Type:** A dropdown menu with the selected option 'there are public ip'.
- Name:** A text input field containing 'PostgreSQL_source_ip'.
- Description:** A text input field containing 'PostgreSQL'.
- JDBC URL:** A text input field containing 'jdbc:postgresql://ServerIP:Port/database'.
- Username:** A text input field containing 'PostgreSQL'.
- Password:** A password input field with masked characters.
- Test Connectivity:** A button labeled 'Test Connectivity'.
- Instructions:** A list of instructions for ensuring network access:
 - Ensure that the database can be network access
 - Ensure that the database is not a firewall prohibits
 - Ensure that the database can be parsed by the domain name
 - Ensure that the database has been launched
- Navigation:** 'Previous' and 'Complete' buttons at the bottom right.

Configurations:

- Type: With a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- JDBC URL: Format: jdbc:mysql://ServerIP:Port/database.

- Username/Password: The user name and password used to connect to the database.

Consider the new **PostgreSQL > Data Source with no public network IP** type.

Configurations:

- Type: data source without a public IP address.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Resource Group: It is used to run synchronization tasks, and generally multiple machines can be bound when you add a resource group. For details, see [Add scheduling resources](#).
- JDBC URL: Format: jdbc:mysql://ServerIP:Port/database.
- Username/Password: The user name and password used to connect to the database.

6. Click Test Connectivity

7. When the connectivity test is passed, click Complete.

Connectivity test description

- The connectivity test is available in the classic network arrangement, to identify whether the input JDBC URL, user name, and password are correct.
- Private Network and no public network IP, data source connectivity test is currently not supported, click **Complete**.

Next step

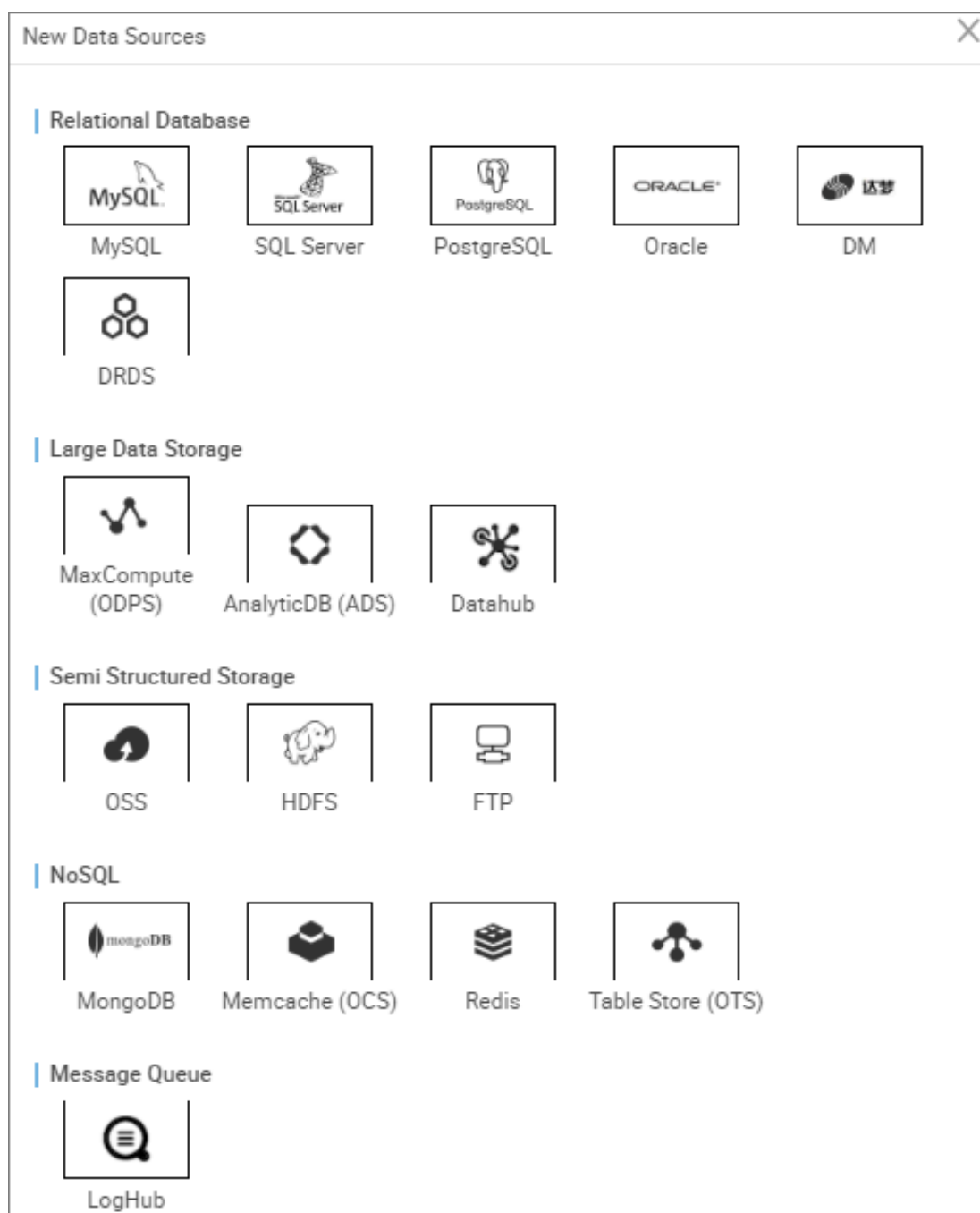
Now you have learned how to configure the PostgreSQL data source. The document explains how to configure the PostgreSQL Writer plug-in later. For more information, see [Configure PostgreSQL Writer](#).

2.2.18 Configure Redis data source

Redis is a document-based NoSQL database that provides persistent memory database services. Based on its highly reliable master/slave hot backup architecture and seamlessly scalable cluster architecture, this service can meet the needs of businesses that require high read/write performance and flexible capacity configuration. The Redis data source allows you to read data from and write data to Redis, and supports configuring synchronization tasks in script mode.

Procedure

1. Log on to the [DataWorks console](#) as an administrator and click **Enter Workspace** in the actions column of the relevant project in the Project List.
2. Click **Data Integration** in the top navigation bar to go to the **Data Source** page.
3. Click **New Source** to pop up the supported data source.



4. In the new data source dialog box, select a data source type of **Redis**.
5. Complete the configuration items for the Redis data source.

The data source types of redis are divided into **Alibaba cloud database** and **public network IP self-built database**.

- Alibaba Cloud databases: These databases generally use classic networks. Classic networks within the same region can connect to each other, but those in different regions may not.

- User-created databases with public IPs: These databases generally use public networks, which may cause a certain cost.

Consider a data source that adds a new **Redis > Ali cloud database** type.

The screenshot shows a configuration window titled "New Redis Data Sources". It includes the following fields and controls:

- * Type:** A dropdown menu with "ali cloud database" selected.
- * Name:** A text input field containing "Redis_source".
- Description:** A text input field containing "Redis".
- * area:** A dropdown menu with a selected region (partially obscured).
- * Instance ID of Redis:** A text input field containing a blurred instance ID, with a help icon (?) to its right.
- redis access password:** A text input field filled with dots to mask the password.
- Test Connectivity:** A button with the text "Test Connectivity".
- Navigation:** "Previous" and "Complete" buttons at the bottom right.

Configurations:

- Type: the currently selected data source type is Redis> Ali cloud database.



Note:

If you have not already authorized the default role for the data integration system, you need the master account to go to ram to authorize the role, then refresh the page.

- Name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Region: refers to the region selected when purchasing redis.
- Instance ID of Reids: You can go to the Redis console to view the redis instance ID.
- Redis access password: the access password for the Redis Server, and does not fill in if not .

Consider a data source that adds a new **Redis > Alibaba cloud database** type.

New Redis Data Sources

* Type: there are public ip

* Name: Redis_source_ip

Description: Redis

* Server address: 10.10.10.10 6379

add visit address

* redis access password:

Test Connectivity: Test Connectivity

Previous Complete

Configurations:

- Type: the currently selected data source type is redis > self-built database with public network IP.
- Name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Description: It is a brief description of the data source with no more than 80 characters.
- Access address: The format is host:port.
- Add an Access Address: Add an access address in the format of host:port.
- Redis access password: Redis's service access password.

6. Click Test Connectivity

7. When the connectivity test is passed, click Complete.

Next step

Now you have learned how to configure the Redis data source. The document explains how to configure the Redis Writer-plugin in later. For more information, see [Configure Redis Writer](#).

2.3 Task Configuration

2.3.1 Data Synchronization task configuration

2.3.2 Configure Reader plug-in

2.3.2.1 Script mode configuration

This article will show you how to configure tasks through the data integration Script Mode.

The steps for task configuration are as follows:

1. Create data source
2. Create a synchronization task
3. Import a template.
4. Configure the synchronization task reader.
5. Configure the synchronization task writer.
6. Configure the mapping between the synchronization task reader and the synchronization task writer.
7. Configure the DMUs, concurrency, transmission rate, dirty data records, resource groups, and other information of synchronization tasks.
8. Configure the scheduling attribute of the synchronization task.

**Note:**

You will be introduced below to the specific implementation of the operation steps, each of the following steps jumps to the corresponding guidance document, after completing the current step, click the link back to this article to continue to the next step.

Create data source

Synchronization tasks support data transmission between various homogenous data sources and heterogeneous data sources. First, register the target data source at Data Integration. Then you can select the data source directly when configuring a synchronization task on Data Integration. Data integration the data source types that support synchronization are shown in [Supported data sources](#).

After confirming that the target data source is supported by Data Integration, you can register the data source at Data Integration. For detailed data source registration, see [configuring data source information](#).

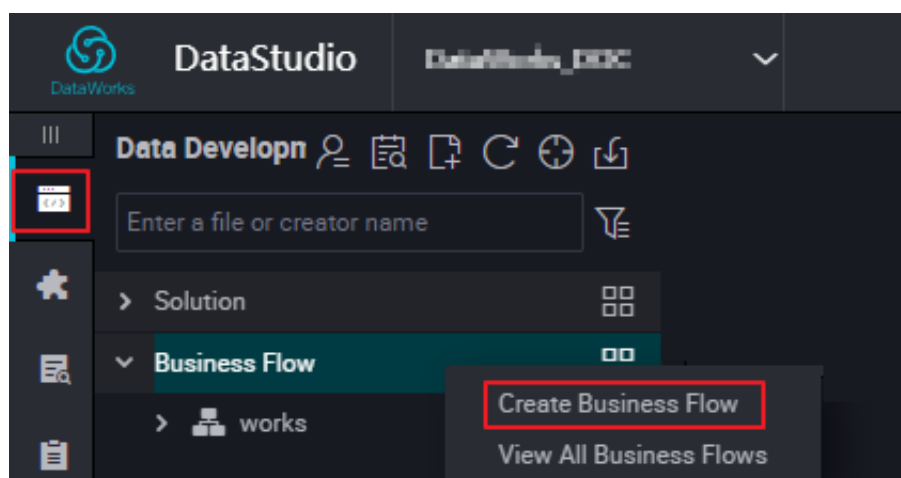
**Note:**

- For some data sources Data Integration does not support test connectivity. For more information on data source test connectivity, see [Data source testing connectivity](#).
- Many times, data sources are created locally and cannot be connected without a public network IP or network. In this case, testing connectivity at the time of configuration of the data source fails directly. Data Integration supports [Add scheduling resources](#) to solve this type of network inaccessibility.

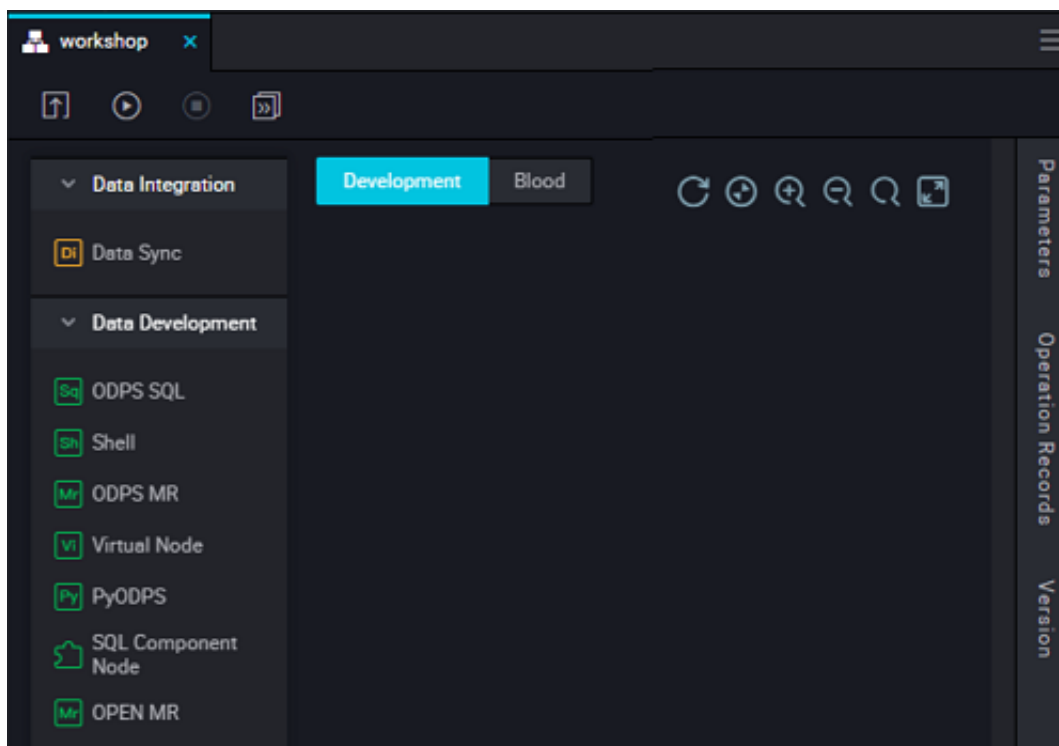
Create a synchronization task and the synchronization task reader**Note:**

This article mainly introduces you to the configuration of synchronization tasks in script mode, select **Script Mode** when creating new synchronization tasks in dataset generation.

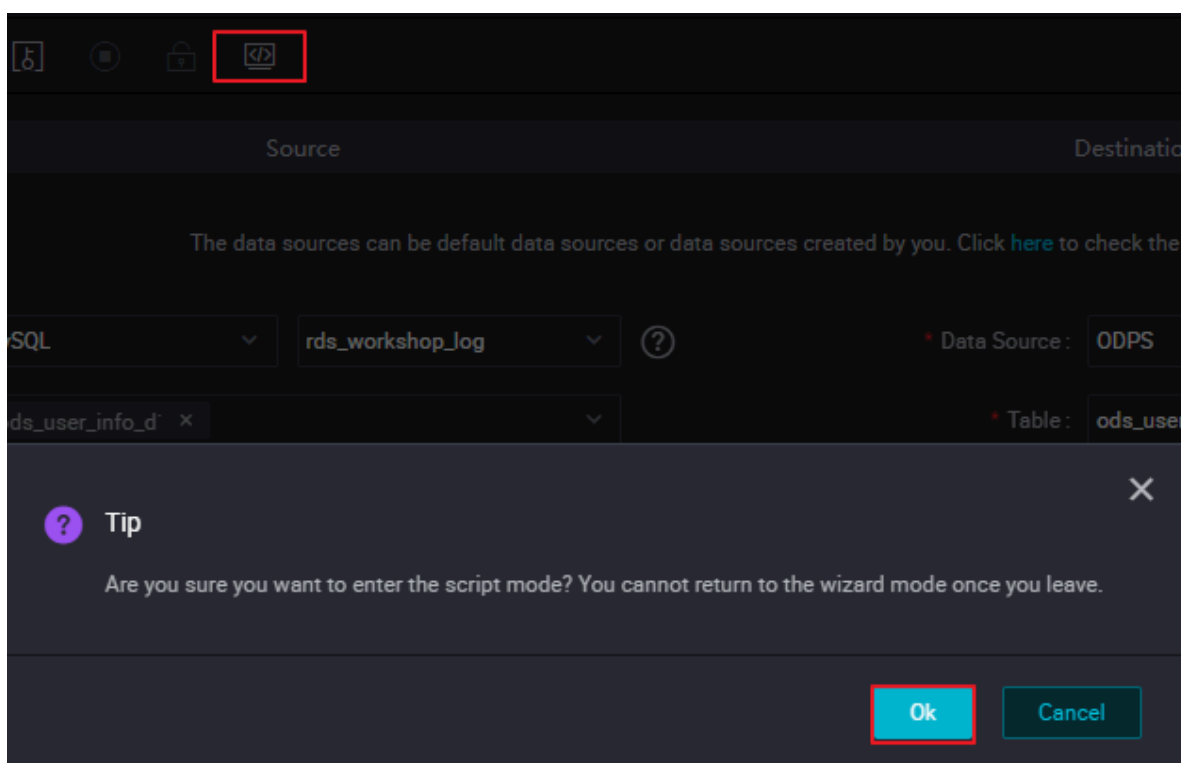
1. Enter the [DataWorks management console](#) as a developer, and click **Data Development** in the corresponding project action bar.
2. Click **Data Development** in the left-hand menu bar to open the Business Process navigator.



3. Right-click **Business Flow** in the navigation bar, create **Data Integration > Data Sync**, and enter the synchronization Task Name.



4. After successfully creating the synchronization node, click the **Switch to Script Mode** in the upper-right corner of the new synchronization node, select **Ok** to enter Script Mode.

**Note:**

Script Mode supports more features, such as synchronous task editing if the network is not up to date.

5. Click **Import template** in the upper-right corner of the script pattern, in the bullet box, select the source type of the read and the data source, the target type of the write, and the data source respectively, click **confirm** to generate the initial script.

The screenshot shows the 'Import Template' dialog box. It has a title bar with a close button (X). Below the title bar, there are two tabs: 'mysql Reader Help Document' and 'odps Writer Help Doc'. The main content area contains four fields:

- * Source Type : ODPS (dropdown menu with a help icon)
- * Data Source : 请选择 (dropdown menu with a help icon and a link 'Add Data Source' below it)
- * Destination Type : ODPS (dropdown menu with a help icon)
- * Data Source : 请选择 (dropdown menu with a help icon and a link 'Add Data Source' below it)

At the bottom right, there are 'OK' and 'Cancel' buttons. A red rectangle highlights the four fields.

Configure the synchronization task reader

After the synchronization task is created, basic configurations of the reader are generated with the imported template. Now you can manually configure the reader data source and the target table information for the data synchronization task.

```
{ "type": "job",  
  "version": "2.0",  
  "Steps": [// above is configured for the entire synchronization  
task header code, do not make modifications. The reader configurations  
are as follows:  
    {  
      "stepType": "mysql",  
      "parameter": {  
        "datasource": "MySQL",  
        "column": [  
          "id",
```

```

        "value",
        "table"
    ],
    "socketTimeout": 3600000,
    "connection": [
        {
            "datasource": "MySQL",
            "table": [
                "`case`"
            ]
        }
    ],
    "where": "",
    "splitPk": "",
    "encoding": "UTF-8"
},
"name": "Reader",
"category": "reader" // description classified as reader
read end
    }, //The above are reader configurations.

```

Configurations:

- **Type:** Specifies the synchronization task for this submission, only the job parameter is supported, so you can only fill in as a job.
- **Version:** the version number currently supported by all jobs is 1.0 or 2.0.

When you configure the read side, for specific parameter settings and code descriptions, see the Script Mode section in [configuring reader](#).



Note:

Many tasks require incremental synchronization of data when configuring read-side data sources, you can now get the date in conjunction with what DataWorks provides to complete the requirement [Parameter configuration](#) to get the incremental data.

Configure the synchronization task writer

After the reader data source is configured, you can manually configure the writer data source and the target table information for the data synchronization task.

```

{ //The writer configurations are as follows:
  "stepType": "odps",
  "parameter": {
    "partition": ""
    "truncate": true,
    "compress": false,
    "datasource": "odps_first",
    "column": [
      "*"
    ],
    "emptyAsNull": false,
    "table": ""
  },
  "name": "Writer",

```



```

        "category": "writer" // instructions are classified as writer
    write end
    }
}, //The above are reader configurations.

```

When you configure write-side information, see the Script Mode section of [configuring writer](#).



Note:

For most tasks, you need to select a write mode based on data sources, such as overwrite mode or append mode. If you have write control requirements, see [configuring writer](#) to choose write mode properly.

Configure mapping

The script mode only supports in-row mapping, that is, the reader "columns" correspond to the writer "columns" one by one from top to bottom.



Note:

Note whether the field types mapped between columns are data compatible.

Synchronous task efficiency settings

When the above steps are configured, the efficiency configuration is required. The **setting** domain describes the job configuration parameters in addition to the source and destination, configuration parameters for job global information. Efficiency can be configured in the setting field, including DUM setting, synchronization concurrency setting, synchronization rate setting, dirty data setting, and resource group setting.

```

"setting": {
    "errorLimit": {
        "record": "1024" // dirty data entry settings
    },
    "speed": {
        "throttle": false, // do you want to limit the speed?
        "concurrent": 1, // synchronous concurrency number
    },
    "dmu": 1 // DMU quantity settings
},

```

Configurations:

- DMU: the unit of charge for data integration.



Note:

When setting up a DMU, be aware that the value of the DMU limits the value of the maximum number of concurrency, please configure properly.

- When you configure ****Synchronization Concurrency****, the data records are split into several tasks based on the specified reader splitting key. These tasks run simultaneously to improve the transmission rate.
- Synchronous rate: Setting the synchronous rate protects the read-side database from excessive extraction speed, put too much pressure on the source library. It is recommended to throttle the synchronization rate and configure the extraction rate properly based on source database configurations.
- Dirty Data is mainly set to control the quality of synchronized data. It supports setting a threshold of dirty data records. If the number of dirty data records exceeds the threshold during job transmission, the job is aborted with an error. For example, in the configuration above, you specified a maximum **error limit** of 1024 records, when the job has a dirty data record number greater than 1024 during the transfer process, the job reports an error to exit.
- You can specify a resource group configuration by clicking **configure task resource groups** in the upper-right corner of the current page.

When you configure a synchronization task, you specify the resource group in which the task runs, default runs on the default Resource Group. When the project has a tight schedule of resources, you can also expand a scheduled resource by adding a Custom Resource Group, the synchronization task is then specified to run on a Custom Resource Group, to add a Custom Resource Group, see adding a scheduled resource. You can make a reasonable configuration based on data source network conditions, project scheduling resource conditions, and business importance.

**Note:**

When synchronizing data is not efficient, see [Optimizing configuration](#) to optimize your synchronization tasks.

Configure scheduling properties

In the scheduling properties, you can set the synchronization task run cycle, run time, task dependency, and so on. Since the synchronization task is the beginning of the ETL job, there are no upstream nodes, at this point, it is recommended to use the project root node as upstream.

After completing the configuration of the synchronization task, save the node and submit.

2.3.2.2 Wizard mode configuration

This article will show you how to configure tasks through the Data Integration wizard mode.

The steps for task configuration are as follows:

1. Create data source
2. Create a synchronization task and configure the synchronization task reader.
3. Configure the synchronization task writer.
4. Configure the mapping between the synchronization task reader and the synchronization task writer.
5. Configure the concurrency, transmission rate, dirty data records, resource groups, and other information of the synchronization task.
6. Configure the scheduling attribute of the synchronization task.



Note:

You will be introduced to the specific implementation of the operation steps, each of the following steps jumps to the corresponding guidance document, after completing the current step, click the link back to this article to continue to the next step.

Create data source

Synchronization tasks support data transmission between various homogenous data sources and heterogeneous data sources. First, register the target data source at Data Integration. Then you can select the data source directly when configuring a synchronization task on Data Integration.

After confirming that the target data source is supported by Data Integration, you can register the data source at Data Integration. For detailed data source registration, see [configuring data source information](#).



Note:

- For some data sources Data Integration does not support test connectivity. For more information on data source test connectivity, see [Data source testing connectivity](#).
- Many times, data sources are created locally and cannot be connected without a public network IP or network. In this case, testing connectivity at the time of configuration of the data source fails directly. Data Integration supports [Add scheduling resources](#) to solve this type of network inaccessibility.

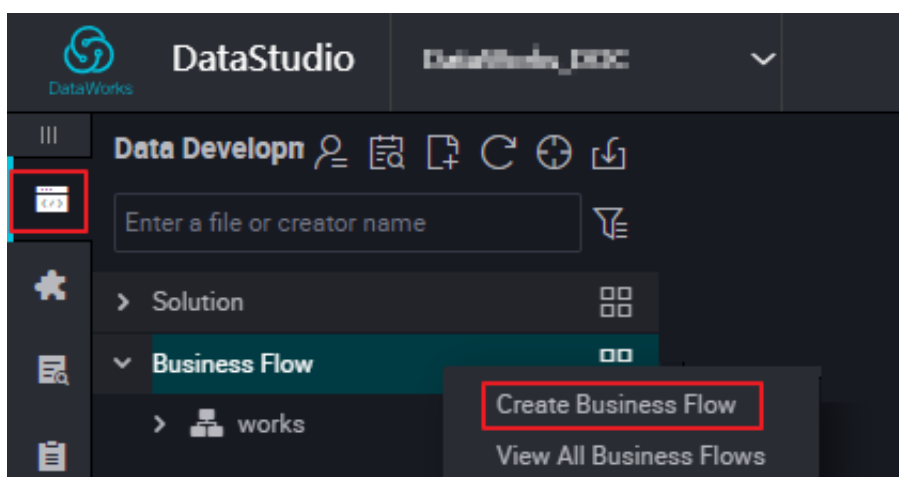
Create a synchronization task and the reader



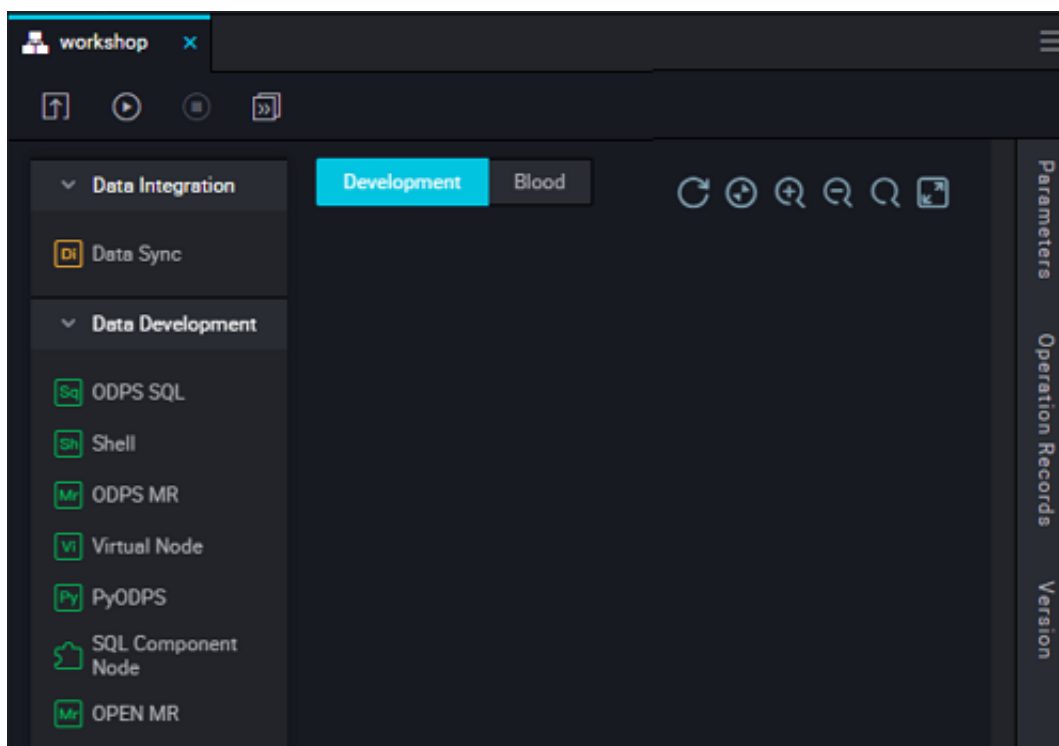
Note:

This article mainly introduces you to the synchronization task configuration in wizard mode. Select **wizard mode** when creating new synchronization tasks.

1. Enter the [DataWorks management console](#) as a developer, and click **Data Development** in the corresponding project action bar.
2. Click **Data Development** in the left-hand menu bar to open the Business Process navigator.



3. Right-click **Business Flow** in the navigation bar, create **Data Integration node** > **Data Sync**, and enter the synchronization task's name.



4. After the synchronization task is created, you can continue to manually configure the reader data source and the target table information for the data synchronization task. When you are selecting a data source to read from, see [configuring reader](#).

**Note:**

Many tasks require incremental synchronization of data when configuring read-side data sources, you can now get the relative date in conjunction with [Parameter configuration](#) to complete the requirement to get the incremental data.

Configure the writer

After the reader data source is configured, you can continue to manually configure the writer data source and the target table information for the data synchronization task. When you are selecting the data source to write on, see [configuring writer](#).

**Note:**

For most tasks, you need to select a write mode based on data sources, such as overwrite mode or append mode. For students with write control requirements, refer to the [Configuring writer](#) documentation to choose the write mode properly.

Configure mapping

When the configuration of both the Read and Write side is complete, you need to specify a mapping relationship between the read and write end columns, and you can choose **Map of the same name** or **Enable Same-line Mapping**.

- Enable Same-line Mapping: automatically sets the mapping relationship for the same row of data.
- Automatic Layout: After the mapping relationship is set, the field ordering display is displayed.

**Note:**

The field types mapped between columns should be data compatible.

Channel

When the above steps are configured, the efficiency configuration is required. Efficiency configuration mainly includes DMU settings, synchronous concurrency number settings, synchronous rate settings, synchronous dirty data settings and synchronizing information such as resource group settings.

Parameters:

- DMU: the unit of charge for Data Integration.

**Note:**

When setting up a DMU, be aware that the value of the DMU limits the value of the maximum number of concurrency, please configure properly.

- When you configure Synchronization Concurrency, the data records are split into several tasks based on the specified reader splitting key. These tasks run simultaneously to improve the transmission rate.
- Synchronous rate: Setting the synchronous rate protects the read-side database from excessive extraction speed, put too much pressure on the source library. It is recommended

to throttle the synchronization rate and configure the extraction rate properly based on source database configurations.

- For example, if the source has varchar type data but is written to a destination column having int type data, a data conversion exception occurs and the data cannot be written to the destination column. The dirty data is mainly set to control the quality of synchronized data. You should set an appropriate number of dirty data based on your business requirements.
- When you configure a synchronization task, you specify the resource group in which the task runs, default runs on the Default Resource Group. When the project has a tight schedule of resources, you can also expand a scheduled resource by adding a Custom Resource Group, the synchronization task is then specified to run on a Custom Resource Group. See [Add scheduling resources](#) for more information. You can make a reasonable configuration based on data source network conditions, project scheduling resource conditions, and business importance.

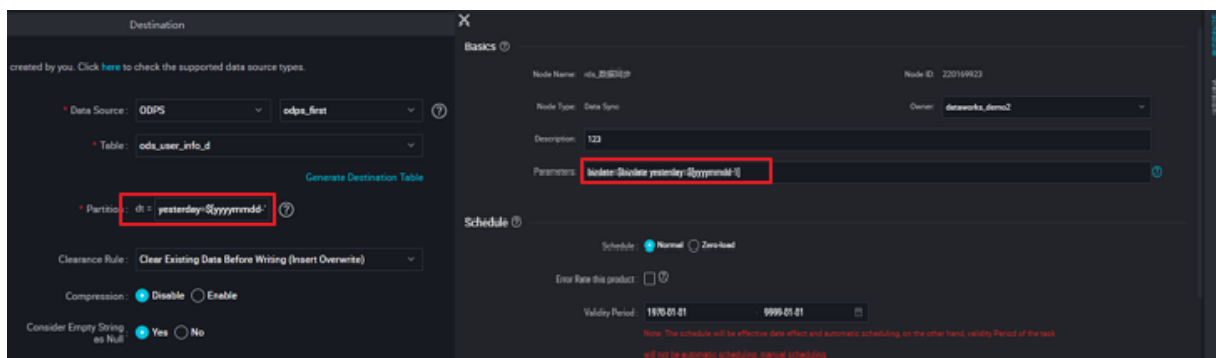


Note:

When synchronizing data is not efficient, see [Optimizing configuration](#) to optimize your synchronization tasks.

Scheduling parameters

You often need to use scheduling parameters to filter your data in synchronization tasks, you will be shown below how to configure scheduling parameters in the synchronization task.



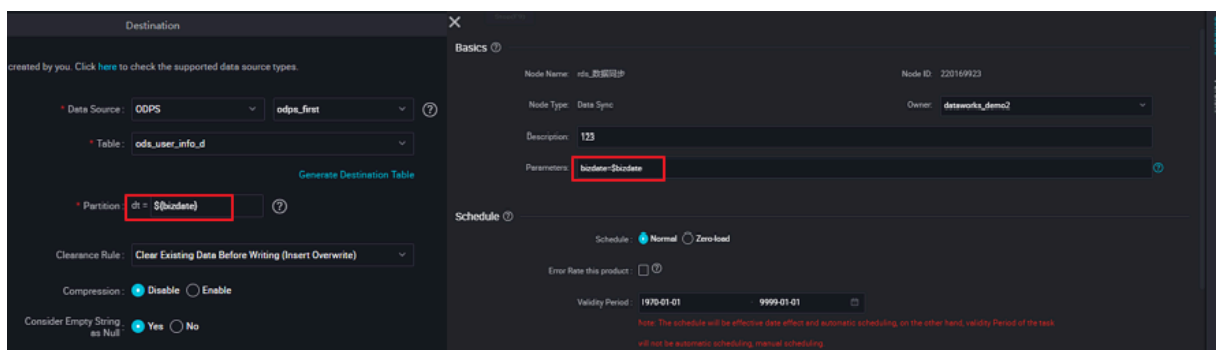
As shown in the figure above, you can declare a schedule parameter variable in the form of a \$ {variable name, when the variable declaration is complete, write the initialization value of the variable in the scheduled parameter properties, the value initialized here by the variable is identified by \$, the content can be either a time expression or a constant.

For example, \$ {today} was written in code, by assigning today = \$ [yyyymmdd] in the scheduling parameter, you can get the date of the day, to add-minus to a date, see [Parameter configuration](#).

Using custom schedule parameters in synchronization tasks

All you need to do in the synchronization task is declare the following parameters in your code.

- bizdate: Get to the business date, run date-1.
- cycetime: gets the current run time, in the form of yyyyymmddhhmiss.
- Dataworks provides two system default scheduling parameters, bizdate and cycetime.



Configure scheduling Properties

In the scheduling properties, you can set the synchronization task run cycle, run time, task dependency, and so on. Since the synchronization task is the beginning of the ETL job, there are no upstream nodes, at this point, it is recommended to use the project root node as upstream.

After completing the configuration of the synchronization task, save the node and submit.

2.3.2.3 Configure DRDS Reader

The DRDS Reader plug-in allows your to read data from DRDS (distributed RDS). At the underlying implementation level, DRDS Reader connects to a remote DRDS database through JDBC and runs corresponding SQL statements to SELECT data from the DRDS database.

Currently, the DRDS plug-in is only adapted to the MySQL engine. DRDS is a distributed MySQL database, and most of the communication protocols are applicable to MySQL use cases.

Specifically, DRDS Reader connects to a remote DRDS database through the JDBC connector. The SELECT SQL query statements are generated and sent to the remote DRDS database based on your configuration. Then, the SQL statements are run and the returned results are assembled into abstract datasets using the custom data types of data synchronization. Datasets are passed to the downstream writer for processing.

DRDS Reader concatenates the table, column, and WHERE information you configured into SQL statements and sends them to the DRDS database. Unlike the MySQL database, DRDS as a distributed database is unable to adapt to all MySQL protocols, and does not support complex clauses such as Join.

DRDS Reader supports most data types in MySQL. Check whether your data type is supported.

DRDS Reader converts MySQL data types as follows:

MySQL data type	DRDS Data Management
Integer	Int, tinyint, smallint, mediumint, and bigint
Floating point	Float, double, decimal
String	varchar, char, tinytext, text, mediumtext, or longtext
Date and time	date, datetime, timestamp, time, or year
Boolean	bit or bool
Binary	tinyblob, mediumblob, blob, longblob, or varbinary

Parameter description

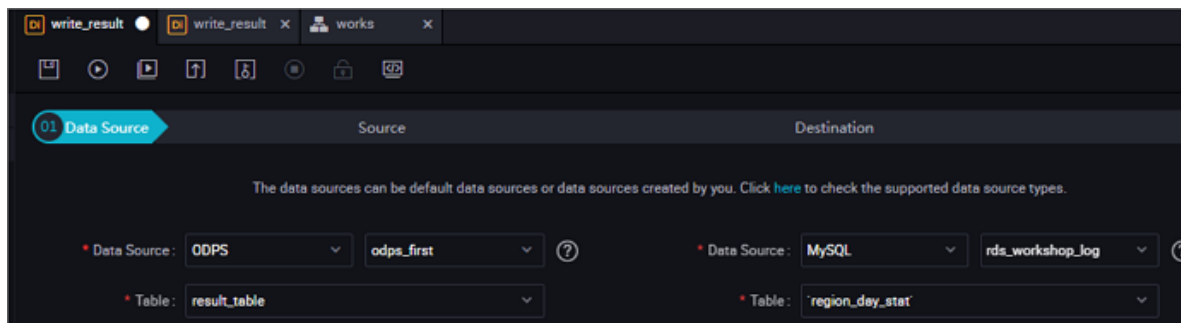
Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
table	The table selected for extraction.	Yes	N/A

Attribute	Description	Required	Default Value
column	<p>The column name set to be synchronized in the configured table. Field information is described with arrays in JSON. ["*"] indicates all columns by default.</p> <ul style="list-style-type: none"> - Column pruning is supported, which means you can select some columns to export. - Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. - Constant configuration is supported. You must follow the MySQL SQL syntax format, for example ["id", "`table`", "1", "bazhen.csy", "null", "to_char(a + 1)", "2.3", "true"], <ul style="list-style-type: none"> - where id refers to the ordinary column name, - `table` is the name of the column containing reserved words, - 1 is an integer constant, - 'bazhen.csy' is a string constant, - null refers to null pointer, - CHARLENGTH(s) is the function expression to calculate the string length, - 2.3 is a floating point, - and true is a boolean value. - Column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	N/A
where	<p>Filtering condition. DRDS Reader concatenates an SQL command based on specified column, table, and WHERE conditions and extracts data according to the SQL statement. For example, you can set the WHERE condition during a test. In actual business scenarios, the data on the current day is usually required to be synchronized, in which case you can set the WHERE condition to <code>STRTO_DATE('\${bdp.system.bizdate}', '%Y%m%d') <= today AND today < DATEADD(STRTO_DATE('\${bdp.system.bizdate}', '%Y%m%d'), interval 1 day)</code>.</p> <ul style="list-style-type: none"> - The where condition can be effectively used for incremental synchronization. - If the where condition is not set or is left null, full table data synchronization is applied. 	No	N/A

Development in wizard mode

1. Choose source

Configuration item descriptions:



Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: The table in the preceding parameter description. Select the table to be synchronized.
- Data filtering: You should synchronize the filter for the data. Limit keyword filter is not supported yet. SQL syntaxes vary with data sources.
- Splitting key: You can use a column in the source table as the splitting key. It is recommended to use a primary key or an indexed column as the splitting key. Only integer fields are supported.

During data reading, the data is split based on the configured fields to achieve concurrent reading, improving data synchronization efficiency. The configuration of splitting key is related to the source selection in data synchronization.

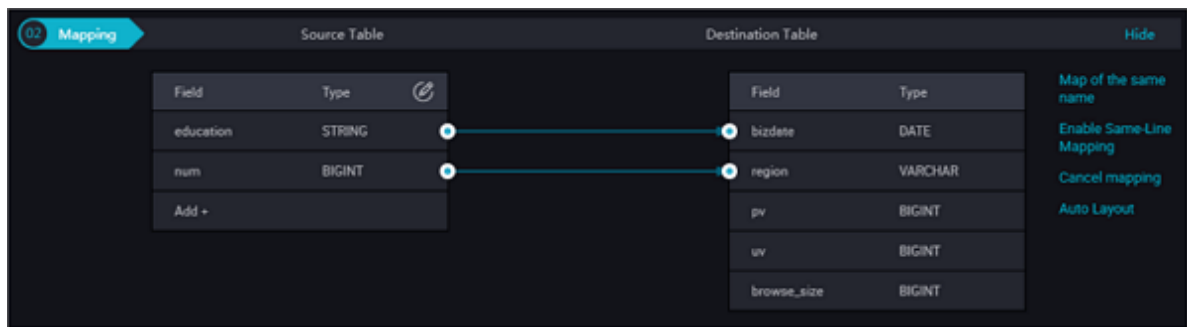


Note:

The splitting key configuration item is displayed only when you configure the data source.

2. Field mapping: The column in the preceding parameter description.

The Source Table Field on the left maps with the Target Table Field on the right. Click **Add Line**, and then a field is added. Hover the cursor over a line, click **Delete**, and then the line is deleted.

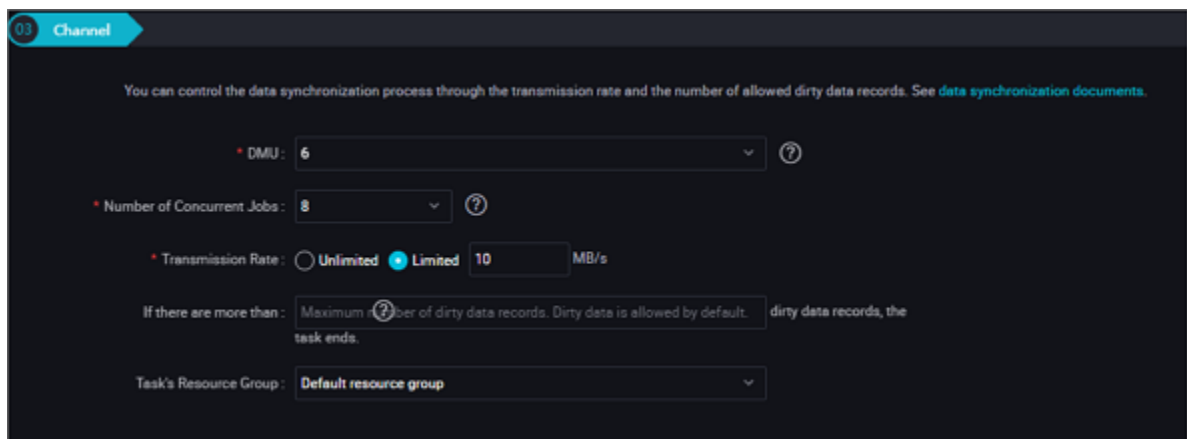


- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

By clicking Add Row,

- you can enter constants. Each constant must be enclosed in a pair of single quotes, such as 'abc' and '123'.
- Use this function with scheduling parameters, such as \${bizdate}.
- Enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as 'Not Identified'.

3. Channel control



Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.

- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- Number of error records: The maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups), see [Add scheduling resources](#).

Development in script mode

Configure a job to synchronously extract data from an RDBMS database:

```
{
  "type": "job",
  "version": "2.0", //Indicates the version.
  "steps": [
    {
      "stepType": "drds", //plug-in name
      "parameter": {
        "datasource": "", //Name of the data source
        "column": [ //column name
          "id",
          "name"
        ],
        "where": "", //Filtering condition
        "table": "", //The name of the target table.
        "splitPk": "", //Splitting key
      },
      "Name": "Reader ",
      "category": "reader"
    },
    { //You can locate the corresponding writer plug-in documentat
      ion among the following documentations.
      "stepType": "stream", //plug-in name
      "parameter": {},
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, // do you want to limit the flow?
      "concurrent": "1", // Number of concurrency
      "DMU": 1 // DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "from": "Reader",
        "to": "Writer"
      }
    ]
  }
}
```

```
    ]  
  }  
  }:"Writer"  
    }  
  ]  
}  
}
```

Additional information

Consistency view

As a distributed database, DRDS cannot provide a consistency view of multiple tables in multiple databases. Different from MySQL where data is synchronized in a single table of one database, DRDS Reader cannot extract the snapshot of database/table sharding at the same time slice, that is to say, DRDS Reader obtains different snapshots of table shards when extracting data from different underlying table shards. Therefore, strong consistency cannot be ensured.

Database coding

Drds provides flexible encoding options, including database-level, table-level, and field-level encoding. Different encodings can also be configured. The priority (from high to low) is field, table, database, and instance. We recommend that you use UTF-8 for database encoding at the database level.

DRDS Reader extracts data using JDBC at the underlying level. JDBC is applicable to all types of encodings and can complete transcoding at the underlying level. Therefore, DRDS Reader can identify the encoding and complete transcoding automatically without the need to specify the encoding.

DRDS Reader cannot identify the inconsistency between the encoding written to the underlying layer of DRDS and the configured encoding, nor provide a solution. Due to this issue, the exported codes may contain garbage codes.

Incremental Synchronization

Since DRDS Reader extracts data using JDBC SELECT statements, you can extract incremental data using the SELECT and WHERE conditions in the following ways:

- When database online applications write data into the database, the modify field is filled with the modification timestamp, including addition, update, and deletion (logical deletion). For this type of applications, DRDS Reader only requires the WHERE condition followed by the timestamp of the last synchronization phase.
- For new streamline data, DRDS Reader requires the WHERE condition followed by the maximum auto-increment ID of the last synchronization phase.

In case that no field is provided for the business to identify the addition or modification of data, DRDS Reader cannot perform incremental data synchronization and can only perform full data synchronization.

SQL security

DRDS Reader provides query SQL statements for you to SELECT data by yourself. DRDS Reader conducts no security verification on query SQL. The security during use is ensured by the data synchronization users.

2.3.2.4 Configure HBase Reader

The HBase Reader plug-in provides the ability to read data from HBase. At the underlying implementation level, HBase Reader connects to remote HBase service with HBase's Java client, reads data within the rowkey range you specified by means of Scan, then assembles data into an abstract dataset using custom data type for Data Integration, and passes the dataset to downstream Writer for processing.

Supported features

- **HBase 0.94.x and HBase 1.1.x versions are supported**

- If you use HBase 0.94.x, choose HBase094x as the reader plug-in, as shown in the following figure:

```
"reader": {
  "plugin": "hbase094x"
}
```

- If you use HBase 1.1.x, choose HBase11x as the reader plug-in, as shown in the following figure:

```
"reader": {
  "plugin": "hbase11x"
}
```

- **The normal and multiVersionFixedColumn modes are supported**

- normal mode: Read the latest version of data from an HBase table which is used as an ordinary two-dimensional table (horizontal table). For example:

```
hbase(main):017:0 is greater than scan 'users'
ROW COLUMN+CELL
lisi column=address:city, timestamp=1457101972764, value=beijing
lisi column=address:contry, timestamp=1457102773908, value=china
lisi column=address:province, timestamp=1457101972736, value=
beijing
lisi column=info:age, timestamp=1457101972548, value=27
lisi column=info:birthday, timestamp=1457101972604, value=1987-06-
17
```

```
lisi column=info:company, timestamp=1457101972653, value=baidu
xiaoming column=address:city, timestamp=1457082196082, value=
hangzhou
xiaoming column=address:contry, timestamp=1457082195729, value=
china
xiaoming column=address:province, timestamp=1457082195773, value=
zhejiang
xiaoming column=info:age, timestamp=1457082218735, value=29
xiaoming column=info:birthday, timestamp=1457082186830, value=1987
-06-17
xiaoming column=info:company, timestamp=1457082189826, value=
alibaba
2 row(s) in 0.0580 seconds }
```

The data read from the table is shown as follows:

rowKey	addres: city	address: contry	address: province	info: age	info:birthday	info: company
Lisi	beijing	china	beijing	27	1987-06-17	baidu
xiaoming	hangzhou	china	zhejiang	29	1987-06-17	alibaba

- multiVersionFixedColumn mode: Read data from an HBase table which is used as an vertical table. Each record read from the table is shown in the form of the following four columns: rowKey, family:qualifier, timestamp, and value. You must specify the column to be read when reading data. The value of each cell is a record.

```
hbase(main):018:0 is greater than scan 'users',{VERSIONS=>5}
ROW COLUMN+CELL
lisi column=address:city, timestamp=1457101972764, value=beijing
lisi column=address:contry, timestamp=1457102773908, value=china
lisi column=address:province, timestamp=1457101972736, value=
beijing
lisi column=info:age, timestamp=1457101972548, value=27
lisi column=info:birthday, timestamp=1457101972604, value=1987-06-
17
lisi column=info:company, timestamp=1457101972653, value=baidu
xiaoming column=address:city, timestamp=1457082196082, value=
hangzhou
xiaoming column=address:contry, timestamp=1457082195729, value=
china
xiaoming column=address:province, timestamp=1457082195773, value=
zhejiang
xiaoming column=info:age, timestamp=1457082218735, value=29
xiaoming column=info:age, timestamp=1457082178630, value=24
xiaoming column=info:birthday, timestamp=1457082186830, value=1987
-06-17
xiaoming column=info:company, timestamp=1457082189826, value=
alibaba
2 row(s) in 0.0260 seconds }
```

The data read from the table (in four columns)


rowKey	column:qualifier	timestamp	value
lisi	address:city	1457101972764	beijing

rowKey	column:qualifier	timestamp	value
Lisi	address:contry	1457102773908	china
lisi	address:province	1457101972736	beijing
lisi	Info: Age	1457101972548	27
lisi	info:birthday	1457101972604	1987-06-17
lisi	info:company	1457101972653	beijing
Aging	address:city	1457082196082	hangzhou
xiaoming	address:contry	1457082195729	china
xiaoming	address:province	1457082195773	zhejiang
xiaoming	info:age	1457082218735	29
xiaoming	info:age	1457082178630	24
xiaoming	info:birthday	1457082186830	1987-06-17
xiaoming	info:company	1457082189826	alibaba

HBase Reader supports HBase data types and converts HBase data types as follows:

Data integration internal types	HBase data type
Long	Int, short, and long
Double	Float and double
String	String and binarystring
Date	Date
Boolean	Boolean

Parameter description

Attribute	Description	Required	Default Value
haveKerberos	<p>Description: If haveKerberos is True, the HBase cluster needs to be authenticated using kerberos.</p> <div>  Note: <ul style="list-style-type: none"> NOTE: If this value is configured as true, the following five parameters related to kerberos authentication must be configured: Kerberoskeytabfilepath, kerberosprincipal, hbasemasterkerberosprincipal, hbaseregionserverkerberosprincipal and hbaserpcprotection. If the HBase cluster is not authenticated using kerberos, these six parameters are not required. </div>	No	false
hbaseConfig	<p>Description: Configuration required for connecting to the HBase cluster, in JSON format. The required item is hbase.zookeeper.quorum, which indicates the URL of HBase ZK. In addition, more HBase client configurations can be added. For example, you can configure the cache and batch of scan to optimize the interaction with servers.</p>	Yes	N/A
mode	<p>Description: Read mode of HBase. The normal and multiVersionFixedColumn modes are supported.</p>	Yes	N/A
table.	<p>Name of HBase table to be read (case-sensitive).</p>	Yes	N/A
encoding	<p>Description: Encoding method (UTF-8 or GBK). This is used when HBase byte[] stored in binary form is converted into String.</p>	No	UTF-8

Attribute	Description	Required	Default Value
column	<p>Description: HBase field to be read. This item is required in both normal and multiVersionFixedColumn modes.</p> <ul style="list-style-type: none"> In normal mode: <p>In normal mode: Except for rowkey, the HBase columns specified by the item name for reading must be in the format of column family:column name. The item type specifies the type of source data. The item format specifies the format of date. The item value specifies the current type as a constant. The system does not read data from HBase, but generates corresponding columns based on the value. The configuration format is as follows:</p> <pre> "column": [{ "name": "rowkey", "type": "string", }, { "value": "test", "type": "string", }] </pre> <p>In normal mode, for the specified Column information, you must enter type and choose one from name/value.</p> <ul style="list-style-type: none"> In multiVersionFixedColumn mode <p>Except for rowkey, the HBase columns specified by the item name for reading must be in the format of column family:column name. The constant column is not supported in multiVersionFixedColumn mode. The configuration is as follows:</p> <pre> "column": [{ "Name": "rowkey ", "type": "string", }, { "name": "info: age", "type": "string", }] </pre>	Yes	N/A
Issue: 20190117			107
maxVersion	<p>Description: Specify the number of versions of data to be read from HBase. The value is an integer greater than 0.</p>	Required	N/A

Attribute	Description	Required	Default Value
range	<p>Specifies the rowkey range that the hbase reader reads.</p> <ul style="list-style-type: none"> startRowkey: Specify start rowkey. endRowkey: Specify end rowkey. isBinaryRowkey: Specify the method for converting configured startRowkey and endRowkey to byte[]. The default value is false. If it is true, Bytes.toBytesBinary(rowkey) is called for conversion. If it is false, Bytes.toBytes(rowkey) is called. If it is true, Bytes.toBytesBinary(rowkey) is called for conversion. If it is false, Bytes.toBytes(rowkey) is called. The configuration format is as follows: <pre> "range": { "startRowkey": "aaa", "endRowkey": "ccc", "isBinaryRowkey": false } </pre>	No	N/A
scanCacheSize	Description: Number of lines read by the HBase client from the server every time when RPC is performed.	No	256
scanBatchSize	Description: Number of columns read by the HBase client from the server every time when RPC is performed.	No	1,000

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

Configure a job to extract data from HBase to local machine: (normal mode).

```

{
  "type": "job",
  "version": "2.0", //Indicates the version.
  "steps": [
    {
      "stepType": "hbase", //plug-in name
      "parameter": {
        "mode": "normal", //read HBase mode, supports normal
        mode, multiVersionFixedColumn Mode
        "scanCacheSize": 256, //Number of lines read by the
        HBase client from the server every time when RPC is performed.
        "scanBatchSize": 100, //The number of columns that
        the HBase client reads per rpc from the server.
        "hbaseVersion": "9.4x/11x", //hbase version
        "column": [ //Field
          {
            "name": "rowkey", //field name
            "type": "string" //data type
          }
        ]
      }
    }
  ]
}

```

```

        },
        {
            "name": "columnFamilyName1: columnName1 ",
            "type": "string",
        },
        {
            "name": "columnFamilyName2: columnName2",
            "format": "yyyy-MM-dd",
            "type": "date",
        },
        {
            "name": "columnFamilyName3: columnName3",
            "type": "long"
        }
    ],
    "range": { //specify the rowkey range that the HBase
Reader reads.
        "endRowkey": "", //specify end rowkey.
        "isBinaryRowkey": true, //Specify the method for
converting configured startRowkey and endRowkey to byte[]. The default
value is false. If it is true, Bytes.toBytesBinary(rowkey) is called
for conversion. If it is false, Bytes.toBytes(rowkey) is called.
        "startRowkey": "", //specify the start rowkey.
    },
    "maxVersion": "", //specify the number of versions read
by hbase reader in Multi-version Mode
    "encoding": "UTF-8", //encoding format
    "table": "ok", //The name of the target table.
    "hbaseConfig": { // configuration information required
to connect to the hbase cluster, JSON format.
        "hbase.zookeeper.quorum": "hostname",
        "hbase.rootdir": "hdfs://ip:port/database",
        "hbase.cluster.distributed": "true"
    }
},
    "name": "Reader",
    "category": "reader"
},
    { //The following is a reader template. You can find the
corresponding reader plug-in documentations.
        "stepType": "stream",
        "parameter": {},
        "name": "Writer",
        "category": "writer"
    }
],
    "setting": {
        "errorLimit": {
            "record": "0" //Number of error records
        },
        "speed": {
            "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
            "concurrent": "1", //Number of concurrent tasks
            "dmu": "1" //DMU Value
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}

```

```
    }
  }
}
```

2.3.2.5 Configuring HDFS Reader

HDFS Reader provides the ability to read the data stored by distributed file systems. At the underlying implementation level, HDFS Reader retrieves the file data on the distributed file system, converts the data into Data Integration transport protocol, and transfers it to the Writer.

HDFS Reader provides the ability to read file data from the Hadoop distributed file system (HDFS) and converts the data into Data Integration transport protocol.

For example:

TextFile is the default storage format for creating Hive tables without data compression.

Essentially, TextFile stores data in HDFS as text, and the implementation of HDFS Reader is quite similar to that of OSS Reader for Data Integration. ORCFile refers to Optimized Row Columnar File, which is the optimized RCFFile. This file format provides an efficient method for storing Hive data. HDFS Reader utilizes the OrcSerde class provided by Hive to read and parse the data of ORCFile files.



Note:

For data synchronization, admin account and read/write permissions for the files are required,

```
[root@wh0 hadoop]# useradd -m -G supergroup -g hadoop -p admin admin
[root@wh0 hadoop]# su admin
[admin@wh0 hadoop]$ hadoop fs -ls /user/hive/warehouse/hive_p_partner_native
17/05/15 18:13:11 UTIL util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rwxr-xr-x 3 hadoop supergroup 922 2017-05-15 16:17 /user/hive/warehouse/hive_p_partner_native/part-00000
[admin@wh0 hadoop]$ cd
[admin@wh0 ~]$ hadoop fs -get /user/hive/warehouse/hive_p_partner_native/part-00000
17/05/15 18:13:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[admin@wh0 ~]$ vim part-00000
[admin@wh0 ~]$ exit
exit
[root@wh0 hadoop]# pssh -h /home/hadoop/slave4pssh useradd -m -G supergroup -g hadoop -p admin admin
1) 18:14:22 SUCCESS wh1
2) 18:14:23 SUCCESS wh2
3) 18:14:23 SUCCESS wh3
```

Usage:

- Create an admin user and home directory, specify a user group and additional group, and grant the permissions for the files.

```
useradd -m -G supergroup -g hadoop -p admin admin
```

- `-G supergroup`: Specifies the additional group to which the user belongs.
- `-g hadoop`: Specifies the user group to which the user belongs.
- `-p admin admin`: Add a password to the admin user.

- View the contents of the files in this directory.

```
hadoop fs -ls /user/hive/warehouse/hive_p_partner_native
```

When using hadoop commands, the format is `hadoop fs -command`, where `command` represents the command.

- Copies the file `part-00000` to the local file system.

```
hadoop fs -get /user/hive/warehouse/hive_p_partner_native/part-00000
```

- Edit the file you just copied.

```
vim part-00000
```

- Exits the current user.

```
exit
```

- Connect to the host from the list and create an admin account on each attached host.

```
pssh -h /home/hadoop/slave4pssh useradd -m -G supergroup -g hadoop -p admin admin
```

- `pssh -h /home/hadoop/slave4pssh: connect to the host from the manifest file.`
- `useradd -m -G supergroup -g hadoop -p admin admin: Create admin account.`

Supported functions

Currently, HDFS Reader supports the following features:

- Supports TextFile, ORCFile, rcfile, sequence file, csv, and parquet file formats, and what is stored in the file must be a two-dimensional table in a logic sense.
- Supports reading multiple types of data (represented by String) and supports column pruning and column constants.
- Supports recursive reading and regular expressions "*" and "?".
- Supports ORCFile data compression, and currently supports the SNAPPY and ZLIB compression modes.
- Supports data compression for sequence files, and currently supports the lzo compression mode.
- Supports concurrent reading of multiple files.
- Supports the following compression formats for the csv type: gzip, bz2, zip, lzo, lzo_deflate, and snappy.

- In the current plugin, the Hive version is 1.1.1, and the Hadoop version is 2.7.1 (Apache [is compatible with JDK 1.6]). Data can be written normally in the testing environments of Hadoop 2.5.0, Hadoop 2.6.0, and Hive 1.2.0.

**Note:**

Temporarily, HDFS Reader does not support the multi-thread concurrent reading of a single file, which involves the internal splitting algorithm of the single file.

Supported data types

RCfile

If the file type of the HDFS file being synchronized is rcfile, you must specify the data type of the column in the Hive table in "column type" because the data storage mode varies with the data type during rcfile underlying storage and the HDFS Reader does not support accessing and querying Hive metadata databases. If the column type is bigint, double, or float, enter bigint, double, or float accordingly as the data type. If the column type is varchar or char, enter string for the same purpose.

RCFile data types are converted into the internal types supported by Data Integration by default, as shown in the following comparison table:

Type Classification	HDFS Data Type
Integer	Tinyint, smallint, Int, and bigint
Float	Float, double, decimal
String type	String, Char, and varchar
Date and time type	Date and timestamp
Boolean class	Boolean
Binary class	BINARY

Parquetfile

ParquetFile data types are converted into the internal types supported by Data Integration by default, as shown in the following comparison table:

Type Classification	HDFS Data Type
Integer	Int32, int64, and int96
Floating point	Float and double
String type	FIXED_LEN_BYTE_ARRAY

Type Classification	HDFS Data Type
Date and time type	Date and timestamp
Boolean	Boolean
Binary	BINARY

TextFile, ORCfile, and SequenceFile

Given that the metadata of TextFile and ORCFile file tables is maintained by and stored in the database maintained by Hive itself (such as MySQL), HDFS Reader currently does not support the access and query to the Hive metadata database, so you must specify a data type for type conversion.

TextFile, ORCFile, and SequenceFile data types are converted into the internal types supported by Data Integration by default, as shown in the following comparison table:

Category	HDFS Data Type
Integer	Tinyint, smallint, Int, and bigint
Floating point	float and double
String type	String, Char, varchar, struct, MAP, array, union , binary
Date and time	date and timestamp
Boolean	Boolean

Notes:



- LONG: Represents with an integer string in the HDFS file, such as 123456789.
- DOUBLE: Represents with a double string in the HDFS file, such as 3.1415.
- BOOLEAN: Represents with a boolean string in the HDFS file, such as true or false (case-insensitive).
- DATE: Represents with a date and time string in the HDFS file, such as 2014-12-31 00:00:00.




Note:


The Timestamp data type supported by Hive can be accurate to nanoseconds, so the data content of Timestamp stored in TextFile and ORCFile can be in the format like "2015-08-21 22:40:47.397898389". If the converted data type is set as Date for Data Integration, the nanosecond part is truncated after conversion. If you want to retain the nanosecond part, set the converted data type as String for Data Integration.


Parameter description

Attribute	Description	Required	Default Value
path	<p>Description: It refers to the file path to be read. If you want to read multiple files, use a regular expression to match all of them, such as /hadoop/data_201704*.</p> <ul style="list-style-type: none"> If a single HDFS file is specified, HDFS Reader only supports single-threaded data extraction. If multiple HDFS files are specified, HDFS Reader supports multiple-threaded data extraction, and the number of concurrent threads is determined by the job speed (mbps). The actual number of initiated concurrent threads is the smaller of the number of HDFS files to be read and the set job speed. <div>  Note: The actual number of initiated concurrent threads is the smaller of the number of HDFS files to be read and the set job speed. </div> <ul style="list-style-type: none"> When the wildcard is specified, HDFS Reader attempts to traverse multiple files. For example: When "/" is specified, HDFS Reader reads all the files under the "/" directory. When "/bazhen/" is specified, HDFS Reader reads all the files under the bazhen directory. Currently, HDFS Reader only supports "*" and "?" as file wildcards, and the syntax is similar to that of the file wildcards of common Linux command lines. <div>  Note: <ul style="list-style-type: none"> Data Integration regards all the files to be read in the same synchronization job as one data table. For this reason, you must ensure that all those files adapt to the same schema information and grant the read permission to Data Integration. Note on reading partitions: During Hive table creation, you can specify partitions. For example, after creating the partition(day="20150820",hour="09"), two directories with the name of /20150820 and /09 respectively are created in the table catalog of the HDFS file system and /20150820 is the parent directory of /09. <p>Given that partitions are organized in a directory</p> </div> <p>structure, we can set the value of path in JSON when reading all the data of a table by partitions. For example, if you want to read all the data of the table</p>	Yes	N/A
114			Issue: 20190117

Attribute	Description	Required	Default Value
fileType	<p>- Description: File type. Currently, only text, orc, rc, seq, csv, and parquet are supported. HDFS reader automatically recognizes the type of file, and uses the corresponding file type read policy. Before synchronizing data, HDFS Reader checks whether the types of all the target files under the specified path are consistent with fileType. If not, the synchronization task fails.</p> <p>The list of parameter values that can be configured by fileType is as follows:</p> <ul style="list-style-type: none"> • text: The format of TextFile. • orc: The format of ORCFile. • rc: The format of RCFile • seq: The format of sequence file • csv: The format of common HDFS file (logical two-dimensional table). • parquet: The format of common parquet file. <div>  <p>Note:</p> <p>Because TextFile and ORCFile are totally different file formats, HDFS Reader parses the two file types in different ways. For this reason, the formats of the converted results varies when complex compound types supported by Hive (such as map, array, struct, and union) are converted to the String type supported by Data Integration. The following takes use the map type as an example:</p> <ul style="list-style-type: none"> • After being parsed and converted to the String type supported by Data Integration, the ORCFile map type is {job=80, team=60, person=70}. • After being parsed and converted to the String type supported by Data Integration, the TextFile map type is job:80,team:60,person:70. <p>From the preceding results, the data itself remains unchanged but the representation formats differ slightly . For this reason, if the fields to be synchronized under the file path configured are compound in Hive, we recommended that you set a unified format for the files.</p> <p>Recommended best practices:</p> </div>	Yes	N/A
Issue 20190117	<ul style="list-style-type: none"> • To unify the file types parsed from compound types, we recommended that you export TextFile tables as ORCFile tables on the Hive client. • If the file type is Parquet, the parquetSchema is required 		115

Attribute	Description	Required	Default Value
column	<p>Description: It refers to the list of fields read, where type indicates the type of the source data, index indicates the number of the column in the text (starts from 0), and value indicates that the current type is constant, which means the corresponding column is automatically generated according to the value instead of the data read from the source file. By default, you can all read the data according to the string type, configured as <code>"column": ["*"]</code>.</p> <p>You also can configure the column field as follows:</p> <pre>{ "type": "long", "index": 0 // Retrieves the int field from the first column of the local file text }, { "type": "string", "value": "alibaba" // HDFS Reader internally generates the alibaba string field as the current field }</pre>	Yes	N/A
fieldDelimiter	<p>Description: It refers to the field delimiter read. When HDFS Reader reads the TextFile data, a file delimiter is required, which defaults to ',' if no delimiter is specified. When HDFS Reader reads the ORCFile data, no field delimiter is required. The default delimiter of Hive is <code>\u0001</code>.</p> <ul style="list-style-type: none"> To use each row as a column of the target, use characters that are not included in the content of rows as the delimiter, such as the invisible characters <code>\u0001</code>. Additionally, <code>\n</code> cannot be used as the delimiter. 	No	,
encoding	Description: Encoding of the read files.	No	UTF-8
nullFormat	<p>Description: Defining null (null pointer) with a standard string is not allowed in text files. Data Integration provides nullFormat to define which strings can be expressed as null. For example, when nullFormat: "null" is configured, if the source data is "null", it is considered as a null field in Data Integration.</p>	No	N/A

Attribute	Description	Required	Default Value
compress	<p>Description: It refers to the possible file compression formats when the fileType is csv, which currently support gzip, bz2, zip, lzo, lzo_deflate, hadoop-snappy, and framing-snappy.</p> <div> Note:<ul style="list-style-type: none">• Two lzo compression formats are available: lzo and lzo_deflate. In actual configuration scenarios, make sure to select the appropriate one.• Given that no unified stream format is now available to snappy, Data Integration currently only supports the most popular two compression formats: hadoop-snappy (the snappy stream format in Hadoop) and framing-snappy (the snappy stream format recommended by Google).• rc is the format of rcfile.• No entry is required for the orc file type.</div>	No	N/A

Attribute	Description	Required	Default Value
parquetSchema	<p>Description: Required when the file is in parquet format. It is used to specify the structure of the target file, and takes effect only when the fileType is parquet. The format is as follows:</p> <pre>message MessageType { Required, data type, column name; ; }</pre> <p>Notes:</p> <ul style="list-style-type: none"> • MessageType: Any supported value • Required: Required or Optional. Optional is recommended. Optional is recommended. • Data Type: Parquet files support the following data types: boolean, int32, int64, int96, float, double, binary (select binary if the data type is string), and fixed_len_byte_array. <p> Note: Note that each configuration row and column, including the last one, must end with a semicolon.</p> <p>Configuration example:</p> <pre>message m { optional int64 id; optional int64 date_id; optional binary datetimestring; optional int32 dspId; optional int32 advertiserId; optional int32 status; optional int64 bidding_req_num; optional int64 imp; optional int64 click_num; }</pre>	No	N/A
csvReaderConfig	<p>Description: Reads the parameter configurations of CSV files. It is the Map type. This reading is performed by the CsvReader for reading CSV files and involves many configuration items, whose defaults are used if they are not configured.</p> <p>Common configuration:</p> <pre>csvReaderConfig "safetySwitch": false, "skipEmptyRecords": false, "useTextQualifier": false }</pre>	No	N/A
118	<p>For all the configuration items and default values, you must configure the map of csvReaderConfig strictly in accordance with the following field names:</p>		Issue: 20190117

Development in script mode

A script template can be imported for development. The following is a script configuration sample. For relevant parameters, see Parameter Description.

```
{
  "type": "job",
  "version": "2.0",
  "steps": [
    {
      "stepType": "hdfs", // plug-in name
      "parameter": {
        "path": "", // file path to read
        "datasource": "", //Name of the data source
        "column": [
          {
            "index": 0, //serial number
            "type": "string" //Field Type
          },
          {
            "index": 1,
            "type": "long"
          },
          {
            "index": 2,
            "type": "double",
          },
          {
            "index": 3,
            "type": "boolean"
          },
          {
            "format": "yyyy-MM-dd HH:mm:ss", // time format
            "index": 4,
            "type": "date",
          }
        ],
        "fieldDelimiter": ",", //Delimiter of each column
        "Encoding": "UTF-8", // encoding format
        "fileType": "// text type
      },
      "name": "Reader",
      "category": "reader"
    },
    {
      //The following is a writer template. You can find the
      //corresponding writer plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Writer",
      "Category": "Writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "" //Number of error records
    },
    "speed": {
      "concurrent": "3", //Number of concurrent tasks
      "throttle": false, //False indicates that the traffic is
      not throttled and the following throttling speed is invalid. True
      indicates that the traffic is throttled.
      "dmu": 1 // DMU Value
    }
  }
}
```

```
    },
    "order": {
      "hops": [
        {
          "from": "Reader",
          "to": "Writer"
        }
      ]
    }
  }
}
```

2.3.2.6 Configure MaxCompute Reader

The MaxCompute Reader plugin provides the ability to read data from MaxCompute. For details about MaxCompute, see [MaxCompute Overview](#).

At the underlying implementation level, it reads data from the MaxCompute system by using Tunnel based on the source project/table/partition/table fields and other information you configured. For common Tunnel commands, see [Tunnel Command Operations](#).

MaxCompute Reader can read both partition and non-partition tables, but cannot read virtual views. To read a partition table, you must specify the partition configuration. For example, to read table t0 with a partition configuration of "pt=1, ds=hangzhou", you must set the value in the configuration. For a non-partition table, the partition configuration is left empty. For table fields, you can specify all or some of the columns sequentially, change the order in which columns are arranged, and specify constant fields and partition columns. (A partition column is not a table field).



Supported data types

MaxCompute Writer supports the following data types in MaxCompute:

Data type	MaxCompute data type
Integer	bigint
Floating point	double and decimal
String	string
Date and time type	Datetime
Boolean	Boolean

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
table.	Reads the table name of the data table (not case sensitive).	Yes	N/A
partition	<p>The information of the partition from which you read data. Linux shell wildcard is allowed ("" represents 0 or multiple characters, and "?" represents any character.) For example, a partition table named "test" has four partitions: pt=1/ds=hangzhou, pt=1/ds=shanghai, pt=2/ds=hangzhou, and pt=2/ds=beijing.</p> <ul style="list-style-type: none">• To read data from partition pt=1/ds=shanghai, configure it to <code>"partition": "pt=1/ds=shanghai"</code>.• To read data from all the partitions under pt=1, configure it to <code>"partition": "pt=1/ds=*" "</code>.• To read data from all the partitions of the "test" table, configure it to <code>"partition": "pt=*/ds=*" "</code>.	Required if the table is a partition table . For non-partition tables, it is left empty.	N/A

Attribute	Description	Required	Default Value
column	<p>The column information of the MaxCompute source table.</p> <p>For example, the fields of a table named "test" are id, name, and age.</p> <ul style="list-style-type: none"> To read the fields in turn, configure it to <code>"column": ["id", "name", "age"]</code> or <code>"column": ["*"]</code>. <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;">  Note: We don't recommend that you configure the extracted field to "*", because it indicates that every field in the table is read in turn. If you change the order or types of the table fields, or add or delete some table fields, it is likely that the source table columns cannot be aligned with the target table columns, causing incorrect results or even failure. </div> <ul style="list-style-type: none"> To read name and id in sequence, configure it to <code>"column": ["name", "id"]</code>. To add a constant field to the fields to be extracted from the source table (to match the field order of the target table), for example, if the data values you want to extract are values of age, name, constant date "1988-08-08 08:08:08", and id columns, configure it to: <code>"column": ["age", "name", "'1988-08-08 08:08:08'", "id"]</code>, with the constant value enclosed by <code>'</code>. In internal implementation, any field enclosed by <code>'</code> is considered as a constant field, and its value is the content in <code>'</code>. <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;">  Note: <ul style="list-style-type: none"> MaxCompute Reader does not use Select SQL statement of MaxCompute for extracting data from a table. Therefore, you cannot specify functions in fields. Column must contain the specified column set to be synchronized and it cannot be blank. </div>	Yes	N/A

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.

The data sources can be default data sources or data sources created by you. Click [here](#) to check the supported data source types.

Source

* Data Source: **ODPS** **odps_first** ?

* Table: **sys_log_storage_maxcompute**

* Partition: dt = **\${bizdate}** ?

Compression: ☒ Disable ☐ Enable

Consider Empty String as Null: ☒ Yes ☐ No

Destination

* Data Source: **Oracle** **luz_oracle** ?

* Table: **luz_oracle**

Statements Run: **select * from PERSON PERSON** ?

Before Import

Statements Run: **select * from PERSON PERSON** ?

After Import

Preview

Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.



Note:

If you specify all columns, you can configure them in column, for example, "column ": [""]. Partition supports configuration methods that configure multiple partitions and wildcard characters.

- "partition": "pt=20140501/ds=*": Reads the data from all partitions in ds.
- "partition": "pt=top?" The mark ? indicates whether the character in front of it exists.

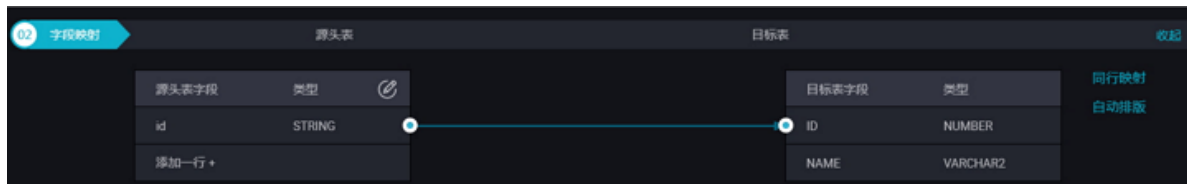
This configuration specifies the two partitions with pt=top and pt=to.

You can enter the partition columns to be synchronized, such as partition columns with pt.

Example: Assuming that the value of each MaxCompute partition is pt=\${bdp.system.bizdate}, add the partition name pt to a field in the source table, ignore the unrecognized mark if any, and proceed with the next step. To synchronize all partitions, configure the partition value to pt=\${*}. To synchronize a certain partition, select a time value for the partition.

2. Field mapping: The column in the preceding parameter description.

The Source Table Field on the left maps with the Target Table Field on the right. Click **Add Line**, and then a field is added. Hover the cursor over a line, click **Delete**, and then the line is deleted.

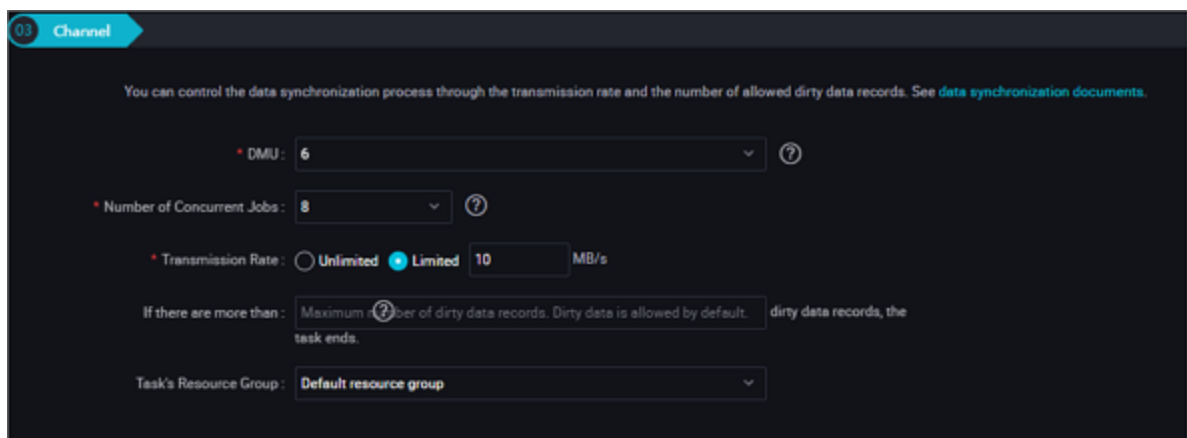


- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

By clicking Add Row,

- Each constant must be enclosed in a pair of single quotes, such as 'abc' and '123'.
- Use this function with scheduling parameters, such as \${bizdate}.
- Enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as 'Not Identified'.

3. Control the tunnel



Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.

- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

To configure a job to extract data locally from MaxCompute, please refer to the above parameter descriptions for details.

```
{
  "type": "job",
  "version": "2.0",
  "steps": [
    {
      "stepType": "odps", // plug-in name
      "parameter": {
        "partition": [], the partition where the read data is
located
        "isCompress": false, //do you Want to compress?
        "datasource": "", //Data Source
        "column": column information for [//source table
          "id",
        ],
        "emptyAsNull": true,
        "table": "//table name
      },
      "name": "Reader ",
      "category": "reader"
    },
    {
      //The following is a writer template. You can find the
corresponding writer plug-in documentations.
      "stepType": "stream ",
      "parameter": {
      },
      "name": "Writer ",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
      "concurrent": "1", //Number of concurrent tasks
      "dmu": "1" //DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "from": "Reader",
        "to": "Writer"
      }
    ]
  }
}
```

```
}
```

2.3.2.7 Configure MongoDB Reader

The MongoDB Reader plugin uses Mongo Client, the Java client of MongoDB, to read data from MongoDB. In the latest version of Mongo, the granularity of the DB lock has been reduced from the DB level to the document level. Combined with the powerful indexing function of MongoDB, it allows a high-performance reading of MongoDB.

**Note:**

- If you are using ApsaraDB for MongoDB, a root account is provided by default. To ensure security, Data Integration only supports using the relevant account of MongoDB for connection. Avoid using the root account as access account when adding and using the MongoDB data source.
- Query does not support the JS syntax.

MongoDB Reader reads data in parallel from MongoDB by means of Data Integration framework. Based on the specified rules, it partitions the data in MongoDB into multiple data fragments, reads them in parallel using the controlling Job program based on the specified rules, and then converts the data types supported by MongoDB to the ones supported by Data Integration individually.

Type conversion list

MongoDB Reader supports most data types in MongoDB. Check whether your data type is supported before using it.

MongoDB Writer converts the MongoDB data types as follows:

Type Classification	MongoDB data type
Integer	int and long
Floating point	double
String	string
Date and time	date
Boolean	bool
Binary	bytes

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A-
collection Name	The collection name of MonogoDB.	Yes	N/A
column	Description: An array of multiple column names of a document in MongoDB. <ul style="list-style-type: none"> name: Column name. type: Column type. splitter: MongoDB supports array, but the CDP framework does not. Therefore, the data items read from MongoDB in an array format are joined into a string using this delimiter. 	Yes	N/A
query	Used to define the range of returned MongoDB data. For example, if you set it to <code>"query": "{ 'operationTime': { '\$gte': ISODate('\${last_day}'T00:00:00.424+0800') } }"</code> , only the data with an operationTime later than or equal to 00:00 of <code>\${last_day}</code> is returned. <code>\${last_day}</code> is the scheduling parameter of DataWorks in the format of <code>\$(yyyy-mm-dd)</code> . You can use conditional operators (<code>\$gt</code> , <code>\$lt</code> , <code>\$gte</code> , <code>\$lte</code>), logical operators (<code>and</code> , <code>or</code>), and functions (<code>max</code> , <code>min</code> , <code>sum</code> , <code>avg</code> , <code>ISODat</code>) supported by MongoDB as needed. For details, see the query syntax of MongoDB.	No	N/A

Development in wizard mode

Development in wizard mode is unavailable currently.

Development in script mode

To configure a job to extract data locally from MongoDB, please refer to the above parameter descriptions for details.

```
{
  "type": "job",
  "version": "2.0", //Indicates the version.
  "steps": [
    {
      "stepType": "hdfs", //plug-in name
      "parameter": {
        "path": "", // path

```

```

        "datasource": "", // Data Source
        "query": "",
        "column": [
            {
                "index": 0, // serial number
                "type": "string" // Field Type
            },
            {
                "index": 1,
                "type": "long"
            },
            {
                "index": 2,
                "type": "double"
            },
            {
                "index": 3,
                "type": "boolean"
            },
            {
                "format": "yyyy-MM-dd HH:mm:ss", //time format
                "index": 4,
                "type": "date",
            },
            {
                "name": "taglist",
                "type": "Array",
                "splitter": " "
            },
            {
                "name": "a.b.c",
                "type": "document.array",
                "splitter": " "
            }
        ],
        "fieldDelimiter": ",", //Delimiter of each column
        "encoding": "UTF-8", // encoding format
        "fileType": "text" // file type
    },
    "name": "Reader ",
    "category": "reader"
},
{
    //The following is a writer template. You can find the
    corresponding writer plug-in documentations.
    "stepType": "stream ",
    "parameter": {}
    "name": "Writer",
    "category": "writer"
}
],
"setting": {
    "errorLimit": {
        "record": "0" //Number of error records
    },
    "speed": {
        "throttle": false, //False indicates that the traffic is
        not throttled and the following throttling speed is invalid. True
        indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order": {
    "hops": [

```



```

    {
      "from": "Reader ",
      "to": "Writer"
    }
  ]
}

```

2.3.2.8 Configure DB2 Reader

The DB2 Reader plug-in enables data reading from DB2. At the underlying implementation level, DB2 Reader connects to a remote DB2 database through JDBC and runs corresponding SQL statements to select data from the DB2 database.

Specifically, DB2 Reader connects to a remote DB2 database through the JDBC connector. The SELECT SQL query statements are generated and sent to the remote DB2 database based on your configuration. Then, the SQL statements are run and the returned results are assembled into abstract datasets using the custom data types of data integration. Datasets are passed to the downstream writer for processing.

- DB2 Reader concatenates the table, column, and WHERE information you configured into SQL statements and sends them to the DB2 database.
- DB2 Reader directly sends the query SQL information you configured to the DB2 database.

DB2 Reader supports most of the DB2 data types. Check whether your data type is supported.

DB2 Reader converts DB2 data types as follows:


Type Classification	DB2 data type
Integer	SMALLINT
Floating point	decimal, real, or double
String	char, character, varchar, graphic, vargraphic, long varchar, clob, long vargraphic, or dbclob
Date and time	Date, time, and timestamp
Boolean	—
Binary	blob

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A

Attribute	Description	Required	Default Value
jdbcUrl	Information of the JDBC connection to the DB2 database. In accordance with the DB2 official specification, jdbcUrl in the DB2 format is jdbc:db2://ip:port/database, and the connection accessory control information can be filled in.	Yes	N/A
username	User name for the data source.	Yes	N/A
password	Description: Password corresponding to the specified username for the data source.	Yes	N/A
table.	You select a table that needs to be synchronized, and one operation can only support one table synchronization.	Yes	N/A
column	<p>The configured table requires a collection of column names that are synchronized, using an array of JSON to describe the field information, all column configurations, such as [*], are used by default.</p> <ul style="list-style-type: none"> Column pruning is supported, which means you can select some columns to export. Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. Constant configuration is supported. You must follow the DB2 SQL syntax format, for example, <code>["id", "1", "'const name'", "null", "upper('abc_lower')", "2.3", "true"]</code>, <ul style="list-style-type: none"> - where id refers to the ordinary column name - 1 is an integer numeric constant - 'const name' is a String constant (requires a pair of single quotes) - null is a null pointer - upper('abc _ down') is a function expression - 2.3 is a floating point number - True is a Boolean Value Column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	N/A

Attribute	Description	Required	Default Value
Splitpk	<p>Description: If you specify the splitPk when using RDBMSReader to extract data, it means that you want to use the fields represented by splitPk for data sharding. Then, the DataX starts concurrent tasks to synchronize data, which greatly improves the efficiency of data synchronization.</p> <ul style="list-style-type: none"> We recommend that the splitPk users use the primary keys of tables, because the primary keys are generally even and data hot spots are less prone to split data fragments. Currently, splitPk only supports data sharding for integer data types. Other types such as floating point, string, and date are not supported. If you specify an unsupported data type, DB2 Reader reports an error. 	No	Blank
where	<p>Filtering condition. DB2 Reader concatenates an SQL statement based on specified column, table, and where conditions and extracts data according to the SQL statement. In actual business scenarios, the data on the current day is usually required to be synchronized. You can specify the where condition as <code>gmt_create > \$bizdate</code>. The where condition can be used to synchronize incremental business data effectively. If the value is null, it means synchronizing all the information in the table.</p>	No	N/A
Querysql	<p>In some business scenarios, the where condition is insufficient for filtration. In such cases, you can customize a filter SQL statement using this configuration item. When this item is configured, the data synchronization system filters data using this configuration item directly instead of such configuration items as table and column.</p> <p>For example, for data synchronization after multi-table join, use <code>select a,b from table_a join table_b on table_a.id = table_b.id</code>. When query SQL is configured, DB2 Reader directly ignores the configuration of table, column, and where conditions.</p>	No	N/A

Attribute	Description	Required	Default Value
Fetchsize	<p>It defines the pieces of batch data that the plug-in and database server can fetch each time. The value determines the number of network interactions between the data synchronization system and the server, which can greatly improve data extraction performance.</p> <div>  Note: A value greater than 2048 may lead to OOM for data synchronization. </div>	No	1,024

Development in wizard mode

Development in wizard mode is unavailable currently.

Development in script mode

Configure a job to synchronously extract data from a DB2 database:

```
{
  "type": "job",
  "version": "2.0", //Indicates the version.
  "steps": [
    {
      "stepType": "DB2", // plug-in name
      "parameter": {
        "password": "", //Password
        "jdbcUrl": "", //DB2 database's JDBC connection
        "column": [
          "id"
        ],
        "where": "", //Filtering condition
        "splitPk": "", //the field represented by/splitpk
        "table": "", //The name of the target table
        "username": "" //User Name
      },
      "name": "Reader ",
      "category": "reader"
    },
    {
      //The following is a writer template. You can find the
      //corresponding writer plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
```

```

        "throttle":false,//False indicates that the traffic is
        not throttled and the following throttling speed is invalid. True
        indicates that the traffic is throttled.
        "concurrent": "1",//Number of concurrent tasks
        "dmu": 1//DMU Value
    }
},
"order":{
    "hops":[
        {
            "from": "Reader ",
            "to": "Writer"
        }
    ]
}
}

```

Additional instructions

Master standby synchronous data recovery problem

Master/slave synchronization means that DB2 uses a master/slave disaster recovery mode in which the slave database continuously restores data from the master database through binlog . Due to the time difference in master/slave data synchronization, especially in some special situations such as network latency, the restored data in the slave database after synchronization are significantly different from the data of the master database, that is to say, the data synchronized in the slave database are not a full image of the master database at the current time.

Consistency Constraints

DB2 is a RDBMS system in terms of data storage, which can provide APIs for querying strong consistency data. For example, if another data writer writes data to the database during a synchronization task, DB2 Reader does not get the newly written data because of the snapshot features of the database. For the characteristics of database snapshots, see [MVCC Wikipedia](#).

These are the features of data synchronization consistency in the single-threaded model of DB2 Reader. Because DB2 Reader uses concurrent data extraction according to your configuration information, robust data consistency cannot be guaranteed. After DB2 Reader completes data sharding based on splitPk, multiple concurrent tasks are successively enabled to synchronize data . Since multiple concurrent tasks do not belong to the same read transaction and time intervals exist between the concurrent tasks, The data is not complete and consistent data snapshot information.

Currently, the consistency snapshot demands in the multi-thread model cannot be met technically, which can only be solved from the engineering point of view. The engineering approaches have both advantages and disadvantages. The following solutions are provided for your consideration:

- Use single-threaded synchronization without data sharding. This is slow but can ensure robust data consistency.
- Close other data writers to ensure the current data is static. For example, you can lock the table or disable backup database synchronization. The disadvantage is that the online businesses may be affected.

Database coding problem

DB2 Reader extracts data using JDBC at the underlying level. JDBC is applicable to all types of encodings and can complete transcoding at the underlying level. Therefore, DB2 Reader can identify the encoding and complete transcoding automatically without the need to specify the encoding.

Incremental Synchronization

Since Oracle Reader extracts data using JDBC SELECT statements, you can extract incremental data using the SELECT...WHERE... in either of the following ways:

- When online database applications write data into the database, the modify field is filled with the modification timestamp, including addition, update, and deletion (logical deletion). For this type of applications, DB2 Reader only requires the WHERE condition followed by the timestamp of the last synchronization phase.
- For new streamline data, DB2 Reader requires the WHERE condition followed by the maximum auto-increment ID of the last synchronization phase.

In case that no field is provided for the business to identify the addition or modification of data, DB2 Reader cannot perform incremental data synchronization and can only perform full data synchronization.

SQL Security

DB2 Reader provides query SQL statements for you to SELECT data by yourself. DB2 Reader conducts no security verification on query SQL.

2.3.2.9 Configure MySQL Reader

MySQL Reader connects to a remote MySQL database through the JDBC connector. The SQL query statements are generated and sent to the remote MySQL database based on your configuration. Then, the SQL statements are run and the returned results are assembled into abstract datasets using the custom data types of data synchronization. Datasets are passed to the downstream writer for processing.

In short, with the JDBC connector, MySQL Reader connects to the remote MySQL database and runs SQL statements to select data from the MySQL database, achieving data reading from the MySQL database at the underlying level.

MySQL Reader supports table and view reading. In table field, you can specify all columns in sequence, specify certain columns, adjust the column order, specify constant fields, and configure MySQL functions, such as now().

MySQL Reader supports the following MySQL data types.

Type Classification	MySQL data type
Integer	int, tinyint, smallint, mediumint, int, bigint
Floating point	float, double, decimal
String	varchar, char, tinytext, text, mediumtext, longtext
Date and time	date, datetime, timestamp, time, year
Boolean	bit, bool
Binary	tinyblob, mediumblob, blob, longblob, varbinary

**Note:**

- Apart from the field types listed here, other types are not supported.
- MySQL Reader classifies tinyint(1) as the integer type.

Type conversion list

MySQL Writer converts the MySQL data types as follows:

Type Classification	MySQL data type
Integer	Int, Tinyint, Smallint, Mediumint, Bigint
Float	Float, Double, Decimal
String type	Varchar, Char, Tinytext, Text, Mediumtext, LongText
Date and time type	Date, Datetime, Timestamp, Time, Year
boolean	Bool
Binary	Tinyblob, Mediumblob, Blob, LongBlob, Varbinary

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
table.	You select a table name that requires synchronization, and a data integration Job can only synchronize one table.	Yes	N/A
column	<p>Description: The column name set to be synchronized in the configured table. Field information is described with arrays in JSON. [*] indicates all columns by default.</p> <ul style="list-style-type: none"> Column pruning is supported, which means you can select some columns to export. Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. Constant configuration is supported. You must follow the MySQL SQL syntax format, for example <code>["id", "table", "1", "'mingya.wmy'", "'null'", "to_char(a + 1)", "2.3", "true"]</code>. <ul style="list-style-type: none"> ID is normal column name Table is a column name that contains Reserved Words 1 for plastic digital Constants 'mingya. wmy' is a String constant (note that a pair of single quotes is required) Null is a null pointer CHAR_LENGTH(s) is the computed String Length Function 2.3 is a floating point number true is a Boolean Value column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	N/A

Attribute	Description	Required	Default Value
splitPk	<p> splitPk If you specify the splitPk when using the MySQL Reader to extract data, it means that you want to use the fields represented by splitPk for data sharding. Then, the data synchronization starts concurrent tasks to synchronize data, which greatly improves the efficiency of data synchronization.</p> <ul style="list-style-type: none"> If you are using splitPk, we recommend that you use the primary keys of tables, because the primary keys are generally even and data hot spots are less prone to split data fragments. Currently, splitPk only supports data sharding for integer data types. Other types such as string, floating point, and date are not supported. If you specify an unsupported data type, the splitPk is ignored and the data is synchronized using a single channel. If the splitPk is not specified (for example, splitPk is not provided or splitPk value is null), the table data is synchronized using a single channel. 	No	N/A
where	<p>In actual business scenarios, the data on the current day is usually required to be synchronized. You can specify the WHERE condition as <code>gmt_create > \$bizdate</code>.</p> <ul style="list-style-type: none"> The where condition can be effectively used for incremental synchronization. If the where is not specified (for example, the key or value of the where is not provided), full synchronization is performed. You cannot specify where condition as <code>limit 10</code>, which does not conform to requirements for MySQL SQL where clause. 	No	N/A

Attribute	Description	Required	Default Value
querySql(advanced mode, wizard mode not available)	In some business scenarios, the where condition is insufficient for filtration. In such cases, the user can customize a filter SQL using this configuration item. When this item is configured, the data synchronization system filters data using this configuration item directly instead of such configuration items as table and column. For example, for data synchronization after multi-table join, use <code>select a, b from table_a join table_b on table_a.id = table_b.id</code> . When querySql is configured, MySQL Reader directly ignores the configuration of table, column, where, and splitPk conditions. The priority of querySql is higher than table, column, WHERE, and splitPk. The datasource uses it to parse out information such as a user name and password.	No	N/A
Singleormulti (applies only to split-up tables)	Represents a sub-library table, and the wizard mode is converted into Script Mode to actively generate this configuration <code>"singleOrMulti": "multi"</code> , but the configuration script task template does not directly generate this configuration must be added manually, otherwise, only the first data source is recognized. Singleormulti is just a frontend, and the back-end does not use this as a split-Table judgment.	Yes	multi

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.

The screenshot shows the '01 Data Source' configuration step in the wizard. It is split into two main panels: 'Source' and 'Destination'.

- Source Panel:**
 - Data Source:** MySQL (dropdown)
 - Table:** bird_rds (dropdown)
 - Data Filtering:** id=1 (text input)
 - Sharding Key:** id (dropdown)
 - Action:** Add Data Source + (button)
- Destination Panel:**
 - Data Source:** DRDS (dropdown)
 - Table:** px_31 (dropdown)
 - Statements Run:** select * from px_31 (text input)
 - Action:** Review (button)

At the bottom, there is a 'Next' button.

Configurations:

- **Data source:** The data source in the preceding parameter description. Enter the data source name you configured.
- **Table:** table in the preceding parameter description. Select the table to be synchronized.
- **Data Filtering:** you are about to synchronize the filtering criteria for the data, and the limit keyword filtering is not supported for the time being. The SQL syntax is consistent with the selected data source.
- **Cut key:** You can use a column in the source data table as a cut key, it is recommended that you use a primary key or an indexed column as a split key, and that only fields of type Integer be supported.

During data reading, the data is split based on the configured fields to achieve concurrent reading, improving data synchronization efficiency.

**Note:**

The configuration of splitting key is related to the source selection in data synchronization. The splitting key configuration item is displayed only when you configure the data source.

2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.



- **Peer mapping:** Click **peer mapping** to establish a corresponding mapping relationship in the peer, note that match the data type.
- **Automatic formatting:** The fields are automatically sorted based on corresponding rules.
- **Manually edit source table field:** Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

The function of adding a row is as follows:

- You can enter constants. The value must be enclosed by a pair of single quotes, such as 'abc' and '123'.

- Use this function with scheduling parameters, such as `${bizdate}`.
- You can enter functions supported by relational databases, such as `now()` and `count(1)`.
- If the value you entered cannot be parsed, the type is displayed as Not identified.

3. Control the tunnel

Configurations:

- **DMU:** A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- **Concurrent job count:** Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- **The maximum number of errors indicates the maximum number of dirty data records.**
- **Task Resource Group:** the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

A script sample for a Single-library single-table, for example, can be found in the above parameter descriptions.

```
{
  "type": "job",
  "version": "1.0"} //Indicates the version.
  "steps": [
    {
      "stepType": "mysql", // plug-in name
      "parameter": {
        "Column": [// column name
          "id",
        ],
      },
    },
  ],
}
```

```

        "connection": [
            {
                "datasource": "", // Data Source
                "table": [// table name
                    "xxx"
                ]
            }
        ],
        "where": "", //Filtering condition
        "Splitpk": "ID", // cut key
        "encoding": "UTF-8", // encoding format
    },
    "name": "Reader ",
    "category": "reader"
},
{ //The following is a writer template. You can find the
corresponding writer plug-in documentations.
    "stepType": "stream ",
    "parameter": {}
    "name": "Writer ",
    "category": "writer"
}
],
"setting": {
    "errorLimit": {
        "record": "0" //Number of error records
    },
    "speed": {
        "throttle": false, //false stands for open current, the
speed of the lower limit does not work, and true stands for current
limit
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 //DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader",
            "to": "Writer"
        }
    ]
}
}

```

2.3.2.10 Configure Oracle Reader

The Oracle Reader plug-in provides the ability to read data from Oracle. At the underlying implementation level, Oracle Reader connects to a remote Oracle database through JDBC and runs the SELECT statements to extract data from the database.

On the public cloud, RDS or DRDS does not provide the Oracle storage engine. Currently, Oracle Reader is mainly used for private cloud data migration and Data Integration projects.

In short, Oracle Reader connects to a remote Oracle database through the JDBC connector. The SELECT SQL query statements are generated and sent to the remote Oracle database based on your configuration. Then, the SQL statements are run and the returned results are assembled into

abstract datasets using the custom data types of data synchronization. Datasets are passed to the downstream writer for processing.

- Oracle Reader concatenates the table, column, and WHERE information you configured into SQL statements and sends them to the Oracle database.
- Oracle directly sends the querySQL information you configured to the Oracle database.

Type conversion list

Oracle Reader supports most data types in DB2. Check whether your data type is supported.


Oracle Reader converts Oracle data types as follows:

Type Classification	Oracle data type
Integer	Number, rawd, integer, Int, and smallint
Float	Numeric, decimal, float, double precisioon, real
String type	Long, Char, NChar, Varchar, Varchar2,NVar2 , Clob, NClob, character, character varying, char varying, national character, National char , National Character varying, national char varying and nchar varying
Date and time type	Timestamp and Date
boolean	Bit and Bool
Binary	Blob, BFile, Raw, and Long Raw

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
table	The name of the selected table that needs to be synchronized.	Yes	N/A

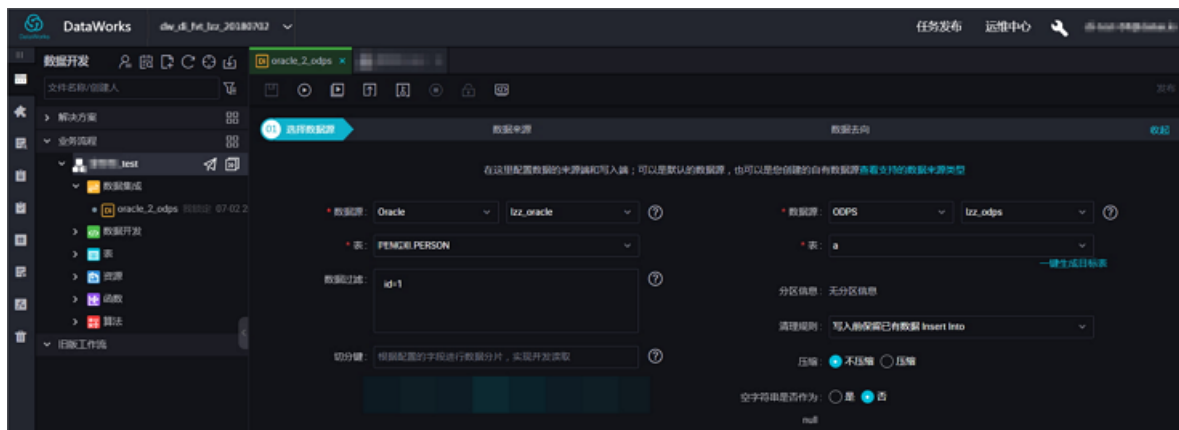
Attribute	Description	Required	Default Value
column	<p>Description: The column name set to be synchronized in the configured table. Field information is described with arrays in JSON. ["****"] indicates all columns by default.</p> <ul style="list-style-type: none"> Column pruning is supported, which means you can select some columns to export. Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. Constant configuration is supported, and you need to configure in JSON format. <pre>["id", "1", "'mingya.wmy'", "null", "to_char(a + 1)", "2.3" , "true"]</pre> <ul style="list-style-type: none"> ID is normal column name 1 is an integer numeric constant 'Mingya.wmy' is a String constant (note that a pair of single quotes is required) Null is a null pointer to_char(a + 1) is an expression 2.3 is a floating point number True is a Boolean Value <ul style="list-style-type: none"> Column is required and cannot be blank. 	Yes	N/A
splitPk	<p>If you specify the splitPk when using RDBMSReader to extract data, it means that you want to use the fields represented by splitPk for data sharding. Then, the DataX starts concurrent tasks to synchronize data, which greatly improves the efficiency of data synchronization.</p> <ul style="list-style-type: none"> If you are using splitPk, we recommend that you use the primary keys of tables, because the primary keys are generally even and data hot spots are less prone to split data fragments. The data types supported by splitPk include the integer, string, floating point, and date. If splitPk is left blank, it indicates that no table splitting is required and Oracle Reader synchronizes full data through a single channel. 	No	N/A

Attribute	Description	Required	Default Value
where	<p>Filtering condition. Oracle Reader concatenates an SQL command based on specified column, table, and WHERE conditions and extracts data according to the SQL command. For example, you can set the WHERE condition as <code>row_number()</code> during a test. In actual service scenarios, incremental synchronization typically synchronizes the data generated on the current day. You can specify the WHERE condition as <code>id > 2 and sex = 1</code>.</p> <ul style="list-style-type: none"> The where condition can be effectively used for incremental synchronization. The WHERE condition can be effectively used for incremental synchronization. 	No	N/A
querySQL (advanced mode, wizard mode not available)	<p>In some service scenarios, the WHERE condition is insufficient for filtering. In such cases, you can customize a SQL filter using this parameter. When this item is configured, the data synchronization system filters data using this configuration item directly instead of such configuration items as table and column. For example, for data synchronization after multi-table join, use <code>select a ,b from table_a join table_b on table_a.id = table_b.id</code>. When querySQL is configured, Oracle Reader directly ignores the configuration of table, column, and WHERE conditions.</p>	No	N/A
fetchSize	<p>It defines the pieces of batch data that the plug-in and database server can fetch each time. The value determines the number of network interactions between the DataX system and the server, which can greatly improve data extraction performance.</p> <div>  Note: The fetchsize value (> 2048) may cause the data synchronization process OOM. </div>	No	1,024

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.



Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.
- Data Filtering: you are about to synchronize the filtering criteria for the data, and the limit keyword filtering is not supported for the time being. The SQL syntax is consistent with the selected data source.
- Cut key: You can use a column in the source data table as a cut key, it is recommended that you use a primary key or an indexed column as a split key, and that only fields of type Integer be supported.

During data reading, the data is split based on the configured fields to achieve concurrent reading, improving data synchronization efficiency.



Note:

The configuration of splitting key is related to the source selection in data synchronization. The splitting key configuration item is displayed only when you configure the data source.

2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.



- Peer mapping: Click **Enable Same-Line Mapping** to establish a corresponding mapping relationship in the peer, note that match the data type.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

The function of adding a row is as follows:

- You can enter constants. The value must be enclosed by a pair of single quotes, such as 'abc' and '123'.
- Use this function with scheduling parameters, such as \${bizdate}.
- You can enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as Not identified.

3. Control the tunnel

Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

Configure a job to synchronously extract data from an Oracle database:

```
{
  "type": "job",
  "version": "2.0"} //Indicates the version.
  "steps":[
    {
      "stepType":"oracle",
      "parameter": {
        "fetchSize: 1024, // The configuration item defines
the number of plug-ins and database server-side data acquisition lines
per volume
        "datasource": "", // fill in the added Data Source
Name
        "column": [// column name
          "id",
          "name"
        ],
        "where": "", //Filtering condition
        "splitPk": "", // cut key
        "table": "// table name
      },
      "name": "Reader ",
      "category": "reader"
    },
    { // Below is a stream example, if it is the other plug-in, you
can find the corresponding plug-in, fill in the corresponding content
      "stepType": "stream ",
      "parameter": {}
      "name": "writer",
      "category": "writer"
    }
  ],
  "setting":{
    "errorLimit": {
      "record": "0"//Maximum number of error records
    },
    "speed": {
      "throttle":false,//False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
      "concurrent": "1",//Number of concurrent tasks
      "dmu": 1 // DMU Value
    }
  },
  "order":{
    "hops":[
      {
        "from": "Reader ",
        "to": "Writer"
      }
    ]
  }
} "to": "Writer"
}
```

```
}
```

Additional instructions

Master standby synchronous data recovery problem

Master/slave synchronization means that Oracle uses a master/slave disaster recovery mode in which the slave database continuously restores data from the master database through binlog . Due to the time difference in master/slave data synchronization, especially in some special situations such as network latency, the restored data in the slave database after synchronization are significantly different from the data of the master database, that is to say, the data synchronized in the slave database are not a full image of the master database at the current time.

Consistency Constraints

Oracle is an RDBMS system in terms of data storage, which can provide APIs for querying strong consistency data. For example, if another data writer writes data to the database during a synchronization task, Oracle Reader does not get the newly written data because of the snapshot features of the database. For the snapshot features of the database, see [MVCC Wikipedia](#) .

Above are the characteristics of data synchronization consistency under the Oracle reader single-threaded model, since Oracle reader can use Concurrent Data Extraction Based on your configuration information, data consistency is not strictly guaranteed. When the Oracle reader is split Based on the splitpk data, multiple concurrent tasks are initiated to complete the synchronization of data. Since multiple concurrent tasks do not belong to the same read transaction and time intervals exist between the concurrent tasks, the data is not complete and consistent data snapshot information.

For the need of multithread consistent snapshot, it can not be realized technically at present, but can only be solved from the perspective of engineering. The method of engineering exists, and the following solutions are provided here, and you can choose according to your own circumstances.

- - Use single-threaded synchronization without data sharding. This is slow but can ensure robust data consistency.
- - Close other data writers to ensure the current data is static. For example, you can lock the table or close slave database synchronization. The disadvantage is that the online businesses may be affected.

Database coding problem

Oracle Reader extracts data using JDBC at the underlying level. JDBC is applicable to all types of encodings and can complete transcoding at the underlying level. Therefore, Oracle Reader

can obtain the encoding and complete transcoding automatically without the need to specify the encoding.

Oracle Reader cannot identify the inconsistency between the encoding written to the underlying layer of the Oracle system and the configured encoding, nor provide a solution. Due to this issue, ****the exported codes may contain garbage codes****.

Incremental Synchronization

Since Oracle Reader extracts data using JDBC SELECT statements, you can extract incremental data using the SELECT and WHERE conditions in either of the following ways:

- When online database applications write data into the database, the modify field is filled with the modification timestamp, including addition, update, and deletion (logical deletion). For this type of applications, Oracle Reader only requires the WHERE condition followed by the timestamp of the last synchronization phase.
- For new streamline data, Oracle Reader requires the WHERE condition followed by the maximum auto-increment ID of the last synchronization phase.

In case no field is provided for the business to identify the addition or modification of data, Oracle Reader cannot perform incremental data synchronization and can only perform full data synchronization.

SQL Security

Oracle Reader provides querySQL statements for you to SELECT data by yourself. Oracle Reader conducts no security verification on querySQL.

2.3.2.11 Configure OSS Reader

The OSS Reader plug-in provides the ability to read data from OSS data storage. In terms of underlying implementation, OSS Reader acquires the OSS data using official OSS Java SDK, converts the data to the data synchronization protocol, and passes it to Writer.

- If you want to learn more about OSS products, see the [OSS Product Overview](#).
- For details about OSS Java SDKs, see [Alibaba Cloud OSS Java SDK](#).
- For details on processing non-structured data such as the OSS data, see [Process Non-structured Data](#).

OSS Reader provides the ability to read data from a remote OSS file and convert the data to the Data Integration/datx protocol. OSS file itself is a non-structured data storage. For Data Integration/datx, OSS Reader currently supports the following features:

- Only supports reading TXT files and the shema in the TXT file must be a two-dimensional table .
- Supports CSV-like format files with custom delimiters.
- Supports reading multiple types of data (represented by String) and supports column pruning and column constants.
- Supports recursive reading and filtering by File Name.
- Supports text compression. The available compression formats include gzip, bzip2, and zip.

**Note:**

Note: Multiple files cannot be compressed into one package.

- Supports concurrent reading of multiple objects.


The following are not supported currently:



- Multi-thread concurrent reading of a single object (file).
- Technically, the multi-thread concurrent reading of a single compressed object is not supported.

OSS Reader supports the following data types of OSS: BIGINT, DOUBLE, STRING, DATETIME, and BOOLEAN.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A

Attribute	Description	Required	Default Value
Object	<p>The object information for the OSS, where you can support filling in multiple objects. For example, if the bucket of xxx contains yunshi folder which has ll.txt file, the object is directly specified as yunshi/ll.txt.</p> <ul style="list-style-type: none">• If a single OSS object is specified, OSS Reader only supports single-threaded data extraction. We are planning to provide the function to concurrently read a single non-compressed object with multiple threads.• If multiple OSS objects are specified, OSS Reader can extract data with multiple threads. The number of concurrent threads is specified based on the number of channels.• - If a wildcard is specified, OSS Reader attempts to traverse multiple objects. For details, see OSS Product Overview. <div> Note: > Data synchronization system identifies all objects synchronized in a job as a same data table. You must ensure that all objects are applicable to the same schema information.</div>	Yes	N/A

Attribute	Description	Required	Default Value
column	<p>Description: It refers to the list of fields read, where the type indicates the type of source data, the index indicates the column in which the current column locates (starts from 0), and the value indicates that the current type is constant and the data is not read from the source file but the corresponding column is automatically generated according to the value.</p> <p>By default, you can read data by taking String as the only type. The configuration is as follows:</p> <pre>json "column": ["*"]</pre> <p>You can configure the column field as follows:</p> <pre>json "column": { "type": "long", "index": 0 //Retrieves the int field from the first column of the local file text }, { "type": "string", "value": "alibaba" // HDFS Reader internally generates the alibaba string field as the current field }</pre> <div>  Note: For the specified column information, you must enter type and choose one from index/value. </div>	Yes	Read all according to string type
fieldDelimiter	<p>The read field separator.</p> <div>  Note: The OSS reader needs to specify a field partition when reading data, if you do not specify a default of ';', the interface configuration also defaults to ';'. </div>	Yes	,
compress	Compression type of files. It is left empty by default, which means no compression is performed. Supports the following compression types: gzip, bzip2, and zip.	No	Do not compress
encoding	Description: Encoding of the read files.	No	UTF-8

Attribute	Description	Required	Default Value
nullFormat	Description: Defining null (null pointer) with a standard string is not allowed in text files. Data Synchronization system provides nullFormat to define which strings can be expressed as null. For example, if the source data is "null", if you configure the <code>nullformat = "null "</code> , the data synchronization system is treated as a null field.	No	N/A
Skipheader	Description: The header of a file in CSV-like format is skipped if it is a title. Headers are not skipped by default. skipHeader is not supported for file compression.	No	false
csvReaderConfig	Description: Reads the parameter configurations of CSV files. It is the Map type. This reading is performed by the CsvReader for reading CSV files and involves many configuration items, whose defaults are used if they are not configured.	No	N/A

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.

The screenshot shows the 'Data Source' configuration wizard. It has two main sections: 'Source' and 'Destination'. The 'Source' section includes fields for 'Data Source' (set to OSS), 'Object Prefix' (with an 'Add +' button), 'File Type' (set to csv), 'Column Separator' (set to comma), 'Encoding' (set to UTF-8), 'Null String' (with a placeholder 'Enter the sting that represents null'), 'Compression' (set to None), and 'Include Header' (set to No). The 'Destination' section includes fields for 'Data Source' (set to OSS), 'Object Prefix', 'File Type' (set to csv), 'Column Separator', 'Encoding' (set to UTF-8), 'Null String' (with a placeholder 'Enter the sting that represents null'), 'Time Format' (with a placeholder 'Enter the time format'), and 'Solution to Duplicate' (set to Replace the Original File). A 'Preview' button is located at the bottom of the Source section.

Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Object Prefix: Object in the preceding parameter description.

**Note:**

If your OSS file name has a section named according to the time of day, such as `aaa/20171024abc.txt`, about the object system parameters, `aaa/${bdp.system.bizdate}abc.txt` can be set.

- Column delimiter: fieldDelimiter in the preceding parameter description, which defaults to ",".
- Encoding format: encoding in the preceding parameter description, which defaults to utf-8.
- null Value: nullFormat in the preceding parameter description. Enter the field to be expressed as null into a text box. If source end exists, the corresponding field is converted to null.
- Compression Format: compress in the preceding parameter description, which defaults to "no compression".
- Whether to Include the Table Header: skipHeader in the preceding parameter description, which defaults to "No".

2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.

Mapping			Source Table	Destination Table	Hide
Location/Value	Type		Sequence in destination table	Identified	
Column 0	string	●	Column 0	Unidentified	Map of the same name
Column 1	string	●	Column 1	Unidentified	Enable Same-Line Mapping
Column 2	string	●	Column 2	Unidentified	Cancel mapping
Column 3	string	●	Column 3	Unidentified	
Column 4	string	●	Column 4	Unidentified	

- Peer mapping: Click **Enable Same-Line Mapping** to establish a corresponding mapping relationship in the peer, note that match the data type.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

3. Control the tunnel

03 Channel

You can control the data synchronization process through the transmission rate and the number of allowed dirty data records. See [data synchronization documents](#).

* DMU: 6

* Number of Concurrent Jobs: 8

* Transmission Rate: ☐ Unlimited ☒ Limited 10 MB/s

If there are more than: Maximum number of dirty data records. Dirty data is allowed by default. dirty data records, the task ends.

Task's Resource Group: Default resource group

Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.

Development in script mode

The following is a script configuration sample. For details about parameters, see the preceding [Parameter Description](#).

```
{
  "type": "job",
  "version": "2.0", // Indicates the version.
  "steps": [
    {
      "stepType": "oss", // plug-in name
      "parameter": {
        "nullFormat": "", // nullformat defines which strings
        can be expressed as null?
        "compress": "", // text compression type
        "datasource": "", // Data Source
        "column": [ // Field
          {
            "index": 0, // column sequence number
            "type": "string" // data type
          },
          {
            "index": 1,
            "type": "long"
          },
          {
            "index": 2,
            "type": "double"
          }
        ]
      }
    }
  ]
}
```

```

        "index": 3,
        "type": "boolean"
    },
    {
        "format": "yyyy-MM-dd HH:mm:ss", // time format
        "index": 4,
        "type": "date"
    }
],
    "skipHeader": "", // the class CSV format file may have
a header as a header condition, need to skip
    "encoding": "", // encoding format
    "fieldDelimiter": ",", // Separator
    "fileFormat": "", // File type
    "object": [] // object prefix
},
    "name": "Reader",
    "category": "reader"
},
    { // The following is a writer template. You can find the
corresponding writer plug-in documentations.
        "stepType": "stream ",
        "parameter": {},
        "name": "Writer ",
        "category": "writer"
    }
],
    "setting": {
        "errorLimit": {
            "record": "" // Number of error records
        },
        "speed": {
            "throttle": false, // False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
            "concurrent": "1", // Number of concurrent tasks
            "dmu": 1 // DMU Value
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader ",
                "to": "Writer"
            }
        ]
    }
}

```

2.3.2.12 Configuring FTP Reader

FTP Reader provides the ability to read data from a remote FTP file system. At the underlying implementation level, FTP Reader acquires the remote FTP file data, converts the data to the data synchronization and transmission protocol, and transmits it to Writer.

What is saved to the local file is a two-dimensional table in a logic sense, for example, text information in a CSV format.

FTP Reader allows you to read data from a remote FTP file and convert the data to the data synchronization protocol. Remote FTP file itself is a non-structured data storage file. For data synchronization, FTP Reader currently supports the following features:

- Only supports reading TXT files and the schema in the TXT file must be a two-dimensional table.
- Supports CSV-like format files with custom delimiters.
- Supports reading multiple types of data (represented by String) and supports column pruning and column constants.
- Supports recursive reading and filtering by File Name.
- Supports text compression. The available compression formats include gzip, bzip2, zip, lzo, and lzo_deflate.
- Supports concurrent reading of multiple files.

The following two features are not supported currently:


- Multi-thread concurrent reading of a single file. This feature involves the internal splitting algorithm of a single file (under planning).
- Technically, the multi-thread concurrent reading of a single compressed file is not supported.


The remote FTP file itself does not provide data types, which are defined by DataX FtpReader:

Internal DataX type	Data type of a remote FTP file
Long	Long
Double	Double
String	String
Boolean	Boolean
Date	Date

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A

Attribute	Description	Required	Default Value
path	<p>Description: The path of the remote FTP file system. Multiple paths can be specified.</p> <ul style="list-style-type: none"> If a single remote FTP file is specified, FTP Reader only supports single-threaded data extraction. We are planning to provide the function to concurrently read a single non-compressed file with multiple threads. If multiple remote FTP files are specified, FTP Reader can extract data with multiple threads. The number of concurrent threads is specified based on the number of channels. If a wildcard is specified, FTP Reader attempts to traverse multiple files. For example, when / is specified, FTP Reader reads all the files under the / directory. When /bazhen/ is specified, FTP Reader reads all the files under the bazhen directory. Currently, FTP Reader only supports * as the file wildcard. <div>  Note: <ul style="list-style-type: none"> Data synchronization system identifies all text files synchronized in a job as a same data table. You must ensure that all files are applicable to the same schema information. You must ensure that the file to be read is in CSV-like format, and the read permission must be granted to the data synchronization system. If no matching file exists for extraction in the path specified by Path, an error may occur in the synchronization task. </div>	Yes-	N/A

Attribute	Description	Required	Default Value
column	<p>Description: It refers to the list of fields read, where the type indicates the type of source data, the index indicates the column in which the current column locates (starts from 0), and the value indicates that the current type is constant and the data is not read from the source file but the corresponding column is automatically generated according to the value.</p> <p>By default, you can read data by taking String as the only type. The configuration is as follows: <code>"column": ["*"]</code>. You can configure the column field as follows:</p> <pre>{ "type": "long", "index": 0 //Read the int field from the first column of the remote FTP file text }, { "type": "string", "value": "alibaba" //FtpReader internally generates the alibaba string field as the current field }</pre> <p>For the specified column information, you must enter type and choose one from index/value.</p>	Yes	Read all according to string type
fieldDelimiter	<p>The delimiter used to separate the read fields.</p> <div>  Note: Note that a field delimiter must be specified when FTP Reader reads data. By default, if "," is not specified, it is entered in the interface configuration. </div>	Yes	,
Skipheader	<p>Description: The header of a file in CSV-like format is skipped if it is a title. Headers are not skipped by default. skipHeader is not supported for file compression.</p>	No	false
encoding	<p>Description: Encoding of the read files.</p>	No	utf-8
nullFormat	<p>Description: Defining null (null pointer) with a standard string is not allowed in text files. Data synchronization provides nullFormat to define which strings can be expressed as null.</p> <p>For example, when <code>nullFormat: "null"</code>, is configured, if the source data is "null", it is considered as a null field in data synchronization.</p>	No	N/A

Attribute	Description	Required	Default Value
markDoneFileName	Description: The name of the file marked as "done". Check MarkDoneFile before data synchronization. If the file does not exist, wait for a while and check again. If the file exists, start the data synchronization task.	No	N/A
MaxRetryTime	Description: The number of attempts made to check MarkDoneFile. Default value is 60. Try once every one minute for 60 minutes in total.	No	600
csvReaderConfig	Description: Reads the parameter configurations of CSV files. It is the Map type. This reading is performed by the CsvReader for reading CSV files and involves many configuration items, whose defaults are used if they are not configured.	No	N/A
fileFormat	Description: Type of the read file. By default, the file is read as a CVS file and the file content is parsed to a logical two-dimensional table for processing. If you set this filed to binary, the file is copied and transmitted in the binary format. Such setting is applicable for peer-to-peer copy of directories between FTP and OSS files. You do not need to configure this item generally.	No	N/A

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.

Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- File Path: that is, path in the above parameter description.
- Column delimiter: fieldDelimiter in the preceding parameter description, which defaults to ",".
- Encoding format: encoding in the preceding parameter description, which defaults to utf-8.
- null value: nullFormat in the preceding parameter description, to define a string that represents the null value.
- Compression Format: compress in the preceding parameter description, which defaults to "no compression".
- Whether to Include the Table Header: skipHeader in the preceding parameter description, which defaults to "No".

2. Field mapping: The column in the preceding parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.

- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

3. Channel control

Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. It represents a unit of data synchronization processing capability given limited CPU, memory, and network resources.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

Configure a synchronous Extraction Data job from the FTP database.

```
{
  "type": "job",
  "version": "2.0" } //Indicates the version.
  "steps": [
    {
      "stepType": "ftp", // plug-in name
      "parameter": {
        "path": [], //File path
        "nullFormat": "", // Null Value
        "compress": "", // compression format
        "datasource": "", // Data Source
      }
    }
  ]
}
```

```

        "column": [// Field
            {
                "index": 0, // serial number
                "type": "// Field Type"
            }
        ],
        "skipHeader": "", // contains a header?
        "fieldDelimiter": ",", //Delimiter of each column
        "encoding": "UTF-8", // encoding format
        "fileFormat": "csv"//File type
    },
    "name": "Reader ",
    "category": "reader"
},
{//The following is a reader template. You can find the
corresponding reader plug-in documentations.
    "stepType": "stream ",
    "parameter": {}
    "name": "Writer ",
    "category": "writer"
}
],
"setting": {
    "errorLimit": {
        "record": "0"//Number of error records
    },
    "speed": {
        "throttle": false, //False indicates that the traffic is
        not throttled and the following throttling speed is invalid. True
        indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader ",
            "to": "Writer"
        }
    ]
}
}

```

2.3.2.13 Configure Table Store(OTS) Reader

In this article we will show you the data types and parameters supported by OTS Reader and how to configure Reader in script mode.

The OTS Reader plug-in provides the ability to read data from Table Store(OTS), which allows incremental data extraction within the specified data extraction range. Currently, the following three extraction methods are supported:

- Full table extraction
- Specified range extraction
- Specified partition extraction

Table Store is a NoSQL database service built upon Alibaba Cloud's Apsara distributed system, enabling you to store and access massive structured data in real time. Table Store organizes data into instances and tables. Using data partition and server load balancing technology, it provides seamless scaling.

In short, OTS Reader connects to OTS server by using official Table Store Java SDK, reads and transfers data to data synchronization field information according to official data synchronization protocol standard, and then transmits the information to downstream Writer side.

Based on Table Store table range, OTS Reader divides the range into multiple tasks according to the number of data synchronization concurrencies. Each task is implemented with an OTS Reader thread.

Currently, OTS Reader supports all Table Store types. The conversion of Table Store types in the OTS Reader is as follows:

Category	MySQL data type
Integer	Integer
Float	Double
String type	String
Boolean	Boolean
Binary	Binary

**Note:**

Table Store itself does not support "date" type. Long value is generally used as Unix TimeStamp at application layer when an error is reported.

Parameter description

Attribute	Description	Required	Default Value
endpoint	The endpoint for the OTS server (service address). For more information, see Endpoint .	Yes	N/A
accessId	The accessId of the Table Store.	Yes	N/A
accessKey	The accesskey of the Table Store.	Yes	N/A

Attribute	Description	Required	Default Value
Instance name	Description: The name of Table Store instance. The instance is an entity for using and managing OTS service. After you enable the Table Store service, you can create an instance in the console to create and manage tables. The instance is the basic unit for Table Store resource management. All access control and resource measurement done by the Table Store for applications are completed at the instance level.	Yes	N/A
table.	Description: The name of the table to be extracted. Only one table can be filled in. Multi-table synchronization is not required for Table Store.	Yes	N/A
column	<p>Description: The column name set to be synchronized in the configured table. Field information is described with arrays in JSON. Because Table Store itself is a NoSQL system, the corresponding field name must be specified when OTS Reader extracts data.</p> <ul style="list-style-type: none"> Reading of ordinary columns is supported, for example, {"name":"col1"}. Reading of partial columns is supported. OTS Reader does not read unconfigured columns. Reading of constant columns is supported, for example, {"type":"STRING", "value" : "DataX"}. "type" is used to describe constant types. Currently supported types include STRING, INT, DOUBLE, BOOL, BINARY (entered with a value encoded using Base64), INF_MIN (minimum system limit value for Table Store. You cannot enter the value attribute if this value is specified, otherwise an error may occur), and INF_MAX (maximum system limit value for Table Store. You cannot enter the value attribute if this value is specified, otherwise an error may occur). Function or custom expression is not supported. Because Table Store itself does not provide function or expression similar to SQL, OTS Reader does not provide function or expression either. 	Yes	N/A

Attribute	Description	Required	Default Value
begin/end	<p>Description: This configuration item that must be used in pairs allows data to be extracted from OTS table range. "begin/end" describes the distribution of OTS PrimaryKeys within the range which must cover all PrimaryKeys. The range of PrimaryKeys under the OTS table requires to be specified. For the range with infinite limit, use {"type": "INF_MIN"} and {"type": "INF_MAX"}. For example, if you want to extract data from an OTS table with the primary key of [DeviceID, SellerID], begin/end is configured as follows:</p> <pre> "range":{ "begin":[{"Type": "inf_min"}, // specify the minimum value of ergonomic ID], "end":[{"type": "INF_MAX"}, // specify the maximum value for ergonomic ID Extraction] } </pre> <p>To extract data from the entire table, use the following configuration:</p> <pre> "range":{ "begin":[{"type": "INF_MIN"}, // specify the minimum value of ergonomic ID], "end":[{"type": "INF_MAX"}, // specify the maximum value for ergonomic deviceID Extraction] } </pre>	Yes	Blank
split	<p>Description: This is an advanced configuration item for the configuration of custom splitting, which is generally not recommended.</p> <p>Application scenario: The custom splitting rule is generally used when OTS Reader's auto splitting policy is invalid in the hotspot where Table Store data is stored.</p> <p>"split" specifies a splitting point within the range between Begin and End and only the information of splitting point for partitionKey, which means that only partitionKey is configured for split, but not all PrimaryKeys require to be specified.</p> <p>If you want to extract data from an OTS table with the</p>	No	N/A

Development in script mode

Configure a job to extract data synchronously from the entire Table Store table to local machine.

```
{
  "type": "job",
  "version": "2.0", //Indicates the version.
  "steps": [
    {
      "stepType": "ots", //plug-in name
      "parameter": {
        "datasource": "", //Data Source
        "column": [// Field
          {
            "name": "columnn1" // field name
          },
          {
            "name": "column2"
          },
          {
            "name": "column3"
          },
          {
            "name": "column4"
          },
          {
            "name": "column5"
          }
        ],
        "range": {
          "split": [
            {
              "type": "INF_MIN"
            },
            {
              "type": "STRING",
              "value": "splitPoint1"
            },
            {
              "type": "STRING",
              "value": "splitPoint2"
            },
            {
              "type": "STRING",
              "value": "splitPoint3"
            },
            {
              "type": "INF_MAX"
            }
          ],
          "end": [
            {
              "type": "INF_MAX"
            },
            {
              "type": "INF_MAX"
            },
            {
              "type": "STRING",
              "value": "end1"
            },
            {
              "type": "INT",
```

```

        "Value": "100"
      },
    ],
    "begin": [
      {
        "type": "INF_MIN"
      },
      {
        "type": "INF_MIN"
      },
      {
        "type": "STRING",
        "value": "begin1"
      },
      {
        "type": "INT",
        "value": "0"
      }
    ]
  },
  "table": "// table name",
  "name": "Reader ",
  "category": "reader"
},
{
  //The following is a writer template. You can find the
  corresponding writer plug-in documentations.
  "stepType": "stream",
  "parameter": {},
  "name": "writer",
  "category": "writer"
},
],
"setting": {
  "errorLimit": {
    "record": "0" //Number of error records
  },
  "speed": {
    "throttle": false, //False indicates that the traffic is
    not throttled and the following throttling speed is invalid. True
    indicates that the traffic is throttled.
    "concurrent": "1", //Number of concurrent tasks
    "dmu": 1 // DMU Value
  }
},
"order": {
  "hops": [
    {
      "from": "Reader ",
      "to": "Writer"
    }
  ]
}
}

```



```
}
```

2.3.2.14 Configuring PostgreSQL Reader

In this article we will show you the data types and parameters supported by PostgreSQL Reader and how to configure Reader in both wizard mode and script mode.

The PostgreSQL Reader plug-in reads data from PostgreSQL databases. At the underlying implementation level, PostgreSQL Reader connects to a remote PostgreSQL database through JDBC and runs SELECT statements to extract data from the database. On the public cloud, RDS provides a PostgreSQL storage engine.

Specifically, PostgreSQL Reader connects to a remote PostgreSQL database through the JDBC connector. The SELECT SQL query statements are generated and sent to the remote PostgreSQL database based on your configuration. Then, the SQL statements are run and the returned results are assembled into abstract datasets using the custom data types of data integration. Datasets are passed to the downstream writer for processing.

- PostgreSQL Reader concatenates the table, column, and where information you configured into SQL statements, and sends them to the PostgreSQL database.
- PostgreSQL directly sends the configured querySQL information to the PostgreSQL database.

Type conversion list

PostgreSQL Reader supports most data types in PostgreSQL. Check whether your data type is supported.

The PostgreSQL reader has a list of Type transformations for PostgreSQL, as shown below.

Category	PostgreSQL data type
Integer	bigint, bigserial, integer, smallint, and serial
Floating point	double precision, money, numeric, and real
String	varchar, char, text, bit, and inet
Date and time	date, time, and timestamp
Boolean	bool
Binary	bytea



Note:


- Except the preceding field types, other types are not supported.

- For "money", "inet", and "bit", you need to use syntaxes such as "a_inet::varchar" to convert data types.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
table.	The column name set to be synchronized in the configured table.	Yes	N/A
column	<p>Field information is described with arrays in JSON. [*] indicates all columns by default.</p> <ul style="list-style-type: none"> Column pruning is supported, which means you can select some columns to export. Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. Constant configuration is supported. You must follow the MySQL SQL syntax format, for example <code>[["id", "table", "1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3", "true"]]</code>. <ul style="list-style-type: none"> ID is normal column name Table is a column name that contains Reserved Words 1 For plastic digital Constants 'mingya.wmy' is a String constant (note that a pair of single quotes is required) Null is a null pointer Char_length (s) is the computed String Length Function 2.3 is a floating point number True is a Boolean Value Column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	N/A

Attribute	Description	Required	Default Value
SplitPk	<p>- Split key: If you specify the splitPk when using PostgreSQLReader to extract data, it means that you want to use the fields represented by the splitPk for data sharding. In this case, the Data Integration initiates concurrent jobs to synchronize data, which greatly improves the efficiency of data synchronization.</p> <ul style="list-style-type: none"> If you are using splitPk, we recommend that you use the primary keys of tables, because the primary keys are generally even and data hot spots are less prone to split data fragments. Currently, splitPk only supports data sharding for integer data types. Other types such as string, floating point, and date are not supported. If you specify an unsupported data type, the splitPk is ignored and the data is synchronized using a single channel. If the splitPk is not specified (for example, splitPk is not provided or splitPk value is null), the table data is synchronized using a single channel. 	No	N/A
where	<p>PostgreSQLReader concatenates an SQL statement based on the specified column, table, and WHERE conditions and extracts data according to the SQL statement. For example, you can set the WHERE condition during a test. In actual service scenarios, the data on the current day are usually required to be synchronized, in which case you can set the WHERE condition as id > 2 and sex = 1.</p> <ul style="list-style-type: none"> The where condition can be effectively used for incremental synchronization. If the where condition is not set or is left null, full table data synchronization is applied. 	No	N/A

Attribute	Description	Required	Default Value
querySQL (advanced mode, wizard mode not available)	In some business scenarios, the where condition is insufficient for filtration. In such cases, the user can customize a filter SQL using this configuration item. When this item is configured, the data synchronization system filters data using this configuration item directly instead of such configuration items as tables, columns, and splitPk. For example, for data synchronization after multi-table join, use <code>select a,b from table_a join table_b on table_a.id = table_b.id</code> . When querySQL is configured, PostgreSQL Reader directly ignores the configuration of table, column, and WHERE conditions.	No	N/A
Fetchsize	<p>Description: It defines the pieces of batch data that the plug-in and database server can fetch each time. The value determines the number of network interactions between the DataX system and the server, which can greatly improve data extraction performance.</p> <div>  Note: The fetchsize value (> 2048) may cause the data synchronization process oom. </div>	No	512 MB

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.

Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.
- Data Filtering: you are about to synchronize the filtering criteria for the data, and the limit keyword filtering is not supported for the time being. The SQL syntax is consistent with the selected data source.
- Cut key: You can use a column in the source data table as a cut key, it is recommended that you use a primary key or an indexed column as a split key, and that only fields of type Integer be supported.

During data reading, the data is split based on the configured fields to achieve concurrent reading, improving data synchronization efficiency.

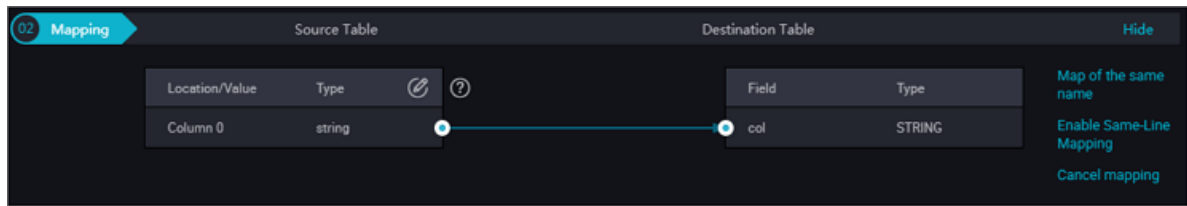


Note:

The configuration of splitting key is related to the source selection in data synchronization. The splitting key configuration item is displayed only when you configure the data source.

2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.

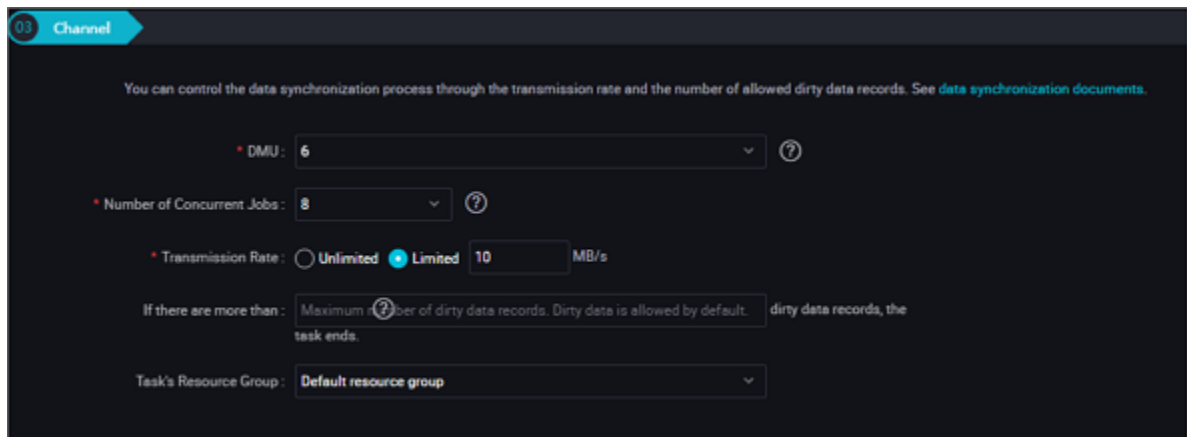


- Peer mapping: Click **Enable Same-Line Mapping** to establish a corresponding mapping relationship in the peer, note that match the data type.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

The function of adding a row is as follows:

- You can enter constants. The value must be enclosed by a pair of single quotes, such as 'abc' and '123'.
- Use this function with scheduling parameters, such as \${bizdate}.
- You can enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as Not identified.

3. Control the tunnel



Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.

- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group. For more information, see [Add scheduling resources](#).

Development in script mode

Configure a job to synchronously extract data from a PostgreSQL database.

```
{
  "type": "job",
  "version": "2.0"} //Indicates the version
  "steps": [
    {
      "stepType": "postgresql", //plug-in name
      "parameter": {
        "datasource": "", // Data Source
        "column": [// Field
          "col1",
          "col2"
        ],
        "where": "", //Filtering condition
        "splitPk": "", //using the fields represented by splitpk
        for Data Division, data Synchronization thus starts concurrent tasks
        for Data Synchronization
        "table": "// table name
      },
      "name": "Reader ",
      "category": "reader"
    },
    { //The following is a reader template. You can find correspond
      ing writer plug-in documentations
      "stepType": "stream ",
      "parameter": {},
      "name": "Writer ",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, //False indicates that the traffic is
      not throttled and the following throttling speed is invalid. True
      indicates that the traffic is throttled.
      "concurrent": "1", //Number of concurrent tasks
      "dmu": 1 //DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "from": "Reader ",
        "to": "Writer"
      }
    ]
  }
}
```

```
}
```

Additional instructions

Master standby synchronous data recovery problem

Master/slave synchronization means that PostgreSQL uses a master/slave disaster recovery mode in which the slave database continuously restores data from the master database through binlog. Due to the time difference in primary/backup data synchronization, especially in some special situations such as network latency, the restored data in the backup database after synchronization is significantly different from the data of the primary database, that is to say, the data synchronized from the backup database is not a full image of the primary database at the current time.

If the data integration system synchronizes data of the RDS provided by Alibaba Cloud, the data is directly read from the primary database without data restoration concerns. However, this may cause concerns on the master database load. Configure it properly for throttling.

Consistency Constraints

PostgreSQL is an RDBMS system in terms of data storage, which can provide APIs for querying strong consistency data. For example, if another data writer writes data to the database during a synchronization task, PostgreSQL Reader does not get the newly written data because of the snapshot features of the database. For the characteristics of database snapshots, see [MVCC Wikipedia](#).

Above are the characteristics of data synchronization consistency under the PostgreSQL reader single-threaded model, since PostgreSQL reader can use Concurrent Data Extraction Based on your configuration information, therefore, data consistency cannot be strictly guaranteed. When the PostgreSQL reader is split Based on the splitpk data, multiple concurrent tasks are initiated to complete the synchronization of data. Since multiple concurrent tasks do not belong to the same read transaction with each other, there are time intervals for multiple concurrent tasks at the same time, therefore, this data is not a complete, consistent snapshot of the data.

For the need of multithreaded consistent snapshot, it can not be realized technically at present, but can only be solved from the perspective of engineering. The method of engineering exists, and the following solutions are provided here, and you can choose according to your own circumstances.

- Use single-threaded synchronization without data sharding. This is slow but can ensure robust data consistency.

- Close other data writers to ensure the current data is static. For example, you can lock the table or close slave database synchronization. The disadvantage is that the online businesses may be affected.

Database coding problem

PostgreSQL supports EUC_CN and UTF-8 encoding for simplified Chinese. PostgreSQL Reader extracts data using JDBC at the underlying level. JDBC is applicable to all types of encodings and can complete the transcoding at the underlying level. Therefore, PostgreSQL Reader can acquire the encoding and complete transcoding automatically without the need to specify the encoding.

PostgreSQL Reader cannot identify the inconsistency between the encoding written to the underlying layer of PostgreSQL and the configured encoding, nor provide a solution. Due to this issue, the exported codes may contain garbage codes.

Incremental Synchronization

PostgreSQL reader uses a jdbc select statement for data extraction, so you can use select... Where... in either of the following ways:

- When online database applications write data into the database, the modify field is filled with the modification timestamp, including addition, update, and deletion (logical deletion). For this type of applications, PostgreSQL Reader only requires the WHERE condition followed by the timestamp of the last synchronization phase.
- For new streamline data, PostgreSQL Reader requires the WHERE condition followed by the maximum auto-increment ID of the last synchronization phase.

In case that no field is provided for the business to identify the addition or modification of data, PostgreSQL Reader cannot perform incremental data synchronization and can only perform full data synchronization.

SQL Security

PostgreSQL Reader provides querySQL statements for you to SELECT data by yourself. PostgreSQL Reader conducts no security verification on querySQL.

2.3.2.15 Configuring SQL server Reader

In this article we will show you the data types and parameters supported by SQL server Reader and how to configure Reader in both wizard mode and script mode.

The SQL Server Reader plug-in provides the ability to read data from SQL Server. At the underlying implementation level, SQL Server Reader connects to a remote SQL Server database through JDBC and runs SELECT statements to extract data from the database.

Specifically, SQL Server Reader connects to a remote SQL Server database through the JDBC connector. The SELECT SQL query statements are generated and sent to the remote SQL Server database based on your configuration. Then, the SQL statements are run and the returned results are assembled into abstract datasets using the custom data types of data integration. Datasets are passed to the downstream writer for processing.

- SQL Server Reader concatenates the table, column, and WHERE information you configured into SQL statements and sends them to the SQL Server database.
- SQL Server directly sends the querySQL information you configured to the SQL Server database.

SQL Server Reader supports most data types in SQL Server. Check whether your data type is supported.


SQL Server Reader converts SQL Server data types as follows:

Category	SQL Server data type
Integer	bigint, int, smallint, and tinyint
Float	float, decimal, real, and numeric
String type	char, nchar, ntext, nvarchar, text, varchar, nvarchar (MAX), and varchar (MAX)
Date and time type	date, datetime, and time
boolean	bit
binary, varbinary, varbinary (MAX), and timestamp	Binary, varbinary, varbinary (max), and timestamp

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
table.	The table selected for synchronization. One job can only synchronize one table.	Yes	N/A

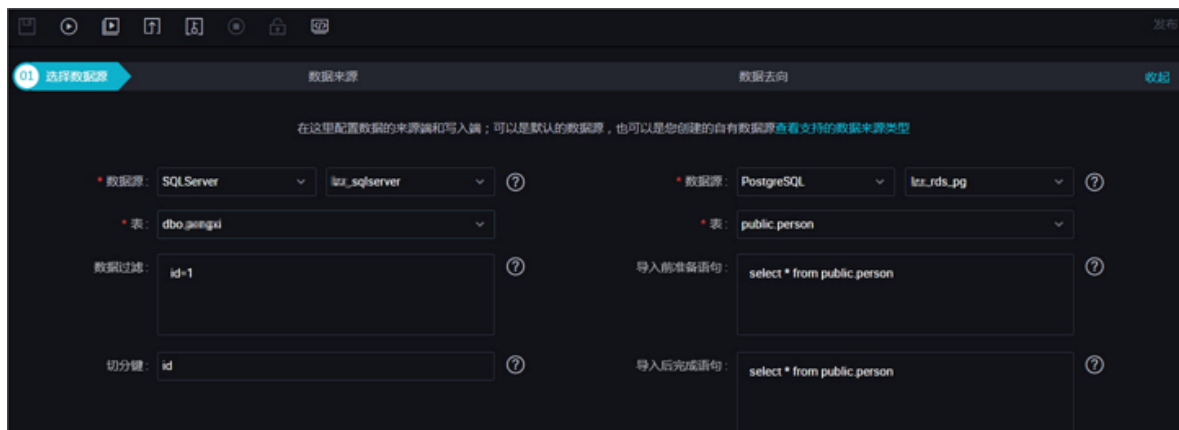
Attribute	Description	Required	Default Value
column	<p>Description: The column name set to be synchronized in the configured table. Field information is described with arrays in JSON. [""] indicates all columns by default.</p> <ul style="list-style-type: none"> Column pruning is supported, which means you can select some columns to export. Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. Constant configuration is supported. You must follow the MySQL SQL syntax format, for example <code>["id", "table", "1", "'mingya.wmy'", "'null'", "to_char(a + 1)", "2.3", "true"]</code>. <ul style="list-style-type: none"> ID is normal column name Table is a column name that contains Reserved Words 1 For plastic digital Constants 'mingya.wmy' is a String constant (note that a pair of single quotes is required) null refers to the null pointer to_char(a + 1) is a function expression 2.3 is a floating point number true is a Boolean value Column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	N/A
splitPk	<p>If you specify the splitPk when using SQL Server Reader to extract data, it means that you want to use the fields represented by splitPk for data sharding. Then, the data synchronization system starts concurrent tasks to synchronize data, which greatly improves the efficiency of data synchronization.</p> <ul style="list-style-type: none"> We recommend that the splitPk users use the primary keys of tables, because the primary keys are generally even and data hot spots are less prone to split data fragments. Currently, splitPk only supports data sharding for integer data types. Other types such as float point, string, and date are not supported. If you specify an unsupported data type, SQL Server Reader reports an error. 	No	N/A

Attribute	Description	Required	Default Value
where	<p>Description: Filtering condition. SQL Server Reader concatenates an SQL command based on the specified column, table, and WHERE conditions and extracts data according to the SQL command. For example, you can specify the where condition as limit 10 during a test. In actual business scenarios, the data on the current day is usually required to be synchronized. You can specify the WHERE condition as <code>gmt_create > \$bizdate</code>.</p> <ul style="list-style-type: none"> The where condition can be effectively used for incremental synchronization. The WHERE condition can be effectively used for incremental synchronization. If the value is null, it means synchronizing all the information in the table. 	No	N/A
querySQL	<p>In some business scenarios, the WHERE condition is insufficient for filtration. In such cases, you can customize a filter SQL statement using this configuration item. When this item is configured, the data synchronization system filters data using this configuration item directly instead of such configuration items as table and column. For example, for data synchronization after multi-table join, use <code>select a,b from table_a join table_b on table_a.id = table_b.id</code>. When querySQL is configured, SQL Server Reader directly ignores the configuration of table, column, and WHERE conditions.</p>	No	N/A
fetchSize	<p>It defines the pieces of batch data that the plug-in and database server can fetch each time. The value determines the number of network interactions between the data synchronization system and the server, which can greatly improve data extraction performance.</p> <div>  Note: A value greater than 2048 may lead to OOM for data synchronization. </div>	No	1,024

Development in wizard mode

1. Choose source

Data source and destination

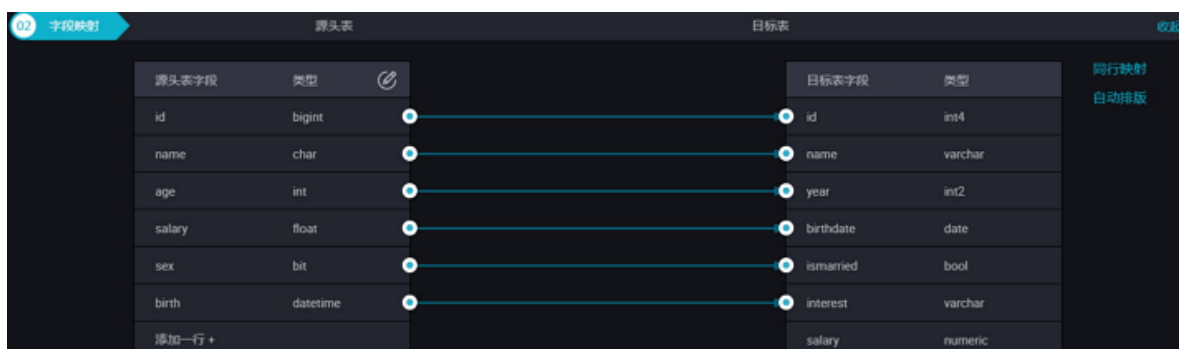


Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: The table in the preceding parameter description. Select the table to be synchronized.
- Filtering condition: You should synchronize the filtering conditions for the data. Limit keyword filter is not supported yet. SQL syntaxes vary with data sources.
- Splitting key: You can use a column in the source table as the splitting key. It is recommended to use a primary key or an indexed column as the splitting key.

2. Field mapping: column in the preceding parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.



- Peer mapping: Click **Enable Same-Line Mapping** to establish a corresponding mapping relationship in the peer, note that match the data type.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

The function of adding a row is as follows:

- you can enter constants. Each constant must be enclosed in a pair of single quotes, such as 'abc' and '123'.
- Use this function with scheduling parameters, such as \${bizdate}.
- Enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as 'Not Identified'.

3. Channel control

Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.-
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group. For more information, see [Add scheduling resources](#).

Development in script mode

Configure a job to synchronously extract data from an SQL Server database:

```
{
  "type": "job",
  "version": "2.0"} //Indicates the version.
  "steps": [
    {
      "stepType": "SQL Server", // plug-in name
      "parameter": {
```

```

        "datasource": "", // Data Source
        "column": [// column name
            "id",
            "name"
        ],
        "where": "", //Filtering condition
        "splitPk": "", // If split PK is specified, indicates
that you want to slice the data using the fields represented by
splitpk
        "table": "// Data Sheet
    },
    "name": "Reader ",
    "category": "Reader"
},
{//The following is a writer template. You can find the
corresponding writer plug-in documentations.
    "stepType": "stream ",
    "parameter": {}
    "name": "writer",
    "category": "writer"
}
],
"setting": {
    "errorLimit": {
        "record": "0"//Number of error records
    },
    "speed": {
        "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader ",
            "to": "Writer"
        }
    ]
}
}

```

Additional instructions

Master standby synchronous data recovery problem

Master/slave synchronization means that SQL Server uses a master/slave disaster recovery mode in which the slave database continuously restores data from the master database through binlog. Due to the time difference in primary/backup data synchronization, especially in some special situations such as network latency, the restored data in the backup database after synchronization is significantly different from the data of the primary database, that is to say, the data synchronized from the backup database is not a full image of the primary database at the current time.

If the data integration system synchronizes data of the RDS provided by Alibaba Cloud, the data is directly read from the primary database without data restoration concerns. However, this may cause concerns on the master database load. Configure it properly for throttling.

Consistency Constraints

SQL Server is an RDBMS system in terms of data storage, which can provide APIs for querying strong consistency data. For example, if another data writer writes data to the database during a synchronization task, SQL Server Reader does not get the newly written data because of the snapshot features of the database. For more information on the snapshot features of the database, refer to the [MVCC Wikipedia](#).

Above are the characteristics of data synchronization consistency under the SQL Server reader single-threaded model, since SQL Server reader can use Concurrent Data Extraction Based on your configuration information, therefore, data consistency cannot be strictly guaranteed. When the SQL Server reader is split Based on the splitpk data,, multiple concurrent tasks are initiated to complete the synchronization of data. Since multiple concurrent tasks do not belong to the same read transaction with each other, there are time intervals for multiple concurrent tasks at the same time, therefore, this data is not a complete, consistent snapshot of the data.

For the need of multithreaded consistent snapshot, it can not be realized technically at present, but can only be solved from the perspective of engineering. The method of engineering exists, and the following solutions are provided here, and you can choose according to your own circumstances.

- - Use single-threaded synchronization without data sharding. This is slow but can ensure robust data consistency.
- - Close other data writers to ensure the current data is static. For example, you can lock the table or close slave database synchronization. The disadvantage is that the online businesses may be affected.

Database coding problem

SQL Server Reader extracts data using JDBC at the underlying level. JDBC is applicable to all types of encodings and can complete transcoding at the underlying level. Therefore, SQL Server Reader can identify the encoding and complete transcoding automatically without the need to specify the encoding.

Incremental Synchronization

SQL Server reader uses a JDBC SELECT statement for data extraction, so you can use select... Where... in either of the following ways:

- When online database applications write data into the database, the modify field is filled with the modification timestamp, including addition, update, and deletion (logical deletion). For this type of applications, SQL Server Reader only requires the WHERE condition followed by the timestamp of the last synchronization phase.
- For new streamline data, SQL Server Reader requires the WHERE condition followed by the maximum auto-increment ID of the last synchronization phase.

In case no field is provided for the business to identify the addition or modification of data, SQL Server Reader cannot perform incremental data synchronization and can only perform full data synchronization.

SQL Security

SQL Server Reader provides querySQL statements for you to SELECT data by yourself. SQL Server Reader conducts no security verification on querySQL. The security during use is ensured by the data synchronization users.

2.3.2.16 Configure LogHub Reader

In this article we will show you the data types and parameters supported by LogHub Reader and how to configure Reader in both wizard mode and script mode.

Honed originally by the Big Data demands of Alibaba Group, Log Service (or "LOG" for short, formerly "SLS") is an all-in-one service for real-time data. With its capabilities to collect, consume, deliver, query, and analyze log-type data, Log Service allows you to process and analyze massive amounts of data much more efficiently. LogHub Reader uses the Java SDK of the Log Service to consume real-time log data in LogHub, and converts the log data to the Data Integration transfer protocol and sends the converted data to Writer.

Implementation

LogHub Reader consumes real-time log data in LogHub by using the following version of Log Service Java SDK:

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>aliyun-log</artifactId>
  <version>0.6.7</version>
</dependency>
```

Logstore is a component of the Log Service for collecting, storing, and querying log data. Logstore read and write logs are stored on a shard. Each log library consists of several partitions, each of

which consists of the left closed right open interval of MD5, each interval range is not covered by each other, and the range of all the intervals is the entire MD5 range of values, each partition can provide a certain level of service capability.

- Writing: 5 MB/s, 2000 times/s.
- Read: 10 MB/s, 100 times/s.

LogHub Reader consumes logs in shards, and the detailed consumption process (GetCursor and BatchGetLog-related APIs) is as follows:




- Obtains a cursor based on the interval range.
- Reads logs based on the cursor and step parameters and returns the next cursor.
- Moves the cursor continuously to consume logs.
- Splits tasks by shard for concurrent execution.



LogHub Reader supports LogHub type conversion, as shown in the following table:

Datax internal type	Loghub Data Type
String	String

Parameter description

Attribute	Description	Required	Default Value
endpoint	Description: The Log Service endpoint is a URL for accessing a project and its internal log data. It is associated with the Alibaba Cloud region and name of the project. Service entry for each region, see service entry .	Yes	N/A
accessId	Description: It refers to an AccessKey for accessing the Log Service, which is used to identify the accessing user.	Yes	N/A
accessKey	Description: It refers to another AccessKey for accessing the Log Service, which is used to verify the user's key.	Yes	N/A
project	Description: It refers to the project name of the target Log Service, which is the resource management component in the Log Service for isolating and controlling resources.	Yes	N/A
logstore	Description: It refers to the name of the target Logstore. Logstore is a component of the Log Service for collecting, storing, and querying log data.	Yes	N/A
batchSize	Description: It refers to the number of data entries queried from the Log Service at a time.	No	128

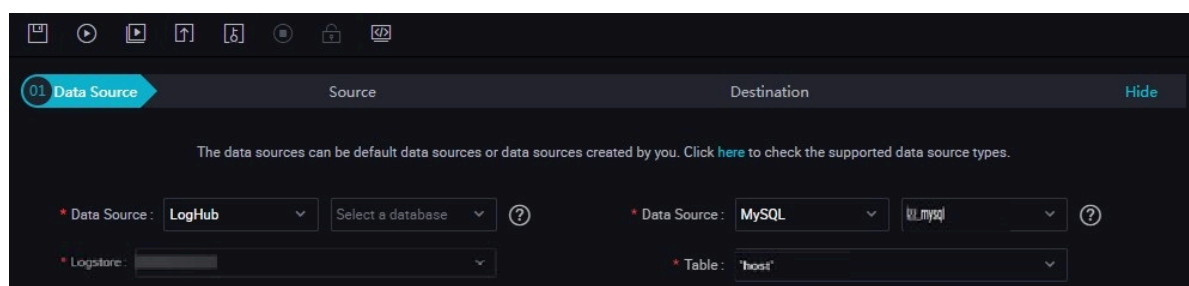
Attribute	Description	Required	Default Value
column	<p>Description: Column names in each data entry. Here, you can set a metadata item in the Log Service as the synchronization column. Supported metadata items include "C_Topic", "C_MachineUUID", "C_HostName", "C_Path", and "C_LogTime", which represent the log topic, unique identifier of the collection machine, host name, path, and log time, respectively.</p> <p>The sub-table represents the log theme, the acquisition machine uniquely identified, the host name, path, log time, and so on.</p> <div>  Note: The values of fields in the format are case insensitive. </div>	Yes	N/A
Begindatetime	<p>Start time of data consumption. The parameter defines the left border of a time range (left closed and right open) in the format of yyyyMMddHHmmss (such as 20180111013000) and can work with the scheduling time parameter in DataWorks.</p> <div>  Note: The maid and enddatetime combinations are used together. </div>	Required : Select either this parameter or endTimestampMillis.	Blank
Enddatetime	<p>End time of data consumption. The parameter defines the right border of a time range (left closed and right open) in the format of yyyyMMddHHmmss (such as 20180111013010) and can work with the scheduling time parameter in DataWorks.</p> <div>  Note: The combination of enddatetime and maid is used together. </div>	No	N/A

Attribute	Description	Required	Default Value
BeginTimestampInMillis	<p>Description: It refers to the start time of data consumption in milliseconds and is the left boundary of the time range (left-closed and right-open).</p> <div>  Note: BeginTimestampInMillis and endTimestampInMillis combination for use. 1 represents the beginning of the log service cursor cursormode.Begin. The beginDateTime mode is recommended. </div>	Required	N/A : Select either this parameter or beginDateTime.
EndTimestampInMillis	<p>Description: It refers to the end time of data consumption in milliseconds and is the right boundary of the time range (left-closed and right-open).</p> <div>  Note: EndTimestampInMillis and beginTimestampInMillis combination for use. -1 represents the last location of the log service cursor, cursormode.End. The endDateTime mode is recommended. </div>	Required	N/A : Select either this parameter or endDateTime.

Development in wizard mode

1. Choose source

Configure the source and destination of the data for the synchronization task.



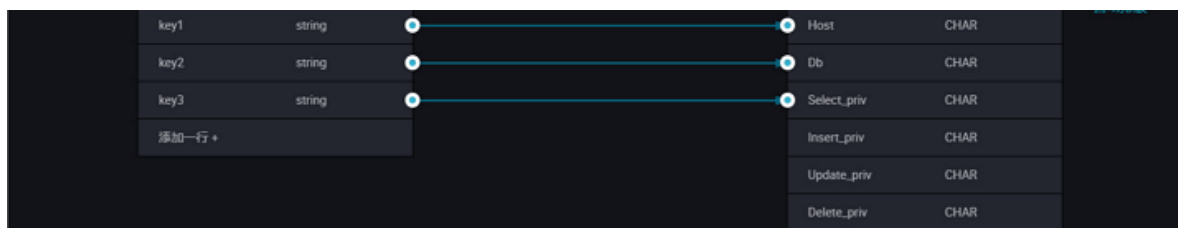
Configurations:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.

- Log start time: Start time of data consumption: It defines the left border of a time range (left closed and right open) in the format of yyyyMMddHHmmss (such as 20180111013000) and can work with the scheduling time parameter in DataWorks.
- Log end time: End time of data consumption. It defines the right border of a time range (left closed and right open) in the format of yyyyMMddHHmmss (such as 20180111013010) and can work with the scheduling time parameter in DataWorks.

2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.



- Peer mapping: Click **Enable Same-Line Mapping** to establish a corresponding mapping relationship in the peer, note that match the data type.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.
- Manually edit source table field: Manually edit the fields. Each line indicates a field. The first and end blank lines are ignored.

The function of adding a row is as follows:

- You can enter constants. The value must be enclosed by a pair of single quotes, such as 'abc' and '123'.
- Use this function with scheduling parameters, such as \${bizdate}.
- You can enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as Not identified.

3. Control the tunnel

03 Channel

You can control the data synchronization process through the transmission rate and the number of allowed dirty data records. See [data synchronization documents](#).

* DMU: 6

* Number of Concurrent Jobs: 8

* Transmission Rate: ☐ Unlimited ☒ Limited 10 MB/s

If there are more than: Maximum number of dirty data records. Dirty data is allowed by default. dirty data records, the task ends.

Task's Resource Group: Default resource group

Configurations:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

The following is a script configuration sample. For details about parameters, see the preceding Parameter Description.

```
{
  "type": "job",
  "version": "1.0"} //Indicates the version.
  "steps": [
    {
      "stepType": "loghub", // plug-in name
      "parameter": {
        "datasource": "", // Data Source
        "column": [// Field
          "col0",
          "col1",
          "col2",
          "col3",
          "col4",
          "C_topic", // log theme
          "C_hostname", // host name
          "C_path", // path
        ]
      }
    }
  ]
}
```

```

        "C_logtime" // log time
    ],
    "beginDateTime": "", // start time of data consumption
    "batchSize": "", // number of data lines to query from
the log service at once
    "endDateTime": "", //end time of data consumption
    "fieldDelimiter": ",", //Delimiter of each column
    "encoding": "UTF-8", // encoding format
    "logstore": "///: name of the target log Library
    },
    "name": "Reader ",
    "category": "Reader"
},
{ //The following is a writer template. You can find the
corresponding writer plug-in documentations.
    "stepType": "stream ",
    "parameter": {}
    "name": "Writer ",
    "category": "writer"
}
],
"setting": {
    "errorLimit": {
        "record": "0" //Number of error records
    },
    "speed": {
        "throttle": false, //false indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order ": {
    "hops ": [
        {
            "from": "Reader ",
            "to": "Writer"
        }
    ]
}
}
}

```

2.3.2.17 Configure OTSReader-Internal

In this article we will show you the data types and parameters supported by OTSReader-Internal and how to configure Reader in script mode

Table Store (originally known as OTS) is a NoSQL database service built upon Alibaba Cloud's Apsara distributed system, enabling you to store and access massive structured data in real time. Table Store organizes data into instances and tables. Using data partition and server load balancing technology, it provides seamless scaling.

OTSReader-Internal is used to export table data for Table Store Internal model while OTS Reader is used to export data for OTS Public model.

Table Store Internal model supports multi-version columns, so OTSReader-Internal also provides two data export modes:

- Multi-version mode: Because Table Store supports multiple versions, a multi-version mode is provided to export data of multiple versions.

Export solution: The Reader plug-in expands a cell of Table Store into a one-dimensional table consisting of four tuples: PrimaryKey (column 1-4), ColumnName, Timestamp, and Value (the principle is similar to the multi-version mode of HBase Reader). The four tuples are passed in to the Writer as four columns in Datax record.

- Normal Mode: consistent with the normal mode of the hbase reader, simply export the latest version of each column in each row of data, for more information, see [Configure HBase Reader](#) the normal mode content that is supported by the hbase reader in.

In short, OTS Reader connects to Table Store server and reads data through Table Store official Java SDK. OTS Reader optimizes the read process using features such as read timeout retry and exceptional read retry.

Currently, OTS Reader supports all Table Store types. The conversion of Table Store types in the OTSReader-Internal is as follows:

Data integration internal types	Table Store data model
Long	Integer
Double	Double
String	String
Boolean	Boolean
Bytes	Binary



Parameter description

Attribute	Description	Required	Default Value
mode	Description: The operation mode of the plug-in, supporting normal and multiVersion, which refers to normal mode and multi-version mode respectively.	Yes	N/A
endpoint	Description: The EndPoint of Table Store Server.	Yes	N/A
accessId	Access ID for Table Store	Yes	N/A
accessKey	Access key for Table Store	Yes	N/A

Attribute	Description	Required	Default Value
Instance name	<p>Description: The name of Table Store instance. The instance is an entity for using and managing Table Store service.</p> <p>After you enable the Table Store service, you can create an instance in the Console to create and manage tables . The instance is the basic unit for Table Store resource management. All access control and resource measurement done by the Table Store for applications are completed at the instance level.</p>	Yes	N/A
table	<p>Description: The name of the table to be extracted. Only one table can be filled in. Multi-table synchronization is not required for Table Store.</p>	Yes	N/A
Range	<p>Description: The export range: [begin,end).</p> <ul style="list-style-type: none">• Begin is less than end, which means reading data in positive sequence.• Begin > end, which means reading data in inverted sequence.• Begin and end cannot be equal.• The following types are supported: string, int, and binary . Binary data is passed in as Base64 strings in binary format. INF_MIN represents an infinitely small value and INF_MAX represents an infinitely large value.	No	Reads from the beginning of the table to the end of the table

Attribute	Description	Required	Default Value
range: {"begin "}	<p>The starting range that is exported, and the value can be an empty array, a PK prefix, or a complete PK. When reading data in positive order, the default fill PK suffix is inf_min, and the reverse order is inf_max, as shown in the example below.</p> <p>If your table has two PrimaryKeys in the type of string and int, the data of the table can be entered in the following three methods:</p> <ul style="list-style-type: none"> • [] Indicates that it is read from the beginning of the table. • [{"type": "string", "value ": "a"}] means from [{"type": "string ", "value": "a"}, {"type": "INF_MIN"}]. • [{"type": "string", "value": "a"}, {"type": "INF_MIN"}] <p>PrimaryKey column in binary type is special. JSON doesn't support directly passing in binary data, so the following rules are defined: To pass in binary data, you must use (Java) Base64.encodeBase64String method to convert binary data into a visualized string and then enter the string in value. The example is as follows (Java):</p> <ul style="list-style-type: none"> • <code>byte[] bytes = "hello".getBytes();</code> :Create binary data. Here the byte value of string hello is used. • <code>String inputValue = Base64.encodeBase64String(bytes);</code> : Call Base64 method to convert binary data into visualized strings. <p>Run the preceding code, and then the inputValue of "aGVsbG8=" can be obtained.</p> <p>Finally, write the value into the configuration: {"type": "binary", "value" : "aGVsbG8="}.</p>	No	Read data from the beginning of the table

Attribute	Description	Required	Default Value
range: {"end"}	<p>The end range that is exported, and the value can be an empty array, a PK prefix, or a complete PK. When reading data in positive order, the default population PK suffix is INF_MAX, and the reverse order is INF_MIN.</p> <p>If your table has two PKs in the type of string and int, the data of the table can be entered in the following three methods:</p> <ul style="list-style-type: none"> • [] Indicates that it is read from the beginning of the table. • [{"type": "string", "value": "a"}] means from [{"type": "string", "value": "a"}, {"type": "INF_MIN"}]. • [{"type": "string", "value": "a"}, {"type": "INF_MIN"}]. <p>PrimaryKey column in binary type is special. JSON doesn't support directly passing in binary data, so the following rules are defined: To pass in binary data, you must use (Java) Base64.encodeBase64String method to convert binary data into a visualized string and then enter the string in value. The example is as follows (Java):</p> <ul style="list-style-type: none"> • <code>byte[] bytes = "hello".getBytes();</code> Create binary data. Here the byte value of string hello is used. • <code>String inputValue = Base64.encodeBase64String(bytes);</code> Call Base64 method to convert binary data into visualized strings. <p>Run the preceding code, and then the inputValue of "aGVsbG8=" can be obtained.</p> <p>Finally, write the value into the configuration: {"type": "binary", "value": "aGVsbG8="}.</p>	No	Read to end of table

Attribute	Description	Required	Default Value
range: {"split"}	<p>Description: If too much data needs to be exported, you can enable concurrent export. Split can split the data in the current range into multiple concurrent tasks according to split points.</p> <div>  Note: <ul style="list-style-type: none"> The value entered in split must be in the first column of PrimaryKey (partition key) and the value type must be consistent with that of PartitionKey. The range of values must be between begin and end. The value within the split must increase or decrease progressively depending on the positive and inverted relationship between begin and end. </div>	No	Empty cut point
column	<p>Specifies the columns to export, supporting common and constant columns.</p> <p>Format (multi-version mode is supported)</p> <p>Regular column format: {"name": "{your column name}"}</p>		
timeRange (only multi-version mode is supported)	<p>Description: The time range of the request data. The read range is [begin,end).</p> <div>  Note: <p>Begin must be smaller than end.</p> </div>	No	Read all versions by default
timeRange: {"begin"} (only multi-version mode is supported)	<p>Description: The start time of the time range of request data . The value range is 0-LONG_MAX.</p>	No	10 by default
timeRange: {"end"} (only multi-version mode is supported)	<p>Description: the end time of the time range of request data. The value range is 0-LONG_MAX.</p>	No	- Default value: Long Max(9223372036854775806)
maxVersion (only multi-version mode is supported)	<p>Description: The specified version of the request. The value range is 1-INT32_MAX.</p>	No	Read all versions by default

Development in wizard mode

Currently, development in wizard mode is not supported.

Development in script mode

Multi-version Mode

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "reader": {
      "plugin": "otsreader-internalreader ",
      "parameter": {
        "mode": "multiversion ",
        "endpoint": "",
        "accessId": "",
        "accessKey": "",
        "instanceName": "",
        "table": "",
        "range": {
          "begin": [
            {
              "type": "string",
              "value": "a"
            },
            {
              "type": "INF_MIN"
            }
          ],
          "end": [
            {
              "type": "string",
              "value": "g"
            },
            {
              "type": "INF_MAX"
            }
          ],
          "split": [
            {
              "type": "string",
              "value": "b"
            },
            {
              "type": "string",
              "value": "c"
            }
          ]
        }
      },
      "column": [
        {
          "name": "attr1"
        }
      ],
      "timeRange": {
        "begin": 1400000000,
        "end": 1600000000
      },
      "maxVersion": 10
    }
  }
}
```

```

    },
    "writer": {
  }
}

```

Normal Mode

```

{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "reader": {
      "plugin": "otsreader-internalreader ",
      "parameter": {
        "mode": "normal",
        "endpoint": "",
        "accessId": "",
        "accessKey": "",
        "instanceName": "",
        "table": "",
        "range": {
          "begin": [
            {
              "type": "string",
              "value": "a"
            },
            {
              "type": "INF_MIN"
            }
          ],
          "end": [
            {
              "type": "string",
              "value": "g"
            },
            {
              "type": "INF_MAX"
            }
          ],
          "split": [
            {
              "type": "string",
              "value": "b"
            },
            {
              "type": "string",
              "value": "c"
            }
          ]
        }
      },
      "column": [
        {
          "name": "pk1"
        },
        {
          "name": "pk2"
        },
        {
          "name": "attr1"
        },
        {
          "type": "string",
          "value": ""
        }
      ]
    }
  }
}

```

```

    },
    {
      "type": "int",
      "value": ""
    },
    {
      "type": "double",
      "value": ""
    },
    {
      "type": "binary",
      "value": "aGVsbG8="
    }
  ]
},
"writer": {}
}

```

2.3.2.18 Configure OTSStream Reader

In this article we will show you the data types and parameters supported by OTSStream Reader and how to configure Reader in script mode.

OTSStream Reader plug-in is mainly used for exporting Table Store incremental data. Incremental data can be seen as operation logs which include data and operation information.

Different from full export plug-in, incremental export plug-in only has multi-version mode and it doesn't support specified columns. This is related to the principle of incremental export. See the following for more information about export format.

Before using the plug-in, ensure that the Stream feature is enabled. You can enable the feature when creating the table or enable it using SDK UpdateTable API.

How to enable Stream:

```

Syncclient client = new syncclient ("","","","");
Enable Stream when you create the table:
CreateTableRequest createTableRequest = new CreateTableRequest(
tableMeta);
createTableRequest.setStreamSpecification(new StreamSpecification(true
, 24)); // 24 means that the incremental data is retained for 24 hours
client.createTable(createTableRequest);
If Stream is not enabled when the table is created, you can enable it
with UpdateTable:
UpdateTableRequest updateTableRequest = new UpdateTableRequest("
tableName");
createTableRequest.setStreamSpecification(new StreamSpecification(true
, 24)); // 24 means that the incremental data is retained for 24 hours

```

```
client.updateTable(updateTableRequest);
```

Implementation

You can enable Stream and set expiration time by using SDK UpdateTable feature to enable incremental feature. When incremental feature is enabled, Table Store server saves your operation logs additionally. Each partition has a sequential operation log queue. Each operation log is moved by garbage collection after a period of time which is the expiration time you specified.

Table Store SDK provides several Stream-related APIs for reading these operation logs. The incremental plug-in also gets incremental data with Table Store SDK API, transforms incremental data into multiple 6-tuples (pk, colName, version, colValue, opType, sequenceInfo), and imports them into MaxCompute.

The format of the export data

In Table Store multi-version mode, the format of table data is in three-level mode, namely row > column > version. One row can have multiple columns. The column name is not fixed, and each column can have multiple versions. Each version has a specific timestamp (version number).

You can perform read/write operations with Table Store API. Table Store records incremental data by recording your recent write operations to the table (or data change operation). Therefore, incremental data can also be seen as a series of operation records.

Table Store has three types of data change operations: PutRow, UpdateRow, and DeleteRow:

- PutRow: write a row. If the row already exists, it is overwritten.
- UpdateRow: Updates a row without changing other data of the original row. Update may include adding or overwriting (if the corresponding version of the corresponding column already exists) some column values, deleting all the versions of a column, and deleting a version of a column.
- DeleteRow: Delete a row.

Table Store generates corresponding incremental data records according to each type of operation. Reader plug-in reads the records and exports the data in the format of Datax.

Because Table Store has the feature of dynamic column and multi-version, a row exported by Reader plug-in doesn't correspond to a row in Table Store but a version of a column in Table Store. A row in Table Store can be exported as multiple rows. Each row includes primary key value, the name of the column, the timestamp of the version under the column (version number), the value of the version, and operation type. If isExportSequenceInfo is set as true, time sequence information is also included.

When the data is transformed into Daxx format, we define four types of operations as follows:

- U (UPDATE): Writes a version of a column.
- DO (DELETE_ONE_VERSION): Deletes a version of a column.
- DA (DELETE_ALL_VERSION): Deletes all the versions of a column. Delete all the versions of the corresponding column according to primary key and column name.
- DR (DELETE_ROW): Deletes a row. Delete all the data of the row according to primary key.

Assuming that the table has two primary key columns. The names of the two primary key columns are pkName1 and pkName2. The example is as follows:

pkName1	pkName2	columnName	timestamp	columnValue	opType
pk1_V1	pk2_V1	col_a	1441803688001	col_val1	U
pk1_V1	pk2_V1	col_a	1441803688002	col_val2	U
pk1_V1	pk2_V1	col_b	1441803688003	col_val3	U
pk1_V2	pk2_V2	col_a	1441803688000	—	Do
pk1_V2	pk2_V2	col_b	—	—	Da
pk1_V3	pk2_V3	—	—	—	Dr
pk1_V3	pk2_V3	col_a	1441803688005	col_val1	U

Assuming that the export data has seven rows as in the shown preceding example, corresponding to three rows in Table Store table. The primary keys are (pk1_V1, pk2_V1), (pk1_V2, pk2_V2), and (pk1_V3, pk2_V3).

- For the row whose primary key is (pk1_V1, pk2_V1), three operations are required, respectively writing two versions of col_a column and one version of col_b column.
- For the row whose primary key is (pk1_V2, pk2_V2), two operations are required, respectively deleting one version of col_a column and all versions of col_b column.
- For the row whose primary key is (pk1_V3, pk2_V3), two operations are required, respectively deleting the whole row and writing one version of col_a column.

Currently OTSStream Reader supports all OTS types. The conversion list for Table Store types is as follows:

Type Classification	OTSstream Data Type
Integer	Integer
Float	Double
String type	String-
Boolean	Boolean
Binary	Binary

Parameter description

Attribute	Description	Required	Default Value
dataSource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	N/A
dataTable	The name of the table from which the incremental data is exported. The table needs to enable the Stream feature . You can enable the feature when creating the table or enable it using UpdateTable API.	Yes	N/A

Attribute	Description	Required	Default Value
statusTable	<p>The name of the table used by the Reader plug-in to record the status, these States can be used to reduce scanning of data in non-target ranges to speed up export. statusTable is the table for recording status in Reader. If the table doesn't exist, Reader creates the table automatically. When an offline export task is completed, you must not delete the table. The statuses recorded in the table can be used for the next export task.</p> <ul style="list-style-type: none"> You don't need to create the table and you only need to provide a name for the table. Reader plug-in tries to create the table under your instance. If the table doesn't exist, it is created. If the table already exists, it judges whether the Meta of the table is consistent with expectation. If it is not consistent, an exception is thrown. When an export is completed, you must not delete the table. The statuses of the table can be used for the next export task. The table enables TTL and data expire automatically, therefore we can consider that the data volume is small. For the Reader configurations of different dataTables under one instance, you can use the same statusTable. The status messages recorded are independent of each other. <p>In conclusion, you must configure a name such as TableStoreStreamReaderStatusTable. Note that the name must not be duplicate with that of business-related tables.</p>	Yes	N/A
startTimes stampMillis	<p>The left boundary of the time range of the incremental data (left closed right), in milliseconds.</p> <ul style="list-style-type: none"> Reader finds the point corresponding to startTimes stampMillis in statusTable, and reads and exports data from that point. If the corresponding point is not found in statusTable, the system reads from the first entry of the incremental data retained in the system and skips the data whose write time is earlier than startTimestampMillis. 	No	N/A

Attribute	Description	Required	Default Value
endTimeStampMillis	<p>The right border of the time range (left closed and right open) of incremental data, in milliseconds.</p> <ul style="list-style-type: none"> After exporting data from the point of startTimes tstampMillis, Reader finishes data export at the first entry of data whose timestamp is later than endTimeStampMillis. When all the incremental data are read, the read is completed, even if endTimeStampMillis is not reached. 	No	N/A
date	The data format is yyyyMMdd, for example 20151111, which means exporting the data of the date. If you do not specify a date, you must specify a maid and a maid, and vice versa. For example, Alibaba Cloud Data Process Center scheduling only supports day level. Therefore, the function of the configuration is similar to startTimes tstampMillis and endTimeStampMillis.	No	N/A
isExportSequenceInfo	Whether to export time sequence information. Time sequence information includes the write time of data. The default value is false which means not to export data.	No	N/A
maxRetries	The maximum number of retries of each request when incremental data is read from TableStore. The default value is 30. There are intervals between retries. The total time of 30 retries is approximately 5 minutes which generally doesn't require changes.	No	N/A
startTimeString	The left border of the time range (left closed and right open) of incremental data, in milliseconds (in the format of yyyyymmddhh24miss).	No	N/A
endTimeString	The right border of the time range (left closed and right open) of incremental data, in millisecond (in the format of yyyyymmddhh24miss).	No	N/A

Development in wizard mode

Currently, development in wizard mode is not supported.

Development in script mode

The following is a script configuration sample. For details about parameters, see the preceding Parameter Description.

```
{
```

```

"type": "job",
"version": "2.0"} //Indicates the version.
"steps":[
{
  "stepType": "otdsstream", // plug-in name
  "parameter": {
    "statusTable": "TableStoreStreamReaderStatusTable",//
The name of the table for recording the status.
    "maxRetries": 30, // when you read incremental data
from the tablestore, maximum number of retries per request, by default
    30
    "isExportSequenceInfo": false, // do you want to
export timing information?
    "datasource": "$ srcdatasource", // Data Source
    "startTimeString": "$ {starttime }", // The left
boundary of the time range of the incremental data (left closed right
on)
    "table": "ok",//Target table name
    "endTimeString": "$ {endtime}" // time range of
incremental data (left closed right) right Border
  },
  "name": "Reader ",
  "category": "Reader"
},
{ //The following is a writer template. You can find the
corresponding writer plug-in documentations.
  "stepType": "stream ",
  "parameter": {}
  "name": "Writer ",
  "category": "Writer"
}
],
"setting":{
  "errorLimit": {
    "record": "0"//Number of error records
  },
  "speed": {
    "throttle":false,//False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
    "concurrent": "1",//Number of concurrent tasks
    "dmu": 1 // DMU Value
  }
},
"order":{
  "hops":[
    {
      "from": "Reader ",
      "to": "Writer"
    }
  ]
}
}

```

2.3.2.19 Configure RDBMS Reader

In this article we will show you the data types and parameters supported by RDBMS Reader and how to configure Reader in script mode.

The RDBMS Reader plug-in allows your to read data from RDBMS (distributed RDS). At the underlying implementation level, RDBMS Reader connects to a remote RDBMS database through

JDBC and runs corresponding SQL statements to SELECT data from the RDBMS database . Currently it supports reading data from databases including DM, DB2, PPAS, and Sybase. Currently, the RDBMS plug-in is only adapted to the MySQL engine. RDBMS is a distributed MySQL database, and most of the communication protocols are applicable to MySQL use cases. Specifically, RDBMS Reader connects to a remote RDBMS database through the JDBC connector . The SELECT SQL query statements are generated and sent to the remote RDBMS database based on your configuration. Then, the SQL statements are run and the returned results are assembled into abstract datasets using the custom data types of data synchronization. Datasets are passed to the downstream writer for processing.

RDBMS Reader concatenates the table, column, and WHERE information you configured into SQL statements and sends them to the RDBMS database. For the querysql information that you configure, the RDBMS sends it directly to the RDBMS database.


RDBMS Reader supports most generic rational database types such as numbers and characters. Check whether your data type is supported and select a reader based on a specific database.

Parameter description

Attribute	Description	Required	Default Value
jdbcUrl	<p>Description: Information of the JDBC connection to the opposite-end database. The format of jdbcUrl is in accordance with the RDBMS official specification, and the URL attachment control information can be entered. Note that JDBC formats vary with databases and DataX selects an appropriate database driver for data reading based on a specific JDBC format.</p> <ul style="list-style-type: none"> DM: jdbc:dm://ip:port/database DB2 jdbc:db2://ip:port/database PPAS jdbc:edb://ip:port/database <p>RDBMS Writer adds new database support in the following ways.</p> <ul style="list-style-type: none"> Enter the corresponding directory of RDBMSWriter. \${DATA_X_HOME} is the main directory of DataX, that is, \${DATA_X_HOME}/plugin/writer/rdbmswriter. Under the RDBMS Reader directory, you can find the plugin .json configuration file. Use this file to register your specific database driver, which is placed in the drivers array. The RDBMS Reader plug-in dynamically selects the appropriate database driver to connect to the database when executing the job. <pre>{ "name": "RDBMS Reader ", "class": "com.alibaba.datax.plugin.reader.RDBMS Reader.RDBMS Reader", "description": "useScene: prod. mechanism : Jdbc connection using the database, execute select sql, retrieve data from the ResultSet . warn: The more you know about the database , the less problems you encounter.", "developer": "alibaba", "drivers": ["dm.jdbc.driver.DmDriver", "com.ibm.db2.jcc.DB2Driver", "com.sybase.jdbc3.jdbc.SybDriver", "com.edb.Driver"] }</pre> <p>The RDBMS Reader directory contains the libs sub-directory, under which you need to put your specific database driver.</p> <pre>\$tree . -- libs -- Dm7JdbcDriver16.jar -- commons-collections-3.0.jar -- commons-io-2.4.jar -- commons-lang3-3.3.2.jar -- commons-math3-3.1.1.jar -- datax-common-0.0.1-SNAPSHOT.jar</pre>	Yes	N/A
Issue 20190117			207

Attribute	Description	Required	Default Value
password	Description: Password corresponding to the specified username for the data source.	Yes	N/A
table.	The selected table that needs to be synchronized.	Yes	N/A
column	<p>The configured table requires a collection of column names that are synchronized, using an array of JSON to describe the field information, all column configurations, such as <code>[*]</code>, are used by default.</p> <ul style="list-style-type: none"> • Column pruning is supported, which means you can select some columns to export. • Change of column order is supported, which means you can export the columns in an order different from the schema order of the table. • Constant configuration is supported, and you need to follow the JSON format <code>["id", "1", "'bazhen.csy'", "null", "to_char(a + 1)", "2.3", "true"]</code>. <ul style="list-style-type: none"> - ID is normal column name - 1 For plastic digital Constants - 'Bazarn. CSY 'is a String constant - Null is a null pointer - To_char (a + 1) is a function expression - 2.3 is a floating point number - True is a Boolean Value • Column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	N/A

Attribute	Description	Required	Default Value
splitPk	<p>Description: If you specify the splitPk when using RDBMS Reader to extract data, it means that you want to use the fields represented by splitPk for data sharding. Then, the DataX starts concurrent tasks to synchronize data, which greatly improves the efficiency of data synchronization.</p> <ul style="list-style-type: none"> If you are using splitPk, we recommend that you use the primary keys of tables, because the primary keys are generally even and data hot spots are less prone to split data fragments. Currently, splitPk only supports data sharding for integer data types. Other types such as floating point, string, and date are not supported. If you specify an unsupported data type, DB2 Reader reports an error. If you do not fill in splitpk, you will be treated as if you do not split the single table, RDBMS reader uses a single channel to synchronize full data. 	No	Blank
where	<p>Description: Filtering condition. RDBMS Reader concatenates an SQL command based on specified column, table, and WHERE conditions and extracts data according to the SQL. For example, you can specify the where condition as limit 10 during a test. In actual business scenarios, the data on the current day is usually required to be synchronized. You can specify the WHERE condition as gmt_create > \$bizdate.</p> <ul style="list-style-type: none"> The where condition can be effectively used for incremental synchronization. If the where condition is not set or is left null, full table data synchronization is applied. 	No	N/A
querySql	<p>In some business scenarios, the where condition is insufficient for filtration. In such cases, the user can customize a filter SQL using this configuration item. When you configure this, the data synchronization system ignores the Table, column, and so on, filter the data directly using the contents of this configuration item.</p> <p>For example, you need to synchronize the data after a multi-table join, using <code>select a,b from table_a join table_b on table_a.id = table_b.id</code>. When querySql is configured, RDBMS Reader directly ignores the configuration of table, column, and where conditions.</p>	No	N/A

Attribute	Description	Required	Default Value
fetchSize	<p>Description: It defines the pieces of batch data that the plug-in and database server can fetch each time. The value determines the number of network interactions between the DataX system and the server, which can greatly improve data extraction performance.</p> <div>  Note: The fetchsize value (> 2048) may cause the data synchronization process oom. </div>	No	1,024

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

Configure a job to synchronously extract data from an RDBMS database:

```
{
  "job": {
    "setting": {
      "speed": {
        "byte": 1048576//Speed
      },
      "errorLimit": {
        "record": "0",
        "percentage": 0.02
      }
    },
    "content": [
      {
        "reader": {
          "name": "RDBMS Reader ",
          "parameter": {
            "username": "xxx",
            "password": "xxx",
            "column": [
              "id",
              "name"
            ],
            "splitPk": "pk",
            "connection": [
              {
                "table": [
                  "table"
                ],
                "jdbcUrl": [
                  "jdbc:dm://ip:port/database"
                ]
              }
            ]
          },
          "fetchSize": 1024,
          "where": "1 = 1"
        }
      ]
    }
  }
}
```

```

    },
    "writer": {
      "name": "streamwriter",
      "parameter": {
        "print": true
      }
    }
  }
]
}

```

Configure a Database Synchronization task for custom SQL to the job for MaxCompute (formerly ODPS).

```

{
  "job": {
    "setting": {
      "speed": {
        "byte": 1048576//Speed
      },
      "errorLimit": {
        "record": "0"
        "Percentage": 0.02
      }
    },
    "content": [
      {
        "reader": {
          "name": "RDBMS Reader",
          "parameter": {
            "username": "xxx",
            "password": "xxx",
            "column": [
              "id",
              "name"
            ],
            "splitPk": "pk",
            "connection": [
              {
                "querySql": [
                  "SELECT * from dual"
                ],
                "jdbcUrl": [
                  "jdbc:dm://ip:port/database"
                ]
              }
            ],
            "fetchSize": 1024,
            "where": "1 = 1"Where": "1 = 1"
          }
        },
        "writer": {
          "name": "streamwriter",
          "parameter": {
            "print": true
          }
        }
      }
    ]
  }
}

```

}

2.3.2.20 Configure Stream Reader

In this article we will show you the data types and parameters supported by Stream Reader and how to configure Reader in script mode.

The Stream Reader plug-in provides the ability to automatically generate data from the memory. It is mainly applicable to performance testing for data synchronization and basic functional testing.

The data types supported by stream reader are shown below.

Data type	Type description
string	Characters
long	Long Integer
date	Date type
bool	boolean
bytes	Bytes type

Parameter description

Attribute	Description	Required	Default Value
column	<p>Description: The column data and type of generated source data. Multiple columns can be configured. You can set to generate random strings and specify the corresponding range. The example is as follows:</p> <pre>"column": [{ "random": "8, 15" }, { "random": "10, 10" }]</pre> <p>Configurations:</p> <ul style="list-style-type: none"> "random": "8,15": means to generate a random string with a length of 8-15 bytes. "random": "10,10": means to generate a random string with a length of 10 bytes. 	Yes	N/A
sliceRecordCount	Represents the number of copies that the loop generates column.	Yes	N/A

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

Configure a synchronization job to read data from memory:

```
{
  "type": "job",
  "version": "1.0"} //Indicates the version.
  "steps":[
    {
      "stepType": "stream", //plug-in name
      "parameter": {
        "column": [// Field
          {
            "type": "string", //Value Type
            "value": "field" //Value
          },
          {
            "type": "long",
            "value": 100
          },
          {
            "dateFormat": "yyyy-MM-dd HH:mm:ss", //time
            "type": "date",
            "value": "2014-12-12 12:12:12"
          },
          {
            "type": "bool",
            "value": true
          },
          {
            "type": "bytes",
            "value": "byte string"
          }
        ],
        "sliceRecordCount": "100000" //Represents the number of
        column generated by the loop.
      },
      "name": "Reader ",
      "category": "reader"
    },
    {
      //The following is a writer template. You can find the
      corresponding writer plug-in documentations.
      "stepType": "stream ",
      "Parameter ": {}
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, //false stands for open current, the
      speed of the lower limit does not work, and true stands for current
      limit
    }
  }
}
```

```
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 //DMU Value
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

2.3.3 Configure Writer plug-in

2.3.3.1 Configure DataHub Writer

In this article we will show you the data types and parameters supported by DataHub Writer and how to configure Writer in script mode.

DataHub is a real-time data distribution and streaming data processing platform. It can publish, subscribe to, and distribute streaming data. It enables you to easily create analysis programs and applications based on streaming data.

Based on Alibaba Cloud's Apsara platform, DataHub delivers high availability, low latency, high scalability, and high throughput. Seamlessly connected to Alibaba Cloud's stream computing engine, StreamCompute, DataHub allows you to easily use SQL statements to analyze streaming data. DataHub provides the function to distribute streaming data to cloud products, currently including MaxComputer and OSS.

**Note:**


String The string can only be UTF-8 encoded and the maximum length of a single string column is 1 MB.

Parameter configuration

The source is connected to the sink through a channel. The channel type at the writer must be consistent with that at the reader. Two types of channels are provided generally: memory channel and file channel. The following example describes how to configure a file channel.

```
"agent.sinks.dataXSinkWrapper.channel": "file"
```

Parameter description

Attribute	Description	Required	Default Value
accessId	The accessId of the datahub.	Yes	N/A
accessKey	The accesskey of the DataHub.	Yes	N/A
endpoint	, For an access request to a datahub resource, select the correct domain name based on the service that the resource belongs.	Yes	N/A
maxRetryCount	Description: the maximum number of retries for task failure.	No	N/A
mode	Description: The write mode when the value type is string.	Yes	N/A
parseContent	Analysis Content	Yes	N/A
project	Description: Project is the basic unit of DataHub data, which contains multiple topics. <div>  Note: DataHub projects are independent from MaxCompute projects. Projects you created in MaxCompute cannot be used in DataHub. </div>	Yes	N/A
topic	Topic is the smallest unit of the datahub subscription and publication, you can use topic to represent one type or one type of streaming data.	Yes	N/A
maxCommitSize	Description: To improve writing efficiency, DataX-On-Flume collects the buffer data and submits it to the target end in batches when the collected data size reaches maxCommitSize (in MB). The maxCommitSize is 1048576 (1 MB) by default.	No	1 MB

Attribute	Description	Required	Default Value
batchSize	Description: To improve writing efficiency, DataX-On-Flume collects the buffer data and submits it to the target end in batches when the number of collected data entries reaches batchSize (in entry). The batchSize is 1024 (1024 entries) by default.	No	1,024
maxCommitInterval	Description: To improve writing efficiency, DataX-On-Flume collects the buffer data and submits it to the target end in batches when the number of collected data entries reaches the limit of maxCommitSize and batchSize. If the data collection source does not produce data for a long time, to ensure the timely delivery of data, the maxCommitInterval parameter (the maximum time allowed for the buffer data preservation, beyond which the data is compulsively delivered) (in milliseconds) is increased. The maxCommitInterval is 30000 (30 seconds) by default.	No	30
parseMode	Description: Log parsing mode, including non-parsing default mode and csv mode. In the non-parsing mode, one collected log line is written directly as a column of DataX Record. CSV mode supports configuring one column separator which separates one log line into multiple columns of DataX Record.	No	default

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

Configure a synchronization job to read data from memory:

```
{
  "type": "job",
  "version": "2.0", //version size
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader ",
      "category": "reader"
    },
    {
      "stepType": "datahub", //plug-in name
      "parameter": {
        "datasource": "", //Name of the data source
      }
    }
  ]
}
```



```

        "topic": "", //Topic is the smallest unit of DataHub
subscription and publishing. You can use Topic to represent a class or
a kind of streaming data.
        "maxRetryCount":500, //Number of retries
        "maxCommitSize": 1048576 //data to be saved to buffer
size reaches maxrefersize size (in MB) when, batch submitted to the
destination
    },
    "name": "Writer ",
    "category": "writer"
}
],
"setting": {
    "errorLimit": {
        "record": "" //Number of error records
    },
    "speed": {
        "concurrent": 20, // Number of concurrent jobs
        "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
        "dmu": 20 //DMU values
    }
},
"order": {
    "hops": [
        {
            "from": "Reader ",
            "to": "Writer"
        }
    ]
}
}

```

2.3.3.2 Configure DB2 writer

In this article we will show you the data types and parameters supported by DB2 Writer and how to configure Writer in script mode.

The DB2 Writer plug-in can write data into the target tables of DB2 databases. At the underlying implementation level, DB2 Writer connects to a remote DB2 database through JDBC, and runs the `insert into ...` SQL statement to write data into DB2. Data is submitted and written into the database in batches within DB2.

DB2 Writer is designed for ETL developers to import data from data warehouses to DB2. DB2 Writer can also be used as a data migration tool by DBA and other users.

DB2 Writer acquires the protocol data generated by Reader by means of the Data Integration framework. When the `insert into ...` SQL statement is run, if the primary key conflicts with the unique index, data cannot be written into the conflicting lines. To improve performance, we use `PreparedStatement + Batch` and configure `rewriteBatchedStatements=true` to buffer data to the thread context buffer. A write request is submitted only when the amount of data in the buffer reaches the threshold.

**Note:**

The task should at least have the insert into... permission. Whether other permissions are required depends on the statements specified in PreSQL and PostSQL when you configure the task.

DB2 Writer supports most data types in DB2. Check whether your data type is supported.

DB2 Writer converts DB2 data types as follows:

Category	DB2 Data Types
Integer	SMALLINT
Float	Decimal, real, and double
String	char, character, varchar, graphic, vargraphic, long varchar, clob, long vargraphic, or dbclob
Date and time type	decimal, real, and double
Boolean	—
Binary	blob

Parameter description

Attribute	Description	Required	Default Value
jdbcUrl	Description: Information of the JDBC connection to the DB2 database. In accordance with the DB2 official specification , jdbcUrl in the DB2 format is jdbc:db2://ip:port/database, and the URL attachment control information can be entered .	Yes	N/A
username	The User Name of the data source.	Yes	N/A
password	Description: Password corresponding to the specified user name for the data source.	Yes	N/A
table	Description: The table selected for synchronization.	Yes	N/A
column	Description: The fields of the target table into which data is required to be written. These fields are separated by commas. For example: "column": ["id", "name", "age"]. Use if it is required to write data into all columns in sequence. For example: "column": ["*"]. For example: "column": ["*"]	Yes	None

Attribute	Description	Required	Default Value
preSql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement, for example, clear old data.	No	N/A
postSql	Description: The SQL statement that is run after the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, add a timestamp.	No	N/A
batchSize	Description: The quantity of records submitted in batch at a time. This parameter can greatly reduce the interactions between Data Integration and DB2 over the network, and increase the overall throughput. However, the running process of Data Integration may become out of memory (OOM) if the value is too large.	No	1,024

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

Configure the data synchronization job to write data to DB2:

```
{
  "type": "job",
  "version": "2.0 ", // version number
  "steps": [
    { // The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {}
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "db2", // plug-in name
      "parameter": {
        "postSql": [], // SQL statement that was first executed
        before the data synchronization task was executed
        "password": "", // Password
        "jdbcUrl": "jdbc:db2://ip:port/database", // JDBC
        connection information for DB2 database
        "column": [
          "id",
        ],
        "batchSize": 1024, // number of records submitted in one
        batch size
        "table": "", // table name
        "username": "", // User Name
        "preSql": [] // SQL statement executed after the data
        synchronization task is executed
      }
    }
  ]
}
```

```

        },
        "name": "Writer",
        "category": "writer"
    }
],
"setting": {
    "errorLimit": {
        "record": "0" // Number of error records
    },
    "speed": {
        "throttle": false, // False indicates that the traffic is
        not throttled and the following throttling speed is invalid. True
        indicates that the traffic is throttled.
        "concurrent": "1", // Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader",
            "to": "Writer"
        }
    ]
}
}

```

2.3.3.3 Configure DRDS Writer

In this article we will show you the data types and parameters supported by DRDS Writer and how to configure Writer in both wizard mode and script mode.

The DRDS Writer plug-in provides the ability to write data to DRDS tables. At the underlying implementation level, DRDS Writer connects to the proxy of a remote DRDS database through JDBC, and writes data into DRDS by running the corresponding SQL statement `replace into` The SQL statement in writes the data to the DRDS.



Note:

Note that the SQL statement you run is `replace into`, and your table must have a primary key or a unique index to avoid data duplication. You must configure the data source before configuring the DRDS Writer plug-in. For more information, see [Configure DRDS data sources](#) Configure the DRDS data source.

DRDS Writer is designed for ETL developers to import data from data warehouses to DRDS. DRDS Writer can also be used as a data migration tool by DBA and other users.

DRDS Writer acquires the protocol data generated by Reader by means of the CDP framework, and writes data into DRDS by running the statement `replace into`.... If the primary key does not conflict with the unique index, the system performs the same action with `insert into`. When a conflict exists, all the fields in the original line are replaced with the fields in the new line.

DRDS Writer commits the accumulated data to DRDS's proxy, which then determines whether the data is written into one table or multiple tables, and how to route the data when it is written into multiple tables.

**Note:**

The entire task should at least have the permission replace into.... Whether other permissions are required depends on the statements you specified in PreSQL and PostSQL when you configure the task.

Similar to MySQL Writer, DRDS Writer currently supports most data types in MySQL. Check whether your data type is supported.

DRDS Writer converts DRDS data types as follows:

Type Classification	DRDS data type
Integer	int, tinyint, smallint, mediumint, int, bigint, and year
Floating point	float, double, and decimal
String	varchar, char, tinytext, text, mediumtext, and longtext
Date and time	date, datetime, timestamp, and time
Boolean	bit, and bool
Binary	tinyblob, mediumblob, blob, longblob, and varbinary

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
table	Description: The table selected for synchronization.	Yes	None
writeMode	Description: Select an import mode. The replace mode and insert ignore mode are supported. <ul style="list-style-type: none">replace: If the primary key does not conflict with the unique index, the system performs the same operation with insert into. When a conflict exists, all the fields in the original line are replaced with the fields in the new line.insert ignore: If the primary key conflicts with the unique index, Data Integration ignores and discards the updated data with no logs.	No	Insert ignore

Attribute	Description	Required	Default Value
column	Description: The fields of the target table into which data is required to be written. These fields are separated by commas . For example: "column": ["id", "name", "age"]. Use * if it is required to write data into all columns in sequence. For example : "column": ["*"].	Yes	None
preSql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, clear old data.	No	None
postSql	Description: The SQL statement that is run after the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, add a timestamp.	No	None
batchSize	Description: The quantity of records submitted in one operation . This parameter can greatly reduce the interactions between Data Integration and MySQL over the network, and increase the overall throughput. However, the running process of Data Integration may become out of memory (OOM) if the value is too large.	No	1,024

Development in wizard mode

1. Data source:

Configuration item descriptions:

The screenshot shows the '01 Data Source' configuration step. It is split into two columns: 'Source' and 'Destination'.

- Source Column:**
 - Data Source:** MySQL (dropdown)
 - Table:** bird_rds (dropdown)
 - Data Filtering:** id=1 (text input)
 - Sharding Key:** id (text input)
- Destination Column:**
 - Data Source:** DRDS (dropdown)
 - Table:** px_31 (dropdown)
 - Statements Run (Before Import):** select * from px_31 (text input)
 - Statements Run (After Import):** select * from px_31 (text input)

Buttons for 'Add Data Source +', 'Preview', and 'Next' are visible at the bottom.

Parameters:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.
- Prepared statement before import: preSql in the preceding parameter description, namely, the SQL statement that is run before the data synchronization task is run.
- Post-import completion statement: postSql in the preceding parameter description, which is the SQL statement that is run after the data synchronization task is run.

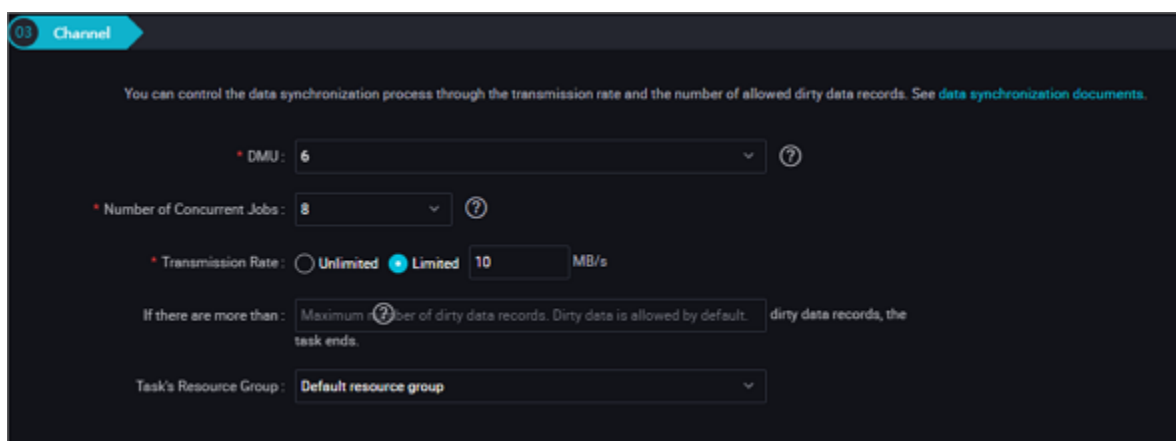
2. Field mapping: The column in the preceding parameter description.

The Source Table Field on the left maps with the Target Table Field on the right. Click **Add Line**, and then a field is added. Hover the cursor over a line, click **Delete**, and then the line is deleted.



- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.

3. Channel control



Parameters:

- **DMU:** A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- **Concurrent count:** Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- **Number of error records:** The maximum number of dirty data records.
- **Task Resource Group:** the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

Configure a job to write data into DRDS:

```
{
  "type": "job",
  "version": "2.0", //version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {}
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "drds", //plug-in name
      "parameter": {
        "postSql": [], // SQL statement executed after the
data synchronization task is executed
        "datasource": "", // Data Source
        "column": [ // column name
          "id",
        ],
        "writeMode": "insert ignore ",
        "batchSize": "1024", //number of records submitted in
one batch size
        "table": "test", //table name
        "postSql": [], //SQL statement executed after the data
synchronization task is executed
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
```



```

        "throttle":false,//False indicates that the traffic is
        not throttled and the following throttling speed is invalid. True
        indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrency
        "dmu": 1 // Number of DMU
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to": "Writer"
            }
        ]
    }
}

```

2.3.3.4 Configure FTP Writer

In this article we will show you the data types and parameters supported by FTP Writer and how to configure Writer in both wizard mode and script mode.

FTP Writer is used to write one or more files in CSV format to a remote FTP file. At the underlying implementation level, FTP Writer converts the data under the Data Integration transfer protocol to CSV files and writes these files to the remote FTP server using FTP-related network protocols. You must configure the data source before configuring the FTP Writer plug-in.



Note:

For more information, see [Configure the FTP data source](#) Configure the FTP data source.

What is written and saved to the FTP file is a two-dimensional table in a logic sense, for example, text information in CSV format.

FTP Writer provides the function to convert the Data Integration protocol to a FTP file. The FTP file is a non-structured data storage file. FTP Writer supports the following features:

- Only supports writing text files (BLOB, for example, video data, is not supported) and schema in the text file must be a two-dimensional table.
- Supports CSV and text files with custom delimiters.
- Does not support text compression during writing.
- Supports multi-thread writing, with different subfiles written using different threads.

The following two features are not supported for the time being.

- FTP Writer does not support the following features currently:
- FTP itself does not provide data types. FTP Writer writes data of String type to FTP file.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
timeout	Description: Time-out period (in milliseconds) of the connection to the FTP server.	No	60000 (1 minute)
path	Description: The path of the FTP file system. FTP Writer writes multiple files under the path directory.	Yes	None
FileName	Description: The name of the file written by FTP Writer. A random suffix is appended to the file name to form the actual name of the file written with each thread.	Yes	None
writeMode	Description: The mode in which FTP Writer clears existing data before writing data. Options include: <ul style="list-style-type: none"> truncate: Clear all the files prefixed by fileName in the directory before writing. append: The file is not processed before writing, and Data Integration FTP Writer writes data directly using fileName without conflict of file names. nonConflict: An error is reported if a file prefixed by fileName exists under the path directory. 	Yes	None
fieldDelimiter	Description: The delimiter used to separate the written fields.	Yes. A single character is used.	None
compress	Description: The gzip and bzip2 compression modes are supported.	No	Do Compress
encoding	Description: Encoding of the read files.	No	UTF-8
nullFormat	Description: Defining null (null pointer) with a standard string is not allowed in text files. Data Integration provides nullFormat to define which strings can be expressed as null. For example, if you configure <code>nullFormat="null"</code> , then if the source data is null, data integration is considered a null field.	No	None

Attribute	Description	Required	Default Value
dateFormat	Description: The format in which data of Date type is serialized into file, for example, "dateFormat": "yyyy-MM-dd".	No	None
fileFormat	The format written by the file includes both CSV and text, and the CSV is a strict CSV format, if you want to write the data that includes the column separator, It is escaped in the escape syntax of the CSV, the escape symbol is double quotes. The text format is a simple division of the data to be written using the column separator, do not escape for data to be written, including column separator.	No	text
header	Description: The header used when a txt file is written, for example, 'id', 'name', 'age'].	No	None
Markdonefilename	Description: The name of the file marked as "done". After a synchronization task is completed, a MarkDoneFile is generated, based on which whether the task is executed successfully is determined.	No	None

Development in wizard mode

1. Choose source

Configuration item descriptions:

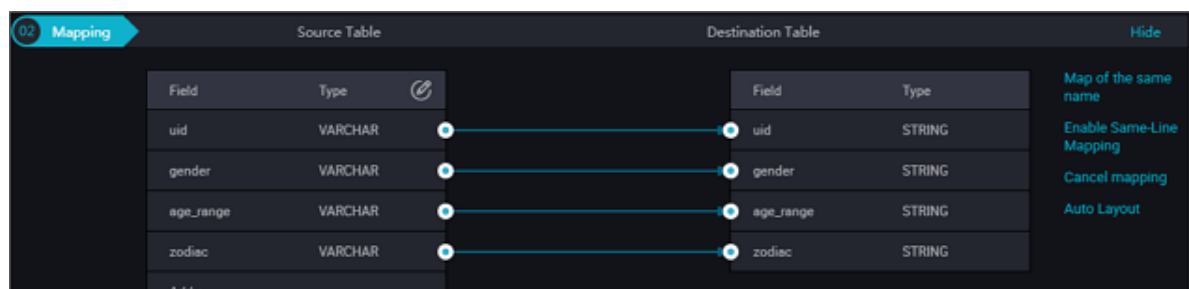
The screenshot shows the '01 Data Source' configuration wizard. It has two main sections: 'Source' and 'Destination'. The 'Source' section includes fields for 'Data Source' (FTP), 'File Path' (/home/workshop/user_log.txt), 'File Type' (text), 'Column' (I), 'Separator', 'Encoding' (UTF-8), 'Null String', 'Compression' (None), and 'Include Header' (No). The 'Destination' section includes fields for 'Data Source' (ODPS), 'Table' (ods_raw_log_d), 'Partition' (dt = \${bizdate}), 'Clearance Rule' (Clear Existing Data Before Writing), 'Compression' (Disable), and 'Consider Empty String as Null' (Yes). A 'Generate Destination Table' button is located between the two sections. A 'Preview' button is at the bottom of the 'Source' section.

Parameters:

- Data Source: datasource in the preceding parameter description. Select the FTP data source.
- File Path: path in the preceding parameter description.
- Column delimiter: fieldDelimiter in the preceding parameter description, which defaults to ",".
- Encoding format: encoding in the preceding parameter description, which defaults to utf-8.
- null Value: nullFormat in the preceding parameter description, which is used to define a string that represents the null value.
- Compression Format: compress in the preceding parameter description, which defaults to "no compression".
- Whether to Include the Table Header: **skipHeader** in the preceding parameter description, which defaults to "No".
- Prefix Conflict: The writemode in the above parameter description defines a string that represents a null value.

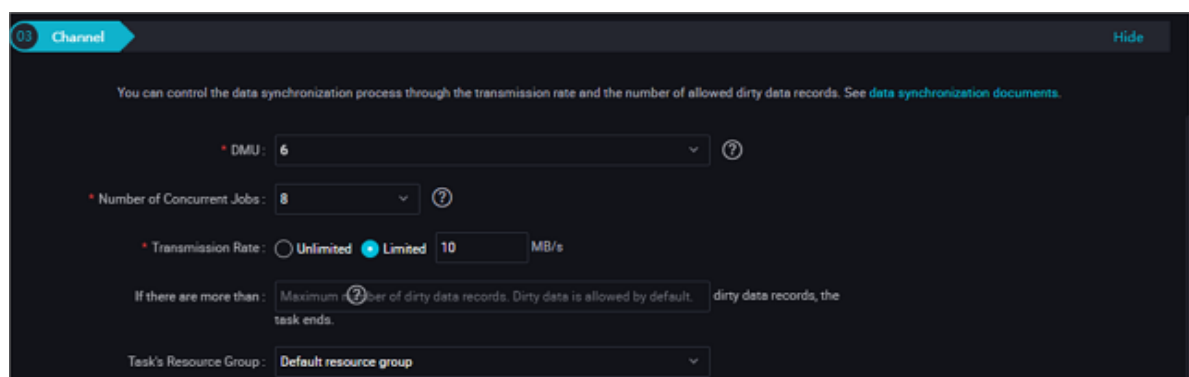
2. Field mapping: The column in the preceding parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add Line** to add a single field and click **Delete** to delete the current field.



In-row mapping: You can click In-row Mapping to create a mapping for the same row. Note that the data type must be consistent.

3. Channel control



Parameters:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

Configure synchronization jobs written to the FTP database.

```
{
  "type": "job",
  "version": "2.0", //version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "ftp", // plug-in name
      "parameter": {
        "path": "", //File path
        "fileName": "", //File name
        "nullFormat": "null", // Null Value
        "dateFormat": "yyyy-MM-dd HH:mm:ss", // time format
        "datasource": "", // Data Source
        "writeMode": "", //Write mode
        "fieldDelimiter": ",", //Delimiter of each column
        "encoding": "UTF-8", // encoding format
        "fileFormat": "", //File type
      },
      "name": "Writer ",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, //False indicates that the traffic is
        not throttled and the following throttling speed is invalid. True
        indicates that the traffic is throttled.
      "concurrent": "1", //Number of concurrent tasks
    }
  }
}
```

```

        "dmu": 1 // DMU Value
      }
    },
    "order": {
      "hops": [
        {
          "from": "Reader",
          "to": "Writer"
        }
      ]
    }
  }
}

```

2.3.3.5 Configure HBase Writer

In this article we will show you the data types and parameters supported by Stream Writer and how to configure Writer in script mode.

The HBase Writer plug-in provides the function to write data into HBase. At the underlying implementation level, HBase Writer connects to a remote HBase service through the HBase Java client, and writes data into HBase in put mode.

Supported features

- **HBase0.94.x and HBase1.1.x versions are supported**
 - If you use HBase 0.94.x, choose HBase094x as the Writer plug-in. For example:

```

"writer": {
  "plugin": "hbase094x"
}

```

- If you use HBase 1.1.x, choose HBase11x as the Writer plug-in. For example:

```

"writer": {
  "plugin": "hbase11x"
}

```

- **Multiple fields in the source end can be concatenated into a rowkey**

Currently, HBase Writer can concatenate multiple fields in the source end into the rowkey of an HBase table. For details, see the rowkeyColumn configuration.

- **Support to versions of data written into HBase**

Supported timestamps (versions) for data written into HBase include:

- Current time
- Specified source column
- Specified time

HBase Reader supports HBase data types and converts HBase data types as follows:

Data integration internal types	Hbase Data Type
Long	int,short,long
float,double	float,double
String	String
Boolean	Boolean

**Note:**

Apart from the field types listed here, other types are not supported.

Parameter description

Attribute	Description	Required	Default Value
haveKerberos	<p>Description: If haveKerberos is True, the HBase cluster needs to be authenticated using kerberos.</p> <div> Note: <ul style="list-style-type: none"> NOTE: If this value is configured as true, the following five parameters related to kerberos authentication must be configured: kerberosKeytabFilePath、kerberosPrincipal、hbaseMasterKerberosPrincipal、hbaseRegionserverKerberosPrincipal and hbaseRpcProtection. If the HBase cluster is not authenticated using kerberos, these six parameters are not required. </div>	No	false
hbaseConfig	<p>Description: Configuration required for connecting to the HBase cluster, in JSON format. The required item is hbase.zookeeper.quorum, which indicates the URL of HBase ZK. In addition, more HBase client configurations can be added. For example, you can configure the cache and batch of scan to optimize the interaction with servers.</p>	Yes	None
mode	<p>Description: The mode in which data is written into HBase. Currently, only the normal mode is supported. The dynamic column mode will be available later.</p>	Yes	None
table	<p>Description: Name of the HBase table to be written. The name is case sensitive.</p>	Yes	None
encoding	<p>Description: The encoding method is UTF-8 or GBK, which is used when data in string is converted to HBase byte[].</p>	No	UTF-8

Attribute	Description	Required	Default Value
column	Description: The HBase field to be written. <ul style="list-style-type: none">• index: Specify the index of the column that corresponds to the column of the Reader, starting from 0.• name: Specifies the column in the HBase table, which must be in column family:column name format.• type: Specifies the type of data to be written, which is used to convert HBase byte[].	Yes	N/A
maxVersion	Description: Specify the number of versions of data to be read by HBase Reader in multi-version mode, which can only be -1 (to read all versions) or a number larger than 1.	The configuration format is as follows:	None

Attribute	Description	Required	Default Value
range	<p>Specifies the rowkey range that the hbase reader reads.</p> <ul style="list-style-type: none"> startRowkey: Specify start rowkey. endRowkey: Specify end rowkey. isBinaryRowkey: Specifies the way in which the configured startrowkey and endrowkey are converted to byte, the default is false. If it is true, Bytes.toBytesBinary(rowkey) is called for conversion. If it is false, Bytes.toBytes(rowkey) is called. The configuration format is as follows: <pre>"range": { "startRowkey": "aaa", "endRowkey": "ccc", "isBinaryRowkey": false }</pre> <p>The format of the configuration file is as follows:</p> <pre>"column": [{ "index": 1, "name": "cf1:q1", "type": "string", }, { "index": 2, "name": "cf1:q2", "type": "string", }]</pre>	No	N/A
rowkeyColumn	<p>Rowkey column of the hbase to write.</p> <ul style="list-style-type: none"> index: Specify the index of the column that corresponds to the column of the Reader, starting from 0. If it is a constant, index is-1. type: Specifies the type of data to be written, which is used to convert HBase byte[]. value: A configuration constant, which is usually used as the concatenation operator of multiple fields. HBase Writer concatenates all columns of the rowkeyColumn into a rowkey in the configuration sequence to write data into HBase. The rowkey cannot contain constants only. <p>The format of the configuration file is as follows:</p> <pre>"rowkeyColumn": [{ "index": 0, "type": "string" }, { "index": -1, "type": "string", </pre>	Yes	None

Attribute	Description	Required	Default Value
walFlag	Description: When committing data to the RegionServer in the cluster (Put/Delete operation), the HBase client writes the WAL (Write Ahead Log, which is an HLog shared by all Regions on a RegionServer). The HBase client writes data into MemStore only after it successfully writes data into WAL. In this case, the client is notified that the data is successfully committed. In case of failure to write the WAL, HBase Client is notified that the commit is failed. Disable walFlag (false) to stop writing the WAL so as to improve the performance of data writing.	No	false
writeBufferSize	Description: Set the buffer size (in byte) of the HBase client. Use it with autoflush. autoflush: <ul style="list-style-type: none"> autoflush: If it is set to true, the HBase client performs an update operation for each put request. I If it is set to false, the HBase client initiates a write request to the HBase server only when the client write buffer is filled up with the put requests. 	No	8 MB

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

Configure a job to write data from a local machine into hbase1.1.x:

```
{
  "type": "job",
  "version": 2.0, // version number
  "steps": [
    {
      //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "hbase", // plug-in name
      "parameter": {
        "mode": "normal", // mode written to hbase
        "walFlag": "false", // close (false) give up writing Wal
        "hbaseVersion": "094x", // Hbase version
        "rowkeyColumn": [ // The rowkey column of the hbase to
          write.
        ]
      }
    }
  ]
}
```

```

        "index": 0, //serial number
        "type": "string" // data type
    },
    {
        "index": "-1",
        "type": "string",
        "value": "_"
    }
],

"nullMode": "skip", //How do I handle null values read by "Skip?
"column": [// The hbase field to write.
    {
        "name": "columnFamilyName1:columnName1", //
field name
        "index": "0", // Index Number
        "type": "string" // data type
    },
    {
        "name": "columnFamilyName2:columnName2",
        "index": "1",
        "type": "string"
    },
    {
        "name": "columnFamilyName3:columnName3",
        "index": "2",
        "type": "string",
    }
],
"writeMode": "api", // write mode is
"encoding": "utf-8", // encoding format
"table": "", // table name
"hbaseConfig": { // configuration information required
to connect to the hbase cluster, JSON format.
    "hbase.zookeeper.quorum": "hostname",
    "hbase.rootdir": "hdfs: //ip:port/database",
    "hbase.cluster.distributed": "true"
},
    "name": "Writer",
    "category": "writer"
},
],
"setting": {
    "errorLimit": {
        "record": "0" //Number of error records
    },
    "speed": {
        "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader",
            "to": "Writer"
        }
    ]
}
}

```

```
}
```

2.3.3.6 Configure HBase11xsq Writer

In this article we will show you the data types and parameters supported by HBase11xsq Writer and how to configure Writer in script mode.

HBase11xsq Writer provides the function to import data in batch to an SQL table (Phoenix) in HBase. The rowkey has been encoded by Phoenix. Therefore, you need to manually convert the data when you directly use HBase APIs for data writing, which is troublesome and error-prone.

This plug-in provides a method for you to import data to a single SQL table.

At the underlying implementation level, the JDBC drive of Phoenix executes the UPSERT statement to write data to HBase.

Supported functions

The writer supports importing data from an indexed table and simultaneously updating all indexed tables.

Limits

The limitations of the glaswriter plug-in are shown below.



- Only HBases of the 1.x version are supported.
- Only tables created by Phoenix are supported. Native HBase tables are not supported.
- Data with a timestamp cannot be imported.

Implementation principles

The JDBC drive of Phoenix executes the UPSERT statement to write data in batch to a table. Because an upper-layer API is used, the indexed tables can be updated simultaneously.

Parameter description

Attribute	Description	Required	Default Value
plugin:	Name of the plug-in, which must be hbase11xsq	Yes	None
table	Name of the table to be imported. The name is case sensitive and the name of Phoenix tables is generally in upper case.	Yes	None

Attribute	Description	Required	Default Value
column	<p>Name of the column. The name is case sensitive and the name of Phoenix tables is generally in upper case.</p> <div>  Note: <ul style="list-style-type: none"> The column sequence must exactly correspond to the sequence of columns output by the reader. The data type does not need to be entered, and the column metadata is automatically retrieved from Phoenix. </div>	Yes	None
hbaseConfig	<p>The address of the HBase cluster, in the format of ip1,ip2,ip3. The zk is required.</p> <div>  Note: <ul style="list-style-type: none"> NOTE: Separate multiple IP addresses by commas (,). znode is optional and the default value is /hbase. </div>	Yes	None
batchSize	Maximum number of rows written in bulk.	No	256
nullMode	<p>Specifies the processing mode when the column value read is null. There are currently two methods:</p> <ul style="list-style-type: none"> - skip: Skip this column. That is, this column is not inserted. If this column of the row already exists, the column is deleted. - empty: Insert a null value. 0 is inserted for value of the numeric type and a null string is inserted for a varchar value. 	No	skip

Development in script mode

The script configuration example is as follows.

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "setting": {
      "errorLimit": {
        "record": "0"
      },
      "speed": {
        "mbps": "1",
        "concurrent": "1"
      }
    }
  },
}
```

```

    "reader": {
      "plugin": "odps",
      "parameter": {
        "datasource": "",
        "table": "",
        "Column ":[
          "partition": ""
        ]
      },
    },
    "plugin": "hbasellxsql",
    "parameter": {
      "table": "Name of the target HBase table, which is case
sensitive",
      "hbaseConfig": {
        "hbase.zookeeper.quorum": "Address of the ZK server of the
target HBase cluster. Ask PE for the address",
        "zookeeper.znode.parent": "znode of the ZK server of the
target HBase cluster. Ask PE for the znode"
      },
      "column": [
        "columnName"
      ],
      "batchSize": 256,
      "nullMode": "skip"
    }
  }
}

```

Limits

The column sequence in the Writer must match that in the Reader. The column sequence in the Reader defines the organizational sequence of columns in each row. The column sequence in the Writer defines the column sequence of the received data that is expected by the Writer. Example:

If the column sequence in the Reader is c1, c2, c3, c4, and the column sequence in the Writer is x1, x2, x3, x4, the column c1 output by the Reader is the column x1 in the Writer. If the column sequence in the Writer is x1, x2, x4, x3, c1 is assigned to x4 and c4 is assigned to x3.

FAQ

Q: How many concurrent settings are appropriate? Can I increase the concurrency to accelerate the import speed?

A: The size of the default JVM stack for the data import process is 2 GB, and the concurrency (number of channels) is realized by multiple threads. Too many threads sometimes cannot accelerate the import speed but may result in performance deterioration due to frequent GC. A recommended concurrency (number of channels) is 5 to 10.

Q: What should the batchSize value be?

A: The default value is 256. You should set an appropriate batchSize according to the data volume in each row. Generally, the data volume at one operation is about 2 MB to 4 MB. You should divide this value by the data volume in the row and set the batchSize accordingly.

2.3.3.7 Configure HDFS Writer

In this article we will show you the data types and parameters supported by HDFS Writer and how to configure Writer in script mode.

HDFS Writer is used to write TextFile, ORCFile, and ParquetFile to the specified path to HDFS. The files can be associated with Hive tables. You must configure the data source before configuring the HDFS Writer plug-in. For more information, see [Configure the FTP data source](#) Configure the HDFS data source.

How to implement HDFS Writer

The implementation process for HDFS writer is shown below.

1. Create a temporary directory that does not exist in HDFS based on the path you specified.

Naming rule: path_random

2. Write the files that have been read to this temporary directory.
3. When all the files are written to the temporary directory, move these files to the directory you specified. The file names should be unique.
4. Delete the temporary directory. If you are unable to connect to HDFS for reasons such as network interruption during the process, delete the temporary directory and the files written to it manually.



Note:

For data synchronization, admin account and read/write permissions for the files are required.

```
[root@wh0 hadoop]# useradd -m -G supergroup -q hadoop -p admin admin
[root@wh0 hadoop]# su admin
[admin@wh0 hadoop]$ hadoop fs -ls /user/hive/warehouse/hive_p_partner_native
17/05/15 18:13:11 UTIL:util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rwxr-xr-x 3 hadoop supergroup 922 2017-05-15 16:17 /user/hive/warehouse/hive_p_partner_native/part-00000
[admin@wh0 hadoop]$ cd
[admin@wh0 ~]$ hadoop fs -get /user/hive/warehouse/hive_p_partner_native/part-00000
17/05/15 18:13:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[admin@wh0 ~]$ vim part-00000
[admin@wh0 ~]$ exit
exit
[root@wh0 hadoop]# pssh -h /home/hadoop/slave4pssh useradd -m -G supergroup -q hadoop -p admin admin
1) 18:14:22 SUCCESS wh1
2) 18:14:23 SUCCESS wh2
3) 18:14:23 SUCCESS wh3
```

as shown in the following figure:

- Create an admin user and home directory, specify a user group and additional group, and grant the permissions for the files.

```
useradd -m -G supergroup -g hadoop -p admin admin
```

- `-G supergroup`: Specifies the additional group to which the user belongs.
- `-g hadoop`: Specifies the user group to which the user belongs.
- `-p admin admin`: Add a password to the admin user.
- View the contents of the files in this directory.

```
hadoop fs -ls /user/hive/warehouse/hive_p_partner_native
```

When using hadoop commands, the format is `hadoop fs -command`, where `command` represents the command.

- Copies the file `part-00000` to the local file system.

```
hadoop fs -get /user/hive/warehouse/hive_p_partner_native/part-00000
```

- Edit the file you just copied.

```
vim part-00000
```

- Exits the current user.

```
exit
```

- Connect to the host from the list and create an admin account on each attached host.

```
pssh -h /home/hadoop/slave4pssh useradd -m -G supergroup -g hadoop -p admin admin
```

- `pssh -h /home/hadoop/slave4pssh`: connect to the host from the manifest file.
- `useradd -m -G supergroup -g hadoop -p admin admin`: Create admin account.

Functional restrictions

- It only supports TextFile, ORCFile, and ParquetFile formats, and what is stored in the file must be a two-dimensional table in a logic sense.
- HDFS is a file system and has no schema. Therefore, it does not support writing columns partially.
- Only the following Hive data types are supported:
 - Numeric: TINYINT, SMALLINT, INT, BIGINT, FLOAT, and DOUBLE
 - String: STRING, VARCHAR, and CHAR
 - Boolean: BOOLEAN

- Time type: date, timestamp.
- Hive data types such as decimal, binary, arrays, maps, ovens, and union are not currently supported.
- For Hive partition tables, the data can only be written to one partition at a time.
- For the TextFile format, ensure the delimiters in the files written to HDFS are identical to the ones used in the tables created in Hive, so that the data written to HDFS is associated with the Hive table fields.
- In the current plug-in, the Hive version is 1.1.1, and the Hadoop version is 2.7.1. Apache is compatible with JDK1.7. Data can be written normally in the testing environments of Hadoop 2.5.0, Hadoop 2.6.0, and Hive 1.2.0. For other versions, further test is needed.

Data type conversion

Currently, HDFS Writer supports most data types in Hive. Check whether the Hive type you are using is supported.


HDFS Writer converts the data types in Hive as follows:


Data Integration category	HDFS/Hive data type
long	TINYINT, SMALLINT, INT, BIGINT
double	FLOAT, DOUBLE
string	STRING, VARCHAR, CHAR
boolean	BOOLEAN
date	DATE, TIMESTAMP

Parameter description

Attribute	Description	Required	Default Value
defaultFS	Description: The namenode address in Hadoop HDFS, for example, <code>hdfs://127.0.0.1:9000</code> . The default resource group does not support the configuration of the advanced Hadoop parameter HA.	Yes	None

Attribute	Description	Required	Default Value
fileType	<p>Description: File type. Currently, only text, orc, and parquet are supported.</p> <ul style="list-style-type: none"> • text: Indicates TextFile. • orc: Indicates ORCFile. • parquet: Indicates ParquetFile. 	Yes	None
path	<p>Description: The path under which the files are written to Hadoop HDFS. HDFS Writer writes multiple files under the path based on the concurrent writing configurations.</p> <p>For association with a Hive table, enter the path under the Hive table stored in HDFS. For example, if the path to the data warehouse set in Hive is <code>/user/hive/warehouse/</code> and you have created the database test table named hello, the path of the Hive table is <code>/user/hive/warehouse/test.db/hello</code>.</p>	Yes	None
FileName	<p>Description: Name of the file written by HDFS Writer. A random suffix is appended to the file name to form the actual name of the file written using each thread.</p>	Yes	None

Attribute	Description	Required	Default Value
column	<p>Description: Fields of the written data. Some columns cannot be written.</p> <p>For association with a Hive table, you must specify all the field names and types in the table, with name and type specifying the field name and field type respectively.</p> <p>You can configure the column field as follows:</p> <pre>"column": [{ "name": "userName", "type": "string" }, { "name": "age", "type": "long" }]</pre>	Yes (if filetype is parquet, this entry is not required)	None
writeMode	<p>Description: The mode in which HDFS Writer clears the existing data before data writing:</p> <ul style="list-style-type: none"> append: The file is not processed before writing, and Data Integration HDFS Writer writes data directly using fileName without conflict of file names. nonConflict: An error is reported if a file prefixed by fileName exists under the path directory. <div>  Note: NOTE: Parquet files only support the nonConflict mode, and does not support the Append mode. </div>	Yes	None
fieldDelimiter	<p>Description: The field delimiter used for the fields written by HDFS Writer. Ensure the field delimiter is identical to the one used in the Hive table created. Otherwise, you are unable to locate the data in the Hive table.</p>	Yes. If the filetype is parquet, it is optional.	None
compress	<p>Description: Compression type of HDFS files. It is left empty by default, which means no compression is performed. Text files support gzip and bzip2 compression types. Orc files support SNAPPY compression. SnappyCodec is needed.</p>	No	None

Attribute	Description	Required	Default Value
encoding	The encoding configuration for the Write File.	No	No compression
parquetSchema	<p>Description: Required when the file is in parquet format. It is used to specify the structure of the target file, and takes effect only when the fileType is parquet. The format is as follows:</p> <pre>message MessageType { Required, data type, column name; ; }</pre> <p>Parameters:</p> <ul style="list-style-type: none"> • MessageType: Any supported value • Required: Required or Optional. Optional is recommended. • Data Type: Parquet files support the following data types: boolean, int32, int64, int96, float, double, binary (select binary if the data type is string), and fixed_len_byte_array. <p> Note: Each configuration row and column, including the last one, must end with a semicolon.</p> <p>Example:</p> <pre>message m { optional int64 id; optional int64 date_id; optional binary datetimestring; optional int32 dspId; optional int32 advertiserId; optional int32 status; optional int64 bidding_req_num; optional int64 imp; optional int64 click_num; }</pre>	否	N/A

Development in wizard mode

Development in wizard mode is not supported currently.

Development in script mode

The script configuration example is as follows, please refer to the above parameter descriptions for details.

```
{
  "type": "job",
  "version": 2.0, // version number
  "steps": [
    {
      //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "hdfs", // plug-in name
      "parameter": {
        "path": "", // path information stored to hadoop HDFS
        "fileName": "", //HDFS writer file name when writing
        "compress": "", // HDFS File compression type
        "datasource": "", //Name of the data source
        "column": [
          {
            "name": "col1", // field name
            "type": "string" // Field Type
          },
          {
            "name": "col2",
            "type": "int"
          },
          {
            "name": "col3",
            "type": "double"
          },
          {
            "name": "col4",
            "type": "boolean"
          },
          {
            "name": "col5",
            "type": "date",
          }
        ],
        "writeMode": "insert", //Write mode
        "fieldDelimiter": ",", //Delimiter of each column
        "Encoding": "UTF-8", // encoding format
        "fileType": "text" // text type
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "concurrent": "3", //Number of concurrent tasks
    }
  }
}
```

```
"throttle":false,//False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
    "dmu": 1 // DMU Value
  }
},
"order": {
  "hops": [
    {
      "from": "Reader",
      "to": "Writer"
    }
  ]
}
}
```

2.3.3.8 Configure MaxCompute Writer

In this article we will show you the data types and parameters supported by MaxCompute Writer and how to configure Writer in both wizard mode and script mode.

The MaxCompute Writer plug-in is designed for ETL developers to insert or update data in MaxCompute. With the ability to import business data to MaxCompute, this plug-in is suitable for TB and GB-level data transmission.



Note:

Before you start configuring the MaxCompute writer plug-in, first configure the data source. For more information, see [Configure MaxCompute data source](#).

For a detailed introduction to MaxCompute, see [Introduction to MaxCompute](#).

At the underlying implementation level, it writes data into MaxCompute by using Tunnel based on the source project/table/partition/table field and other information you configured. For common Tunnel commands, see [Tunnel Command Operations](#).

Supported data type

MaxCompute Writer supports the following data types in MaxCompute:

Data	MaxCompute data
Integer	Bigint
Float	Double and decimal
String type	String
Date and time	Datetime
Boolean	Boolean

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
table	Description: the name of the data table to write data into (case-insensitive). Writing data into multiple tables is not supported.	Yes	None
partition	<p>The partition information of the data table must be written. Specify the parameter until the last-level partition. For example, if you want to write data to a three-level partition table, configure through to a last-level partition, for example, pt=20150101, type=1, biz=2.</p> <ul style="list-style-type: none"> For non-partition tables, this value must not be entered, which means that the data is directly imported to the target table. MaxCompute Writer does not support writing data by routing. For partition tables, always ensure the data is written through to a last-level partition. 	Required if the table is a partition table. For non-partition tables, it is left empty.	None
column	<p>A list of fields that need to be imported, which can be configured as "column": ["*"] when all fields are imported. When you need to insert a partial MaxCompute column, fill in a partial column, for example, "column": ["id", "name"].</p> <ul style="list-style-type: none"> MaxCompute writer supports Column Filtering, column switching, for example, there are three fields in a table, A, B, and C. You can configure to "column": ["c", "b"] by synchronizing only the C and B fields. During the import process, field a is automatically empty, set to null. Column must contain the specified column set to be synchronized and it cannot be blank. 	Yes	None

Attribute	Description	Required	Default Value
truncate	<p>Description: "truncate": "true" is configured to ensure the idempotence of write operations. When a reattempt is made after a failed write attempt, MaxCompute Writer cleans up this data and imports the new data. This ensures the data is consistent after each rerunning.</p> <p>The option truncate is not an atomic operation. Because MaxCompute SQL is used for data cleansing, SQL cannot be atomic. Therefore, when multiple tasks clean up a Table /Partition at the same time, the concurrency and timing problem may occur. So proceed with caution.</p> <p>To avoid this problem, we recommend that you try not to operate on one partition with multiple job DDLs at the same time, or that you create partitions before starting multiple concurrent jobs.</p>	Yes	None

Development in wizard mode

1. Choose source

Configuration item descriptions:

Parameters:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.
- Partition information: If all columns are specified, you can configure them in column, for example, "column ": [""]. Partition supports configuration methods that configure multiple partitions and wildcard characters.

- `"partition": "pt=20140501/ds=*"` represents all partitions in DS.
- `"partition": "pt=top?"` In? indicates whether the character in front of it exists. This configuration specifies the two partitions with `pt=top` and `pt=to`.

You can enter the partition columns to be synchronized, such as partition columns with `pt`. Example: Assuming that the value of each MaxCompute partition is `pt=${bdp.system.bizdate}`, add the partition name `pt` to a field in the source table, ignore the unrecognized mark if any, and proceed with the next step. To synchronize all partitions, configure the partition value to `pt=${*}`. To synchronize a certain partition, select a time value for the partition.

- Cleaning rules:
 - Clean up Existing Data Before Import: All data in the table or partition is cleaned up before import, which is equivalent to insert overwrite.
 - 2) Keep existing data before writing: No data needs to be cleared before data importing. New data is always appended with each run, which is equivalent to "Insert into".
- Compression: Default selection is not compressed.
- Whether the empty string is null: the default is yes.

2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click Add row to **add a single** field and click **Delete** to delete the current field.



- In-row mapping: You can click **In-row Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.

3. Control the tunnel

Parameters:

- **DMU:** A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- **Concurrent job count:** Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- **The maximum number of errors indicates the maximum number of dirty data records.**
- **Task Resource Group:** the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

For example, under the script configuration example, please refer to the above parameter descriptions for details.

```
{
  "type": "job",
  "version": "2.0 ", // version number
  "steps": [
    { // You can locate the corresponding writer plug-in documentation among the following documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader ",
      "category": "reader"
    },
    {
      "stepType": "odps", // plug-in name
      "parameter": {
        "partition": "", // Shard information
        "truncate": true, // write Rule
        "compress": false, // do you want to compress?
      }
    }
  ]
}
```

```

        "datasource": "odps_first", //The data source name.
        "column": [ // column name
            "*"
        ],
        "emptyAsNull": false, if the empty string is null?
        "table": "" // table name
    },
    "name": "Writer",
    "category": "writer"
}
],
"Setting": {
    "errorLimit": {
        "record": "0" //Maximum number of error records
    },
    "speed": {
        "throttle": false, // do you want to limit the flow?
        "concurrent": "1", //Number of concurrent tasks
        "dmu": 1 // DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader",
            "to": "Writer"
        }
    ]
}
}

```

Additional instructions

Questions about Column Filtering

MaxCompute itself does not support column filtering, reordering, and null filling, but MaxCompute Writer does. For example, a list of fields that need to be imported, which can be configured as `"column": ["*"]` when all fields are imported `"": [""]`.

The MaxCompute table has three fields, A, B, and C, you can configure the column as `"column": ["c", "b"]` by synchronizing only the C and B fields `"": ["C", "B"]`, indicates that the first and second columns of reader will be imported into the C and B fields of MaxCompute, the newly inserted a field in the MaxCompute table is set to null.

Column configuration error handling

To ensure data is written in a reliable manner, data loss from redundant columns must be prevented to avoid data quality failure. When redundant columns are written, MaxCompute Writer produces an error. For example, if the MaxCompute table has fields a, b, and c, but MaxCompute Writer writes more than three fields, MaxCompute Writer produces an error.

Partition configuration

MaxCompute Writer only provides the write through to a last-level partition function, and does not support partition routing of writing based on a specific field. For a table that has three levels of partition, you must specify writing data to a level-3 partition. For example, to write data to the level -3 partition of a table, you can configure it to `pt=20150101`, `type=1`, `biz=2`, but not `pt=20150101`, `type=1` or `pt=20150101`.

Task rerunning and failover

In MaxCompute Writer, `"truncate": true` is configured to ensure the idempotence of write operations. When a reattempt is made after a failed write attempt, MaxCompute Writer cleans up this data and imports the new data. This ensures the data is consistent after each rerunning. If the task is interrupted by any exceptions during the running process, the atomicity of the data cannot be guaranteed, nor will the data be rolled back or rerun automatically. It is required that you use this idempotence to rerun the task to ensure data integrity.



Note:

Setting "truncate" to "true" cleans up all the data of the specified partition or table, so proceed with caution.

2.3.3.9 Configure Memcache (OCS) Writer

In this article we will show you the data types and parameters supported by Memcache (OCS) Writer and how to configure Writer in script mode.

ApsaraDB for Memcache (formerly known as OCS) is a seamlessly scalable distributed memory database service with high performance and reliability. Based on the Apsara distributed system and high performance storage, ApsaraDB for Memcache provides a complete set of solutions for master/slave hot standby, disaster recovery, business monitoring, data migration, and other scenarios.

ApsaraDB for Memcache supports the out-of-the-box deployment mode, and relieves the database load for dynamic web applications using the cache service, thus accelerating the overall response of the website.

Similar to the local Memcache databases, ApsaraDB for Memcache is compatible with the Memcached protocol. You can use it directly in your operating environment. The difference is that the hardware and data of ApsaraDB for Memcache are deployed in the cloud, providing complete infrastructure, network security, and system maintenance services. All these services are billed on a Pay-As-You-Go basis.


Memcache Writer writes data into Memcache channels based on the Memcached protocol.

Currently, Memcache Writer supports only one write mode. Data types written in different modes are converted differently:

- text: Memcache Writer serializes source data to the String type, and uses your fieldDelimiter as the delimiter.
- Binary: not supported.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
writeMode	Memcache Writer writes data in the following modes: <ul style="list-style-type: none">• set: Store the data.• add: Store the data only when this key does not exist (not supported currently).• replace: Store the data only when this key exists (not supported currently).• append: Store data after the existing key, and ignore exptime (not supported currently).• prepend: Store data before the existing key, and ignore exptime (not supported currently).	Yes	None

Attribute	Description	Required	Default Value
writeFormat	<p>Currently, Memcache Writer supports writing data in only one format:</p> <p>TEXT: Serialize the source data to the text format with the first field being the key written into Memcache, and all subsequent fields to the String type. Use fieldDelimiter you specified as the delimiter to concatenate the text data into a complete string and write it into Memcache.</p> <p>For example, source data is:</p> <pre> ID NAME COUNT --- :--- :--- 23 "AMC" 100 </pre> <p>If fieldDelimiter is specified as ^, the data format written into Memcache is:</p> <pre> KEY (OCS) VALUE(OCS) :--- :--- 23 CDP^100 </pre>	No	None
ExpireTime	<p>The cache invalidation time for the Memcache value. Currently, Memcache supports two types of the invalidation time.</p> <ul style="list-style-type: none"> • Unix time (number of seconds since January 1, 1970) indicates that data is invalid at a certain time point in the future. • The relative time (in seconds) starting from the current time point, which indicates the time length from the current time before data is invalid. <div>  Note: If the invalidation time is larger than 60*60*24*30 (30 days), the server identifies the invalidation time as the Unix time. </div>	No	0. 0 permanently valid
batchSize	<p>Description: The quantity of records submitted in one operation. Setting this parameter can greatly reduce the interactions between CDP and Memcache over the network, and increase the overall throughput. However, an excessively large value may cause the running process of CDP to become out of memory. (Writing in batches is not supported for the current Memcache version.)</p>	No	1,024

Development in wizard mode

Currently, development in wizard mode is not supported.

Development in script mode

Use the data generated from memory and imported into Memcache.

```
{
  "type": "job",
  "version": 2.0, // version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "Oss", // plug-in name
      "parameter": {
        "Writeformat": "text", // memcache writer writes data
        "expireTime": 1000, // memcache value cache failure
        "indexes": 0,
        "datasource": "", // Data Source
        "writeMode": "insert", // Write mode
        "batchSize": "1000", // number of records submitted in
        one batch size
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "Setting": {
    "errorLimit": {
      "record": "0" // Number of error records
    },
    "speed": {
      "throttle": false, // False indicates that the traffic is
      not throttled and the following throttling speed is invalid. True
      indicates that the traffic is throttled.
      "concurrent": "1", // Number of concurrent tasks
      "dmu": 1 // DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "from": "Reader",
        "to": "Writer"
      }
    ]
  }
}
```

```
}
```

2.3.3.10 Configure MongoDB Writer

In this article we will show you the data types and parameters supported by MongoDB Writer and how to configure Writer in script mode.

The MongoDB Writer plug-in uses MongoClient, the Java client of MongoDB, to write data into MongoDB. The latest version of Mongo has reduced the granularity of DB locks from the DB level to the document level, with the powerful indexing capabilities of MongoDB, data sources are basically able to meet the requirements of writing data to MongoDB. The requirements for data updates can also be implemented by configuring the business primary key.

**Note:**

- Before you start configuring the MongoDB writer plug-in, first configure the data source. For more information, see [Configure MongoDB data source](#).
- If you are using ApsaraDB for MongoDB, a root account is provided by default.
- To ensure security, Data Integration only supports using the relevant account of MongoDB for connection. Avoid using the root account as an access account when adding and using the MongoDB data source.

MongoDB Writer acquires the protocol data generated by Reader by means of the Data Integration framework, and converts the data types supported by Data Integration to the ones supported by MongoDB individually. The data integration itself does not support array types, but MongoDB supports array types, and the index of the array type is strong.

To use the MongoDB array type, you must convert the string to the array in MongoDB by using special configurations of parameters before writing data into MongoDB.

Type conversion list

MongoDB Writer supports most data types in MongoDB. Check whether your data type is supported before using it.

MongoDB Writer converts the MongoDB data types as follows:

Type Classification	MongoDB Data
Integer	INT and Long
Float	Double
String type	String and array
Date and time	Date

Type Classification	MongoDB Data
boolean	bool
Binary	Bytes

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
Collection name	The collection name of monogodb.	Yes	None
column	Description: An array of multiple column names of a document in MongoDB. <ul style="list-style-type: none"> name: The column name. type: The Column type. splitter: Special delimiter. It is used only when a string to be processed is split into character arrays by delimiters. Strings are split using the delimiter specified by this parameter and stored into MongoDB arrays. 	Yes	None
Writemode	Description: It specifies whether to overwrite data during transmission. <ul style="list-style-type: none"> isReplace: If this parameter is set to true, the data of the same replaceKey is overwritten. If it is set to false, the data is not overwritten. replaceKey: It specifies the business primary key for each record entry and is used to overwrite data (ReplaceKey must be unique and is generally the primary key in Mongo). 	No	None

Development in wizard mode

Development in wizard mode is unavailable currently.

Development in script mode

To configure data synchronization jobs written to MongoDB, please refer to the above parameter descriptions for details.

```
{
```

```

    "type": "job",
    "version": "2.0 ", // version number
    "steps": [
        { //The following is a reader template. You can find the
        corresponding reader plug-in documentations.
            "stepType": "stream",
            "parameter": {},
            "name": "Reader ",
            "category": "reader"
        },
        {
            "stepType": "hdfs", //plug-in name
            "parameter": {
                "path": "", //path
                "fileName": "ww", //File name
                "compress": "", // File compression type
                "datasource": "", // Data Source
                "column": [
                    {
                        "name": "col1", // field name
                        "type": "string" // Field Type
                    },
                    {
                        "name": "col2 ",
                        "type": "int"
                    },
                    {
                        "name": "col3",
                        "type": "double"
                    },
                    {
                        "name": "col4",
                        "type": "boolean"
                    },
                    {
                        "name": "col5",
                        "type": "date"
                    }
                ],
                "writeMode": "insert", //Write mode
                "fieldDelimiter": ",", //Delimiter of each column
                "encoding": "UTF-8", // encoding format
                "fileType": "// text type
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": "0" //Number of error records
        },
        "speed": {
            "throttle": false, //False indicates that the traffic is
            not throttled and the following throttling speed is invalid. True
            indicates that the traffic is throttled.
            "concurrent": 1, // Number of job concurrency
            "dmu": 1 // DMU Value
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",

```

```
        "to": "Writer"
      }
    ]
  }
}
```

2.3.3.11 Configure MySQL Writer

In this article we will show you the data types and parameters supported by MySQL Writer and how to configure Writer in both wizard mode and script mode.

The MySQL Writer plug-in can write data into a target table of a MySQL database. At the underlying implementation level, MySQL Reader connects to a remote MySQL database through JDBC, and runs the `insert into...` or `replace into...` SQL statement to write data into MySQL. Data is written into the database in batches within MySQL, and the database must use InnoDB engine.

**Note:**

You must configure the data source before configuring the MySQL Writer plug-in. For more information, see [Configure MySQL data source](#) Configure the MySQL Data Source.

MySQL Writer is designed for ETL developers to import data from data warehouses to MySQL. MySQL Writer can also be used as a data migration tool by DBA and other users. MySQL Writer acquires the protocol data generated by Reader based on writeMode by means of the Data Synchronization framework.

**Note:**

The entire task requires at least the `insert/replace into...` permission. Whether other permissions are required depends on the statements specified in PreSQL and PostSQL when you configure the task.

Type conversion list

Similar to MySQL Reader, MySQL Writer currently supports most data types in MySQL. Check whether your data type is supported.

MySQL Writer converts the MySQL data types as follows:

Category	MySQL data type
Integer	int, tinyint, smallint, mediumint, int, bigint, and year
Floating point	float, double, and decimal

Category	MySQL data type
String	varchar, char, tinytext, text, mediumtext, and longtext
Date and time	date, datetime, timestamp, and time
Boolean	bool
Binary	tinyblob, mediumblob, blob, longblob, and varbinary

Parameter description

Attribute	Description	Required	Default Value
datasource	Description: Data source name. The name entered here must be same to the name of the added data source. You can add a data source in script mode.	Yes	N/A
table	Description: The table selected for synchronization.	Yes	None
writeMode	<p>Description: Selects an import mode. The insert/replace mode is supported.</p> <ul style="list-style-type: none"> replace into... (If the primary key does not conflict with the unique index, the system performs the same action as insert into. When a conflict exists, all the fields in the original line are replaced with the fields in the new line.) insert into...(If the primary key conflicts with the unique index, data cannot be written into the conflicting lines and is regarded as dirty data.) INSERT INTO table (a,b,c) VALUES (1,2,3) ON DUPLICATE KEY UPDATE...;(If the primary key does not conflict with the unique index, the system performs the same action as insert into. When a conflict exists, the specified field in the original line is replaced with the field in the new line.) 	No	insert
column	Description: The fields of the target table into which data is required to be written. These fields are separated by commas. For example: "column": ["id", "name", "age"]. Use * if it is required to write data into all columns in sequence. For example, "column": ["*"].	Yes	None

Attribute	Description	Required	Default Value
preSql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, clear old data.	No	None
postSql	Description: The SQL statement that is run after the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, add a timestamp.	No	None
batchSize	Description: The quantity of records submitted in one operation. Setting this parameter can greatly reduce the interactions between Data Synchronization and MySQL, and increase the overall throughput. However, an excessively large value may cause the running process of Data Synchronization to become out of memory (OOM).	No	1,024

Development in wizard mode

1. Choose source

Configuration item descriptions:

The screenshot shows the '01 Data Source' configuration step in DataWorks. It is split into two main columns: 'Source' and 'Destination'.
 In the 'Source' column:
 - 'Data Source' is set to 'ODPS' with a dropdown arrow.
 - 'Table' is set to 'ods_first' with a dropdown arrow.
 - 'Data Filtering' has a field with a filter icon and a '?' help icon.
 - 'Sharding Key' has a field with a filter icon and a '?' help icon.
 In the 'Destination' column:
 - 'Data Source' is set to 'MySQL' with a dropdown arrow.
 - 'Table' is set to 'person' with a dropdown arrow.
 - 'Statements Run : Before Import' has a text area for SQL statements.
 - 'Statements Run : After Import' has a text area for SQL statements.
 At the bottom of the 'Source' column is a 'Preview' button.

Parameters:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.

- Prepared statement before import: preSql in the preceding parameter description, namely, the SQL statement that is run before the data synchronization task is run.
- Post-import completion statement: postSql in the preceding parameter description, which is the SQL statement that is run after the data synchronization task is run.
- Primary key conflict: writeMode in the preceding parameter description. You can select the expected import mode.

2. Field mapping: The column in the preceding parameter description.

The Source Table Field on the left maps with the Target Table Field on the right. Click **Add Line**, and then a field is added. Hover the cursor over a line, click **Delete**, and then the line is deleted.



- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.

3. Channel control

Parameters:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.

- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

The following is a script configuration sample. For relevant parameters, see Parameter Description.

```
{
  "type": "job",
  "version": 2.0, //version number
  "steps": [//below is the template for reader, you can find the
appropriate read plug-in documentation.
    {
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "mysql", // plug-in name
      "parameter": {
        "postSql": [], //Post-import preparation statement
        "datasource": "", // Data Source
        "column": [// column name
          "id",
          "value"
        ],
        "writeMode": "insert", //Write mode
        "batchSize": "1024", // number of records submitted in
one batch size
        "table": "", // table name
        "preSql": [], //Pre-import preparation statement
      },
      "name": "Reader",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": { // Number of error records
      "record": "0"
    },
    "speed": {
      "throttle": false, // do you want to limit the flow?
      "concurrent": "1", // Number of concurrency
      "dmu": 1 // DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "name": "Reader",
        "to": "Writer"
      }
    ]
  }
}
```

```
}  
}
```

2.3.3.12 Configuring Oracle Writer

In this article we will show you the data types and parameters supported by Oracle Writer and how to configure Writer in both wizard mode and script mode.

The Oracle Writer plug-in provides the ability to write data into the target tables of the primary Oracle database. At the underlying implementation level, Oracle Writer connects to a remote Oracle database through JDBC, and runs the `insert into...` SQL statement to write data into Oracle.

**Note:**

You must configure the data source before configuring the Oracle Writer plug-in. For more information, see [Configure Oracle data source](#) Configure the Oracle Data Source.

Oracle Writer is designed for ETL developers to import data from data warehouses to Oracle. Oracle Writer can also be used as a data migration tool by DBA and other users.

Oracle Writer uses the data synchronization framework to get the protocol data generated by Oracle Reader. Then it connects to a remote Oracle database through JDBC, and runs the `insert into ...` SQL statement to write data into Oracle.

Type conversion list

Similar to Oracle Reader, Oracle Writer currently supports most data types in Oracle. Check whether your data type is supported.

Oracle Writer converts the data types in Oracle as follows:

Type Classification	Oracle data type
Integer	NUMBER, RAWID, INTEGER, INT, and SMALLINT
Floating point	NUMERIC, DECIMAL, FLOAT, DOUBLE PRECISION, and REAL
String	LONG, CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, CLOB, NCLOB, CHARACTER, CHARACTER VARYING, CHAR VARYING, NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, and NCHAR VARYING

Type Classification	Oracle data type
TIMESTAMP and DATE	Timestamp and date
Boolean	BIT and BOOL
Binary	BLOB, BFILE, RAW, and LONG RAW

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
table	Description: Target table name. If the schema information of table is not consistent with the username in the preceding configuration, enter the table information in the schema. table format.	Yes	N/A
column	Description: The fields of the target table into which data is required to be written. These fields are separated by commas. For example: "column": ["id", "name", "age"]. Use * if it is required to write data into all columns in sequence. For example: "column": ["*"].	Yes	None
preSql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, clear old data.	No	None
postSql	Description: The SQL statement that is run after the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, add a timestamp.	No	None
batchSize	Description: The quantity of records submitted in one operation. Setting this parameter can greatly reduce the interactions between CDP and Oracle over the network, and increase the overall throughput. However, an excessively large value may cause the running process of CDP to become out of memory (OOM).	No	1,024

Development in wizard mode

1. Choose source

Configuration item descriptions:

Parameters:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: The table in the preceding parameter description. Select the table to be synchronized.
- Prepared statement before import: preSql in the preceding parameter description, namely, the SQL statement that is run before the data synchronization task is run.
- Post-import completion statement: postSql in the preceding parameter description, which is the SQL statement that is run after the data synchronization task is run.
- Primary key conflict: writeMode in the preceding parameter description. You can select the expected import mode.

2. Field mapping: The column in the preceding parameter description.

The Source Table Field on the left maps with the Target Table Field on the right. Click **Add Line**, and then a field is added. Hover the cursor over a line, click **Delete**, and then the line is deleted.

id	STRING	●	ID	NUMBER
			NAME	VARCHAR2

- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.

3. Channel control

Parameters:

- **DMU:** A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- **Concurrent job count:** Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- **The maximum number of errors** indicates the maximum number of dirty data records.
- **Task Resource Group:** the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

Configure a job to write data into Oracle:

```
{
  "type": "job",
  "version": "2.0", // version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "oracle", // plug-in name
      "parameter": {
        "postSql": [], // SQL statement executed after the
data synchronization task is executed
        "datasource": "",
        "session": [], // database connection session
      }
    }
  ]
}
```

```

        "column": [// Field
            "id",
            "name"
        ],
        "encoding": "UTF-8", // encoding format
        "batchSize": "1024", // number of records submitted in
one batch size
        "table": "", // table name
        "postSql": []// SQL statement executed after the data
synchronization task is executed
    },
    "name": "Writer",
    "category": "writer"
},
],
"setting": {
    "errorLimit": {
        "record": "0"//Number of error records
    },
    "speed": {
        "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
        "concurrent": "1", //Number of concurrency
        "dmu": 1 //DMU Value
    }
},
"order": {
    "hops": [
        {
            "from": "Reader",
            "to": "Writer"
        }
    ]
}
}

```

2.3.3.13 Configure OSS Writer

In this article we will show you the data types and parameters supported by OSS Writer and how to configure Writer in both wizard mode and script mode.

OSS Writer provides the ability to write one or more table files in CSV-like format into OSS.



Note:

You must configure the data source before configuring the OSS Writer plug-in. For more information, see [Configure OSS data source](#) Configure the OSS data source.

What is written and saved to the OSS file is a two-dimensional table in a logic sense, for example, text information in a CSV format.

- If you want to learn more about OSS products, see the [OSS Product Overview](#).

OSS Writer provides the ability to convert the data synchronization protocol to a text file in OSS, which itself is a non-structured data storage. Currently, OSS Writer supports the following features:

- Only supports writing text files and the schema in the text file must be a two-dimensional table.
- Supports CSV-like format files with custom delimiters.
- Supports multi-thread writing, with different subfiles written using different threads.
- Supports file rollover. A file exceeding a specific size value must be switched. A file that contains lines exceeding a specific number of lines must be switched.

OSS Writer does not support the following features temporarily:

- Concurrent writing is not supported for a single file.
- OSS itself does not provide data types. OSS Writer writes data of the String type to OSS.

OSS itself does not provide data types, which are defined by DataX OSS Writer.

Type Classification	OSS data type
Integer	Long
Float	Double
String	String
Boolean	bool
Date and time	Date

Parameter description

Attribute	Description	Required	Default Value
datasource	Description: Data source name. The name entered here must be same to the name of the added data source. You can add a data source in script mode.	Yes	None
Object	<p>Description: The file name written by OSS Writer. It enables the simulation of directories with file names in OSS. If the bucket in the OSS data source for data synchronization is the test folder of test118, as shown in the following figure:</p> <p>only test needs to be specified for object, without the bucket name. The file name synchronized to the OSS end is identical to the one entered in the source end.</p> <p>If "object": "test/DI" is specified, the object written in OSS begins with test/DI, in which test is a folder, DI is the prefix of the file name (suffix is a random string), and a forward slash (/) is used as the delimiter of the simulated OSS directory.</p>	Yes	None

Attribute	Description	Required	Default Value
writeMode	<p>Description: The mode in which OSS Writer clears the existing data before writing data.</p> <ul style="list-style-type: none"> truncate: All objects with matched object name prefixes are cleared before writing. For example, if "object" : "abc" is specified, all objects beginning with abc are cleared. append: No processing is done before writing. Data Integration OSS Writer writes data directly using the object name, and appends a random UUID suffix name to ensure no conflict of file names. For example, if the object name you specified is Data Integration, the name is actually entered as DI_XXXXXX_XXXX_XXXX. nonConflict: If an object with matched prefix exists in a specified path, an error is reported directly. For example, if "object" : "abc", is specified, when an object beginning with abc123 exists, an error is reported directly. 	Yes	None
fileFormat	<p>The format written by the file, including both CSV and text.</p> <p>Description: The format in which a file is written. Supported formats are CSV and text. If the data to be written contains column delimiters, the column delimiters are escaped to double quotation marks (") in CSV escape syntax. For text format, the data to be written is separated by column delimiters without being escaped.</p>	No	text
fieldDelimiter	Description: The delimiter used to separate the read fields.	No	,
encoding	Description: Encoding of the written files.	No	UTF-8
nullFormat	<p>Description: Defining null (null pointer) with a standard string is not allowed in text files. Data Synchronization system provides nullFormat to define which strings can be expressed as null. For example, when nullFormat = "null" is configured, if the source data is null, it is considered as a null field in Data Synchronization.</p>	No	None

Attribute	Description	Required	Default Value
header (advanced configuration, which is not supported in wizard mode)	Description: Header used when a file is written in OSS. For example, ['id', 'name', 'age'].	No	None
maxFileSize (advanced configuration, which is not supported in wizard mode)	Description: The maximum size of a single object file written in OSS, which defaults to 10,000 x 10 MB. It is similar to the log rotation based on the log size in log4j log printing. For multipart upload in OSS, the size of each part is 10 MB (which is the minimum file granularity for log rotation, and maxFileSize smaller than 10 MB is also taken as 10 MB), and the maximum number of parts supported for each OSS InitiateMultipartUploadRequest is 10,000. When rotation occurs, the naming rule for object is the original object prefix + a random UUID + a suffix such as _1, _2, _3.	No	1,000 MB

Development in wizard mode

1. Choose source

Configuration item descriptions:

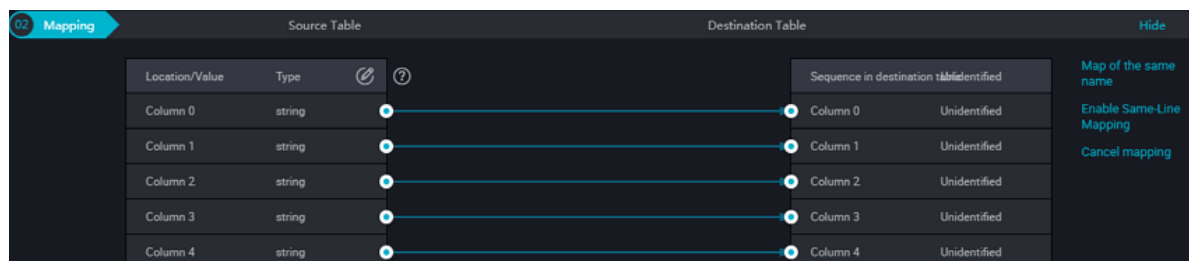
The screenshot shows the 'Data Source' configuration interface in the wizard mode. The interface is divided into two main sections: 'Source' and 'Destination'. The 'Source' section is active, showing configuration fields for OSS. The 'Destination' section is also visible, showing configuration fields for OSS. The 'Source' section has a 'Data Source' dropdown set to 'OSS' and a text input 'OSS_sourceec'. Below it are fields for 'Object Prefix', 'File Type' (set to 'csv'), 'Column Separator' (set to ','), 'Encoding' (set to 'UTF-8'), 'Null String', 'Compression' (set to 'None'), and 'Include Header' (set to 'No'). A 'Preview' button is at the bottom. The 'Destination' section has similar fields, with 'File Type' set to 'csv' and 'Solution to Duplicate' set to 'Replace the Original File'.

Parameters:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Object prefix: Object in the preceding parameter description. Enter a path to the OSS folder without the bucket name.
- Column delimiter: fieldDelimiter in the preceding parameter description, which defaults to ",".
- Encoding format: encoding in the preceding parameter description, which defaults to utf-8.
- null value: nullFormat in the preceding parameter description, to define a string that represents the null value.

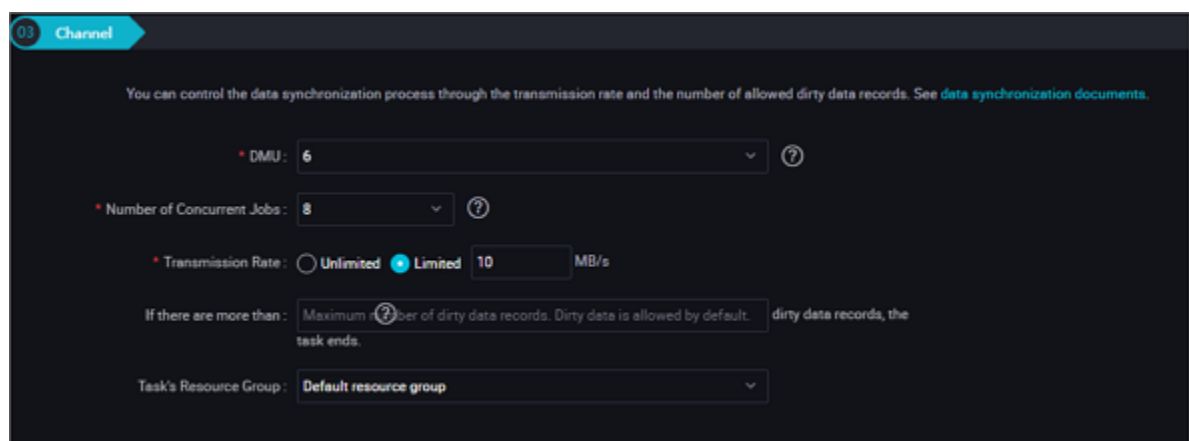
2. Field mapping: The column in the preceding parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click **Add row** to add a single field and click **Delete** to delete the current field.



In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.

3. Channel control



Parameters:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.

- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

The following is a script configuration sample. For details about parameters, see the preceding Parameter Description.

```
{
  "type": "job",
  "version": "2.0",
  "steps": [
    {
      //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "oss", // plug-in name
      "parameter": {
        "nullFormat": "", // The data synchronization system
        provides a nullformat to define which strings can be expressed as null
        .
        "dateFormat": "", // Date Format
        "datasource": "", // Data Source
        "writeMode": "", //Write mode
        "encoding": "UTF-8", // encoding format
        "fieldDelimiter": ",", //Separator
        "fileFormat": "", //File type
        "object": "" // object prefix
      },
      "name": "Writer ",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, //False indicates that the traffic is
      not throttled and the following throttling speed is invalid. True
      indicates that the traffic is throttled.
      "concurrent": "1", //Number of concurrent tasks
      "dmu": 1 //DMU Value
    }
  }
},
```

```
"order": {
  "hops": [
    {
      "from": "Reader ",
      "to": "Writer"
    }
  ]
}
```

2.3.3.14 Configure PostgreSQL Writer

In this article we will show you the data types and parameters supported by PostgreSQL Writer and how to configure Writer in both wizard mode and script mode.

The PostgreSQL Writer plug-in reads data from PostgreSQL. At the underlying implementation level, PostgreSQL Writer connects to a remote PostgreSQL database through Java DataBase Connectivity (JDBC) and runs corresponding SQL statements to select data from the PostgreSQL database. On the public cloud, Relational Database Service (RDS) provides a PostgreSQL storage engine.



Note:

Configure the data source before configuring a PostgreSQL Writer plug-in. For details, see [Configure PostgreSQL data source](#) Configure the PostgreSQL Data Source.

In short, PostgreSQL Writer connects to a remote PostgreSQL database through a JDBC connector, generates SELECT SQL query statements based on configuration, sends the statements to the remote PostgreSQL database, assembles returned results of SQL statement execution into abstract datasets through the custom data types of CDP, and passes the datasets to the downstream writer.

- PostgreSQL Writer concatenates the configured table, column, and WHERE information into SQL statements and sends them to the PostgreSQL database.
- PostgreSQL directly sends the configured querySql information to the PostgreSQL database.

Type conversion list

PostgreSQL Writer supports most PostgreSQL data types. Check whether the data type is supported.

PostgreSQL Writer converts PostgreSQL data types as follows:

Data integration internal types	PostgreSQL data type
Long	bigint, bigserial, integer, smallint, and serial
Double	double precision, money, numeric, and real

Data integration internal types	PostgreSQL data type
String	varchar, char, text, bit, and inet
Date	Date, time, and timestamp
Boolean	bool
Bytes	Bytea

**Note:**

- Except the preceding field types, other types are not supported.
- For "money", "inet", and "bit", you need to use syntaxes such as "a_inet::varchar" to convert data types.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
table	The name of the selected table that needs to be synchronized.	Yes	None
writeMode	Description: Specifies the import mode. Data can be inserted. insert: If the primary key conflicts with the unique index, Data Integration determines the data as dirty data but retains the original data.	No	insert
column	Description: The fields of the target table into which data is required to be written. These fields are separated by commas. For example, "column": ["id", "name", "age"]. If you want to write all columns in turn, use the * representation, for example, "column": ["*"].	Yes	None
preSQL	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, clear old data.	No	None
postSQL	SQL statement executed after the data synchronization task is executed. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, add a timestamp.	No	None

Attribute	Description	Required	Default Value
batchSize	Description: The quantity of records submitted in one operation. This parameter can greatly reduce the interactions between Data Integration and PostgreSQL over the network, and increase the overall throughput. However, an excessively large value may cause the running process of Data Integration to become out of memory (OOM).	No	1,024

Development in wizard mode

1. Choose source

Configuration item descriptions:

Parameters:

- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.
- Prepared statement before import: preSQL in the preceding parameter description, namely, the SQL statement that is run before the data synchronization task is run.
- Post-import completion statement: postSQL in the preceding parameter description, which is the SQL statement that is run after the data synchronization task is run.

2. Field mapping: The column in the preceding parameter description.

The Source Table Field on the left maps with the Target Table Field on the right. Click **Add Line**, and then a field is added. Hover the cursor over a line, click **Delete**, and then the line is deleted.

id	bigint		id	int4
name	char		name	varchar
age	int		year	int2
salary	float		birthdate	date
sex	bit		ismarried	bool
birth	datetime		interest	varchar
添加一行 +			salary	numeric

- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
- Automatic formatting: The fields are automatically sorted based on corresponding rules.

3. Control the tunnel

03 Channel

You can control the data synchronization process through the transmission rate and the number of allowed dirty data records. See [data synchronization documents](#).

* DMU: ?

* Number of Concurrent Jobs: ?

* Transmission Rate: ☐ Unlimited ☒ Limited MB/s

If there are more than: dirty data records, the task ends.

Task's Resource Group:

Parameters:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.
- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

The following is a script configuration sample. For details about parameters, see [Parameter Description](#).

```
{
  "type": "job",
  "version": 2.0, // version number
  "steps": [// below is the template for reader, you can find the
appropriate read plug-in documentation.
    {
      "stepType": "stream",
      "Parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "postgresql", // plug-in name
      "parameter": {
        "postSQL": [], // SQL statement that was first
executed after the data synchronization task was executed
        "datasource": "// Data Source
          "col1",
          "col2",
        ],
        "table": "", // table name
        "postSQL": [], // SQL statement that was first
executed after the data synchronization task was executed
      },
      "name": "Reader",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" // Number of error records
    },
    "speed": {
      "throttle": false, // False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
      "concurrent": "1", // Number of concurrent tasks
      "dmu": 1 // DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "name": "Reader",
        "To": "Writer"
      }
    ]
  }
}
```

2.3.3.15 Configure Redis Writer

Redis Writer is a Redis writing plug-in based on the Data Integration framework. It can import data from a data warehouse or other data sources to a Redis instance. Redis Writer interacts with

Redis Server by Jedis. As a preferred Java client development kit provided by Redis, Jedis has almost all Redis features.


Redis (Remote Dictionary Server) is a high-performance persistent log-based key-value storage system supporting network and based on memory, which can be used as a database, high-speed cache, and message queue (MQ) proxy. Redis supports diverse types of storage values, including string, list, set, zset (sorted set), and hash. For details about Redis, see redis.io.




Note:

- Configure the data source before configuring a Redis Writer plug-in. For details, see [Configure Redis data source](#) Configure the Redis Data Source.
- When data is written to a Redis instance through Redis Writer, if values are lists, the result of the re-run synchronization task is not idempotent. So if the value type is list, you must manually clear the corresponding data on Redis when re-running the synchronization task.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
Keyindexes	<p>Description: keyIndexes indicates which columns of the source table are used as key (starts with 0 for the first column). If the key is the combination of the first and second columns, the value of keyIndexes is [0,1].</p> <div>  <p>Note: When keyindexes is configured, The redis writer takes the remaining columns as value. If you want to synchronize only a few columns of the source table as key, a few columns as value, you do not need to synchronize all the fields, so you can specify column on the Reader plug-in side for column filtering.</p> </div>	Yes	None
Keyfieldde limiter	Writes a key separator to redis. Take key=key1\u0001id as an example. If multiple keys need to be concatenated, the value is required; if only one key exists, this configuration item can be ignored.	No	\u0001

Attribute	Description	Required	Default Value
batchSize	Description: The quantity of records submitted in one operation. This parameter can greatly reduce the interactions between Data Integration and PostgreSQL over the network, and increase the overall throughput. However, an excessively large value may cause the running process of Data Integration to become out of memory (OOM).	No	1,000
expireTime	<p>Description: The Redis value cache expiration time (which is valid permanently if this configuration item is left empty).</p> <ul style="list-style-type: none"> seconds: The relative time (in seconds) starting from the current time point, which indicates the time length from the current time before data is invalid. unixtime: The Unix time (number of seconds from January 1, 1970), specifying a future time point at which data becomes invalid. <div>  Note: If the invalidation time is larger than 60*60*24*30 (30 days), the server identifies the invalidation time as the Unix time. </div>	No	0 (0 indicates permanent validity)
timeout	The time-out, in milliseconds, that was written to redis.	No	30000 (that is, cover 30 seconds of network break time)
dateFormat	The time when data is written into Redis in date format: "yyyy-MM-dd HH:mm:ss".	No	None

Attribute	Description	Parameter type	Description				Required	Default Value
			type	mode	Valuefield	delimiter		
writeMode	Description: Redis supports diverse types of values, including string, list, set, zset, and hash. Redis Writer can write these types of data into a Redis instance. Configuration of writeMode varies slightly based on the value type. writeMode is configured as follows. Only one of the following types can be selected when you configure Redis Writer:	String (string) <pre>"writeMode": { "type": "string", "name": "set", " valueField Delimiter": "\u0001" }</pre>	Description	Description: The write mode when the value type is string.	Description: The delimiter between values when values are strings if there are more than two columns of source data in each row (this configuration item can be ignored if only two columns of source data exist: "key" and "value"), for example, value1\u0001value2\u0001value3.	No	Default value: string	
			Required	Yes	Required: Yes. Available value: set (store the data, and overwrite this data if it already exists)			No
			Default Value	-	-			\u0001
		List of strings <pre>"writeMode": { "type": "list", "mode": "lpush rpush", " valueField Delimiter": "\u0001" }</pre>	Description	Description: The write mode when the value type is list.	Description: The delimiter between values when the value type is string. For example, value1\u0001value2\u0001value3.			
Issue: 20190117		valueField Delimiter": "\u0001"		Required: Yes	Required: Yes	No		281

- Description: Redis supports diverse types of values, including string, list, set, zset, and hash. Redis Writer can also write these types of data into Redis. However, the configuration of writeMode varies slightly with the value type. writeMode is configured as follows. Only one of the following five types can be configured when you configure Redis Writer:

- String (string)

```
"Writemode ":{
  "type": "string",
  "mode": "set",
  "valueFieldDelimiter": "\u0001"
}
```

Parameters:

- type

- Description: value type: string
- Required: Yes

- mode

- Description: The write mode when the value type is string.
- Required: Yes. Available value: set (store the data, and overwrite this data if it already exists)

- valueFieldDelimiter

- Description: The delimiter between values when values are strings if there are more than two columns of source data in each row (this configuration item can be ignored if only two columns of source data exist: "key" and "value"), for example, value1\u0001value2\u0001value3.
- Required: No
- Default value: \u0001

- List of strings

```
"writeMode":{
  "type": "list",
  "mode": "lpush|rpush",
  "Maid": \ u0001"
}
```

Parameters:

- type

- Description: value type: list

- Required: Yes
- mode
 - Description: The write mode when the value type is list.
 - Required: Yes. Available value: lpush (store the data on the far left of list) | rpush (store the data on the far right of list)
- valueFieldDelimiter
 - Description: The delimiter between values when the value type is string. For example, value1\u0001value2\u0001value3.
 - Required: No
 - Default value: \u0001
- String collection (set)

```
"writeMode": {  
  "type": "set",  
  "name": "set",  
  "valueFieldDelimiter": "\u0001"  
}
```

Parameters:

- type
 - Description: value type: set
 - Required: Yes
- mode
 - Description: The write mode when the value type is set.
 - Required: Yes. Available value: sadd (store the data into set, and overwrite this data if it already exists)
- valueFieldDelimiter
 - Description: The delimiter between values when the value type is string. For example, value1\u0001value2\u0001value3.
 - Required: No
 - Default value: \u0001
- String collection (SET)



Note:

NOTE: If values are Zset data, each row of records of the data source must follow this rule: apart from the key, each row only contains one pair of score and value, and score must be located before value, so that Redis Writer can parse the score column and the value column.

```
"writeMode":{
  "type": "zset",
  "mode": "zadd"
}
```

Configuration item descriptions:

- type
 - Description: value type: zset
 - Required: Yes;
- mode
 - Description: The write mode when values are Zset data.
 - Mandatory: Yes; available value: zadd (stored in the Zset sorted set, and overwritten if it already exists)
- Hash (hash)



Note:

NOTE: If values are hashed, each row of records of the data source must follow this rule: apart from the key, each row only contains one pair of attribute and value, and attribute must be located before value, so that Redis Writer can parse the attribute column and the value column.

```
"writeMode":{
  "type": "hash",
  "mode": "hset"
}
```

Parameters:

- type
 - Description: value type: hash
 - Required: Yes
- mode
 - Description: The write mode when values are hashed

- **Mandatory:** Yes. **Optional value:** hmset (stored in the hash sorted set, and overwritten if it already exists)

You need to specify one of the data types. If you leave it empty, the data type is "string" by default.

- **Required:** No
- **Default value:** string

Development in wizard mode

Currently, development in wizard mode is not supported.

Development in script mode

Configure Data Synchronization jobs written to redis, see parameter descriptions for details.

```
{
  "type": "job",
  "version": "2.0", // version number
  "steps": [
    {
      //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "redis", // plug-in name
      "parameter": {
        "expireTime": { // redis value cache failure time
          "seconds": 1000
        },
        "keyFieldDelimiter": "u0001", // key separator written
        to redis.
        "dateFormat": "yyyy-MM-dd HH:mm:ss", // time format of
        date when redis is written
        "datasource": "", // Data Source
        "writeMode": { // write mode
          "mode": " ", // alue is the mode of writing for a
          type
          "valueFieldDelimiter": " ", the separator between
          // Value
          "type": " // Value Type
        },
        "Keyindexes": [ // primary key index
          0,
          1-
        ],
        "batchSize": "1000", // number of records submitted in
        one batch size
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
```

```

        "errorLimit": {
            "record": "0"//Number of error records
        },
        "speed": {
            "throttle":false,//False indicates that the traffic is
            not throttled and the following throttling speed is invalid. True
            indicates that the traffic is throttled.
            "concurrent": "1",//Number of concurrent tasks
            "dmu": 1 // DMU Value
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}Writer"
    }
}
}
}
}

```

2.3.3.16 Configure SQL Server Writer

In this article we will show you the data types and parameters supported by SQL Server Writer and how to configure Writer in both wizard mode and script mode.

The SQL Server Writer plug-in can be used to write data in target tables of the primary SQL Server database. At the underlying implementation level, SQL Server Writer connects to a remote SQL Server database through JDBC, and runs the `insert into...` to write data in an SQL Server instance. Data is submitted to the database in batch within the instance.

SqlServer Writer is designed for ETL developers to import data from data warehouses to SqlServer. SqlServer Writer can also be used as a data migration tool by DBA and other users.

SQL Server Writer obtains protocol data (`insert into...`) generated by Reader through the Data Integration framework. If the primary key conflicts with the unique index, data cannot be written in conflicting lines. To improve the performance, we use `PreparedStatement + Batch` and configure `rewriteBatchedStatements=true` to buffer data to the thread context buffer. Write requests are initiated only when the amount of data in the buffer reaches the threshold.



Note:

- Data can be written into a target table only when the target table resides in the primary database.

- The task should at least have the insert into... permission. Whether other permissions are required depends on the statements specified in PreSQL and PostSQL when you configure the task.

Type conversion list

SqlServer Writer supports most data types in SqlServer. Check whether your data type is supported before using it.

The SQL Server writer converts the list of types for SQL Server, as shown below.

Type Classification	SQL Server Data Types
Integer	Bigint、Int、Smallint和Tinyint
Float point	Float, decimal, real numeric
String type	char, nchar, ntext, nvarchar, text, varchar, nvarchar (MAX), and varchar (MAX)
Date and time type	Date, time, and datetime
Boolean	Bit
Binary	Binary, varbinary, varbinary (max), and timestamp

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
table	The name of the selected table that needs to be synchronized.	Yes	None
column	Description: The fields of the target table into which data is required to be written. These fields are separated by commas. For example, "column":["id","name","age"]. If you want to write all columns in turn, use the * representation, for example, "column":["*"].	Yes	None
preSql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, clear old data.	No	None

Attribute	Description	Required	Default Value
postSql	Description: The SQL statement that is run after the data synchronization task is run. Currently, you can run only one SQL statement in wizard mode, and more than one SQL statement in script mode. For example, add a timestamp.	No	None
writeMode	Description: Specifies the import mode. Data can be inserted. insert: If the primary key conflicts with the unique index, Data Integration deems the data as dirty data, but the original data is retained.	No	insert
batchSize	Description: The number of records submitted in batch at a time can greatly reduce the interactions between Data Integration and SQL Server over the network, and increase the overall throughput. However, an excessively large value may cause the running process of Data Integration to become out of memory (OOM).	No	1,024

Development in wizard mode

1. Choose source

Configuration item descriptions:

The screenshot shows the '01 Data Source' configuration step. It has two tabs: 'Source' and 'Destination'.
Source Tab:
 - * Data Source: SQL Server (dropdown)
 - * Table: public.person (dropdown)
 - Data Filtering: A text area with Chinese instructions: '请多写语句或where过滤语句 (不要写where关键字)。通过该语句来对数据进行增量同步。'
 - Sharding Key: col (text input)
 - Preview button at the bottom.
Destination Tab:
 - * Data Source: SQL Server (dropdown)
 - * Table: dbo.worker (dropdown)
 - Statements Run: Before Import: select * from PERSON_PERSON (text input)
 - Statements Run: After Import: select * from PERSON_PERSON (text input)

Parameters:

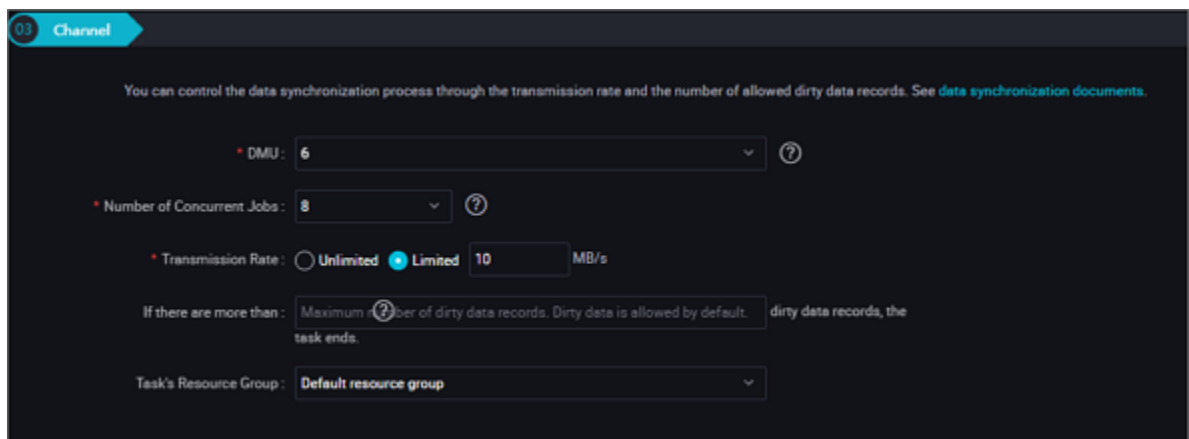
- Data source: The datasource in the preceding parameter description. Enter the data source name you configured.
- Table: table in the preceding parameter description. Select the table to be synchronized.

- Prepared statement before import: preSql in the preceding parameter description, namely, the SQL statement that is run before the data synchronization task is run.
 - Post-import completion statement: postSql in the preceding parameter description, which is the SQL statement that is run after the data synchronization task is run.
 - Primary key conflict: writeMode in the preceding parameter description. You can select the expected import mode.
2. The field mapping, which is the column in the above parameter description.

The source table field on the left and the target table field on the right are one-to-one relationships, click Add row to **add a single** field and click **Delete** to delete the current field.



- In-row mapping: You can click **Enable Same-Line Mapping** to create a mapping for the same row. Note that the data type must be consistent.
 - Automatic formatting: The fields are automatically sorted based on corresponding rules.
3. Control the tunnel



Parameters:

- DMU: A unit which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration. One DMU represents the minimum amount of resources used for a data synchronization task.
- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task. In wizard mode, configure a concurrency for the specified task on the wizard page.
- The maximum number of errors indicates the maximum number of dirty data records.

- Task Resource Group: the machine on which the task runs, if the number of tasks is large, the default Resource Group is used to wait for a resource, it is recommended that you add a Custom Resource Group (currently only 1 East China, east China 2 supports adding custom resource groups). For more information, see [Add scheduling resources](#).

Development in script mode

Configure jobs written to SQL Server, see parameter descriptions for specific parameter completion.

```
{
  "type": "job",
  "version": 2.0, // version number
  "steps": { // The following is a reader template. You can find the
    corresponding reader plug-in documentations.
    {
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "sqlserver", // plug-in name
      "parameter": {
        "postSql": [], // SQL statement that was first
        executed after the data synchronization task was executed
        "datasource": "", // Data Source
        "column": [ // Field
          "id",
          "name"
        ],
        "table": "", // table name
        "preSql": [] // SQL statement that was first executed
        before the data synchronization task was executed
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": 0 // Number of error records
    },
    "speed": {
      "throttle": false, // False indicates that the traffic is
      not throttled and the following throttling speed is invalid. True
      indicates that the traffic is throttled.
      "concurrent": 1, // Number of concurrent tasks
      "dmu": 1 // DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "from": "Reader",
        "to": "Writer"
      }
    ]
  }
}
```

}

2.3.3.17 Configure ElasticSearch Writer

In this article we will show you the data types and parameters supported by ElasticSearch Writer and how to configure Writer in both wizard mode and script mode.

ElasticSearch is a Lucene-based search and data analysis tool that provides a distributed service. ElasticSearch is an open source product that follows Apache's open source terms and is currently the mainstream enterprise-class search engine. The ElasticSearch core concept corresponds to the core concepts of the database as follows.

```
Relational DB (Instance)-> databases (database)-> tables (table) ->
rows (one row of data)-> Columns (one row of data)
Innisearch-> index-> types-> documents-> Fields
```

There can be multiple indexes (INDEX)/(database) in ElasticSearch, each index can contain multiple types (type)/(table), each type can contain multiple document rows, each document can then contain multiple fields (columns). The ElasticSearch writer plug-in uses the rest API interface of ElasticSearch, write the data that is read from the reader in bulk to ElasticSearch.

Parameter description

Attribute	Description	Required	Default Value
endpoint	Description: ElasticSearch URL, in the format of <code>http://xxxx.com:9999</code> .	No	None
accessId	The username of ElasticSearch, which is used for authorization when a connection with ElasticSearch is established.	No	None
accessKey	Description: Password of the ElasticSearch instance.	No	N/A
index	Description: index name in ElasticSearch.	No	None
indexType	Description: type name of index in ElasticSearch.	No	ElasticSearch
cleanup	Whether the data already exists in the index is deleted or not, the method used to clean the data is to delete and rebuild the corresponding index, the default value of false indicates that the data in the existing index is retained.	No	false
batchSize	Description: Number of data entries imported in batch each time.	No	1,000
trySize	Description: Number of retries after failure.	No	30

Attribute	Description	Required	Default Value
timeout-	Client timeout.	No	600,000
discovery	Description: When Node Discovery is enabled, server list in the client is polled and regularly updated.	No	false
compression	Description: Specifies whether compression is enabled for HTTP requests.	No	true
multiThread	Description: http request, multiple threads or not.	No	true
ignoreWriteError	Description: Ignore writing error and keep writing without retries.	No	false
ignoreParseError	Description: Ignore format error of parsing data and keep writing.	No	true
alias	ElasticSearch's alias is similar to the database's view mechanism, creating an alias name for the index my_index, this is like the operation of my_index with respect to maid. Configuring alias means that after the data import is complete, an alias is created for the specified index.	No	N/A
aliasMode	Description: Modes of adding an alias after the data is imported: append and exclusive.	No	append
Settings	If you are inserting a target-side data column type that is an array type, use the specified separator (-, -) split the source data. Example: Source column is string type data a-, -b-, -c-, -d, using the separator-, -after the split is the array ["a", "b", "c", "d"], eventually written into the ElasticSearch corresponding filed column.	No	-, -

Attribute	Description	Required	Default Value
column	<p>Column is used to configure multiple fields of the document, filed information, each specific field item can be configured with a base configuration such as name, type, and so on, and extension configurations such as analyzer, format, and array. Specific instructions are as follows: The Field Types supported by essbase search are as follows.</p> <pre> - id - string - text - keyword - long - integer - short - byte - double - float - date - boolean - binary - integer_range - float_range - long_range - double_range - date_range - geo_point - geo_shape - ip - completion - token_count - array -Object - nested </pre> <p>You can configure analyzer when the column type is text type) the, norms, index_options parameters are as follows.</p> <pre> { "name": "col_text ", "type": "text", "analyzer": "ik_max_word" } </pre> <p>When the column type is a date type, you can configure the format, timezone parameters, represents a date serialization format and a time zone, respectively, as an example.</p> <pre> { "name": "col_date ", "type": "date", "format": "YYYY-MM-dd HH:mm:ss", "timezone": "UTC" } </pre> <p>You can configure tree (geohash or quadtree) when the column type is ge_shape), precision properties, as in the</p>	Yes	N/A

Development in script mode

The following is a script configuration sample. For details about parameters, see the preceding Parameter Description.

```
{
  "job": {
    "setting": {
      ...
    },
    "content": [
      {
        "reader": {
          ...
        },
        "writer": {
          "name": "ElasticSearchwriter",
          "parameter": {
            "endpoint": "http://xxxx.com: 9999 ",
            "accessId": "xxxx",
            "accessKey": "yyyy",
            "index": "test-1",
            "type": "default",
            "cleanup": true,
            "settings": {"index" :{"number_of_shards": 1, "number_of_
replicas": 0}},
            "discovery": false,
            "batchSize": 1000,
            "splitter": ",",
            "column": [
              {"name": "pk", "type": "id"},
              {"name": "col_ip", "type": "ip" },
              {"name": "col_double", "type": "double" },
              {"name": "col_long", "type": "long" },
              {"name": "col_integer", "type": "integer" },
              {"name": "col_keyword", "type": "keyword" },
              {"name": "col_text", "type": "text", "analyzer": "
ik_max_word"},
              {"name": "col_geo_point", "type": "geo_point" },
              {"name": "col_date", "type": "date", "format": "yyyy-MM
-dd HH:mm:ss"},
              {"name": "col_nested1", "type": "nested" },
              {"name": "col_nested2", "type": "nested" },
              {"name": "col_object1", "type": "object" },
              {"name": "col_object2", "type": "object" },
              {"name": "col_integer_array", "type": "integer", "array
":true},
            ]
          }
        }
      ]
    }
  }
}
```



Note:

ElasticSearch for the VPC environment, currently using only custom scheduling resources, if you run in the default Resource Group, there will be a network breakdown. To add a Custom Resource Group, see [Add scheduling resources](#)

2.3.3.18 Configure LogHub Writer

In this article we will show you the data types and parameters supported by LogHub Writer and how to configure Writer in both wizard mode and script mode.

LogHub Writer uses Log Service Java SDK to push data in DataX Reader to the specified Log Service LogHub for consumption by other programs.



Note:

LogHub cannot realize idempotence, and re-execution of the task after FailOver may result in data duplication.

Implementation principles

LogHub Writer uses the DataX framework to obtain data generated by Reader, and converts the data of types supported by DataX into the string type. When the data size reaches the value specified by batchSize, LogHub Writer uses Log Service Java SDK to push all the data to LogHub at a time. By default, 1,024 data entries are pushed. The maximum batchSize value is 4096.

LogHub Writer supports LogHub type conversion, as shown in the following table:

Internal DataX type	LogHub data type
Long	String
Double	String
String	String
Date	String
Boolean	String
Bytes	String

Parameter description

Attribute	Description	Required	Default Value
endpoint	Log Service address	Yes	None
accessKeyId	Description: AccessKeyId for accessing the Log Service instance.	Yes	None

Attribute	Description	Required	Default Value
accessKeySecret	Description: AccessKeySecret for accessing the Log Service instance.	Yes	None
project	Description: Project name of target Log Service.	Yes	None
logstore	Name of the Logstore of the target Log Service instance.	Yes	None
topic	Description: Select a topic	No	Null string
batchSize	Description: Number of data entries that can be pushed at a time.	Required : No. The default value is 1024.	None
column	Description: Name of the column in each data entry	Yes	None

Introduction to script mode

The wizard mode configuration is not supported at this time, you can click on the link to convert to script mode or select import Script Template for development.

Introduction to script mode

The following is a script configuration sample. For details about parameters, see the preceding Parameter Description.

```
{
  "type": "job",
  "version": "2.0", //version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "loghub", // plug-in name
      "parameter": {
        "datasource": "", //Name of the data source
        "column": [ // Field
          "col0",
          "col1",
          "col2",
          "col3",
          "col4",
          "col5"
        ]
      }
    }
  ]
}
```



```

        ],
        "topic": "", // select topic
        "batchSize": "1000", // number of records submitted in
one batch size
        "logstore": "//The name of the target LOL logstore
    },
    "name": "Writer",
    "category": "writer"
    }
    ],
    "setting": {
        "errorLimit": {
            "record": "//Number of error records
        },
        "speed": {
            "concurrent": "3", //Number of concurrent tasks
            "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
            "dmu": 1 // DMU Value
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}

```

2.3.3.19 Configure OpenSearch Writer

In this article we will show you the data types and parameters supported by OpenSearch Writer and how to configure Writer in both wizard mode and script mode.

The OpenSearch Writer plug-in is designed to insert or update data into OpenSearch. Data developers can use it to import processed data into OpenSearch and output data by searching . How fast data can be transmitted depends on the qps of the account corresponding to the OpenSearch table.

Implementation

At the underlying implementation level, OpenSearch Writer provides the openly available OpenSearch API by means of OpenSearch.

- OpenSearch V3 uses internal dependent databases, with POM of com.aliyun.opensearch aliyun-sdk-opensearch 2.1.3.



Note:

- To use the OpenSearch Writer plug-in, you must use JDK 1.6-32 or later versions. You can view the Java version through `java -version`.

- Currently, the default resource group does not support connections to the VPC environment due to possible network problems.

Plug-in features

Column order

The columns in OpenSearch are unordered, so you should use OpenSearch Writer to write data in strict accordance with the order of the specified columns. If the number of specified columns is less than that in OpenSearch, the redundant columns are set to the default value or null.

For example, if the field list to be imported contains fields b and c but the OpenSearch table contains fields a, b, and c, you can configure the column to "column": ["c","b"]. The first two columns in Reader are imported to fields c and b in OpenSearch, and the field a, into which new records are inserted, is set to the default value or null.

- **How to handle column configuration errors**

To ensure data is written in a reliable manner, data loss from redundant columns must be prevented to avoid data quality failure. When redundant columns are written, OpenSearch Writer produces an error.- If the OpenSearch table contains fields a, b, and c, OpenSearch Writer produces an error when more than three fields are written by OpenSearch Writer.

- **Table configuration considerations**

OpenSearch Writer can only write the data from one table at a time.

- **Task rerunning and failover:**

After one task is rerun, the data is automatically overwritten based on IDs. Therefore , OpenSearch must contain one ID column. The ID uniquely identifies a record line in OpenSearch. The data same as the unique ID will be overwritten.

- **Task rerunning and failover:**

After one task is rerun, the data is automatically overwritten based on IDs.


OpenSearch Writer supports most data types in OpenSearch. Check whether your data type is supported. OpenSearch Writer converts the data types in OpenSearch as follows:

Category	Opensearch Data Type
Integer	Int
Float point	Double/Float
String type	TEXT/Literal/SHORT_TEXT
Date and time type	Int

Category	Opensearch Data Type
Boolean	Literal

Parameter description

Attribute	Description	Required	Default Value
accessId	Description: Logon ID for the Alibaba Cloud system.	Yes	None
accessKey	Description: Key for the Alibaba Cloud system.	Yes	None
host		Yes	None
indexName	Description: The name of the OpenSearch project.	Yes	None
table	Description: The table to which the data is written. You cannot enter more than one table, because DataX does not support importing multiple tables at a time.	Yes	None
column	Description: The list of fields to be imported. If you need to import all the fields, it can be configured to "column": ["*"]. If you need to insert some of the OpenSearch columns, enter these columns, for example, "column": ["id", "name"]. OpenSearch supports column filtering and column order changing. For example, a table has three fields: a, b, and c, and you only want to synchronize fields c and b. You can configure it to ["c, b"]. During the import process, field a is automatically inserted with null values and set to null.	Yes	None
batchSize	The number of data lines written in a single note. Data is written into OpenSearch in batches. In general, the advantage of OpenSearch is query, and its write performance (tps) is not impressive. Proceed with the configuration based on the resources applied by your account. For OpenSearch, generally, a single item of data is less than 1 MB, and the data to be written at a time is less than 2 MB.	Required : This option is required for a partition table. Do not enter this field if the target table is not a partition table.	300

Attribute	Description	Required	Default Value
writeMode	<p>Description: In OpenSearch Writer, "writeMode": "add/update" is configured to ensure the idempotence of write operations.</p> <ul style="list-style-type: none"> - "add": When a reattempt is made after a failed write attempt, OpenSearch Writer cleans up this data and imports the new data (atomic operation). - "update": It indicates that the data is inserted in a modified manner (atomic operation). <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;">  Note: In OpenSearch, batch insert is not an atomic operation, which may be partially successful. Therefore, writeMode is a critical option. OpenSearch with version =v3 does not support the update operation currently**. </div>	Yes	None
ignoreWriteError	<p>Description: Ignores write errors.</p> <p>Configuration example: "ignoreWriteError": true.</p> <p>OpenSearch write operations are performed in batches. It indicates whether to ignore the write failure occurred in the current batch. If yes, other write operations keep going. If no, the current task is ended, and an error is returned. The default value is recommended.</p>	No	false
version	<p>Description: The version information of OpenSearch.</p> <p>Configuration sample: "version": "v3". OpenSearch V2 has multiple limitations on push operations, so OpenSearch V3 is preferable.</p>	No	v2

Development in script mode

Configure the data synchronization job to write data to OpenSearch:

```
{ "type": "job", "version": "1.0", "configuration": { "reader": {}, "writer": { "plugin": "opensearch",
"parameter": { "accessId": "*****", "accessKey": "*****", "host": "http://yyyy.aliyuncs.com",
"indexName": "datax_xxx", "table": "datax_yyy", "column": [ "appkey", "id", "title", "gmt_create",
"pic_default" ], "batchSize": 500, "writeMode": add, "version": "v2", "ignoreWriteError": false } } }
```

2.3.3.20 Configure Table Store (OTS) Writer

In this article we will show you the data types and parameters supported by Table Store (OTS) Writer and how to configure Writer in both wizard mode and script mode.

Table Store (originally known as OTS) is a NoSQL database service built on the Alibaba Cloud Apsara distributed system, allowing storage of and real-time access to massive structured data. Table Store organizes data into instances and tables. Using data partition and server load balancing technology, it provides seamless scaling.

In short, Table Store Writer-Internal connects to the Table Store server through official Table Store Java SDKs and writes data in the Table Store server through SDKs. Table Store Writer has greatly optimized the write process, including retry upon write timeout, retry upon exception in writing, batch submission, and other features.

Currently, Table Store Writer-Internal supports all types of Table Store data and converts data types for Table Store as follows:

- PutRow: PutRow for Table Store API, which is used to insert data to a specified row. If this row does not exist, a new row is added. Otherwise, the original row is overwritten.
- UpdateRow: UpdateRow for Table Store API, which is used to update the data of a specified row. If the row does not exist, a new row is added. Otherwise, the values of the specified columns are added, modified, or deleted as request.

Currently, Table Store Writer supports all Table Store data types and converts the data types in Table Store as follows:

Type Classification	Table store data type
Integer	Integer
Float	Double
String	String
Boolean	Boolean
Binary	Binary

**Note:**

You must configure the Integer category to Int in script mode so that it can be converted to the Integer type for Table Store. If you directly configure it to the Integer type for Table Store, an error is reported in the log and causes task failure.

Parameter description

Attribute	Description	Required	Default Value
datasource	Data source name. It must be identical to the data source name added. Adding data source is supported in script mode.	Yes	None
endPoint	The endpoint for the table store server, see access control for details.	Yes	None
accessId	Description: accessId of a Table Store instance	Yes	None
accessKey	AccessKey required for accessing Table Store service	Yes	None
instanceName	Description: Name of the Table Store instance An instance is an object for using and managing Table Store. After activating Table Store, you need to create an instance through the console, and then create and manage tables in the instance. An instance is a basic unit for Table Store resource management. Table Store controls access to applications and measures resources on an instance basis.	Yes	None
table	Description: The name of the table to be extracted. You can enter only one table name. Multi-table synchronization is not required for Table Store.	Yes	None

- **primaryKey**
 - Primary key information of Table Store. Field information is described using JSON array. Table Store itself is a NoSQL system, so the corresponding field name must be specified when Table Store Writer imports data.
 - Required: Yes.
 - PrimaryKey of Table Store only supports STRING and INT types, so only these two types can be entered for Table Store Writer.

Data synchronization system supports data type conversion, so Table Store Writer can convert the non-String and non-Int source data. Configuration example:

```
"primaryKey" : [  
  {"name": "pk1", "type": "string"},  
],
```

- **column**

- Description: The column name set to be synchronized in the configured table. Field information is described with arrays in JSON.
- Required: Yes.
- By default, this field is not specified.

The format is as follows:

```
{"name": "col2", "type": "INT"},
```

"name" specifies the name of Table Store's column to be written, and "type" specifies the type of data to be written. Data types supported by Table Store include STRING, INT, DOUBLE, BOOL, and BINARY.

Constants, functions, or custom statements are not supported during write process.

- writeMode
 - Description: Write mode. The following three modes are supported:
- Single row operation

```
GetRow: Read data from a single row.
PutRow: PutRow for Table Store API, which is used to insert data to
       a specified row. If this row does not exist, a new row is added.
       Otherwise, the original row is overwritten.
UpdateRow: UpdateRow for Table Store API, which is used to update the
          data of a specified row. If the row does not exist, a new row is added
          . Otherwise, the values of the specified columns are added, modified,
          or deleted as request.
DeleteRow: Delete a row.
```

- Batch Operation

```
BatchGetRow: Read data from multiple rows.
```

- Read range

```
GetRange: Read table data within a certain range.
```

- Required: Yes
- By default, this field is not specified.

Development in wizard mode

Currently, development in wizard mode is not supported.

Development in script mode

Configure a job to write data to Table Store:

```
{
  "type": "job",
  "version": 2.0, // version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "ots", //plug-in name
      "parameter": {
        "datasource": "", // Data Source
        "column": [ // Field
          {
            "name": "columnname1", // field name
            "type": "INT" // data type
          },
          {
            "name": "columnname2 ",
            "type": "STRING"
          },
          {
            "name": "columnname3 ",
            "type": "double"
          },
          {
            "name": "columnname4 ",
            "type": "BOOLEAN"
          },
          {
            "name": "columnname5 ",
            "type": "BINARY"
          }
        ],
        "writeMode": "insert", //Write mode
        "table": "", // table name
        "primaryKey": primary key information for [ // table
store
          {
            "name": "pk1",
            "type": "STRING"
          },
          {
            "name": "pk2",
            "type": "INT"
          }
        ]
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    }
  }
}
```



```

        "speed": {
            "throttle": false, // False indicates that the traffic is
            not throttled and the following throttling speed is invalid. True
            indicates that the traffic is throttled.
            "concurrent": "1", // Number of concurrent tasks
            "dmu": 1 // DMU Value
        },
        "order": {
            "hops": [
                {
                    "from": "Reader",
                    "to": "Writer"
                }
            ]
        }
    }
}

```

2.3.3.21 Configure RDBMS Writer

In this article we will show you the data types and parameters supported by RDBMS Writer.

The RDBMS Writer plug-in provides the ability to write data into the target table of the master RDBMS database. At the underlying implementation level, RDBMS Writer connects to a remote RDBMS database through JDBC, and runs the insert into ... to write data into RDBMS. RDBMS Writer is a relational database write plug-in for generic purposes, allowing you to add any relational database write support by registering database drivers or other methods.

RDBMS Writer is designed for ETL developers to import data from data warehouses to RDBMS. RDBMS Writer can also be used as a data migration tool by DBA and other users.

Implementation principles

RDBMS Writer uses the DataX framework to get the protocol data generated by Reader. Then it connects to a remote RDBMS database through JDBC, and runs the insert into ... to write data into RDBMS.

Function description

Configuration sample

- Configure a job for writing data into RDBMS.

```

{
    "job": {
        "setting": {
            "speed": {
                "channel",
            }
        },
        "content": [
            {
                "reader": {
                    "name": "streamreader",
                    "parameter": {

```

```

        "Column": [
            {
                "value": "DataX",
                "type": "string",
            },
            {
                "value": 19880808,
                "type": "long"
            },
            {
                "value": "1988-08-08 08:08:08",
                "type": "date",
            },
            {
                "doc_value": true,
                "type": "bool"
            },
            {
                "value": "test",
                "type": "bytes"
            }
        ],
        "sliceRecordCount": 1000
    },
    "writer": {
        "name": "RDBMS Writer",
        "parameter": {
            "connection": [
                {
                    "jdbcUrl": "jdbc:dm://ip:port/database",
                    "table": [
                        "table"
                    ]
                }
            ],
            "username": "username",
            "password": "password",
            "table": "table",
            "column": [
                "*"
            ],
            "preSql": [
                "delete from XXX;"
            ]
        }
    }
}

```

Parameter description

- jdbcUrl
 - Description: Information of the JDBC connection to the opposite-end database. The format of jdbcUrl is in accordance with the RDBMS official specification, and the URL attachment

control information can be entered. Note that JDBC formats vary with databases and DataX selects an appropriate database driver for data reading based on a specific JDBC format.

- DM: jdbc:dm://ip:port/database
- DB2 format: jdbc:db2://ip:port/database
- PPAS format: jdbc:edb://ip:port/database

How to add database support using RDBMS Writer:

- Enter the corresponding directory of RDBMS Writer. \${DATAX_HOME} is the main directory of DataX, that is, \${DATAX_HOME}/plugin/writer/RDBMS Writer.
- Find the plugin.json file under the directory of RDBMS Writer and register your database driver into the file (keep the database driver in the drivers array). RDBMS Writer dynamically selects an appropriate database driver to connect the database during task execution.

```
{
  "name": "RDBMS Writer",
  "class": "com.alibaba.datax.plugin.reader.RDBMS Writer.RDBMS Writer",
  "description": "useScene: prod. mechanism: Jdbc connection using the database, execute select sql, retrieve data from the ResultSet. warn: The more you know about the database, the less problems you encounter.",
  "developer": "alibaba",
  "Drivers": [
    "dm.jdbc.driver.DmDriver",
    "com.ibm.db2.jcc.DB2Driver",
    "com.sybase.jdbc3.jdbc.SybDriver",
    "com.edb.Driver"
  ]
}
```

- Find the libs subdirectory under the directory of RDBMS Writer and keep your database driver in the libs subdirectory.

```
$tree
.
|-- libs
|   |-- Dm7JdbcDriver16.jar
|   |-- commons-collections-3.0.jar
|   |-- commons-io-2.4.jar
|   |-- commons-lang3-3.3.2.jar
|   |-- commons-math3-3.1.1.jar
|   |-- datax-common-0.0.1-SNAPSHOT.jar
|   |-- datax-service-face-1.0.23-20160120.024328-1.jar
|   |-- db2jcc4.jar
|   |-- druid-1.0.15.jar
|   |-- edb-jdbc16.jar
|   |-- fastjson-1.1.46.sec01.jar
|   |-- guava-r05.jar
|   |-- hamcrest-core-1.3.jar
|   |-- jconn3-1.0.0-SNAPSHOT.jar
|   |-- logback-classic-1.0.13.jar
|   |-- logback-core-1.0.13.jar
```

```

|-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
|-- slf4j-api-1.7.10.jar
|-- plugin.json
|-- plugin_job_template.json
|-- RDBMS Writer-0.0.1-SNAPSHOT.jar

```

- Required: Yes
- By default, this field is not specified.

Attribute	Description	Required	Default Value
username	Data Source User Name	Yes	None
password	Description: Password corresponding to the specified username for the data source.	Yes	None
table	Description: Target table name. If the schema information of table is not consistent with the username in the preceding configuration, enter the table information in the schema.table format.	Yes	None
column	Description: The column name set to be synchronized in the configured table. separated by commas (,). We strongly recommend against the default column configuration.	Yes	None
Presql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement, for example , clear old data.	No	None

Attribute	Description	Required	Default Value
Postsql	Description: The SQL statement that is run before the data synchronization task is run. Currently, you can run only one SQL statement, for example , add a timestamp.	No	None
batchSize	Description: The quantity of records submitted in one operation. Setting this parameter can greatly reduce the interactions between DataX and RDBMS over the network, and increase the overall throughput. However , an excessively large value may cause the running process of DataX to become out of memory (OOM).	No	1,024

Type conversion

RDBMSReader supports most generic rational database types such as numbers and characters. Check whether your data type is supported and select a reader based on a specific database.

2.3.3.22 Configure Stream Writer

In this article we will show you the data types and parameters supported by Stream Writer and how to configure Writer in script mode.

The Stream Writer plug-in provides the ability to read data from Reader and print data on the screen or directly discard data. It is mainly applicable to performance testing for data synchronization and basic functional testing.

Parameter description

- Print
 - Description: Whether to print the outputted data on the screen.

- Required: No
- Default value: true.

Development in wizard mode

Currently, development in wizard mode is not supported.

Development in script mode

Configure a job to read data from the Reader and print the data on the screen:

```
{
  "type": "job",
  "version": "2.0 ", // version number
  "steps": [
    { //The following is a reader template. You can find the
      corresponding reader plug-in documentations.
      "stepType": "stream",
      "parameter": {},
      "name": "Reader",
      "category": "reader"
    },
    {
      "stepType": "stream", //plug-in name
      "parameter": {
        "print": false, // do you want to print output to the
screen?
        "fieldDelimiter": ",", //Delimiter of each column
      },
      "name": "Writer",
      "category": "writer"
    }
  ],
  "setting": {
    "errorLimit": {
      "record": "0" //Number of error records
    },
    "speed": {
      "throttle": false, //False indicates that the traffic is
not throttled and the following throttling speed is invalid. True
indicates that the traffic is throttled.
      "concurrent": "1", //Number of concurrent tasks
      "dmu": "1" //DMU Value
    }
  },
  "order": {
    "hops": [
      {
        "from": "Reader",
        "to": "Writer"
      }
    ]
  }
}
```

```
}
```

2.3.4 Optimizing configuration

Influence factors of the data synchronization speed, differences between speed-limited jobs and speed-not-limited jobs, precautions for custom resource groups, and how to adjust the DMU configuration and concurrent configuration of synchronization jobs for the maximum synchronization speed.

DataWorks Data Integration supports real-time, offline data interconnection between any data sources in any location and any network environment. It is a comprehensive full-stack data synchronization platform that allows you to copy dozens of TBs of data between various cloud and local data storage media.

The super fast data transmission performance and interconnection between over 400 pairs of heterogeneous data sources are the crucial factor that helps users focus on core big data issues only. The service can be used to design advanced analysis solutions with deep insight into all data

Factors affecting the speed of data synchronization

The factors that affect the speed of data synchronization are as follows.

- Source-side data sources
 - Database performance: The performance of the CPU, memory module, SSD, network, and hard disk.
 - Concurrency: A high data source concurrency results in a high database workload.
 - Network: The bandwidth (throughput) and speed of the network. Generally, a database with better performance can bear a higher concurrency. Therefore, the data synchronization job can be configured for high-concurrency data extraction.
- Synchronous task configuration for data integration
 - Synchronization speed: whether a limit is set for the synchronization speed.
 - DMU: the amount of resources used for running the synchronization task.
 - Concurrency: the maximum number of threads that can be used to read data from the data source, or write data to the target data source at the same time in one synchronization task.
 - Wait resource.
 - Bytes setting. If Bytes is set to 1048576, and the network is slow, the data transmission is timed out before it completes. We recommend that you set Bytes to a lower value.
 - Whether to create an index for query statements.

- Objective To end Data Source
 - Performance: The performance of the CPU, memory module, SSD, network, and hard disk.
 - Load: High database load, affecting data write efficiency.
 - Network: The bandwidth (throughput) and speed of the network.

You need to monitor and optimize the performance, load, and network of the originating data source and destination databases. The following mainly describes how to set core configurations of a synchronization task on Data Integration.

DMU

- Configuration

A data synchronization task can run using single or multiple DMUs. In Wizard mode, you can configure a maximum of 20 DMUs for a task. The following is an example of how to set the number of DMUs in Script mode:

```
"Setting ":{  
  "Speed ":{  
    "dmu": 10  
  }  
}
```



Note:

Note: If system performance is good, you can set the number of DMUs to more than 20 using a script. However, this may not improve system performance. Do not assign too many DMUs to a task.

- Relationship between DMU and operation speed

The DMU represents the resource capability, and the synchronization task is configured with a higher DMU, you can allocate more resources, but it does not mean that the speed of the synchronization task must be improved. Speed tuning requires combining concurrent, DMU ratio tuning between the two. For example, a synchronization task that configures 3 concurrency, requires 3dmu, and the synchronization speed is 10 Mb/s. At this time, the number of 3 Concurrent resources required is 3dmu, and the task does not need to use more resources, increasing the DMU does not, therefore, increase the speed of the synchronization task.

Concurrency

- Configuration

In wizard mode, configure a concurrency for the specified task on the wizard page. The following is an example of configuring the number of concurrency with Script Mode.

```
"Setting ":{
  "Speed ":{
    "concurrent": 10
  }
}
```

- Concurrent relationship with DMU

A higher concurrency requires more DMUs. When network conditions and performance of data sources are good, more DMUs and higher concurrency will lead to better synchronization speed.

- To ensure that a task can be successfully executed at high concurrency, in Wizard mode , the highest concurrency allowed must not exceed the number of DMUs you set. For example, do not configure more than 10 concurrent threads when the number of DMUs is set to 10.
- When a high concurrency is set, you need to consider data source capabilities of reading and writing ends. Excessive concurrency may affect the performance of source database. Therefore, you need to tune the database.
- In Script mode you can set a high concurrency. However, the number of DMUs that can be provided for a task are limited. Do not set an excessively high concurrency.

Speed Limit

After the beta phase of Data Integration has ended, throttling is disabled by default. In a synchronization task, data is synchronized at the maximum speed supported by the concurrency and DMUs configured for that task. Considering that excessively fast synchronization may overstress the database and thus affect the production, Data Integration allows you to limit the synchronization speed and optimize the configuration as required. It is recommended that the maximum speed should not exceed 30 MB/s when this option is enabled. The following is a sample example for configuring the speed limit in script mode, in which the transmission bandwidth is 1 MB/s:

```
"Setting ":{
  "Speed ":{
    "throttle": true // Throttling enabled.
    "mbps": 1, // Synchronization speed
  }
}
```

}

**Note:**

- Note: When the throttling parameter is set to false, throttling is disabled, and you do not need to configure the mbps parameter.
- The traffic measured value is a Data Integration metric and does not represent the actual NIC traffic. Generally, the NIC traffic is two to three times of the channel traffic, which depends on the serialization of the data storage system.
- A semi-structured Single file does not have the concept of cutting keys, multiple files can set the maximum job rate to increase the speed of synchronization, however, the maximum job rate is related to the number of files. For example, there are n files with maximum job rate limit set to n mb/s, if you set n + 1 Mb/s or sync at n mb/s speed, if set to n-1 mb/s, synchronization is performed at n-1 mb/s speed.
- Only when a maximum job rate and a splitting key are configured for a relational database, table splitting can be performed according to the set maximum job rate. Relational databases only support numeric splitting keys, but Oracle databases support both numeric and string splitting keys.

Cases of slow data synchronization**Synchronization tasks remain in the waiting status when using public scheduling (WAIT) resources**

- Related examples are as follows

When you test the synchronization tasks in DataWorks, multiple tasks remain in the waiting status and an internal system error occurs.

It takes 800 seconds to synchronize a task from RDS to MaxCompute using the default resource group. But the log shows that the task runs for only 18 seconds and stops. Other synchronization tasks with hundreds of data entries also remain in the waiting status.

The waiting log is displayed as follows:

```
2017-01-03 07: 16: 54: State: 2 (wait) | Total: 0r 0b | speed: 0r/s
0b/S | error: 0r 0b | stage: 0.0%
```

- Solution

In this case, public scheduling resources are used, whose capability is limited because they are shared by many projects but not only two or three tasks of a single user. A 10-second task was

extended to 800 seconds because the required resources were insufficient and must be waited for when you ran the task.

If you have strict requirements on the synchronization speed and the waiting time, we recommend starting the synchronization tasks in non-rush hours. Typically, synchronization tasks are concentrated between 00:00 and 03:00. You can perform synchronization tasks in other time apart from the aforesaid period to avoid resource waiting.

Accelerate the tasks of synchronizing the data in multiple tables to the same table

- Related examples are as follows

Synchronization tasks are serialized to synchronize the tables of multiple data sources to the same table, but the synchronization duration turns out to be a long one.

- Solution

To start multiple tasks to write data to the same database at the same time, pay attention to the followings:

- Ensure that the load capacity of the destination database is sufficient to prevent improper running.
- When you configure workflow tasks, select a single task node and configure database or table sharding tasks, or set multiple nodes to run concurrently in the same workflow.
- If the synchronization tasks encounter resource waiting (WAIT) during running, run them in non-rush hours for a high execution priority.

No index added while using the SQL WHERE clause

- Related examples are as follows

The executed SQL statement is as follows:

```
select bid,inviter,uid,createTime from `relatives` where createTime  
>='2016-10-2300:00:00'and reateTime<'2016-10-24 00:00:00';
```

Query statement execution started at 2016-10-25 11:01:24.875 Beijing Time (UTC+8). Query result return started at 2016-10-25 11:11:05.489 Beijing Time (UTC+8). The synchronization program waited the database to return the SQL query result, and MaxCompute waited for a long time to start.

- Cause analysis:

When the where condition was executed, the createTime column was not indexed and full-table scanning was enforced.

- Solution

We recommend that you add an index to the column you want to scan if you want to use the SQL WHERE clause.

2.4 Common configuration

2.4.1 Add security group

This article describes how to add a corresponding security group when you are using DataWorks (formerly Data IDE) in different regions.

To ascertain the security and stability of databases, you must add the IP addresses or IP segments used for accessing the database to the [Add whitelist](#) or security group of the target instance before using certain database instances. This article describes how to add a corresponding security group when you are using DataWorks (formerly Data IDE) in different regions.

Add a security group

- If your data synchronization tasks run on your own ECS resource group, you should authorize your ECS resource group by adding its private/public IP and port to the ECS security group.
- If your data synchronization tasks run on the default resource group, you should add your security group based on your ECS machine region. For example, if your ECS is North China 2, you should add the security group based on North China 2 (Beijing): 2ze3236e8pcbxw61o9y0 and 1156529087455811, as shown in the following table.

Region	Authorization object	Account ID
China (Hangzhou)	sg-bp13y8iuj33uqpqvgqw2	1156529087455811
China (Shanghai)	sg-uf6ir5g3rlu7thymywna	1156529087455811
China (Shenzhen)	sg-wz9ar9o9jgok5tadj7ll	1156529087455811
Asia Pacific SE 1(Singapore)	sg-t4n222njci99ik5y6dag	1156529087455811
Hong Kong	Sg-j6c28uqpqb27yc3tjmb6	1156529087455811
US West 1 (Silicon Valley)	sg-rj9bowpmdvhy153lza2j	1156529087455811
US East 1	sg-0xienf2ak8gs0puz68i9	1156529087455811
China (Beijing)	sg-2ze3236e8pcbxw61o9y0	1156529087455811

**Note:**

ECS in VPC environment does not support adding the above security groups.

Add an ECS security group

1. Log on to the Administration Console of the cloud server ECS.
2. Select the **network and security** > **groups** in the left-hand navigation bar.
3. Select the target region.
4. Locate the security group where you want to configure authorization rules, and click the **configuration rule** that is listed in the action.
5. Click **Security Groups** and click **Add Rules**.
6. Sets the parameters in the dialog dialog box.
7. Click **Confirm**.

2.4.2 Add whitelist

This article describes how to add a corresponding whitelist and security group when you are using DataWorks in different regions.

To make sure the security and stability of databases, you can add the IP addresses or IP segments used for accessing the database to the whitelist or [Add security group](#) of the target instance before using certain database instances.

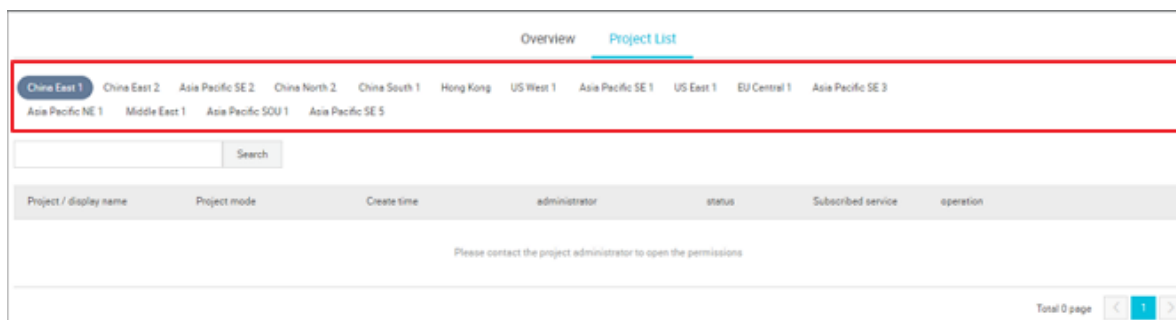
**Note:**

You can only add whitelists for Data integration tasks. For other kinds of tasks, adding whitelists are not supported.

Add a whitelist

1. Enter the [DataWorks management console](#) as a developer and navigate to the **project list** page.
2. Select a project region.

Currently, the supported regions are China East 2 (Shanghai), China South 1 (Shenzhen), Hong Kong, and Asia Pacific SOU 1 (Singapore). The default region is China East 2, and you can switch to other regions where your project is located, as shown in the following figure.



3. Select the whitelist for your project region.

Some data sources have a whitelist restrictions currently and need to add IPs of data integration to whitelists. Common data sources, such as RDS, MongoDB, and Redis, need to add IPs to whitelists in their consoles. Adding a whitelist has the following two case:

- When a sync task is running on the custom resource group. You must authorize machines for the custom resource group, and add machines' intranet IPs and extranet IPs to the whitelist of data source.
- The whitelist entries differs from region to region. Select the whitelist for the selected region from the following table.

region	Whitelist
China East 1(Hangzhou)	100.64.0.0/8,11.193.102.0/24,11.193.215.0/24,11.194.110.0/24,11.194.73.0/24,118.31.157.0/24,47.97.53.0/24,11.196.23.0/24,47.99.12.0/24,47.99.13.0/24,114.55.197.0/24,11.197.246.0/24,11.197.247.0/24
China East 2(Shanghai)	11.193.109.0/24,11.193.252.0/24,47.101.107.0/24,47.100.129.0/24,106.15.14.0/24,10.117.28.203,10.117.39.238,10.143.32.0/24,10.152.69.0/24,10.153.136.0/24,10.27.63.15,10.27.63.38,10.27.63.41,10.27.63.60,10.46.64.81,10.46.67.156,11.192.97.0/24,11.192.98.0/24,11.193.102.0/24,11.218.89.0/24,11.218.96.0/24,11.219.217.0/24,11.219.218.0/24,11.219.219.0/24,11.219.233.0/24,11.219.234.0/24,118.178.142.154,118.178.56.228,118.178.59.233,118.178.84.74,120.27.160.26,120.27.160.81,121.43.110.160,121.43.112.137,100.64.0.0/8
China South 1(Shenzhen)	100.106.46.0/24,100.106.49.0/24,10.152.27.0/24,10.152.28.0/24,11.192.91.0/24,11.192.96.0/24,11.193.103.0/24,100.64.0.0/8,120.76.104.0/24,120.76.91.0/24,120.78.45.0/24
Hong Kong	10.152.162.0/24,11.192.196.0/24,11.193.11.0/24,100.64.0.0/8,11.192.196.0/24,47.89.61.0/24,47.91.171.0/24,11.193.118.0/24,47.75.228.0/24

region	Whitelist
Asia Pacific SE 1(Singapore)	100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11.192.153.0/24,11.192.40.0/24,11.193.8.0/24,100.64.0.0/8,100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11.192.40.0/24,47.88.147.0/24,47.88.235.0/24,11.193.162.0/24,11.193.163.0/24,11.193.220.0/24,11.193.158.0/24,47.74.162.0/24,47.74.203.0/24,47.74.161.0/24,11.197.188.0/24
Asia Pacific SE 2(Sydney)	11.192.100.0/24,11.192.134.0/24,11.192.135.0/24,11.192.184.0/24,11.192.99.0/24,100.64.0.0/8,47.91.49.0/24,47.91.50.0/24,11.193.165.0/24,47.91.60.0/24
China North 2(Beijing)	100.106.48.0/24,10.152.167.0/24,10.152.168.0/24,11.193.50.0/24,11.193.75.0/24,11.193.82.0/24,11.193.99.0/24,100.64.0.0/8,47.93.110.0/24,47.94.185.0/24,47.95.63.0/24,11.197.231.0/24,11.195.172.0/24,47.94.49.0/24,182.92.144.0/24
US West 1	10.152.160.0/24,100.64.0.0/8,47.89.224.0/24,11.193.216.0/24,47.88.108.0/24
US East 1	11.193.203.0/24,11.194.68.0/24,11.194.69.0/24,100.64.0.0/8,47.252.55.0/24,47.252.88.0/24
Asia Pacific SE 3 (Malaysia)	11.193.188.0/24,11.221.205.0/24,11.221.206.0/24,11.221.207.0/24,100.64.0.0/8,11.214.81.0/24,47.254.212.0/24,11.193.189.0/24
EU Central 1(Germany)	11.192.116.0/24,11.192.168.0/24,11.192.169.0/24,11.192.170.0/24,11.193.106.0/24,100.64.0.0/8,11.192.116.14,11.192.116.142,11.192.116.160,11.192.116.75,11.192.170.27,47.91.82.22,47.91.83.74,47.91.83.93,47.91.84.11,47.91.84.110,47.91.84.82,11.193.167.0/24,47.254.138.0/24
Asia Pacific NE1(Japan)	100.105.55.0/24,11.192.147.0/24,11.192.148.0/24,11.192.149.0/24,100.64.0.0/8,47.91.12.0/24,47.91.13.0/24,47.91.9.0/24,11.199.250.0/24,47.91.27.0/24
Middle East 1(Dubai)	11.192.107.0/24,11.192.127.0/24,11.192.88.0/24,11.193.246.0/24,47.91.116.0/24,100.64.0.0/8
Asia Pacific SE 1(Mumbai)	11.194.10.0/24,11.246.70.0/24,11.246.71.0/24,11.246.73.0/24,11.246.74.0/24,100.64.0.0/8,149.129.164.0/24,11.194.11.0/24
UK	11.199.93.0/24,100.64.0.0/8
Asia Pacific SE 5 (Jakarta)	11.194.49.0/24,11.200.93.0/24,11.200.95.0/24,11.200.97.0/24,100.64.0.0/8,149.129.228.0/24,10.143.32.0/24,11.194.50.0/24

Add an RDS whitelist

The RDS data source can be configured in the following two ways.

- RDS instance

In this case, a data source is created by using an RDS instance. Currently, the connectivity test (including the RDS in VPC environments) is supported. If the connectivity test fails, you can try to add the data source using the JDBCURL.

- JDBCURL

For the IP in JDBCURL, enter an intranet IP address or an Internet IP address if no intranet IP address is available. The intranet IP address features faster synchronization because the address is relevant to Alibaba Cloud data centers, while the synchronization speed of the internet IP address is subject to the available internet bandwidth.

RDS whitelist configuration

When Data Integration is connected to RDS for data synchronization, the database standard protocol must be connected to the database. The RDS permits all IP connections by default. If you specify an IP whitelist during RDS configuration, you must add an IP whitelist of Data Integration execution nodes. If no RDS whitelist is specified, no whitelist is provided for Data Integration.

If you have set up an IP white list for your RDS, go to the RDS [Management Console](#), and navigate to **security control** to make the whitelike settings based on the [whitelike list](#) above.



Note:

If you use a custom resource group to schedule the RDS data synchronization task, you must add the IP address of the computer hosting the custom resource group to the RDS whitelist.

2.4.3 Add scheduling resources

Project administrators can create new and modify scheduled resources on the **data integration > synchronous Resource Management > Resource Group** page.

When the default scheduling resource is unable to connect to your complex network environment, with the deployment of the data integration agent, the synchronization of data transfer between any network environment can be reached, see [Data integration when the network of data source \(one side only\) is disconnected](#) and [Data sync when the network of data source \(both sides\) is disconnected](#) for details.



Note:

- Scheduling resources added in Data Integration can only be used for data integration.

- Admin permission is required to customize some files running on a resource group, for example, calling shell files, SQL on custom ECS in a shell script task that you write yourself documents, etc.

Purchase the ECS cloud server

Purchase the ECS cloud server.



Note:

- centos6、centos7 or AliyunOD is recommended.
- If the added ECS instance needs to run MaxCompute or synchronization tasks, verify whether the current Python version of the ECS instance is 2.6 or 2.7 (The Python version of CentOS 5 is 2.4 while those of other operating systems are later than 2.6).
- Ensure that the ECS instance has a public IP address.
- The configuration of the ECS is recommended for the 8-core 16g.

View the ECS host name and the internal network IP address

You can go to the **cloud server ECS > instance** page to view the ECS host name and IP purchased.

Provision 8000 port to read log



Note:

If it is a VPC network type, there is no need to provision a 8000 port.

1. Add security group rules

Navigate to the **cloud server ECS > network and security > Security Group** page, click **configuration rules** , and enter the configuration rules page.

2. Go to the **security group rules > Intranet entry direction** page, and click in the upper right corner to **add security group rules** .

3. Complete the configuration information in the **add security group Rule** dialog box, configure IP as 10.116.134.123, and access port 8000.

Add scheduling resources

1. Enter the DataWorks management console as a developer, and click **Enter workspace** in the corresponding project action bar.
2. Click **data integration** in the top menu bar to navigate to **resource management > new resource groups**.

3. Click **Next** to **add the purchased ECS** cloud server to the Resource Group in the Add Server dialog box.

Configurations:

- Network Type
 - Classic network: IP addresses are allocated in a unified manner by Alibaba Cloud, featuring easy configuration and convenient use. This network type is suitable for users who require high ease-of-use of operations and need to use ECS quickly.
 - This type refers to logically isolated private networks. Users can customize network topology and IP addresses, and the network supports leased line connections. VPC is suitable for users familiar with network management.
- Server name
 - Alibaba cloud Classic Network: log in to ECS, execute the `hostname` command, and get the return value.
 - Private Network: log in to ECS, execute `dmidecode | grep UUID`, and get the return value.
- Maximum concurrency
 - Count concurrency: The concurrency count calculator is based on the CPU number and memory size.
 - Add Server: The content is related to the network type selected above. If you select classic networks, you can only add classic networks. If you select a proprietary network, the content of the proprietary network type is displayed.



Note:

- When you want to make an ECS in a VPC as the server, you should fill in the ECS UUID as the server name. Logging in to the ECS machine to perform `dmidecode | grep UUID` can be obtained.
- For example, to execute `dmidecode | grep UUID`, the return result is
UUID: 713f4718-8446-4433-a8ec-6b5b62d75a24, the corresponding UUID is
713F4718-8446-4433-A8EC-6B5B62D75A24.

4. Install Agent and initialize.

If you are adding a newly added server, follow these steps.

- a. Log into the ECS server as a root user.

b. Execute the following command:

```
chown admin:admin /opt/taobao
wget https://alisaproxy.shuju.aliyun.com/install.sh --no-check-
certificate
sh install.sh --user_name=xxxxxxxxxx19d --password=yyyyyygh1bm --
enable_uuid=false
```

c. Later on the Add Server Page, click **Refresh** to see if the service status becomes **available**.

d. Provision port 8000 of the server.



Note:

If you do install.sh an error occurred during Sh or a re-execution is required at install.sh the same directory of SH runs `rm -rf install.sh` to delete the files that have been generated. Then execute `install.sh`. The initialization interface above is different for each user's command, please execute the relevant commands according to your own initialization interface.

After doing so, if the service status has been **stopped**, you may encounter the following problems.

The error shown in the preceding figure indicates that no host was bound. To fix the error, follow these steps:

1. Switch to the admin database.
2. Execute `hostname -i` to see how the host is bound.
3. Execute `vim/etc/hosts` and add the IP address and host name.
4. Refresh the page service status if the CS Server registration is successful.



Note:

- If you are still stopping after the refresh, You can restart the alisa command.

Switch to the admin account and execute the following command.

```
/home/admin/alisatasknode/target/alisatasknode/bin/serverctl
restart
```

- If your AK information is involved in the command, please do not expose it to others easily.

2.5 Metadata Collection

2.5.1 Overview of metadata collection

Metadata collection means that metadata is collected periodically into the system, and quickly pull the relevant table and field information through the wizard mode.

You can operate Metadata collection in Data Source Management page **New Collection task** and **Managing Collection tasks** with data source type **No public network IP**.



Note:

- No public network IP database (MySQL, SQL Server, Oracle, PostgreSQL) metadata (database table information, field information) is currently supported only. Especially, Metadata add function is only supported in East China 2.
- Only the project administrators have the read access for the relevant metadata collection entry .
- A data source allows only one metadata collection task.

2.5.2 Metadata Collection

This article will show you how to perform metadata collection.

Add datasources



Note:

- Metadata collection only supports source database type **No public network IP**.
- Due to network restriction, data sources with No public network IP need to run on Customized Resource Groups to pull table and column information, for more information, please refer [Add scheduling resources](#).

1. Log into [DataWorks Management Console](#) as project administrator, click **Enter Project** in corresponding project operation bar.
2. Click **Data Integration** in the top navigation bar, Select **Sync Resource > Data Source**.
3. Click **Add Data Source**, Select data source type **MySQL**.
4. Select data source type as **Has No public network IP** in **Add Data Source MySQL** Dialog Box.

Configuration	Instructions
Data Source Type	Without public network IP.
Data Source Name	Data Source Name can contain letters, numbers, and underscores (_). It must begin with a letter, and cannot exceed 60 characters.
Description	The description of the data source, which must not exceed 80 characters.
Resource Group	Resource Group: It is used to run synchronization tasks, and generally multiple machines can be selected when you add a resource group. For more information, please refer Add scheduling resources .
JDBC URL	JDBC URL: JDBC connection information and its format is <code>jdbc://mysql://serverIP:Port/Database</code> .
User Name	User name for corresponding database.
Password	The password for corresponding database.

5. Click **Finish**.

Create a collection task

1. Click **Add a Collection task** after corresponding source data.
2. Complete relevant configuration information in **Add a Collection task** dialog box.



Note:

If the source group is available, the name of source group, which source data belongs to, will display by default.

Configuration	Instructions
Data source to be collected	The data source has already been added with collection point which cannot be added again, and there would be prompts for related action. The options in the drop-down box are the Has no public network IP data sources that you added.
Resource Group	Automatically display the resource groups name you selected when adding source data. For more information please refer Add scheduling resources .
Table to be collected	Including All Tables and Specify tables , the default selection is All Tables . Edit box will pop-up after selecting the specific table. Multiple tables entry is supported, separated by comma(,).
Collection time	You can choose any exactly hour as collection starting timing, from 00 to 23.

3. Click **Confirm**.

Successfully created collection tasks are displayed in collection task list, this list is mainly used for checking database tables and columns information, you can search related collection list by data source name and owner.

Configuration	Instructions
Data Source Name	Data Source Name need to be consistent with name of the newly added data source.
Node ID	Each task node ID needs to be unique.
Collected Tables	Normally, Node name contains 2 parts, which are automatically generated as `data source name_table` and `data source name_column` respectively.
Owner	Owner by default is the user whom created collection task.
Status	Generally , there are 4 kinds of collection status which are collection failure, collection success, waiting for scheduling, and collecting.
Collection time	The timing which metadata is triggered regularly per day for information collection.
Start Time	Generally ,the default format is <code>yyyy-mm-dd hh:mm:ss</code> .
End time	Generally ,the default format is <code>yyyy-mm-dd hh:mm:ss</code> .
Action	<p>The action of task collection can be divided into two parts.</p> <ul style="list-style-type: none"> DataSource actions <ul style="list-style-type: none"> Modify timing: Modify the trigger timing of the current data source collection task. Delete: Delete current data source collection task. Node operation <ul style="list-style-type: none"> Collect now: Immediately trigger the collection task, update relevant datasource table names and field names. Schedule Operations: Navigate into Operation Center > Cycle instance Page by clicking this button, relevant cycle instance would display filtered by specific conditions. Latest logs: View the corresponding process log.

Configure a synchronization task

When meta collection completed, you can configure corresponding tasks through wizard mode.

For more details, please refer [Creating a Synchronization Task](#).



Note:

- Wizard mode can assist for Synchronization Task configuration, but the task is still running under Custom Resource Group. There may be network connection issue if you choose to run under Default Resource Group instead.
- The relevant table and column information had already stored in the metadata service, so there would not be table information and column information collecting problems cause by network connection issue.

When the synchronization task configuration completed, click **Run**, the task runs immediately. Alternatively, click **Submit** to submit the synchronization task to the scheduling system. The scheduling system periodically runs the task starting from the next day according to the task configurations.

2.6 Full-database migration

2.6.1 Full-database migration overview

This article describes the full-database migration feature in terms of its functions and limits.

Full-database Migration is a convenient tool that can improve user efficiency and reduce user cost. It can quickly upload all the tables in a MySQL database to MaxCompute all at the same time, saving on time that is involved in creating batch tasks for initial data migration to cloud.

For example, if a database contains 100 tables, you are supposed to configure 100 data synchronization tasks in a traditional way. With the full-database migration, you can upload all the tables at the same time. However, because of the design normalization of database tables, this tool cannot guarantee to complete the synchronization of all tables at the same time as per your business demands. In other words, it has limits too.

Task generation rules

After the configuration is completed, MaxCompute tables are created and data synchronization tasks are generated based on the selected tables to be synchronized.

The table names, field names, and field types of the MaxCompute tables are generated according to the advanced settings. If no advanced settings are set, the structure of MaxCompute tables are identical to that of MySQL tables. The partition of these tables is pt, and its format is yyyyymmdd.

The generated data synchronization tasks are cyclic tasks scheduled on a daily basis. They run automatically in the morning on the next day with a transfer rate of 1 MB/s. The actual performance of synchronization tasks varies with the selected synchronization mode and concurrency setting. You can locate the generated tasks by clicking **clone_database > Data**

source name > **mysql2odps_table name** in the directory tree of synchronization tasks to customize them as needed.

**Note:**

We recommend that you perform a smoke test on the data synchronization tasks on the same day. You can find all the synchronization tasks generated by a data source in **project_etl_start** > **Full-database Migration** > **Data Source Name** under **O&M Center** > **Task Management**, and then right-click to test corresponding task nodes.

Limits

Full-database migration is subject to certain limitations due to the design normalization of database tables. The limitations include:

- Currently, only the full-database migration from the Mysql data source to MaxCompute is supported. The migration feature for Hadoop/Hive and Oracle data sources will be available in the future.
- Only the daily incremental and daily full upload modes are available.

If you want to synchronize historical data at a time, this feature cannot meet your needs. The following are a few suggestions for you to consider:

- You can configure daily tasks instead of synchronizing historical data at the same time. You can trace the historical data with the provided data completing, which eliminates the need to run temporary SQL tasks to split data after the historical data is fully synchronized.
- You can configure a task on the task development page and click **Run**. After that, convert data using SQL statements. They are both one-time operations.

If your daily incremental upload is subject to a special business logic and cannot be identified by a date field, this feature cannot meet your needs, and we provide the following suggestions:

- The incremental upload of data can be achieved through binlog (available in the DTS product) or the date field for data change provided by databases.

Currently, Data Integration supports the latter method, and thus your database must contain the date field for data change. The system can determine if your data is changed on the same day as the business date using this field. If yes, all the changed data is synchronized.

- To facilitate incremental upload, we recommend that you include the `gmt_create` and `gmt_modify` fields when creating any database tables. Meanwhile, you can set the `id` field as the primary key to improve efficiency.

- Full-database migration supports batch upload and full upload.

Batch upload is configured with time intervals. Currently, the connection pool protection feature for data sources is not provided, which will be available soon.

- To prevent the database from being overloaded, the full-database migration provides the batch upload mode, which enables you to split tables in batches by a time interval and prevents compromised service functionality because of the database overload. We have two suggestions:
 - If you have master and slave databases, we recommend that you synchronize the data of the slave database.
 - In batch tasks, each table pertains to a database connection with the maximum speed of 1 Mbit/s. If you run the synchronization tasks for 100 tables at the same time, 100 database connections are established. For this reason, make sure to select an appropriate concurrency based on your business conditions.
- If you need a specific task transfer rate, this feature cannot meet your needs. The maximum speed of any generated tasks is 1 Mbit/s.
- Only the mapping of all table names, field names, and field types are supported.

During the full-database migration, MaxCompute tables are created automatically, where the partition field is pt, the field type is string, and the format is yyyyymmdd.

**Note:**

When you select tables for synchronization, all fields must be synchronized and none of these fields can be edited.

2.6.2 Configure MySQL full-database migration

This article demonstrates how to migrate a full MySQL database to MaxCompute with the full-database migration feature.

The full-database migration is a fast tool for improving user efficiency and reducing user usage costs, it can quickly upload all the tables in the MySQL database to MaxCompute, for a detailed introduction to the whole library migration, see [Full-database migration overview](#).

Procedure

1. Log in to [dataworks> Data Integration console](#), click **offline sync > data source** on the left to enter the data source management page.

2. Click **Add-in data** source in the upper-right corner to add a Mysql Data Source library for the whole library migration.
3. After you click **test connectivity** and verify that the data source is accessed correctly, confirm and save the data source.
4. After successful addition, the newly added MySQL data source clone_database is displayed in the data source list. Click the entire library migration that corresponds to the MySQL data source, you can go to the **entire library migration** features page for the corresponding data source.

The whole library migration page mainly has three functional areas.

- Filter area of tables to be migrated: It lists all the database tables under the MySQL data source clone_database. You can select database tables to be migrated in batch.
 - Advanced Settings: It provides the conversion rules of table names, column names, and column types between MySQL and MaxCompute data tables.
 - Control area of the migration mode and concurrency: You can select the full-database migration mode (full or incremental) and the concurrency (batch upload or full upload) and check the progress of submitting the migration task.
5. Click **Advanced Settings** to select conversion rules based on specific requirements. For example, the prefix ods_ was added consistently when the MaxCompute table was built.
 6. In the control area of the migration mode and concurrency, select Daily Incremental as the synchronization mode and set gmt_modified for the incremental field. Data Integration generates a where condition of incremental extraction for each task based on the selected incremental field by default and defines a daily data extraction condition by working with a DataWorks scheduling parameter such as \${bdp.system.bizdate},

Data integration is used to extract data from a MySQL library table to connect to a remote MySQL database by JDBC, and execute the corresponding SQL statement to select the data from the MySQL library. Since it is a standard SQL extraction statement, you can configure the WHERE clause to control the scope of data. Here you can view where conditions for incremental extraction are as follows:

```
STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d') <= gmt_modified AND
gmt_modified < DATE_ADD(STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d'), interval 1 day)
```

To protect the MySQL data source from being overloaded by too many data synchronization jobs started at the same point of time, Batch Upload can be selected. You can set to start synchronizing three database tables every one hour from 00:00 everyday.

Finally, click **Submit task**, where you can see the migration progress information and the status of the migration task for each table.

7. Click the migration task for table a1 to jump to the task development page of Data Integration,

As shown in the preceding figure, the table ods_a1 in MaxCompute corresponding to the source table a1 is created successfully, and the column name and type also match the previously set conversion rules. Under the left-hand directory tree clone_database directory, there will be all of the corresponding whole library migration tasks, the task naming rule is the source table name, as shown in the red box section above.

So far, you have completed migrating the full MySQL data source clone_databae to MaxCompute. These tasks are scheduled to run according to the set scheduling cycle (daily scheduling by default). Also, you can transmit historical data by using the data completing feature of DataWorks. The **data integration > whole library migration** function can greatly reduce the configuration and migration costs of your initial cloud.

The whole library migration A1 table task performs a successful log as shown in the following figure:

2.6.3 Configure Oracle full-database migration

This article demonstrates how to migrate a full Oracle database to MaxCompute by using the full-database migration feature.

The whole library migration is a fast tool for improving user efficiency and reducing user usage costs, it can quickly upload all the tables in the Oracle database to maxcompute, for a detailed introduction to the whole library migration, see [Full-database migration overview](#).

Procedure

1. Log in to the [DataWorks management console](#) and select **data integration** in the top menu bar.
2. Select **offline synchronization > data source** in the left navigation bar and go to the data source management page.
3. Click **Add-in data source** in the upper-right corner to add an Oracle Data Source hub for the whole library migration.
4. After you click **test connectivity** and verify that the data source is accessed correctly, confirm and save the data source.
5. After successful addition, the newly added Oracle data source clone_database is displayed in the data source list. Click the entire library migration that corresponds to the Oracle data

source, you can go to the **entire library migration** features page for the corresponding data source.

The whole library migration page mainly has three functional areas.

- Filter area of tables to be migrated: It lists all the database tables under the Oracle data source clone_database. You can select database tables to be migrated in batch.
- Advanced settings: It provides the conversion rules of table names, column names, and column types between Oracle and MaxCompute data tables.
- Control area of the migration mode and concurrency: You can select the full-database migration mode (full or incremental) and the concurrency (batch upload or full upload) and check the progress of submitting the migration task.

6. Click **Advanced Settings** to select conversion rules based on specific requirements.
7. In the control area of the migration mode and concurrency, select Daily Full as the synchronization mode.

**Note:**

If the date field exists in your table, you can select Daily Incremental as the synchronization mode, and set the incremental field as the date field. Data Integration generates a where condition of incremental extraction for each task based on the selected incremental field by default and defines a daily data extraction condition by working with a DataWorks scheduling parameter such as `#{bdp.system.bizdate}`.

To protect the Oracle data source from being overloaded by too many data synchronization jobs started at the same point of time, Batch Upload can be selected. You can set to start synchronizing three database tables every one hour from 00:00 every day.

Finally, click **Submit task**, where you can see the migration progress information and the status of the migration task for each table.

8. Click the **view task** corresponding to the table to jump to the task Development page for data integration, you can view the run details of the task.

So far, you have completed migrating the full Oracle data source clone_databae to MaxCompute. These tasks are scheduled to run according to the set scheduling cycle (daily scheduling by default). Also, you can transmit historical data by using the data completing feature of DataWorks. The **data integration > whole library migration** function can greatly reduce the configuration and migration costs of your initial cloud.

2.7 Bulk Sync

2.7.1 Bulk Sync

This article will show you how to Bulk Sync.

Bulk Sync is a tool that can help you to improve efficiency and reduce the cost. It allows you to quickly upload all tables in MySQL, Oracle, SQL Server databases to MaxCompute in one time which saves a lot of time on the creation of bulk task for initialization data migration.

**Note:**

Currently Bulk Sync function only supports Shanghai region.

you can flexibility configure table name conversion ,field name conversion, field data type conversion, sink table add-on filed, sink table field value, data filter, sink table name prefix rules, etc. to meet your business requirement.

In **The Data Integration > Sync Resources > Bulk Sync** Page, you can check the cloud migration tasks that you configured.

**Note:**

- **Log** and **View Rules**, under the Actions column in the Bulk Sync list, are only readable rather than modifiable.
- The configuration rule you submitted would invalid if the task does not submit accordingly.

Procedure

1. Select the data source for synchronization.

Select the ready successful added synchronous data source. You can select multiple data sources with the same data source type, for example MySQL, Oracle, or SQL Server. Please refer [Add data sources in Bulk Mode](#).

2. Configure synchronization rules.

Currently, nine configuration rules are supported, and you can select the appropriate rule configuration according to your needs, then you can execute the rule, and check DDLs and synchronous scripts to confirm the effect of the configuration rules.

**Note:**

- If the rules in the interface do not meet your needs, you can try the script mode.

- After configure the rules, you must **Execution rules** and **Submitting tasks**, Otherwise the rules you configure would not be recorded after refreshing or closing the browser.

Action	Configuration	Instructions
Add rule	Target table partition field rules	Show the content of the partition, in accordance with the schedule parameter configuration, see Parameter configuration for the details.
	Table name conversion rules	select any word of your database table name then convert to the content you need.
	Field name conversion rules	Select any word for the name of the field in your table to convert to what you need.
	Type conversion rules	Select data type in your source database then convert into the data tyoe you need.
	The rule of create new field in target table	You can add a column to the MaxCompute table with the name according to your needs.
	The rule of assignment in target table	Assign a value in your newly added filed.
	The rule of Data Filtering	Filter data in the table from the source database you selected.
	The rule of target table name prefix	Add a prefix to the table name.
Convert to script	Configuration rule can convert into script mode configuration. Compared with UI mode, each rule in script mode can be specified with scope of action. However, when the UI mode is converted to script mode, it cannot be converted back to UI configuration mode.	
Reset script	Script can be reset only after converted to script mode. Unified script template will pop up when click this icon.	
Execution rules	Click Execution rules , You can see the effect of the rules on the DDL script and the synchronization script, and this action does not create the task, provides only a preview of the DDL and synchronization scripts. You can select a part of the table to check for the corresponding DDL and synchronization scripts to see if they comply with the rules.	

3. Select the tables to synchronize and commit.

You can select multiple tables for bulk commit, and the MaxCompute table will be created based on the above configuration rules. If the execution fails, you can place mouse over the execution result and system will prompt a hint for the cause of failure.

Configuration	Instructions
DDL	After you click, you can only view the related table creation statements, rather than modify them.
Sync configuration	Click Sync configuration to view the tasks that you configured, which are displayed in script mode.
View table	Navigate to the appropriate data management console page, where you can view the create details of MaxCompute table.

4. View tasks.

After task submitted successfully, you can enter **Data Development > Business Processes** Page to view your bulk cloud migration task.

The number of business process is same as the number of source database you selected.

The general naming rule is clone_database _ `data source name`. Each table generates a synchronization task, and the naming rule is the `data source name`2odps_`table name`.

- a. Task configuration: synchronize the MySQL, generated by bulk cloud migration, to odps synchronize task, and the data filter condition is generated by The rule of Data Filtering.
- b. Field mapping: The mapping target field output is based on the relevant field rule, you can view the output depends on your configuration rule.
- c. Tunnel Configuration: You can configure synchronization task DMU, job concurrency, number of error records in Tunnel Configuration. This configuration is closely related to the running speed of the task.

**Note:**

Please refer to [Configure Reader plug-in](#) and [Configure writer plug-in](#) for instruction of task configuration.

5. Run the task.

Click **Run** , the synchronization task will run immediately. Alternatively, You can submit the synchronization task to the scheduling system by click **Submit**. The scheduling system periodically runs the task according to the task configurations starting from the second day. For more detail please refer to [Scheduling Configuration](#).

**Note:**

- **Simple Mode:** Task takes effect in production environment directly after submission .

- **Standard Mode:** Task is submitted into development environment, then publish to the production environment.

2.7.2 Add data sources in Bulk Mode


This article will show you how to add data sources in Bulk Mode.



Note:

- Fast cloud currently only supports three types of data sources: MySQL, Oracle, and SQL Server.
- Add data sources in Bulk Mode is currently only available for Data Source Type **Has Public Network IP**.
- After adding MySQL and Oracle, SQL Server data sources , **Batch testing connectivity** are required. Only when **Connected State** is **Success**, the specific bulk data source would be an available data source option for **Bulk Sync**.

1. Log in to the [DataWorks Console](#) as Project Administrator.
2. Click **The Data Integration** in a specific Workspace.
3. In **Data Integration > Sync Resource > Data Source** Page, click **Add Data Source**.
4. In **Add Data Source** window, you can select **MySQL, Oracle** , or **SQL Server**.

Configuration	Instructions
Data Source Type	Select Has Public Network IP .
Configuration	Select Bulk Mode .
The script upload	Click Template to download the template file, input your data source name, data source description, link address, user name, and password into the downloaded template file. <div>  Note: In general, there is an existing data source, you can just delete and add your own data source information. </div>
Select a file	Click Select a file to choose an existing template in local.
Start new	After file uploaded successfully, click Start new , the information of data source uploading will display in the text box, such as the number of successes, the number of failures, cause of the failures, etc.

5. Click **Finish** when uploading process completes.
6. In **Data Source** Page, select specific data source, click **Bulk testing connectivity**.

**Note:**

Only if the Connected Status of the data source is **Success**, you can operate the Bulk Sync.

7. Select the data sources that you want to upload then click **Bulk Sync**.

2.8 Best practice

2.8.1 Data integration when the network of data source (one side only) is disconnected

This article demonstrates how to migrate a full MySQL database to MaxCompute with the full-database migration feature.

Scenario

Complex network environments are characteristic of the following two conditions.

- Either the data source or the data target is in the private network environment.
 - VPC environment (except the RDS) <-> Public network environment
 - Financial Cloud environment <-> Public network environment
 - Local user-created environment without the public network <-> Public network environment
- Both the data source and target are in the private network environment.
 - VPC environment (except the RDS) <-> VPC environment (except the RDS)
 - Financial Cloud environment <-> Financial Cloud environment
 - Local user-created environment without the public network <-> Local user-created environment without the public network
 - Local user-created environment without the public network <-> VPC environment (except the RDS)
 - Local user-created environment without the public network <-> Financial Cloud environment

Data Integration provides the network penetration ability in the complex network environments. By deploying Data Integration agents, synchronous data transmission can be implemented between any network environments. The following describes the specific implementation logics and procedures and assumes that the network of both ends of data sources cannot be connected.

Implementation logics

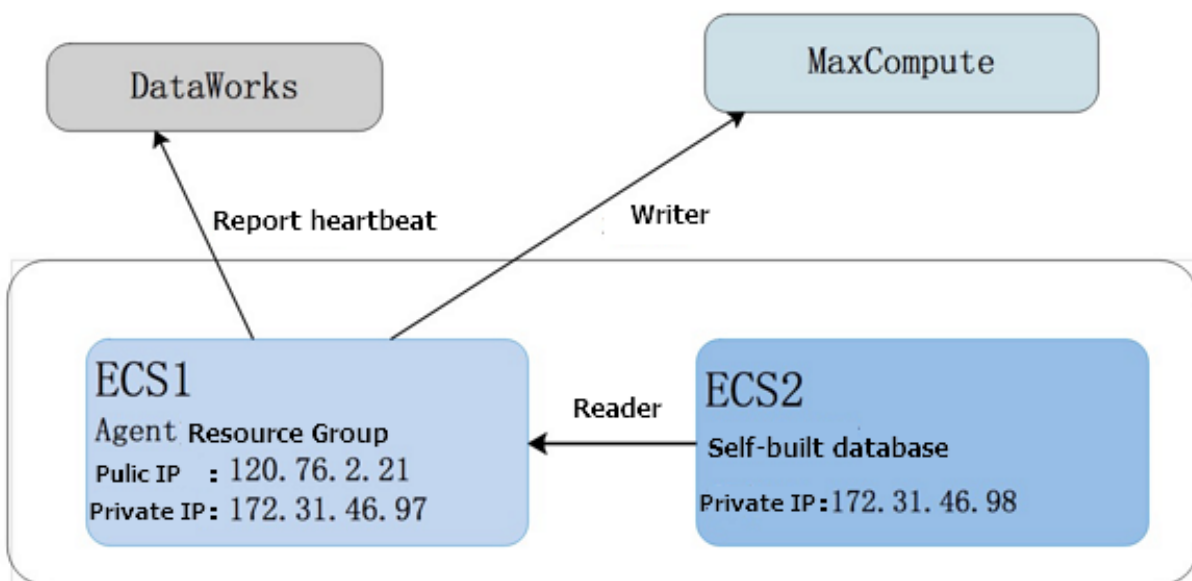
For the complex network environments where either the data source or the data target is in the private network environment, deploy the Data Integration agent on the machine in the same

network environment as that of the end which is in the private environment and connect to the external public network through the agent. Private network environments are characteristic of the following two conditions:

- The database built on ECS is purchased with no public IP address or elastic public IP address assigned.
- Type: Data source without a public IP address.

ECS

The data synchronization method in this scenario is shown in the following figure:



- Because ECS2 server cannot access the public network, an ECS1 machine that is in the same network segment as ECS2 and has the ability to access the public network is required for agent deployment.
- Set ECS1 as the resource group, and run the synchronization task on the machine.



Note:

You need to grant database permissions to the ECS2 server to access relevant database and read the data of the database to ECS1. The command for granting permissions is as follows:

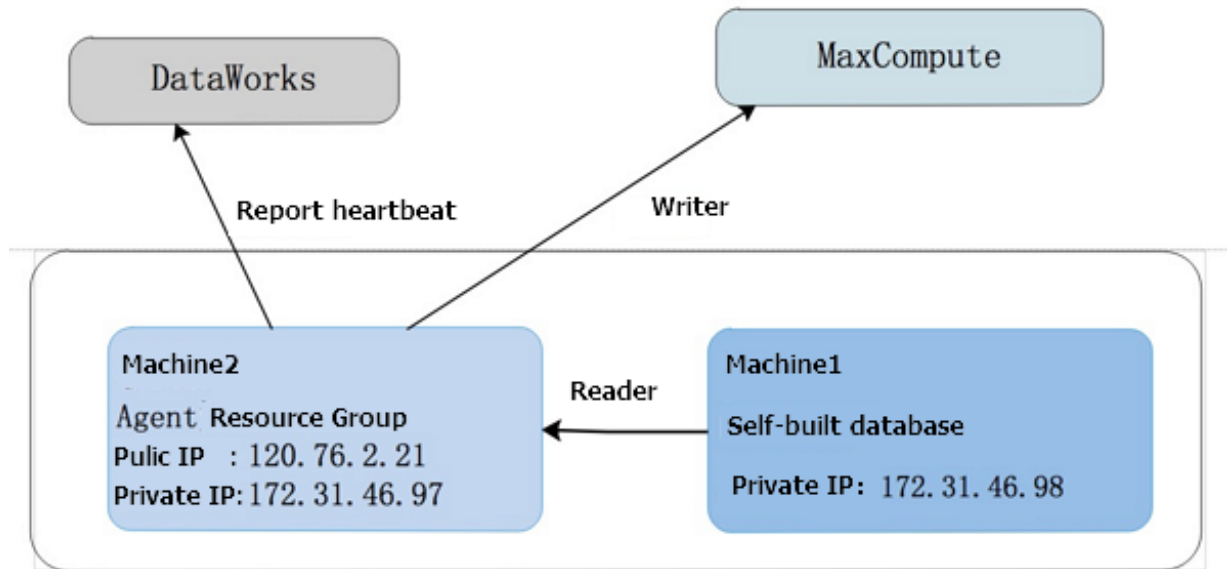
```
grant all privileges on *.* to 'demo_test'@'%' identified by ''
Password'; --> % means granting permissions to any IP addresses<br>.
```

The user-created data source synchronization task on ECS2 runs in the custom resource group. To authorize the machine of the custom resource group, you must add internal and external IP

address and the port of ECS2 to the safety group of ECS1. See [Add security group](#) for more information.

Local IDC with no public IP address

The data synchronization method in this scenario is shown in the following figure:



- Because machine 1 cannot access the public network, an machine 2 that is in the same network segment as machine 1 and has the ability to access the public network is required for agent deployment.
- Set machine 2 as the scheduling resource group, and run the synchronization task on the machine.

Procedure

Configure the Data Source

1. Enter the [DataWorks management console](#) as a developer, and click Enter workspace in the corresponding project action bar.
2. Click Data Integration from the top menu bar and navigate to the Data Source page.
3. Click Add Data Source to show the supported data source types.
4. Select the data source without a public IP address from the data sources for the relational database MySQL.
 - Source data source (with no public IP).

The configuration items are as follows:

- Data source type: data source without a public IP address.

- Data source name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- Resource group: The machine on which the target agent is deployed to connect to the external public network. The synchronization task of data source in special network environment can run in the resource group. To add source group, see [Add Scheduling Resources](#). For more information on adding resource groups, see [Add scheduling resources](#).
- JDBC URL: the JDBC URL. Format: jdbc:mysql://ServerIP:Port/database.
- User name/Password: The user name and password used to connect to the database.
- Test Connectivity: the data source for public network IP does not support test connectivity, just click **Finish**.
- Target data source (with a public network).

Parameters:

- Data source name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- ODPS endpoint: defaults to read-only. The value is automatically read from the system configuration.
- ODPS project name: the corresponding MaxCompute project indicator.
- Access ID: the Access ID corresponding to the MaxCompute project owner's cloud account.
- Access Key: The Access Key of the MaxCompute Project Owner cloud account, used in combination with the Access ID. The access key is equivalent to the logon password.
- Connectivity test: the connectivity test is supported.

Configure a synchronization task

1. Select the source.

Select the source. Because the data source has no public IP, the network of the data source is unavailable. You must run the synchronization task in the script mode. Click Switch Script button directly.

2. Import a template.

Parameter description:

- Source type: The data source name is automatically selected base on the data source selected in the wizard mode.
- Target type: You can select a target data source from the drop-down list.



Note:

If adding data sources on the page is supported by the database, you can select data sources from the template. If not, you must edit relevant data source information in JSON code section of the template and then click Add Data Source directly.

3. An example of how to switch into the script mode.

Configure the resource groups: You can change and view the resource groups for the synchronization task. Collapsed by default.

```
{
  "type": "job",
  "configuration": {
    "setting": {
      "speed": {
        "concurrent": "1", //Number of concurrent tasks
        "mbps": "1" //Maximum task speed
      },
      "errorLimit": {
        "record": "0" //Maximum number of error records
      }
    },
    "reader": {
      "parameter": {
        "Splitpk": "ID", // cut key
        "column": [ //Target column name
          "name",
          "tag",
          "age",
          "balance",
          "gender",
          "birthday"
        ],
        "table": "source", // source name
        "where": "ds = '20171218'", // filter criteria
        "datasource": "private_source" //Data source name, which must be
        consistent with the name of the added data source
      },
      "plugin": "mysql"
    },
    "writer": {
```

```

    "parameter": {
      "partition": "pt=${bdp.system.bizdate}", //The partition
information.
      "truncate": true,
      "column": [//Target column name
        "name",
        "tag",
        "age",
        "balance",
        "gender",
        "birthday"
      ],
      "table": "random_generated_data", //Table name of the target end
      "datasource": "odps_mrtest2222" //Data source name, which must
be consistent with the name of the added data source
    },
    "plugin": "odps"
  }
},
"version": "1.0"
}

```

Run a synchronization task

You can run the synchronization task in the following methods:

- Click Run in the page of the Data Integration.
- Schedule the task. For the configuration of related scheduling, see [scheduling configuration](#).

2.8.2 Data sync when the network of data source (both sides) is disconnected

Scenario

Complex network environments are characteristic of the following two conditions.

- Either the data source or the data target is in the private network environment.
 - VPC environment (except the RDS) <-> Public network environment
 - Financial Cloud environment <-> Public network environment
 - Local user-created environment without the public network <-> Public network environment
- Both the data source and target are in the private network environment.
 - VPC environment (except the RDS) <-> VPC environment (except the RDS)
 - Financial Cloud environment <-> Financial Cloud environment
 - Local user-created environment without the public network <-> Local user-created environment without the public network
 - Local user-created environment without the public network <-> VPC environment (except the RDS)
 - Local user-created environment without the public network <-> Financial Cloud environment

Data Integration provides the network penetration ability in the complex network environments. By deploying Data Integration agents, synchronous data transmission can be implemented between any network environments. The following describes the specific implementation logics and procedures and assumes that the network of both ends of data sources cannot be connected. For the scenarios where only one end is unreachable, see [Data sync when the network of data source \(both sides\) is disconnected](#).

Implementation logics

For the complex network environments where both ends of data sources are in the private network environment, deploy the Data Integration agent for the both ends under the same network environment, where the source agent is for pushing data to the Data Integration server and the target agent is for pulling the data to the local device. During data transmission, the transmission timeliness and security are ensured by data blocking, compression, and encryption.

Procedure

Configure the Data Source

1. Log on to the [DataWorks console](#) as a developer and click Enter Project to enter the project management page.
2. Click **Data Integration** from the upper menu and navigate to the Offline Sync > Data Sources page.
3. Click **New Source** to show the supported data source types.
4. Select the data source without a public IP address from the FTP data sources.

Add a source data source.

Configuration item description:

- Type: Data source without a public IP address.
- Name: It is a combination of letters, numbers, and underscores (.). It must begin with a letter or an underscore (.) and cannot exceed 60 characters.
- Description: It is a brief description of the data source up to 80 characters.
- Select resources group: It is the machine on which the agent is deployed. The source agent is for pushing data to the Data Integration server. To add source group, see [Add scheduling resources](#).
- Protocol: ftp or sftp.
- *Host: The default ftp port is port 21 and the default sftp port is port 22.
- Username/Password: The username and password used to connect to the database.

- **Test Connectivity:** Data sources with public IP addresses do not support connectivity tests. Click **Finish** to complete the source-end configuration.

Add a target data source

Resource group: The machine on which the target agent is deployed. The target agent is for pulling data to the local device. To add source group, see [Add scheduling resources](#).

Select the script mode

1. Click **Data Integration** from the upper menu, and go to **Sync Tasks** page.
2. Choose **New > Script Mode** on the page.

On the script mode page, select an appropriate template that contains key parameters of synchronization tasks, and enter the required information. Note that the script mode cannot be switched to the wizard mode.

3. Select the **ftp-to-ftp** import template.

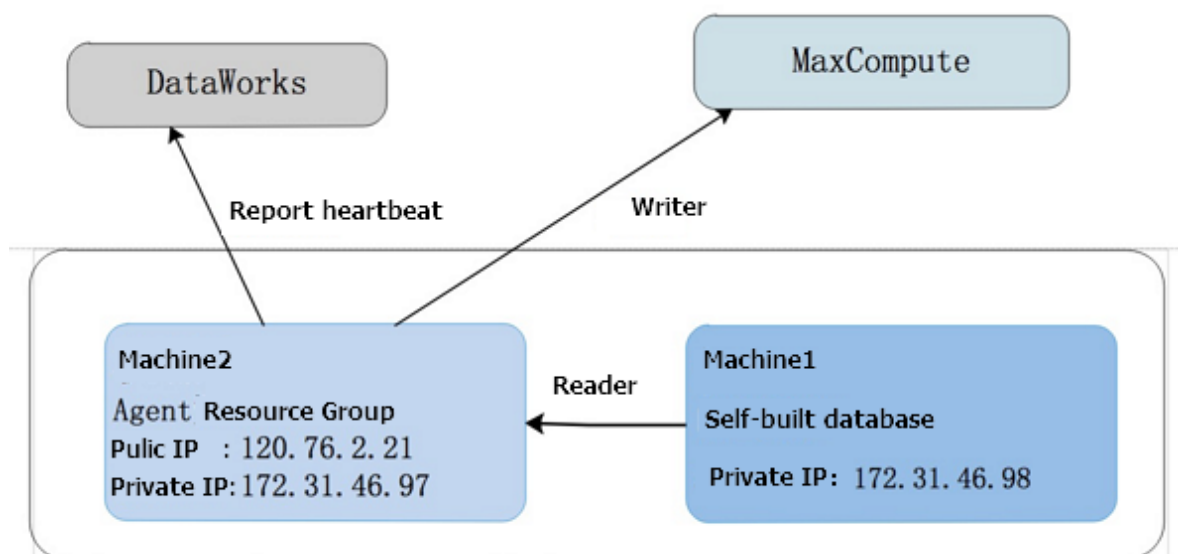
- **Source type:** The data source name is automatically selected base on the data source selected in the wizard mode.
- **Target type:** You can select a target data source from the drop-down list.



Note:

If adding data sources on the page is supported by the database, you can select data sources from the template. If not, you must edit relevant data source information in JSON code section of the template and then click **Add Data Source** directly.

4. Configure a synchronization task.



- Because machine 1 cannot access the public network, an machine 2 that is in the same network segment as machine 1 and has the ability to access the public network is required for agent deployment.
- Set machine 2 as the scheduling resource group, and run the synchronization task on the machine.

Procedure

Configure the Data Source

1. Enter the [DataWorks management console](#) as a developer, and click Enter workspace in the corresponding project action bar.
2. Click Data Integration from the top menu bar and navigate to the Data Source page.
3. Click Add Data Source to show the supported data source types.
4. Select the data source without a public IP address from the data sources for the relational database MySQL.
 - Source data source (with no public IP).

The configuration items are as follows:

- Data source type: data source without a public IP address.
- Data source name: It is a combination of letters, numbers, and underlines It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- Resource group: The machine on which the target agent is deployed to connect to the external public network. The synchronization task of data source in special network environment can run in the resource group. To add source group, see Add Scheduling Resources. For more information on adding resource groups, see [Add scheduling resources](#).
- JDBC URL: the JDBC URL. Format: jdbc:mysql://ServerIP:Port/Database.
- User name/Password: The user name and password used to connect to the database.
- Test Connectivity: the data source for public network IP does not support test connectivity, just click **Finish**.
- Target data source (with a public network).

Parameters:

- Data source name: It is a combination of letters, numbers, and underlines. It must begin with a letter or underline and cannot exceed 60 characters.
- Data source description: It is a brief description of the data source with no more than 80 characters.
- ODPS endpoint: defaults to read-only. The value is automatically read from the system configuration.
- ODPS project name: the corresponding MaxCompute project indicator.
- Access Id: the Access ID corresponding to the MaxCompute project owner's cloud account.
- Access Key: The Access Key of the MaxCompute Project Owner cloud account, used in combination with the Access ID. The access key is equivalent to the logon password.
- Connectivity test: the connectivity test is supported.

Configure a synchronization task

1. Select the source.

Because the data source has no public IP, the network of the data source is unavailable. You must run the synchronization task in the script mode. Click Switch Script button directly.

2. Import a template.

Parameter description:

- Source type: The data source name is automatically selected based on the data source selected in the wizard mode.
- Target type: You can select a target data source from the drop-down list.



Note:

If adding data sources on the page is supported by the database, you can select data sources from the template. If not, you must edit relevant data source information in JSON code section of the template and then click Add Data Source directly.

3. An example of how to switch into the script mode.

Configure the resource groups: You can change and view the resource groups for the synchronization task. The default source and target groups are the resource groups that you selected when adding the data source.

```
{  
  "configuration": {
```

```

"setting": {
  "speed": {
    "concurrent": "1", //Number of concurrent tasks
    "mbps": "1" //Maximum task speed
  },
  "errorLimit": {
    "record": "0" //Maximum number of error records
  }
},
"reader": {
  "parameter": {
    "fieldDelimiter": ",", //Delimiter
    "encoding": "UTF-8", //Encoding format
    "column": //Data source column
    {
      "index": 0,
      "type": "string",
    },
    {
      "index": 1,
      "type": "string",
    }
  ],
  "path": //File path
    "/home/wb-zww354475/ww.txt"
  ],
  "datasource": "lzz_test3" //Data source name, which must be
consistent with the name of the added data source
},
"plugin": "ftp"
},
"writer": {
  "parameter": {
    "writeMode": "truncate", //Writing mode
    "fieldDelimiter": ",", //Delimiter
    "fileName": "ww", //File name
    "path": "/home/wb-zww354475/ww_test", //File path
    "dateFormat": "yyyy-MM-dd HH:mm:ss",
    "datasource": "lzz_test4", //Data source name, which must be
consistent with the name of the added data source
    "fileFormat": "csv" //File type
  },
  "plugin": "ftp"
}
},
"Type": "job ",
"version": "1.0"
}

```

Run a synchronization task

You can run the synchronization task in the following methods:

- Click Run in the page of the Data Integration.
- Schedule the task. For the configuration of related scheduling, see [scheduling configuration](#).

2.8.3 Data increase synchronization

The two types of data to be synchronized

Based on whether the data is changed after being written, the data to be synchronized is classified as unchanged data (generally log data) and changed data (such as the personnel table where the personnel status may change).

Example

You must specify different synchronization policies for each data. The following example shows how to synchronize the data of the RDS database to MaxCompute, which also applies to other data sources.

According to the idempotence (multiple operations of tasks produce the same result. In this way, the task supports re-running scheduling and can easily clear dirty data when an error occurs), data is imported to a separate table or partition, or directly overwrites the historical data in the existing table or partition.

In the example, the task test date is 11/14/2016, full synchronization is performed on the same day, and historical data is synchronized to the partition where ds=20161113. For the incremental synchronization scenario in this example, automatic scheduling is configured to synchronize the incremental data to the partition where ds=20161114 on November 15, 2016. There is a time field optime indicating the modified time of the data, which is used to determine whether the data is incremental or not.

Incremental synchronization of unchanged data

This scenario allows you to partition easily based on the data generation pattern because the data remains unchanged after being generated. Typically, you can partition by date, such as creating one partition on a daily basis.

Data preparation

```
drop table if exists oplog;
create table if not exists oplog(
  optime DATETIME,
  uname varchar(50),
  action varchar(50),
  status varchar(10)
);
Insert into oplog values(str_to_date('2016-11-11','%Y-%m-%d'),'LiLei',
', 'SELECT', 'SUCCESS');
```

```
Insert into oplog values ("2016-11-12 ', '% Y-% m-% d''),' hanmm ', '
desc ', "success ');
```

The two data entries as the historical data are available. Perform full data synchronization first to synchronize the historical data to the partition created yesterday.

Procedure

1. Create a MaxCompute table.

```
Create a good maxcompute table and partition by day
create table if not exists ods_oplog(
  optime datetime,
  uname string,
  action string,
  status string
) partitioned by (ds string);
```

2. Configure a task to synchronize the historical data.

Given that the task is performed only once, only one test is required. After the test is complete, change the status of the task to Paused (in the rightmost scheduling configuration) and submit and release the task again in the “Data Development” module to prevent the task from being scheduled automatically.

3. Write more data to the RDS source table as the incremental data.

```
Insert into oplog values (current_date, "Jim", "Update", "success
');
insert into oplog values(CURRENT_DATE,'Kate','Delete','Failed');
insert into oplog values(CURRENT_DATE,'Lily','Drop','Failed');
```

4. Configure a task to synchronize the incremental data.



Note:

If you configure the “Data Filtering”, all the data added to the source table on November 14 is retrieved and synchronized to the incremental partition in the target table during the synchronization on the early morning the next day, which is November 15.

5. View synchronization results.

If you set the task scheduling cycle as daily scheduling, the task is scheduled automatically the next day after the task is submitted and released, and the data in the MaxCompute target table is changed as follows once the task runs successfully.

Incremental synchronization of changed data

For data in personnel or order tables that is subject to changes, full data synchronization on a daily basis is recommended based on the time variant collection feature of the data warehouse. In

other words, you store full data on a daily basis. In this way, both historical and current data can be retrieved easily.

In actual scenarios, daily incremental synchronization may be required. Because MaxCompute does not support changing data with the Update statement, you must take other measures to implement the synchronization. The following describes how to implement full and incremental synchronization.

Data preparation

```
drop table if exists user ;
create table if not exists user(
    uid int,
    uname varchar(50),
    deptno int,
    gender VARCHAR(1),
    optime DATETIME
);
-- Historical data
insert into user values (1,'LiLei',100,'M',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (2,'HanMM',null,'F',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (3,'Jim',102,'M',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (4,'Kate',103,'F',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (5,'Lily',104,'F',str_to_date('2016-11-11','%Y-%m-%d'));
Incremental data
update user set deptno=101,optime=CURRENT_TIME where uid = 2; --
Change null to non-null
update user set deptno=104,optime=CURRENT_TIME where uid = 3; --
Change non-null to non-null
update user set deptno=104,optime=CURRENT_TIME where uid = 4; --
Change non-null to null
delete from user where uid = 5;
insert into user(uid,uname,deptno,gender,optime) values (6,'Lucy',105,'F',CURRENT_TIME);
```

Daily full synchronization

1. Create a MaxCompute table

Daily full synchronization is relatively simple.

```
create table ods_user_full(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    Optime datetime
) partitioned by (ds string);
```

2. Configure full synchronization tasks.

**Note:**

Set the scheduling cycle of the task as daily scheduling because daily full synchronization is required.

3. Test the task and view the synchronized MaxCompute target table.

Because full synchronization is performed on a daily basis and no incremental synchronization is performed in this case, you can see the following data results after the task is automatically scheduled on the next day.

To query the data results, set `where ds = '20161114'` to retrieve the full data.

Daily incremental synchronization

This mode is not recommended except in specific scenarios. Because the delete statement is not supported in specific scenarios, deleted data cannot be retrieved by filtering conditions of SQL statements. Generally, enterprises' codes are deleted logically, in which case the update statement is applied instead of the delete statement. Now that there are some inapplicable scenarios, using this sync method may cause data inconsistency when some special condition is encountered. Another drawback is that you must merge new data and historical data after the synchronization.

Data preparation

Create two tables, one of which is for writing latest data and the other is for writing incremental data.

```
-- Result table
create table dw_user_inc(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
);
-- Incremental record
create table ods_user_inc(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
)
```

1. Configure a task to write full data directly to the result table.**Note:**

Note: Run this task only once and set the task as Paused in the **Data Development** module after the task runs successfully.

2. Configure a task to write incremental data to the incremental record.
3. Merge the data.

```
insert overwrite table dw_user_inc
select
case when b.uid is not null then b.uid else a.uid end as uid,
Case when B. uid is not null then B. uname else A. uname end as
uname,
case when b.uid is not null then b.deptno else a.deptno end as
deptno,
case when b.uid is not null then b.gender else a.gender end as
gender,
case when b.uid is not null then b.optime else a.optime end as
optime
from
dw_user_inc a
full outer join ods_user_inc b
on a.uid = b.uid ;
```

as you can see in the preceding figure, the deleted data entries are not synchronized.

The daily incremental synchronization is different from the daily full synchronization in that the daily incremental synchronization synchronize only a small amount of incremental data, but with the risk of data inconsistency, and requires extra computing workload for data merging.

If not necessary, change the amount of data that is synchronized throughout the day. In addition , you can set a Lifecycle for the historical data, which can be deleted automatically after a certain period.

2.8.4 Import data into Elasticsearch using Data Integration

This topic describes how to offline import data into Elasticsearch by using Data Integration.

[Data Integration](#) is a data synchronization platform provided by Alibaba Group. Data Integration is a reliable, secure, cost-effective, elastic, scalable data synchronization platform. Data Integration can be used across heterogeneous data storage systems and provides offline (full/incremental) data synchronization channels in different network environments for more than 20 types of data sources. For more information about data source types, see [Supported data sources](#).

Prerequisites

Before importing data using Data Integration, you must:

- [Prepare Alibaba Cloud account](#) Sign up for an Alibaba Cloud account and create AccessKeys for this account.
- Activate MaxCompute, and then a default MaxCompute data source is automatically created.

- [Create a project](#) with the Alibaba Cloud account.

To use DataWorks, first create a project. Then, you can complete the workflow and maintain data and tasks through collaboration within the project.

**Note:**

You can grant RAM users the permissions to create Data Integration tasks. For more information, see [Create a sub-account](#) and [Member management](#).

- Configure data sources. For more information, see [Data source config](#).

Procedure

1. Log on to the [DataWorks console](#) as a developer, find the project, and then click **Data Integration**.
2. Right click **Business Flow** and select **Create Business Flow**.
3. Right click **Data Integration** under the created business flow and choose **Create Data IntegrationNode ID > Data Sync**.
4. Set up configurations in the **Create Node** dialog box and click **Submit**.

Configuration	Description
Node Type	Defaults to Data Sync.
Node Name	The name of the node.
Destination folder	The node is located in the corresponding process by default.

5. Click **Switch to Script Mode** in the navigation bar and click **Ok**.
6. Click **Import Template** in the toolbar and set up configurations in the **Import Template** dialog box.

Configuration	Description
Source Type	In this example, select MySQL .
Data Source	Select a configured data source.
Destination Type	In this example, select Elasticsearch .

7. Click **Ok** to generate an initial script and set up configurations as needed.

```
{
  "configuration": {
    "setting": {
      "speed": {
        "concurrent": "1", //Number of concurrent jobs
        "mbps": "1" //Maximum transmission rate
      }
    }
  },
}
```

```

"reader": {
  "parameter": {
    "connection": [
      {
        "table": [
          "`es_table`" //Source table name
        ],
        "datasource": "px_mysql_OK" //Data source name. We recommend
        you use the same data source name as the one you added.
      }
    ],
    "column": [ //Column names in the source table
      "col_ip",
      "col_double",
      "col_long",
      "col_integer",
      "col_keyword",
      "col_text",
      "col_geo_point",
      "col_date"
    ],
    "where": "", //Filtering condition
  },
  "plugin": "mysql"
},
"writer": {
  "parameter": {
    "cleanup": true, //Whether to clear the original data when
    importing the data to Elasticsearch each time. Set to true when
    performing full import or when rebuilding indexes. Set to false when
    synchronizing incremental data. For the data synchronization in
    this example, set it to false.
    "accessKey": "nimda", //In this example, the password is
    required because the X-Pack plugin is used. If the plugin is not
    used, set it to an empty string.
    "index": "datax_test", //Index name of Elasticsearch. If it is
    unavailable, the plugin will create one automatically.
    "alias": "test-1-alias", //The alias to which the data is
    written after the data is imported.
    "settings": {
      "index": {
        "number_of_replicas": 0,
        "number_of_shards": 1
      }
    },
    "batchSize": 1000, //The number of data entries per batch.
    "accessId": "default", //If the X-PACK plug-in is used, enter
    the username here, and if not, enter an empty string. Because the X-
    PACK plug-in is used for Alibaba Cloud Elasticsearch, a username is
    required here.
    "endpoint": "http://example.com:port", //The address to
    Elasticsearch, which can be found on the console.
    "splitter": ",", //Specify a delimiter if arrays are inserted.
    "indexType": "default", //The type name under the corresponding
    index in Elasticsearch.
    "aliasMode": "append", //The mode of adding an alias after the
    data is imported: append and exclusive.
    "column": [ //Column names in Elasticsearch, whose order is the
    same as that of columns in Reader.
      {
        "name": "col_ip", //Corresponds to the property column "name
        " in TableStore.
        "type": "ip" //Text type, the default analyzer is used.
      },

```

```

    {
      "name": "col_double",
      "type": "string",
    },
    {
      "name": "col_long",
      "type": "long"
    },
    {
      "name": "col_integer",
      "type": "integer"
    },
    {
      "name": "col_keyword",
      "type": "keyword"
    },
    {
      "name": "col_text ",
      "type": "text"
    },
    {
      "name": "col_geo_point",
      "type": "geo_point"
    },
    {
      "name": "col_date ",
      "type": "date"
    }
  ],
  "discovery": false//Set to true to enable automatic discovery.
},
"plugin": "elasticsearch">//Name of the Writer plugin: ElasticSearchWriter, leave it as the default.
},
"type": "job",
"version": "1.0"
}

```

8. Click **Save** and **Run**.



Note:

- Elasticsearch only supports importing data in script mode.
- If you want to use a new template, click **Import Template** in the toolbar. The existing content is overwritten once the script is reset.
- After saving the synchronization task, click **Run** to immediately run the task. Alternatively, click **Submit** to submit the synchronization task to the scheduling system. The scheduling system periodically runs the task starting from the next day according to the task configurations.

Reference

For more information about how to configure synchronization tasks, see the following documents.

- [Configure the Reader plug-in.](#)
- [Configure the Writer plug-in.](#)

2.8.5 Use Data Integration to ship log data collected by LogHub

This topic describes how to use Data Integration to ship data collected by LogHub to supported destinations, such as MaxCompute, Object Storage Service (OSS), Table Store, relational database management systems (RDBMSs), and DataHub. In this topic, we use MaxCompute as an example.



Note:

This feature is available in the China (Beijing), China (Shanghai), China (Shenzhen), Hong Kong, US (Silicon Valley), Singapore, Germany (Frankfurt), Australia (Sydney), Malaysia (Kuala Lumpur), Japan (Tokyo), India (Mumbai) regions.

Scenarios

- Synchronize data across regions between different types of data sources, such as LogHub and MaxCompute data sources.
- Synchronize data using different Alibaba Cloud accounts between different types of data sources, such as LogHub and MaxCompute data sources.
- Synchronize data using one Alibaba Cloud account between different types of data sources, such as LogHub and MaxCompute data sources.
- Synchronize data using a public cloud account and an Alibaba Finance Cloud account between different types of data sources, such as LogHub and MaxCompute data sources.

Note on cross-account data synchronization

If you want to create a Data Integration task using account B to synchronize LogHub data under account A to MaxCompute data source under account B.

1. Create a LogHub data source with the Access Id and the Access Key of account A.

Account B has the permissions to access all Log Service projects created by account A.

2. Create a LogHub data source with the Access Id and the Access Key of RAM user A1.

- Use Alibaba Cloud account A to grant pre-defined Log Service permissions (`AliyunLogFullAccess` and `AliyunLogReadOnlyAccess`) to RAM user A1. For more information, see [Grant RAM subaccounts permissions to access Log Service](#).
- Use Alibaba Cloud account A to assign custom Log Service permissions to RAM user A1.

Choose **RAM console** > **Policies** and choose **Custom Policy** > **Create Authorization Policy** > **Blank Template**.

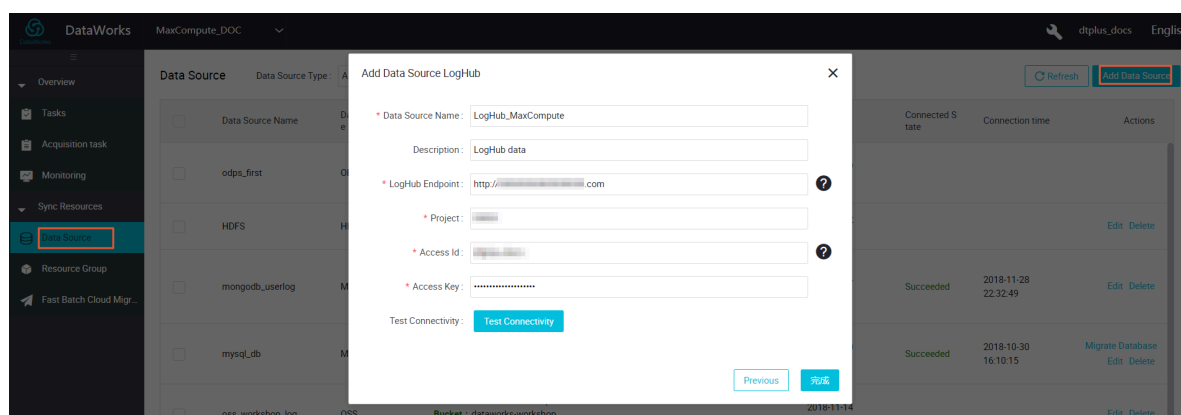
For more information about authorization, see [Access control RAM](#) and [RAM subaccount access](#).

If the following policy is applied to RAM user A1, account B can only read project_name1 and project_name2 data in Log Service through RAM user A1.

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:Get*",
        "log:List*",
        "log:CreateConsumerGroup",
        "log:UpdateConsumerGroup",
        "log:DeleteConsumerGroup",
        "log:ListConsumerGroup",
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ConsumerGroupHeartBeat",
        "log:GetConsumerGroupCheckPoint"
      ],
      "Resource": [
        "acs:log:*:*:project/project_name1",
        "acs:log:*:*:project/project_name1/*",
        "acs:log:*:*:project/project_name2",
        "acs:log:*:*:project/project_name2/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Add a data source

1. Log on to the [DataWorks console](#) as a developer with account B or a RAM user of account B, find the project, and then click **Data Integration**.
2. Choose **Sync Resources** > **Data Source** and click **Add Data Source** in the upper-right corner.
3. Select **LogHub** as the data source type, and then configure the data source in the **Add Data Source LogHub** dialog box.



Configuration	Description
Data Source Name	Can contain letters, numbers, and underscores (_). It must begin with a letter, and cannot exceed 60 characters in length.
Description	The description of the data source, which must not exceed 80 characters in length.
LogHub Endpoint	The endpoint of the LogHub data source in the format of http://example.com.
Project	For more information, see Service endpoints .
Access Id and Access Key	The logon credential, similar to the account name and the password. You may enter the Access Id and the Access Key of an Alibaba Cloud account or a RAM user account.

4. Click **Test Connectivity**.
5. When the connection test is passed, click **OK**.

Configure a synchronization task in wizard mode

1. Choose **Business Flow > Data Integration** and click **Create Integration Node** in the upper-left corner.
2. Set up configurations in the **Create Node** dialog box and click **Submit**. Then, the configuration page of the data synchronization task appears.
3. Select a source.

01 Data Source

Source

The data sources can be default data sources or data source

* Data Source :

LogHub

LogHub_MaxCompute

?

* Logstore :

Please select

* Start Time :

\${startTime}

?

* End Time :

\${endTime}

?

Number of Records :

256

?

Read Per Batch

Preview

Configuration	Description
Data source	Select LogHub and enter the LogHub data source name.
Logstore	The name of the table from which incremental data is exported. You must enable the Stream feature on the table when creating the table or using the UpdateTable operation after the creation.
Start Time	The start (included) of the selected time range for filtering log entries by log time. The format is yyyyMMddHHmmss. For example, 20180111013000. These parameters correspond to the scheduling time of DataWorks tasks.
End Time	The end (excluded) of the selected time range for filtering log entries by log time. The format is yyyyMMddHHmmss. For example, 20180111013000. These parameters correspond to the scheduling time of DataWorks tasks.
Number of Records Read Per Batch	Number of data entries read each time. The default value is 256.

You can click the Data preview button to preview the data .



Note:

Data Preview allows you to view a small number of LogHub data entries in a preview box, which may be different from the data that you synchronize. The data that you synchronize is determined by the Start Time and End Time.

4. Select a destination.

Select a MaxCompute destination and select a table. In this example, select the ok table.

Destination
Hide

ed by you. Click [here](#) to check the supported data source types.

* Data Source : ODPS odps_first ?

* Table : Please select

Clearance Rule : Clear Existing Data Before Writing (Insert Overwrite)

Compression : ☒ Disable ☐ Enable

Consider Empty String as Null : ☐ Yes ☒ No

Configuration	Description
Data Source	Select ODPS and enter a destination name.
Table	Select the table to be synchronized.
Partition information	The table to be synchronized is a non-partitioned table. Therefore, no partition information is displayed.
Clearance Rule	<ul style="list-style-type: none"> Clear Existing Data Before Writing (Insert Overwrite): All data in the table or partition is cleaned up before import. Retain Existing Data (Insert Into): No data is cleared before data importing. New data is always appended with each run.
Compression	The default value is Disable.
Consider Empty String as Null	The default value is No.

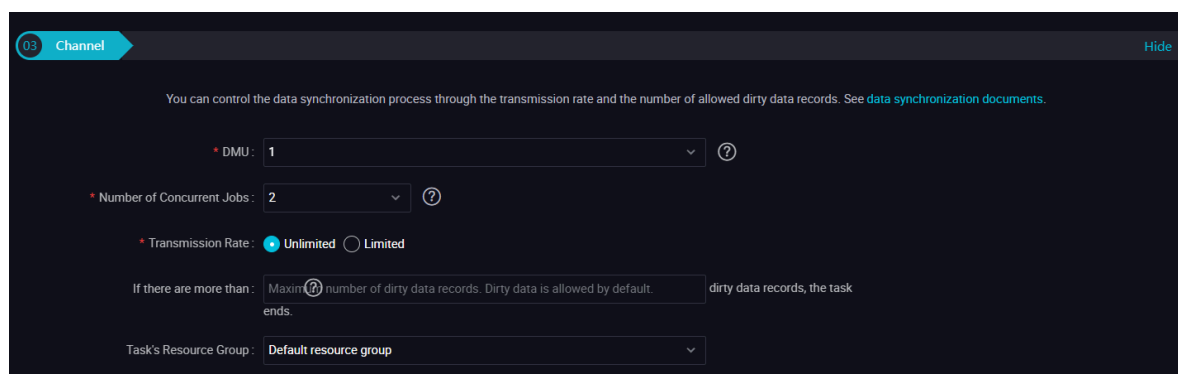
5. Set field mappings.


Map the fields in source and destination tables. Fields in the source table (left) have a one to one correspondence with fields in the destination table. Select **Enable Same Line Mapping**.



6. Configure channel control policies.

Configure the maximum transmission rate and dirty data check rules.



Configuration	Description
DMU	<p>The billing unit of Data Integration.</p> <div>  Note: The DMU value limits the maximum number of concurrent jobs. Ensure that DMU is set to an appropriate value. </div>
Number of Concurrent Jobs	<p>When you configure Synchronization Concurrency, the data records are split into several tasks based on the specified reader splitting key. These tasks run simultaneously to improve the transmission rate.</p>
Transmission Rate	<p>Setting a transmission rate protects the source database from excessive read activity and heavy load. We recommend that you throttle the transmission rate and configure the transmission rate properly based on the source database configurations.</p>
If there are more than	<p>The number of dirty data entries. For example, if varchar type data in the source is to be written into a destination column of the int type, a data conversion exception occurs and the data cannot be written into the destination column. You can set an upper limit for the dirty data entries to control the quality of synchronized data. Set an appropriate upper limit based on your business requirements.</p>

Configuration	Description
Task's Resource Group	<p>The resource group used for running the synchronization task. By default, the task runs with the default resource group. When the project has insufficient resources, you can add a custom resource group and run the synchronization task using the custom resource group. For more information about how to add custom resource groups, see Add scheduling resources.</p> <p>Choose an appropriate resource group based on your data source network conditions, project resources, and business importance.</p>

7. Run the task.

You can run the task using either of the following methods:

- Directly run the task (one-time running).

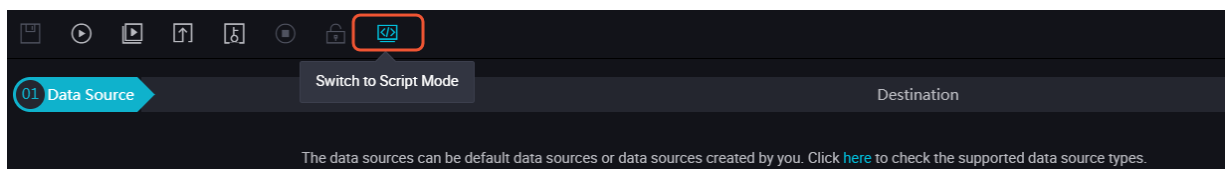
Click **Run** in the tool bar to run the task. After setting certain parameters, you can run the task directly on the DataStudio page.

- Schedule the task.

Click **Submit** to submit the synchronization task to the scheduling system. The scheduling system periodically runs the task starting from the next day according to the task configurations.

Configure a synchronization task in script mode

To configure this task in script mode, click **Switch to Script Mode** in the tool bar and click **OK**.



Script mode allows you to set up configurations as needed. An example script is as follows.

```
{
  "type": "job",
  "version": "1.0",
  "Configuration ": {
    "reader": {
      "plugin": "loghub",
      "parameter": {
        "datasource": "loghub_lzz", //Data source name. Use the name of the
        data resource that you have added.
        "logstore": "logstore-ut2", //Source Logstore name. A Logstore is a log
        data collection, storage, and query unit in LogHub.
        "beginDateTime": "${startTime}", //Start (included) time for filtering
        log entries by log time.
        "endDateTime": "${endTime}", //End (included) time for filtering log
        entries by log time.
      }
    }
  }
}
```

```

"batchSize": 256, //The number of data entries that are read each time
. The default value is 256.
"splitPk": "",
"column": [
  "key1",
  "key2",
  "key3"
]
},
"writer": {
  "plugin": "odps",
  "parameter": {
    "datasource": "odps_first", //Data source name. Use the name of the
    data resource that you have added.
    "table": "ok", //Destination table name
    "truncate": true,
    "partition": "", //Partition information
    "column": [ //Destination column name
      "key1",
      "key2",
      "key3"
    ]
  }
},
"Setting ": {
  "Speed ": {
    "mbps": 8, //Maximum transmission rate
    "concurrent": 7 //Number of concurrent jobs
  }
}
}
}
}

```

2.8.6 Import data into DataHub using Data Integration

This topic explains how to import data into offline DataHub by using Data Integration.

[Data Integration](#) is a data synchronization platform provided by Alibaba Group. Data Integration is a reliable, secure, cost-effective, elastic, scalable data synchronization platform. Data Integration can be used across heterogeneous data storage systems and provides offline (full/incremental) data synchronization channels in different network environments for more than 20 types of data sources. For more information about data source types, see [Supported data sources](#).

Prerequisites

1. [Prepare Alibaba Cloud account](#) An Alibaba Cloud account and logon credentials (AccessID and AccessKey) for the account.
2. Activate MaxCompute, and then a default MaxCompute data source is automatically created. Log on to the DataWorks console using the Alibaba Cloud account.
3. [Create a project](#) Create a project. To use DataWorks, first create a project. Then, you can complete the workflow and maintain data and tasks through collaboration within the project.


**Note:**

If you want to create Data Integration tasks using a RAM user, you must grant required permissions to it. For more information, see [Create a sub-account](#) and [Member management](#).

Procedure

In the following example, the Stream data is synchronized to DataHub and the synchronization task is configured in script mode:

1. Log on to the [DataWorks console](#) as a developer, find the project, and then click **Data Integration**.
2. Choose **Overview > Tasks** and click **Create Task** in the upper-right corner.
3. Complete the configurations in the **Create Node** dialog box and click **Submit**. The configuration page of the data synchronization task appears.
4. Click **Switch to Script Mode** in the toolbar and click **OK** to switch to script mode.
5. Click **Import Template** in the toolbar and set up configurations in the **Import Template** dialog box.

Configuration	Description
Source Type	In this example, select Stream .
Destination Type	In this example, select DataHub .
Data Source	Select a configured data source as the destination. <div>  Note: If no data source is configured, click Add Data Source to add one. </div>

6. Click **OK** to generate an initial script. Then, complete the configurations as needed.

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "setting": {
      "errorLimit": {
        "record": "0"
      },
      "speed": {
        "mbps": "1",
        "concurrent": "1", //Number of concurrent jobs
        "dmu": 1, //Data migration unit (DMU) is a measurement unit,
        which measures the resources (including CPU, memory, and network
        bandwidth) consumed by Data Integration.
        "throttle": false
      }
    },
    "reader": {
```

```

    "plugin": "stream",
    "parameter": {
      "column": [//Column name of the source
        {
          "value": "field",//Column properties
          "type": "string"
        },
        {
          "value": true,
          "type": "bool"
        },
        {
          "value": "byte string",
          "type": "bytes"
        }
      ],
      "sliceRecordCount": "100000"
    }
  },
  "writer": {
    "plugin": "datahub",
    "parameter": {
      "datasource": "datahub",//Data source name
      "topic": "xxxx",//Topic is the minimum unit of DataHub
      subscription and publishing, which can be used to represent a type
      of streaming data.
      "mode": "random",//Random write.
      "shardId": "0",//Shard represents a concurrent channel for data
      transmission of a topic, and each shard has a corresponding ID.
      "maxCommitSize": 524288,//To improve writing performance,
      configure the system to write data to the destination in batches
      when the size of the collected data reaches maxCommitSize (in MB).
      The default value is 1048576 (1 MB).
      "maxRetryCount": 500
    }
  }
}

```

7. Click **Save** and **Run**.



Note:

- DataHub only supports importing data in script mode.
- To use a new template, click **Import Template** in the toolbar. The existing content is overwritten once the script is imported.
- After saving the synchronization task, click **Run** to immediately run the task.

Alternatively, click **Submit** to submit the synchronization task to the scheduling system. The scheduling system periodically runs the task starting from the next day according to the task configurations.

Reference

For more information about how to configure synchronization tasks, see the following topics.

- [Configure the Reader plug-in.](#)
- [Configure the Writer plug-in.](#)

2.8.7 Configure OTSStream data synchronization tasks

The OTSStream plugin is used for exporting Table Store incremental data. The incremental data can be considered as operation logs that contain data and operation information.

Different from full export plugins, the incremental export plugin only has multi-version mode that does not allow you to specify columns. This limit is related to how incremental export works. For more information, see [Configure OTSStream Reader](#).



Note:

When configuring OTSStream data synchronization tasks, note the following:

- The system can only read the data that is generated five minutes ago but over the past 24 hours.
- The end time cannot be later than the current system time. Therefore, the end time must be at least five minutes earlier than the task start time.
- Scheduling a task to run daily may cause data loss.
- Scheduling periodic and monthly tasks is not supported.

Example:

The start time and the end time must cover the time period for operating Table Store tables. For example, if you insert two data entries to Table Store at 20171019162000, the start time and the end time can be set to 20171019161000 and 20171019162600 respectively.

Add a data source

1. Log on to the [DataWorks console](#) as a project administrator, find the project, and then click **Data Integration**.
2. Choose **Sync Resources > Data Source** and click **Add Data Source** in the upper-right corner.
3. Select **Table Store (OTS)** as the data source type and set up the configurations in the dialog box that appears.

Configuration	Description
Data Source Name	Can contain letters, numbers, and underscores (_). It must begin with a letter, and cannot exceed 60 characters in length.
Description	The description of the data source.

Configuration	Description
Endpoint	The endpoint of the LogHub data source in the format of <code>http://example.com</code> .
Table Store instance ID	The instance ID corresponding to the Table Store service.
AccessId/ AccessKey	The logon credential, similar to the account name and the password.

4. Click **Test Connectivity**.
5. When the connection test is passed, click **Complete**.

Configure a synchronization task in wizard mode

1. Choose **Overview > Tasks** and click **Create Task** in the upper-right corner.
2. Set up configurations in the **Create Node** dialog box and click **Submit**. Then, the configuration page of the data synchronization task appears.
3. Select a data source.

Configuration	Description
Data Source	Select OTSStream and enter the OTSStream data source name.
Table	The name of the table from which incremental data is exported. You must enable the Stream feature on the table when creating the table or using the UpdateTable operation after the creation.
Start Time	The start time (included) in milliseconds of the incremental data. The format is yyyyMMddHHmmss.
End time	The end time (excluded) in milliseconds of the incremental data. The format is yyyyMMddHHmmss.
State Table	The name of the table for recording states.
Maximum Retries	The maximum number of retries of each request for reading incremental data from Table Store. The default value is 30.
Export Sequence Information	Whether to export time-series information. Time-series information includes the time when data is written.

4. Select a destination.

Select a MaxCompute destination and select a table.

Configuration	Description
Data Source	Select ODPS and enter a destination name.


Configuration	Description
Table	Select the table to be synchronized.
Partition information	The table to be synchronized is a non-partitioned table. Therefore, no partition information is displayed.
Clearance Rule	<ul style="list-style-type: none"> • Clear Existing Data Before Writing (Insert Overwrite): All data in the table or partition is cleaned up before import. • Retain Existing Data (Insert Into): No data is cleared before data importing. New data is always appended with each run.
Compression	The default value is Disable.
Consider Empty String as Null	The default value is No.

5. Set field mappings.

Map the fields in source and destination tables. Fields in the source table (left) have a one to one correspondence with fields in the destination table.

6. Configure channel control policies.

Configure the maximum transmission rate and dirty data check rules.

Configuration	Description
DMU	<p>The billing unit of Data Integration.</p> <div>  Note: The DMU value limits the maximum number of concurrent jobs. Ensure that DMU is set to an appropriate value. </div>
Number of concurrent jobs	When you configure Synchronization Concurrency, the data records are split into several tasks based on the specified reader splitting key. These tasks run simultaneously to improve the transmission rate.
Transmission Rate	Setting a transmission rate protects the source database from excessive read activity and heavy load. We recommend that you throttle the transmission rate and configure the transmission rate properly based on the source database configurations.

Configuration	Description
If there are more than	The number of dirty data entries. For example, if varchar type data in the source is to be written into a destination column of the int type, a data conversion exception occurs and the data cannot be written into the destination column. You can set an upper limit for the dirty data entries to control the quality of synchronized data. Set an appropriate upper limit based on your business requirements.
Task's Resource Group	The resource group used for running the synchronization task. By default, the task runs with the default resource group. When the project has insufficient resources, you can add a custom resource group and run the synchronization task using the custom resource group. For more information about how to add custom resource groups, see Add scheduling resource . Choose an appropriate resource group based on your data source network conditions, project scheduling resources, and business importance.

7. Click **Save** and **Run**.

Click the **Run** button above the task panel to run the task on the Data Integration page. You need to set the custom parameters before running the task.

Configure a synchronization task in script code

To configure this task in script mode, click Switch to Script Mode in the toolbar and click **OK**.

Script mode allows you to set up configurations as needed. An example script is as follows.

```
{
  "type": "job",
  "version": "1.0",
  "Configuration ":{
    "reader": {
      "plugin": "otsstream",
      "parameter": {
        "datasource": "otsstream",//Data source name. Use the name of
the data resource that you have added.
        "dataTable": "person",//Name of the table from which the
incremental data is exported. You must enable the Stream feature on
the table when creating the table or using the UpdateTable operation
after the creation.
        "startTimeString": "${startTime}",//The start time (included)
in milliseconds of the incremental data. The format is yyyyMMddHHmmss.
        "endTimeString": "${endTime}",//The start time (excluded) in
milliseconds of the incremental data. The format is yyyyMMddHHmmss.
        "statusTable": "TableStoreStreamReaderStatusTable",//The name
of the table for recording the states.
        "maxRetries": 30,//The maximum number of retries of each
request.
        "isExportSequenceInfo": false,
      }
    }
  }
}
```

```

    },
    "writer": {
      "plugin": "odps",
      "parameter": {
        "datasource": "odps_first", //Data source name
        "table": "person", //Destination table name
        "truncate": true,
        "partition": "pt=${bdp.system.bizdate}", //Partition informatio
n
        "column": [ //Destination column name
          "id",
          "colname",
          "version",
          "colvalue",
          "optype",
          "sequenceinfo"
        ]
      }
    },
    "Setting ":{
      "Speed ":{
        "mbps": 7, //Maximum transmission rate
        "concurrent": 7 //Number of concurrent jobs
      }
    }
  }
}

```

**Note:**

- You can configure the time range of the incremental data using either of the following methods.

- "startTimeString": "\${startTime}"

The start time (included) in milliseconds of the incremental data. The format is yyyyMMddHHmmss.

- "endTimeString": "\${endTime}"

The end time (excluded) in milliseconds of the incremental data. The format is yyyyMMddHHmmss.

- "startTimestampMillis": ""

The start time (included) in milliseconds of the incremental data.

The Reader plugin finds a point corresponding to startTimestampMillis from the statusTable, and starts to read and export data from this point.

If the Reader plugin cannot find the corresponding point, it starts to read incremental data retained by the system from the first entry, and skip the data which is written later than startTimestampMillis.

```
"endTimeStampMillis": " "
```

The end time (included) in milliseconds of the incremental data.

The Reader plugin exports data from the startTimestampMillis and ends at the data with the timestamp later than or equal to the endTimeStampMillis.

When the Reader plugin finishes reading all the incremental data, the reading process is ended even if it does not reach the endTimeStampMillis.

This value is a timestamp value, measured in milliseconds.

- If isExportSequenceInfo is set to true ("isExportSequenceInfo": true), the system exports an extra column for time-series information. The time-series information contains data writing time. The default value of isExportSequenceInfo is false, which means no time-series information is exported.

2.9 FAQ

2.9.1 How to troubleshoot data integration problems?

If any problem arises during Data Integration operations, you must identify the relevant information, such as: on what server the tasks are run, the information on data sources, and the region in which the synchronization tasks are configured.

The server performing the tasks

- running on Alibaba's server:

running in Pipeline[basecommon_group_xxxxxxxxx]

- running on your server:

running in Pipeline[basecommon_xxxxxxxxx]

Information on data sources

When Data Integration fails, you must review the information on data sources:

- check the data sources among which the synchronization tasks are run.
- check the environment of the data sources.

For example: Alibaba Cloud database, data sources with/without public IPs or VPC network environment (RDS and other sources), Financial Cloud (VPC and classic network).

- check if the connectivity test of data source is successful.

Compare against the Data Source Configuration document: check if the information on data source is filled up incorrectly (typical situations include mixing up multiple databases, adding spaces or special characters when filling up the information, or the connectivity test is not supported (data source from database without public IPs or a VPC environment except RDS)).

Check the region in which the synchronization tasks are configured

You can see the related regions in the DataWorks console, such as East China 2, North China 1, Hong Kong, Southeast Asia Pacific 1, Central Europe 1, and Southeast Asia Pacific 2. Generally, the default region is the East China 2. You can see the corresponding region after purchasing the MaxCompute.

Copy the troubleshooting code when interface pattern errors are reported

When interface pattern errors are reported, copy the troubleshooting code for relevant personnel.

The log reports exceptions

The log reports an error occurred while running the SQL statement (the column contains the keyword)

```
2017-05-31 14:15:20.282 [33881049-0-0-reader] ERROR ReaderRunner -
Reader runner Received Exceptions:com.alibaba.datax.common.exception.
DataXException: Code:[DBUtilErrorCode-07]
```

Error Details:

Failed to read database data. Check your column/table/where/querySql configuration or ask DBA for help.

The executed SQL statement is as follows:

```
select **index**,plaid,plarm,fget,fot,havm,coer,ines,oumes from xxx
```

The error details are shown as follows:

```
You have an error in your SQL syntax; check the manual that correspond
s to your MySQL Server version for the right syntax to use near Index
, plaid, plarm, fget, fot, havm, coer, Ines, oums from XXX
```

Troubleshooting:

- Then, run another SQL statement:

```
select **index**,plaid,plarm,fget,fot,havm,coer,ines,oumes from
xxx
```

If you look at the results, there will also be corresponding errors.

- If the field contains the keyword index, you can add single quotes or modify the field to resolve the problem.

The log reports that an error occurred while running the SQL statement (the table name is in single quotes within double quotes)

```
com.alibaba.datax.common.exception.DataXException: Code:[DBUtilErrorCode-07]
```

Error Details:

Failed to read database data. Check your column/table/where/querySql configuration or ask DBA for help.

The executed SQL statement is as follows:

```
select /_+read_consistency(weak) query_timeout(100000000)/ _ from** '
ql_ddddd_[0-31]' **where 1=2
```

The error details are shown as follows:

```
You have an error in your SQL syntax; check the manual that correspond
s to your MySQL server version for the right syntax to use near ''
ql_live_speaks[0-31]' where 1=2' at line 1 - com.mysql.jdbc.exceptions
.jdbc4. Mysqlsyntaxerrorexception: You have an error in your SQL
syntax; check the manual that corresponds to your MySQL Server version
for the right syntax to use near '**' 'ql _ dddd _ [0-31] 'where 1 =
2 '**
```

Troubleshooting

If the table name is in single quotes within double quotes, you can delete the single quotes directly in the configuration constant "table":["qldddd[0-31]"].

Connectivity test of data source fails (The exception message "Access denied for..." is reported)

An error occurred while connecting to the database. Database connection string: jdbc:mysql://xx.xx.xx.x:3306/t_demo. User name: fn_test. Exception message: Access denied for user 'fn_test'@'%' to database 't_demo'. Make sure you have added a whitelist in RDS.

Troubleshooting:

- When the exception message Access denied for... is reported, it generally indicates certain problems of the information you entered. Check that information.
- Check whether the whitelist or your account has the permission to access the database. You can add the required whitelist and permissions in the RDS console.

The routing policy has some problems. The running pool are OXS and ECS clusters.

```
2017-08-08 15:58:55 : Start Job[xxxxxxx], traceId **running in
Pipeline[basecommon_group_xxx_cdp_oxs]**ErrorMessage:Code:[DBUtilErrorCode-10]
```

Error Details:

An error occurred while connecting to the database. Check your account, password, database name, IP address and port or ask DBA for help (note the network environment). An error occurred while connecting to the database, because no connecting JDBC URL can be found from jdbc:oracle:thin:@xxx.xxxxx.x.xx:xxxx:prod. Check and modify your configurations. Check your configurations and make changes.

The error message "java.lang.Exception: DataX" indicates that the corresponding database cannot be connected for the following reasons:

- the IP/port/database/JDBC you configured is incorrect and cannot be connected.
- the user name/password you configured is incorrect, and authentication is unsuccessful.
Confirm with DBA whether the connection information of the database is correct.

Troubleshooting:**Scenario 1:**

- To synchronize RDS-PostgreSQL data sources from Oracle, you can click **Run** directly. The tasks cannot be performed by the scheduler, because different pools are required.
- You can add data sources in the form of JDBC to RDS, then the RDS-PostgreSQL data sources can be synchronized from Oracle.

Scenario 2:

- RDS-PostgreSQL data sources in VPC environment cannot run on a custom source group. The RDS in VPC environment provides reverse proxy capability, leading to network problems for the custom resource group. Therefore, RDS in VPC environment can directly run on Alibaba's server. If our server cannot meet your requirements, and you want to run tasks on your server, you must add data source in the form of JDBC to RDS in VPC environment and purchase the ECS in the same network segment.
- - The "jdbc:mysql://100.100.70.1:4309/xxx,100" mapped out by the RDS in VPC environment often begins with an IP mapped out by the background. If it begins with an domain, the RDS is not in a VPC environment.

HBase Writer does not support the Date type

Hbase synchronization to hbase: 2017-08-15 11: 19: 29: State: 4 (fail) | Total: 0r 0b | speed: 0r/s 0b/S | error: 0r 0b | stage: 0.0% errormessage: Code: [fig]

Error Details:

The value of the parameter you entered is invalid.

Hbase writer does not support this type: Date. The types currently supported are: [string, boolean, short, int, long, float, double].

Troubleshooting:

- HBase writer does not support the Date type. You cannot configure any data in the type of Date in the writer.
- You can directly configure the data in string type, because HBase has no limit in terms of data type. The bottom layer of the HBase is generally the byte array.

JSON format configuration error

Column configuration error

Based on the analysis by DataX, the most likely cause of this error is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[Framework-02]
```

Error Details:

The DataX engine encountered an error when running. For details, see the error diagnostic information after DataX stops running

```
java.lang.ClassCastException: com.alibaba.fastjson.Jsonobject cannot be cast to java.lang.String
```

Troubleshooting:

JSON is configured improperly.

```
Writer:
"column":[
{
"name":"busino",
"type": "string"
}
]
Write the statement as follows:
"column":[
{
"Busino"
}
```

```
]
```

- The JSON list is written less []

In using smart analysis of DataX, the most likely reason for error is:

```
com.alibaba.datax.common.exception.DataXException: Code:[Framework-02]
```

Error Details:

The DataX engine encountered an error when running. For details, see the error diagnostic information after DataX stops running

```
java.lang.String cannot be cast to java.util.List - java.lang.String
cannot be cast to java.util.List
at com.alibaba.datax.common.exception.DataXException.asDataXExc
eption(DataXException.java:41)
```

Troubleshooting:

When [] is missing, the list type is changed. You can resolve this by finding where the is missing and adding the.

Permission issues

- Permission issues (no permission for "delete" operation)

For synchronization from MaxCompute to RDS-MySQL, the error message is: Code:DBUtilErrorCode-07

Error Details:

Failed to read database data. Check your column/table/where/querySql configuration or ask DBA for help.

The executed SQL statement is as follows:

```
delete from fact_xxx_d where sy_date=20170903
```

The error details are shown as follows:

```
**DELETE command denied** to user 'xxx_odps'@[xx.xxx.xxx.xxx](
http://xx.xxx.xxx.xxx)' for table 'fact_xxx_d' - com.mysql.jdbc.
exceptions.jdbc4. MySQLSyntaxErrorException: DELETE command denied
to user 'xxx_odps'@[xx.xxx.xxx.xxx](http://xx.xxx.xxx.xxx)' for
table 'fact_xxx_d'
```

Troubleshooting:

The error message "DELETE command denied to" indicates that you have no permission to delete the table, and you must grant the permission required in the corresponding database.

- Permission issues (no permission for "drop" operation)

Code:DBUtilErrorCode-07

Error Details:

Failed to read database data. Check your column/table/where/querySql configuration or ask DBA for help.

The SQL you run is: truncate table be_xx_ch

The error details are shown as follows:

```
**DROP command denied to user** 'xxx'@[xxx.xx.xxx.xxx](http://xxx.xx.xxx.xxx)' for table 'be_xx_ch' - com.mysql.jdbc.exceptions.jdbc4
.MySQLSyntaxErrorException: DROP command denied to user 'xxx'@[xxx.xx.xxx.xxx](http://xxx.xx.xxx.xxx)' for table 'be_xx_ch'
```

Troubleshooting:

The preceding error is reported when the prepared statement "truncate" before MySQLWriter configuration execution is performed to delete the table data, because you have no permission for "drop" operation.

ADS permission issues

```
2016-11-04 19:49:11.504 [job-12485292] INFO OriginalConfPretreat
mentUtil - Available jdbcUrl:jdbc:mysql://100.98.249.103:3306/ads_rdb
? yearIsDateType=false&zeroDateTimeBehavior=convertToNull&tinyIntIs
Bit=false&rewriteBatchedStatements=true.
2016-11-04 19:49:11. 505 [job-12485292] warn maid
```

There is a certain risk of column configuration in your configuration file. Because you do not have columns configured to read database tables, when there is a change in the number and type of your table fields, may affect task correctness or even run errors. Check your configurations and make changes.

```
2016-11-04 19:49:11.528 [job-12485292] INFO Writer$Job
```

If it is MaxCompute > ADS data synchronization, you must complete the following authorizations:

- The ADS official account must have at least the "describe" and "select" permissions for the tables to be synchronized, because the ADS system requires the structure and data information of the table to be synchronized from MaxCompute.
- The account AK you configured to access the ADS data source must have the permission to initiate a request to load data to the specified ADS database. You can add the authorization in the ADS system.

```
2016-11-04 19:49:11.528 [job-12485292] INFO Writer$Job
```

If it is the data synchronization between RDS (or other non-MaxCompute data sources) and ADS, the implementation logic is to first load the data to the MaxCompute temporary table, and then synchronize data from MaxCompute temporary table to ADS (set temporary MaxCompute project as `cdp_ads_project`, and set the temporary project account as `cloud-data-pipeline@aliyun-inner.com`).

Permissions:

- The ADS official account must have at least the "describe" and "select" permissions for the tables (MaxCompute temporary table) to be synchronized, because the ADS system requires the structure and data information of the table to be synchronized from MaxCompute (the authorization has been completed at deployment).
- The account `cloud-data-pipeline@aliyun-inner.com` of temporary MaxCompute must have the permission to initiate a request to load data to the specified ADS database. You can add the authorization in the ADS system.

Troubleshooting:

This problem is due to the lack of permission to load data.

The temporary project account is `cloud-data-pipeline@aliyun-inner.com`. ADS official account must have at least the "describe" and "select" permissions for the tables (MaxCompute temporary table) to be synchronized, because the ADS system requires the structure and data information of the table to be synchronized from MaxCompute (the authorization has been completed at deployment). Log on to the ADS console and grant the "load data" permission to the ADS.

Whitelist issues

- The whitelist has not been added and the connectivity test of data source fails.

Test connection failed. Connectivity test of data source failed:

```
error message: Timed out after 5000 ms while waiting for a server
that matches ReadPreferenceServerSelector{readPreference=primary}.
Client view of cluster state is {type=UNKNOWN, servers=[{address:
3717=dds-bplafbf47fc7e8e41.mongodb.rds.aliyuncs.com}(http://address
:3717=dds-bplafbf47fc7e8e41.mongodb.rds.aliyuncs.com), type=UNKNOWN
, state=CONNECTING, exception={com.mongodb.MongoSocketReadException
: Prematurely reached end of stream}}, {[address:3717=dds-bplafbf47f
c7e8e42.mongodb.rds.aliyuncs.com}(http://address:3717=dds-bplafbf47f
c7e8e42.mongodb.rds.aliyuncs.com), type=UNKNOWN, state=CONNECTING
,** exception={com.mongodb.MongoSocketReadException: Prematurely
reached end of stream**}]]}
```

Troubleshooting

When adding data source to MongoDB in non-VPC environment, if the error message Timed out after 5000 is reported, it means that the whitelist has a problem.



Note:

If you are using ApsaraDB for MongoDB, a root account is provided by default. To ensure security, Data Integration only supports using the relevant account of MongoDB for connection. Avoid using root account as the access account when adding and using the MongoDB data source.

- White List not complete

for Code:[DBUtilErrorCode-10]

Error Details:

An error occurred while connecting to the database. Check your account, password, database name, IP address and port or ask DBA for help (note the network environment).

The error details are shown as follows:

```
java.sql.SQLException: Invalid authorization specification, message
  from server: "#**28000ip not in whitelist, client ip is xx.xx.xx.xx
  ". **
2017-10-18 11:03:00. 673 [job-Newfoundland] Error retryutil-
exception when calling callable
```

Troubleshooting:

The whitelist you added is incomplete. You has not added your server into the whitelist.

The data source information is incorrect

- When configuring the script mode, the corresponding data source information (could not be blank) is missing.

```
2017-09-06 12:47:05 INFO Success to fetch meta data for table with
projectId 43501 project ID and instance ID mongodbdsource name.
**
2017-09-06 12:47:05 [INFO] Data transport tunnel is CDP.
2017-09-06 12:47:05 [INFO] Begin to fetch alisa account info for
3DES encrypt with parameter account: [zz_683cdbcefbal43b7b709067b362
d4385].

2017-09-06 12:47:05 [INFO] Begin to fetch alisa account info for
3DES encrypt with parameter account: [zz_683cdbcefbal43b7b709067b362
d4385].
[Error] exception when running task, message: ** configuration
property [adord] is generally the information to be filled in by
ODPS data source could not be blank! **
```

Troubleshooting:

The error message shows that the corresponding accessId information is blank. This is generally due to script mode issues. Check the JSON code you configured to see whether the corresponding data source name is missing.

- Data source is not configured

```
2017-10-10 10:30:08 INFO
=====

File "/home/admin/synccenter/src/Validate.py", line 16, in notNone
raise Exception("Configuration property [%s] could not be blank!" %
(Context ))
** Exception: configuration property [username] could not be blank!
**
```

Troubleshooting:

- Check with the normal logs:

```
[56810] and instanceId(instanceName) [spfee_test_mysql]...
2017-10-09 21:09:44 [INFO] Success to fetch meta data for table
with projectId [56810] and instanceId [spfee_test_mysql].
```

- Generally, such information shows that an error occurred while calling the data source. If the empty user name is reported, it shows that the data source has not been configured or the location of data source has not been configured correctly. In this case, the user has configured an incorrect position of the data source.
- DRDS data connection time-out

When synchronizing data from MaxCompute to DRDS, the following errors often appear:

```
[2017-09-11 16:17:01. 729 [49892464-0-0-writer] warn maid $ task
```

Roll back the data written this time and write a single row of data each time and submit again.

The reasons are as follows:

```
com.mysql.jdbc.exceptions.jdbc4. CommunicationsException: **
Communications link failure **
The last packet successfully received from the server was 529
milliseconds ago.
The last packet sent successfully to the server was** 528 millisecon
ds ago**.
```

Troubleshooting:

Datx client timeouts can be added when adding DRDs data sources ? `useUnicode=true&characterEncoding=utf-8&socketTimeout=3600000` timeout Parameter

Example:

```
jdbc:mysql://10.183.80.46:3307/ae_coupon? useUnicode=true&characterEncoding=utf-8&socketTimeout=3600000
```

- System internal problems

Troubleshooting:

Generally, system internal problems are reported when the data source in JSON format is mistakenly modified and saved in the development environment. When the page is blank, you can directly provide the project name and the node name to us for background processing.

Dirty data

- Dirty data (the string [""] cannot be converted to long)

```
2017-09-21 16:25:46.125 [51659198-0-26-writer] ERROR WriterRunner -
Writer Runner Received Exceptions:
com.alibaba.datax.common.exception.DataXException: Code:[Common-01]
```

Error Details:

The business dirty data generated during data synchronization is caused by incorrect data type conversion. The string [""] cannot be converted to long.

Troubleshooting:

The String [""] cannot be converted to long: The statements for table creation in two tables are the same. The preceding error is reported because the field type empty cannot be converted to long. You can directly configure it as a string.

- Dirty data (out of range value)

```
2017-11-07 13:58:33.897 [503-0-0-writer] ERROR StdoutPluginCollector

Dirty data:
{"exception":"Data truncation:Out of range value for column 'id' at
row 1","record":{"byteSize":2,"index":0,"rawData":-3,"type":"LONG"},
{"byteSize":2,"index":1,"rawData":-2,"type":"LONG"}, {"byteSize":2,"
index":2,"rawData":"other","type":"STRING"}, {"byteSize":2,"index":3
,"rawData":"other","type":"STRING"}, "type":"writer"}
```

Troubleshooting:

The source data type of mysql2mysql is set as smallint(5) and the target data type is int(11) unsigned. Because the data in the type of smallint(5) contains negative number, and the data in the type of unsigned cannot be negative, the dirty data is generated.

- Dirty data (storing emoji)

The data table is configured to store emoji, and dirty data is reported during data synchronization.

Troubleshooting:

Data integration is supported by default by utf 8, so when you add a data source in JDBC format, you need to modify your settings, such `jdbc:mysql://xxx.x.x.x:3306/database?characterEncoding=utf8&com.mysql.jdbc.faultInjection.serverCharsetIndex=45`, so that you can set the emotability on the data source to synchronize successfully.

- Dirty data caused by empty fields

```
{ "exception": "Column 'xxx_id' cannot be null", "record": [{ "byteSize": 0, "index": 0, "type": "LONG" }, { "byteSize": 8, "index": 1, "rawData": -1, "type": "LONG" }, { "byteSize": 8, "index": 2, "rawData": 641, "type": "LONG" }
```

Based on the analysis by DataX, the most likely cause of this error is as follows:

`com.alibaba.datax.common.exception.DataXException: Code:[Framework-14]`

Error Details:

The dirty data transmitted by DataX exceeds user expectations. This error often occurs when a lot of dirty business data exists within the source data. Please check carefully the dirty data log information reported by DataX, or adjust the dirty data threshold accordingly.

The check on the number of dirty data entries failed. The number of dirty data entries is limited to 1, but seven are captured.

Troubleshooting:

The dirty data is generated because the field "column 'xxx_id' cannot be null" cannot be empty, and empty data is used during data synchronization. You can modify those empty data, or modify the field.

- The field "data too long for column 'flash'" is too short and the dirty data is generated.

```
2017-01-02 17:01:19.308 [16963484-0-0-writer] ERROR StdoutPlug
inCollector
Dirty data:
{"exception": "Data updatation: data Too long for column 'Flash '
at Row 1, "record ": [{ "bytesize": 8, "Index": 0, "rawdata": 1, "
type": "long"}, { "bytesize": 8, "Index ": 3, "rawdata": 2, "type":
"long "}, { "bytesize": 8, "Index": 4, "rawdata": 1, "type": "long
"}, { "bytesize": 8, "Index ": 5, "rawdata": 1, "type": "long "}, { "
bytesize": 8, "Index": 6, "rawdata": 1, type: "Long "}
```

Troubleshooting:

The field "data too long for column 'flash'" is too short, but the data that you synchronized is too long. Therefore, the dirty data is generated. You can modify the data, or the field.

- Read-only permission to database settings

```
2017-11-07 13:58:33.897 503-0-0-writer ERROR StdoutPluginCollector
Dirty data:
{"exception": "the MySQL server is running with the -- read-only
option so it cannot execute this statement", "record": [{"bytesize
": 3, "Index": 0, rawdata: 201, type: Long}, {"bytesize ": 8, "Index
": 1, "rawdata": 1474603200000, "type ": "date"}, {"bytesize": 8
, "Index": 2, rawdata: September 23, "12", "type": "string "}, {"
bytesize": 5, "Index": 3, "rawdata ": "12", "type": "string"}
```

Troubleshooting:

When read-only mode is set, if all the data to be synchronized is dirty data, you can change the "read-only" mode of the database into "writable" mode.

- Logs generated when partition error occurs

An error message is reported when the parameter is configured as \$yyyyymm. The log is generated as follows:

```
[2016-09-13 17:00:43] 2016-09-13 16:21:35. 689 [job-10055875] Error
Engine
```

Based on the analysis by DataX, the most likely cause of this error is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[OdpsWriter-13
]
```

Error Details:

If an exception occurs while running MaxCompute SQL, you can try again. If the MaxCompute target table throws an exception when executing MaxCompute SQL, contact the MaxCompute administrator. The content of SQL is as follows:

```
alter table db_rich_gift_record add IF NOT EXISTS
partition(pt='${thismonth}');
```

Troubleshooting:

The single quotes added leads to invalid scheduling parameter replacing. Solution: remove the single quotes of '\${thismonth}'.

- column is not configured as the array form

```
Run Command failed.
com.alibaba.cdp.sdk.exception.CDPException: com.alibaba.fastjson.
JSONException: syntax error, **expect {,** actual error, pos 0
```

```
at com.alibaba.cdp.sdk.exception.CDPEException.asCDPEException(
CDPEException.java:23)
```

Troubleshooting:

The JSON has the following problem:

```
"plugin": "mysql",**
"parameter":{
"Datasource": "XXXXXX",
** "column": "uid",**
  "where": "",
  "splitPk": "",
  "table": "xxx"
}
"column": "uid",-----has not been configured as the array form
```

- JDBC formatting error

Troubleshooting:

The JDBC format is incorrect. The correct format is: jdbc:mysql://ServerIP:Port/Database.

- Test connectivity failed

Troubleshooting:

- Check whether the firewall limits the IP and port used by your account.
- Check the port development of the security group.

- uid[xxxxxxxx] is reported in the logs

```
Run Command failed.
com.alibaba.cdp.sdk.exception.CDPEException: RequestId[F9FD049B-
xxxx-xxxx-xxx-xxxx] Error: there was an exception in the network
information for the obtained instance, please check the RDS buyer
ID and the RDS Instance name, UID [Newfoundland], instance [rm-
bplcwz5886rmzio92] serviceunavailable: the request has failed due to
a maid failure of the server.
RequestIdF9FD049B-xxxx-xxxx-xxx-xxxx Error:
```

Troubleshooting:

Generally, when synchronizing data from RDS to MaxCompute, if the preceding error is reported, you can directly copy the RequestId:F9FD049B-xxxx-xxxx-xxx-xxxx to the RDS personnel.

- The query parameter in MongoDB is incorrect

When the following error is reported as synchronizing data from MongoDB to MySQL, if you find that it is caused by incorrect JSON, it means that the JSON query parameter is not configured properly.

```
Exception in thread "taskGroup-0" com.alibaba.datax.common.exception.DataXException: Code:[Framework-13]
```

Error Details:

The DataX plug-in encountered an error while running. For the specific causes, refer to the error diagnostic information after DataX stops running.

```
org.bson.json.JsonParseException: Invalid JSON input. Position: 34.
Character : '.'.
```

Troubleshooting:

- Negative example: "query":{"update_date":{"\$gte":new Date().valueOf()/1000}}". The parameter in the form of "new Date()" is not supported.
- Correct example: "query":{"operationTime":{"\$gte":ISODate('\${last_day}T00:00:00.424+0800')}}"
- Cannot allocate memory

```
2017-10-11 20:45:46.544 [taskGroup-0] INFO TaskGroupContainer -
taskGroup[0] taskId[358] attemptCount[1] is started
Java HotSpot™ 64-Bit Server VM warning: INFO: os::commit_memory(
0x00007f15ceaeb000, 12288, 0) failed; error= '**Cannot allocate
memory' ** (errno=12)
```

Troubleshooting:

The memory is insufficient. If it occurs on your server, you must add extra memory; if it occurs on Alibaba's server, directly contact the technical support personnel.

- max_allowed_packet parameter

The error details are shown as follows:

```
Packet for query is too large (70>-1 ). You can change this value on
the server by setting the max_allowed_packet' variable. **com.mysql
.jdbc.PacketTooBigException: Packet for query is too large (70 > -1
). You can change this value on the server by setting the max_allowe
d_packet' variable. **
```

Troubleshooting:

The max_allowed_packet parameter is used to define the maximum length of the communication buffer. MySQL may limit the size of the data packets received by the server based on the

configuration file. Sometimes, insertions and updates in large size may fail due to the limitation of the `max_allowed_packet` parameter.

- If the value of `Max_allowed_packet` parameter is too large, you can change it into a smaller one . 10 MB = 10_1024_1024.
- "HTTP Status 500" is reported and an error occurred while reading the logs.

```
Unexpected Error:
Response is com.alibaba.cdp.sdk.util.http.Response@382db087[proxy
=HTTP/1.1 500 Internal Server Error [Server: Tengine, Date: Fri,
27 Oct 2017 16:43:34 GMT, Content-Type: text/html; charset=utf-8,
Transfer-Encoding: chunked, Connection: close,
**HTTP Status 500** - Read timed out**type** Exception report**
message**++Read timed out++**description**++The server encountered
an internal error that prevented it from fulfilling this request.+
**exception**
java.net.SocketTimeoutException: Read timed out
```

Troubleshooting:

When "HTTP Status 500" is reported while your tasks are running, if an error occurred during log reading of the tasks running on Alibaba's server, contact technical support personnel. If you are running on tasks on your own server, restart the Alisa.



Note:

If the service status remains Stopped after the refreshing, restart the following alisa command to switch to the admin account: `/home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart`.

- `hbasewriter` parameter: `hbase.zookeeper.quorum` configuration error

```
2017-11-08 09:29:28.173 [61401062-0-0-writer] INFO ZooKeeper -
Initiating client connection, connectString=xxx-2:2181,xxx-4:2181
,xxx-5:2181,xxxx-3:2181,xxx-6:2181 sessionTimeout=90000 watcher=
hconnection-0x528825f50x0, quorum=node-2:2181,node-4:2181,node-5:
2181,node-3:2181,node-6:2181, baseZNode=/hbase
Nov 08, 2017 9:29:28 AM org.apache.hadoop.hbase.zookeeper.Recoverabl
eZooKeeper checkZk
WARNING: **Unable to create ZooKeeper Connection**
```

Troubleshooting:

- Error example: "hbase. zookeeper. quorum: "xxx-2, xxx-4, xxx-5, xxxx-3, xxx-6"
- "Hbase.zookeeper.quorum":"your zookeeper IP address"
- No relevant files are found

Based on the analysis by DataX, the most likely cause of this error is as follows:

```
com.alibaba.datax.common.exception.DataXException: Code:[HdfsReader-08]
```

Error Details:

The directory of the file you are trying to read is empty. Failed to locate the file to be read, check your configuration items.

```
Path:/user/hive/warehouse/maid /*  
at com.alibaba.datax.common.exception.DataXException.asDataXExc  
eption(DataXException.java:41)
```

Troubleshooting:

Find the corresponding location using the path to check the corresponding file. If the file is not found, perform the necessary operations on the file.

- Table doesn't exist

Based on the analysis by DataX, the most likely cause of this error is as follows:

com.alibaba.datax.common.exception.DataXException: Code:[MySQLErrCode-04]

Error Details:

The table does not exist. Check the table name or contact DBA to confirm whether the table exists.

Table name: xxxx.

The SQL executed is: `Select * from Newfoundland where 1 = 2;`

The error details are shown as follows:

```
Table 'darkseer-test.xxxx' doesn't exist - com.mysql.jdbc.exceptions  
.jdbc4. MySQLSyntaxErrorException: Table 'darkseer-test.xxxx' doesn'  
t exist
```

Troubleshooting:

`select * from xxxx where 1=2` and check if the table xxxx has a problem. Take appropriate actions if any problem exists.

2.9.2 Synchronous task waiting for slots

Issue Description

The task is not functioning properly, and the log prompts the current instance that it has not yet generated log information, waiting for the slot.

Root cause

The above prompts occur because the configuration schedule for the task uses a custom resource , however, there are currently no custom resources available.

Solution

1. You can go to the **DataWorks > operations center > task operations** page, right-click tasks that are not scheduled as expected, select **view node properties** to view the resource groups used by the task.
2. Go to the **Project Management > scheduling Resource Management** page, locate the scheduling resource that the task uses, and click **server administration**, check to see if the status of the server is stopped or occupied by other tasks.
3. If the above troubleshooting does not resolve the issue, you can restart the service by executing the following command.

```
su - admin`  
/home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart`
```

2.9.3 RDS synchronization failure converted to JDBC format

Issue Description

When synchronizing data from RDS (MySQL/SQL Server/PostgreSQL) to user-created MySQL /SQL Server/PostgreSQL, the error message "DataX cannot connect to the corresponding database" appears.

Solution

Taking data synchronization from RDS (MySQL) to user-created SQL Server as an example, you must complete the following operations:

1. 1. Create a data source, and configure the data source as MySQL->JDBC format;
2. Use the new data source to configure synchronization tasks and re-execute them.



Note:

Note: For data synchronization between RDS (MySQL) -> RDS (SQL Server) and other cloud products, we recommend that you select RDS (MySQL) -> RDS (SQL Server) data source to configure synchronization tasks.

2.9.4 Synchronous table column name is a key and task fails

Issue Description

When you perform a synchronization task, the task fails as the column name of the synchronized table is a keyword.

Solution

Take MySQL data source as an example:

1. Create a new table aliyun, and the table creation statement is as follows:

```
create table aliyun (`table` int ,msg varchar(10));
```

2. Creates a view, giving the table column an alias.

```
create view v_aliyun as select `table` as col1,msg as col2 from aliyun;
```



Note:

- Table is the MySQL keyword, And the mosaic code will be reported wrong when the data is synchronized. So bypass this restriction by creating a view and assigning an alias to the table column.
- Keywords are not recommended as column names for tables.

3. The above statement gives an alias for a column that has a keyword, so when you configure a Data Synchronization task, you can choose the maid view instead of the aliyun table.



Note:

- The Escape Character for MySQL is 'key '.
- The escape characters for Oracle and PostgreSQL are "keywords".
- The Escape Character for SQL Server is the [Key].

2.9.5 How does the data synchronization task customize the table name?

Data backdrop

Data Background: The tables are identified by days (such as orders_20170310, orders_20170311 , and orders_20170312) on a one-table-for-one-day basis with the same table structure.

Achieving demand

Requirement: Create only one data synchronization task to import the table data of the previous day read from the source database into MaxCompute with a custom table name every morning (for example, on March 15, 2017, orders_20170314 table data is read automatically from the source database and imported, and so on).

Implementation

1. Log in to the dataworks console and navigate to the **data integration** page.
2. Create a Data Synchronization task in wizard mode, and select a table name as the name for the data source table when you configure it. Configure and save the synchronization task following the normal procedure.
3. Click **convert script** to convert the wizard mode to script mode.
4. Use a variable as the name of the source table in the script mode, such as orders_`\${tablename}`.

Assign the variable "tablename" a value in parameter settings of the task. Since the table names in "Data Background" are identified by days, which requires reading the table of the previous day, the assigned value is \$yyyymmdd-1.



Note:

Or you can use orders_`\${bdp.system.bizdate}` as the variable to name the source table.

After completing the configuration above, save and submit before following up.

2.9.6 Encoding formatting issues

After the data integration synchronization task is formatted, synchronization failure may occur and result in dirty data, synchronization success, but the data is messy.

Synchronization failed with dirty data generated

Issue Description

The data integration task failed and dirty data is generated due to encoding problem. The error log is shown as follows:

```
016-11-18 14:50:50.766 13350975-0-0-writer ERROR StdoutPluginCollector
- Dirty data:<br>
{"exception":"Incorrect string value: '\\xF0\\x9F\\x98\\x82\\xE8\\xA2...' for column 'introduction' at row 1","record":[{"byteSize":8,"index":0,"rawData":9642,"type":"LONG"}, {"byteSize":33,"index":1,"rawData":" Hello world! (http://docs.aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/pic/56134/cn_zh/1498728641169/%E5%9B%BE%E7%89%877.png)}
```

```
"", "type": "STRING"},
{"byteSize": 8, "index": 4, "rawData": 0, "type": "LONG"}], "type": "writer"}
2016-11-18 14:50:51. 265 [13350975-0-0-writer] warn maid $ task-roll
back this write, commit by writing one row at a time. Because: Java.
SQL. batchupdateexception: incorrect string value: '\ xq0 \ x9f \ x88
\ xB6 \ XeF \ xb8... 'For column' introduction 'at Row 1
```

Root cause

The user does the appropriate encoding formatting for the database, or when adding a data source, no encoding is set to maid, because only chain encoding supports synchronous emotiffs.

Solution

- When you add a data source in JDBC format, you need to modify the settings of the scanner, such as jdbc:mysql://xxx.x.x.x:3306/database? Com. mySQL. JDBC. faultinjection. serverchar setindex = 45, so that you can set the emotability on the data source to synchronize successfully.
- Modify the data source encoding format to utf8mb4. For example, you can modify the database encoding format of the RDS on the RDS console.

Synchronization succeeded with data garbled

Issue Description

The data synchronization task succeeded, but the data is garbled.

Root cause

Three reasons for garbled data:

- Source-side data is already out of order.
- The encoding for the database and the client is not the same;
- Browser encoding is not the same, resulting in preview failure or garbled data.

Solution

You can select a solution for different reasons that cause chaos.

- For the first reason, you must process the original data properly before starting the synchronization task.
- For the second reason, you must modify the encoding format.
- For the third reason, you must unify the encoding format before previewing the data.

2.9.7 Full-database migration data type

Currently, full-database migration only supports synchronizing data from MySQL databases (including MySQL databases on the RDS server) to MaxCompute. You can enter the full-database migration page from the added MySQL data source.

The following is a description of the data types that are set at the advanced level in the whole library migration.

The data source types supported by MySQL for the whole library migration source include tinyint, smallint, mediumint, Int, bigint, varchar, Char, tinytext, text, mediumtext, longtext, year, float, double, decimal, date, datetime, timestamp, time, and LOL.

The data source types supported by the target-side MaxCompute are bigint, String, double, datetime, and Boolean.

All those preceding MySQL-supported data types support converting to MaxCompute data source types.



Note:

Bit in MySQL, if it is more than bit (2), conversion with bigint, String, double, datetime, and Boolean is currently not supported. If it is bit (1), it is converted to a Boolean.

2.9.8 An error occurred when using username root to add MongoDB data source

Issue description

An error occurred when using username root to add MongoDB data source.

Root cause

When adding the MongoDB data source, you must use the username created by the database where the table you are required to synchronize resides, instead of the root.

Solution

For example, to import the name table, which is in the test database, enter test as the database name.

Enter the username created in a specific database, instead of root. For example, if the test database is specified, then use the account created in the test database as the username.

3 Data development

3.1 Solution

The data development mode has been upgraded to the three-level structure(project-solution-business flow), and the traditional directory organization mode is no longer used.

Project-solution-business flow

In the new version of DataWorks, the data development mode is upgraded to integrate different types of node tasks based on business types. Such a structure better facilitates code development by business. In the development process, development can be implemented across multiple business flows from a wider viewing angle. Based on the three-level structure of project-solution-business flow, the development process is re-defined to improve users' development experience.

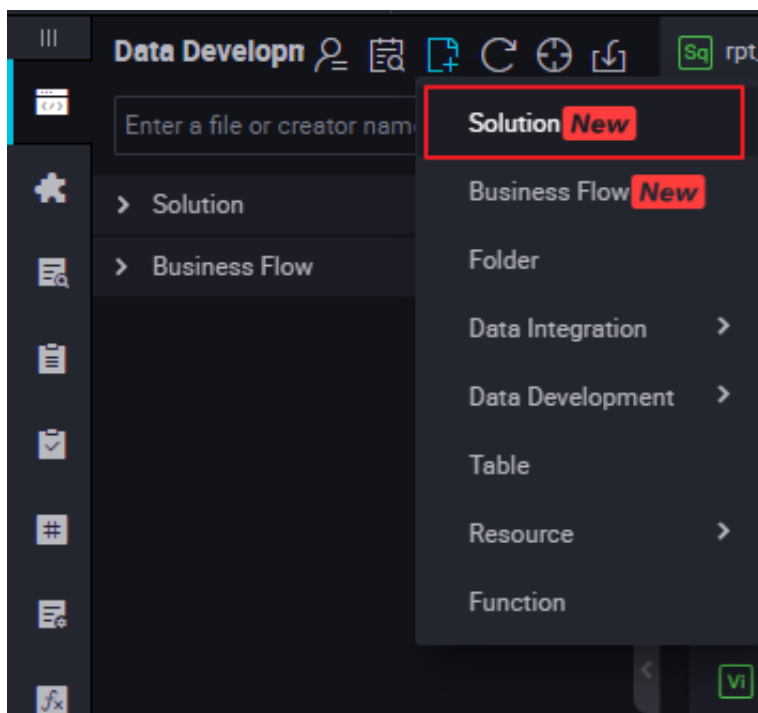
- **Project:** It is the basic unit for permission organization and used to control user permissions, such as development and O&M permissions. In the same project, all codes of project members can be developed and managed in a collaborative manner.
- **Solution:** Users can customize a solution by combining some business flows. Advantages:
 - A solution contains multiple business flows.
 - The same business flow can be reused in different solutions.
 - Immersive development can be implemented for a combined solution.
- **Business flow:** It is an abstract entity of business, which enables users to organize data code development from the business point of view. A business flow can be reused by multiple solutions. Advantages:
 - The business flow helps users better organize codes from the business point of view. It provides the task type-based code organization mode. It supports multiple levels of sub-directories (preferentially up to four levels).
 - The entire workflow can be viewed and optimized from the business point of view.
 - The business flow dashboard is provided to improve the development efficiency.
 - Release and O&M can be organized based on the business flow.

Immersive development experience

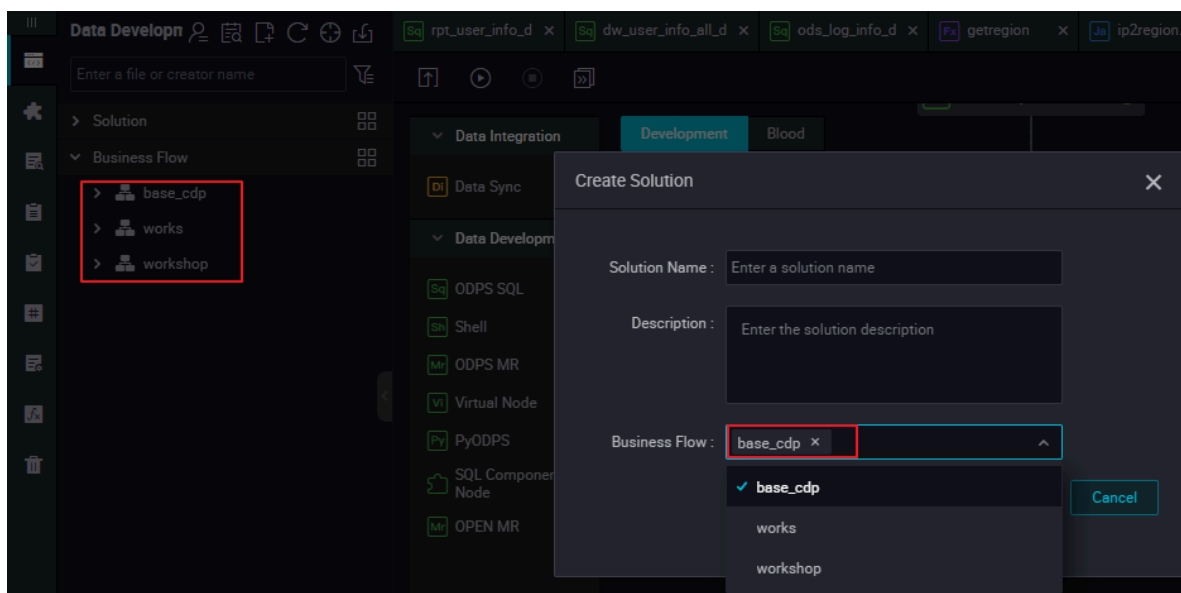
You can double-click any created solution to switch from the development area to the solution area. The directory displays only the content of the current solution. You are provided with a fresh

environment, and will not be troubled by other codes of the project that are irrelevant to the current solution.

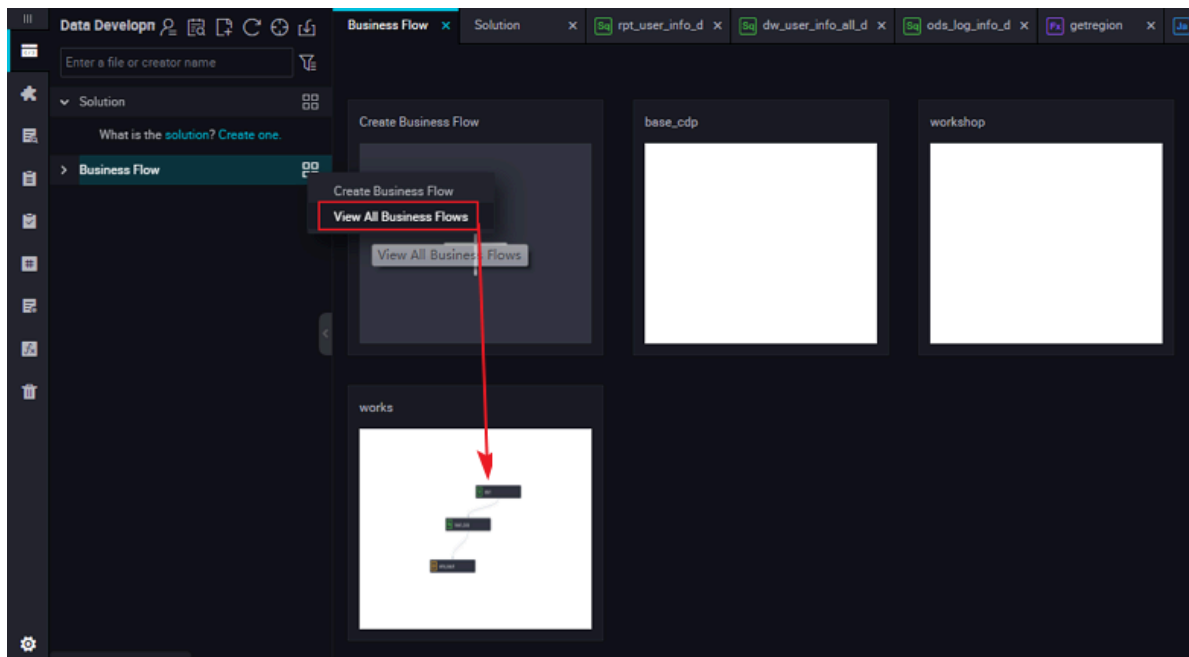
1. Go to the DataStudio page and create a solution.



2. Select the business flow to be viewed from the created solution.



3. Right-click **View All Business Flows** to view the nodes of the selected business flow or modify the solution.



4. Go to another page.

- Click **Publish** to go to the **Task Publish** page. Nodes in the **To be released** state under the current solution are displayed on this page.
- Click **O&M** to go to the **O&M Center > Periodic Instances** page. Periodic instances of all nodes under the current solution are displayed by default on this page.

A business flow can be reused by multiple solutions. You only need to immerse yourself in development of your own solutions. Other users can directly edit business flows referenced by you in other solutions or business flows, implementing collaborative development.

3.2 Encoding principles and standards for the SQL code

Encoding principles

SQL code is encoded as follows:

- Coding principles
- Code lines are clear, neat, and nice looking.
- The code lines are well arranged and have a good hierarchical structure.
- Comments must be provided whenever necessary to enhance readability of codes.
- Requirements in this convention are not required constraints for coding behaviors of developers. In practice, on the precondition that general requirements are not violated, rational deviations from this convention is acceptable if they are beneficial to code development, and this convention will be continuously improved and supplemented.

- All keywords and reserved words used in SQL codes are in lower case. Examples of such words include select, from, where, and, or, union, insert, delete, group, having, and count.
- In addition to keywords and reserved codes used in SQL codes, other codes (such as field names and table alias) must be in lower case.
- Four spaces are equivalent to an indentation unit. All indentions must be the integer multiples of an indentation unit and aligned according to the code hierarchy.
- It is prohibited to use the select * operation. The column name must be specified in all operations.
- The corresponding brackets must be on the same column.

SQL coding specification

The code specification for SQL code is as follows:

- Code header

Information, such as the subject, function description, author, and date, must be added to the code header. The log and title bars must be reserved so that later users can add change records. Note that each line cannot have more than 80 characters. The following is a template:

```
-- *****
-- ** Subject:      AGDS Risk application
-- ** function      Credit index interface
-- ** description:   chenfeng
-- ** creator:      2014-05-23
-- ** create date:   2014-05-23
-- ** Modify the log:
-- ** Modify the date:      Modifier      Modifies the content
-- ** 2014-05-23           chenfeng      create
-- *****
```

```
-- MaxCompute(ODPS) SQL
--
-- *****
-- Subject: Transaction
-- Function description: Transaction refund analysis
-- Author: With code
-- Create time: 20170616
-- Change log:
-- Modified on  Modified by  Content
-- yyyymmdd name comment
-- 20170831 Without code Add a judgment on the transaction biz_type=
1234
--
-- *****
```

- Field arrangement requirements
 - For fields selected for the select statement, each field occupies one line.
 - One indentation next to the word "select" is directly followed by the first selected field. That is, the field is two indentions away from the start of the line.

- Each of other fields starts with two indentions, followed by a comma (,) and then the field name.
- The comma (,) between two fields is right before the second field.
- The as statement must be in the same line as the related fields. We recommended that the "as" statements with multiple fields be aligned in the same column.

```
select  channel_id      as channel_id
        ,trade_channel_desc  as trade_channel_desc
        ,trade_channel_edesc as trade_channel_edesc
        ,inst_date         as inst_date
        ,trade_iswap       as trade_iswap
        ,channel_type      as channel_type
        ,channel_second_desc as channel_second_desc
from    (
```

- Insert sub-statement arrangement requirements

The Insert sub-statement must be written on the same row. Line feed is prohibited.

- Select sub-statement arrangement requirements

Sub-statements used by the select statements, such as from, where, group by, having, order by, join, and union, must conform to the following requirements:

- Line feed.
- The sub-statements must be left aligned with the select statement.
- Two indentions must be reserved between the first letter of a sub-statement and its subsequent code.
- The logical operators (such as "and" and "or") in a "where" sub-statement must be left aligned with where.
- If the length of a sub-statement exceeds two indentions, add a space to the sub-statement and then write the subsequent code, such as "order by" and "group by".

```
select      trim(channel) channel
            ,min(id)      id
from        ods_trd_trade_base_dd
where       channel is not null
and         dt = ${tmp_uuuuummdd}
and         trim(channel) <> ''
group by    trim(channel)
order by    trim(channel)
```

- Requirements for spacing before and after an operator One space must be reserved before and after an arithmetic operator or a logical operator. Operators must be written on the same line unless the line length exceeds 80 characters.

```

select      trim(channel) channel
            ,min(id)      id
from        ods_trd_trade_base_dd
where       channel is not null
and         dt = ${tmp_uuuuumdd}
and         trim(channel) <> ''
group by    trim(channel)
order by    trim(channel)

```

- Compiling of the "case" statement

In a "select" statement, the "case" statement is used to judge or assign values to fields. Correct compiling of the "case" statement is critical for enhancing readability of code lines.

The following conventions are stipulated for compiling of the "case" statement:

- The "when" sub-statement is in the same line as the "case" statement and starts after one indentation.
- One "when" sub-statement occupies one line. Line feed is acceptable if the statement is too long.
- A "case" statement must contain the "else" sub-statement. The "else" sub-statement must be aligned with the "when" sub-statement.

```

, case      when p1.trade_from = '3008' and p1.trade_email is null then 2
            when p1.trade_from = '4000' and p1.trade_email is null then 1
            when p9.trade_from_id is not null then p9.trade_from_id
end         as trade_from_id
,p1.trade_email      as partner_id

```

- Nesting query compiling specification

Nesting sub-query is often used in ETL development of the data warehouse system. Therefore, it is important to arrange codes in a hierarchical manner. Example:

```

select      p.channel
            ,rownumber() order_id
from        (
select      s1.channel
            ,s1.id
from        (
select      trim(channel)      as channel
            ,min(id)          as id
from        ods_trd_trade_base_dd
where       channel is not null
and         dt = ${tmp_yyyymmdd}
and         trim(channel) <> ''
group by    trim(channel)
            ) s1
left outer join
            dim_trade_channel s2
on          s1.channel = s2.trade_channel_edesc
where       s2.trade_channel_edesc is null
order by    id
            ) p
;

```

- Table alias definition convention
 - Alias must be added to all tables. Once an alias is defined for an operation table in a "select" statement, the alias must be used whenever the statement references the table. To facilitate code compiling, alias must be simple and concise whenever possible and keywords must be avoided.
 - The table alias is defined using simple characters. We recommended that aliases be defined in alphabetical order.
 - Before multi-layered nesting sub-query of an alias, the hierarchy must be shown. The SQL statement alias is defined by layer. Layer 1 to layer 4 are represented by P (Part), S (Segment), U (Unit), and D (Detail), respectively. Alternatively, layer 1 to layer 4 can be represented by a, b, c, and d. Sub-statements at the same layer are differentiated from each other by the numbers (such as 1, 2, 3, and 4) behind the letter that represents the layer. A comment can be added for a table alias if necessary.

```

select      p.channel
            ,rownumber() order_id
from        (
            select  s1.channel
                  ,s1.id
            from      (
                    select  trim(channel)      as channel
                          ,min(id)            as id
                    from    ods_trd_trade_base_dd
                    where   channel is not null
                    and     dt = ${tmp_yyyymmdd}
                    and     trim(channel) <> ''
                    group by trim(channel)
                ) s1
            left outer join
                dim_trade_channel s2
            on      s1.channel = s2.trade_channel_edesc
            where   s2.trade_channel_edesc is null
            order by id
        ) p
;

```

- SQL comments
 - A comment must be added for each SQL statement.
 - The comment for each SQL statement exclusively occupies one line, and is placed in front of the statement.
 - The comment of a field comes next to the field.
 - Comments must be added for branch condition expressions that are not easy to understand.
 - Comments must be added for important calculations to describe their functions.
 - If a function is too long, its statement must be segmented based on the implemented functions, and comments must be added to describe each segment.

- For a constant or variable, it is required to add a comment to explain the meaning of the saved value, and optional to add a comment to explain the valid value range.

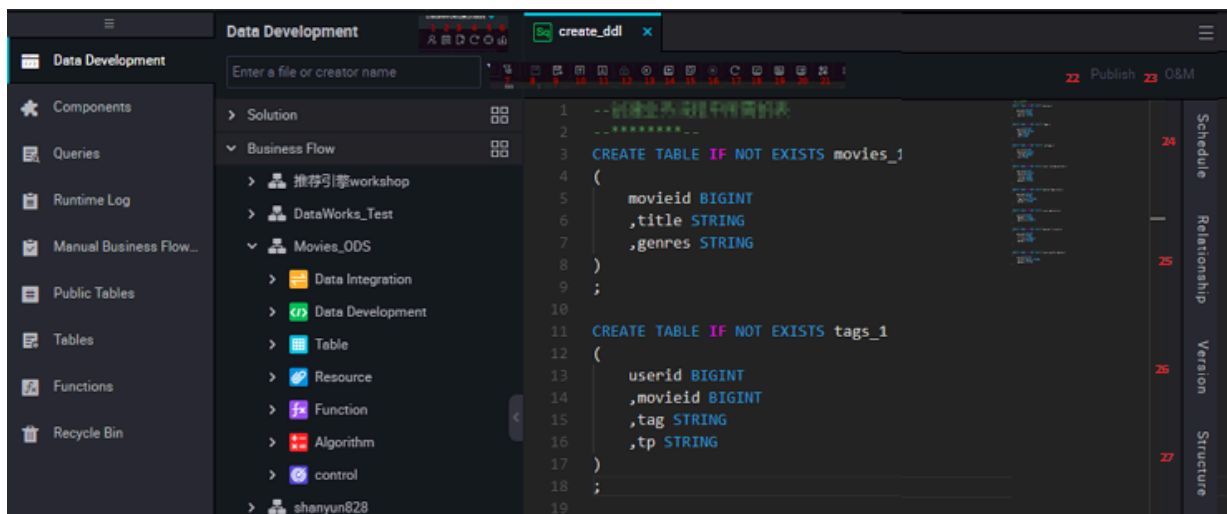
```

-- *****
-- STEP1:      Clean up data partition on      tmp_dws_tbd_alijr_user_relation_dd_5
--              that day.
-- *****

```

3.3 Console functions

3.3.1 Introduction to console



The interface function points are described below:

No.	Feature	Description
1	Show My Files	Click it to view nodes under your own account in the current column.
2	Code Search	Click it to search for a code or a code segment.
3	[+]	Click it to create a solution, business flow, folder, node, table, resource, or function entry.
4	Reload	Click it to refresh the current directory tree.
5	Locate	Click it to locate the position of the selected file.
6	Import	Click it to import local data to an online table. Pay attention to the encoding format.
7	Filter	Click it to filter nodes based on the specified conditions.
8	Save	Click it to save the current code.
9	Save as Query File	Click it to save the current code as a temporary file, which is displayed in the Temporary query column.

No.	Feature	Description
10	Submit	Click it to submit the current node.
11	Submit and Unlock	Click it to submit the current node and unlock the node to edit the code.
12	Steallock	Click it to edit a node if you are not the owner of this node .
13	Run	Click it to run the code of the current node.
14	Run After Setting Parameters	Click it to run the code of the current node using the configured parameters.
15	Precompile	Click it to edit and test parameters of the current node.
16	Stop Run	Click it to stop the code that is being run.
17	Reload	Click it to refresh the page and return to the previously saved page.
18	Run Smoke Test in Development Environment	Click it to test the code of the current node in the development environment.
19	View Smoke Test Log in Development Environment	Click it to view the run log of a node that runs in the development environment.
20	Go to Scheduling System of Development Environment	Click it to go to the O&M center of the development environment.
21	Format	Click it to sequence codes of the current node. It is often used when the code on a single line is too long.
22	Publish	Click it to publish the submitted code. After the code is published, the code is in the production environment.
23	O&M	Click it to go to the O&M center of the production environment.
24	Scheduling Configuration	Click it to configure the scheduling attributes, parameters , and resource groups of a node.
25	Relationship	Click it to view the relationship between tables used by the code.
26	Version	Click it to view the submission and publish records of the current node.

No.	Feature	Description
27	Structure	Click it to view the code structure of the current node. If the code is too long, you can quickly locate a code segment based on the key information in the structure.

3.3.2 Version

A version is a submission and release record of the current node, each submission generates a new version. You can check the related status, change type, and release remarks as required to facilitate operations on the node.



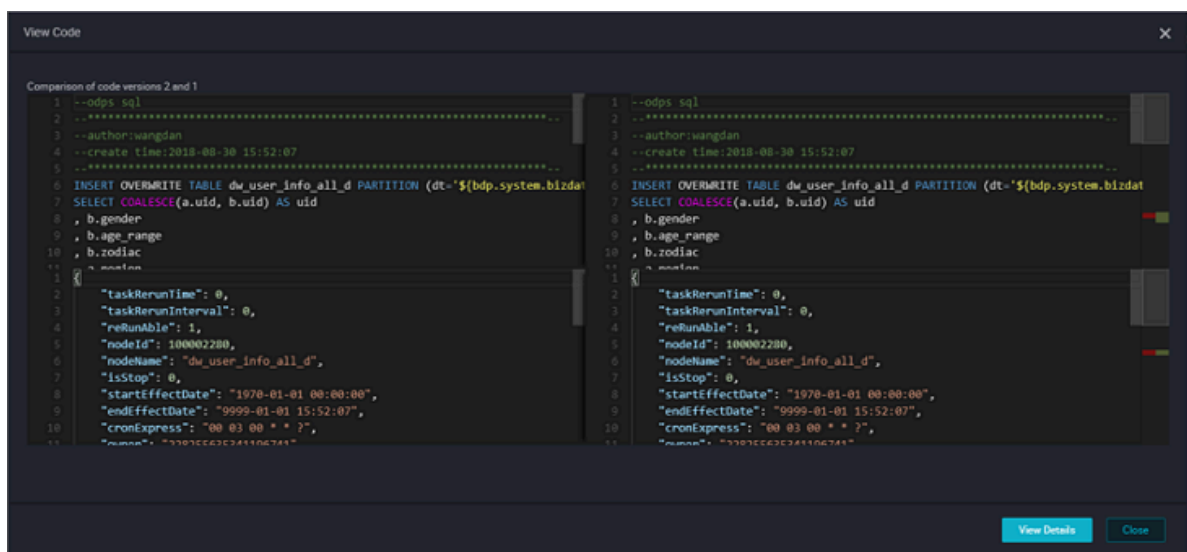
Note:

Only a submitted node has the version information.

<input type="checkbox"/>	5000118 87	V7	dataworks_dem o2	2018-09-02 10:3 9:57	Edit	Published	test	View Code Roll Back		Structure
<input type="checkbox"/>	5000118 87	V6	dataworks_dem o2	2018-09-02 10:3 7:47	Edit	Published	123	View Code Roll Back		Version
<input type="checkbox"/>	5000118 87	V5	dataworks_dem o2	2018-09-02 10:3 6:28	Edit	Published	test	View Code Roll Back		Relationship
<input type="checkbox"/>	5000118 87	V4	dataworks_dem o2	2018-09-02 10:3 3:54	Edit	Published	test	View Code Roll Back		Structure
<input type="checkbox"/>	5000118 87	V3	dataworks_dem o2	2018-09-02 10:3 0:19	Edit	Published	test	View Code Roll Back		Version
<input type="checkbox"/>	5000118 87	V2	wangdan	2018-08-31 10:2 1:19	Edit	Published	workshop user portrait part is w ritten logically.	View Code Roll Back		Relationship
<input type="checkbox"/>	5000118 87	V1	wangdan	2018-08-30 17:3 7:55	Add	Published	workshop user portrait part is w ritten logically.	View Code Roll Back		Structure

- File ID: ID of the current node.
- Version: A new version is generated for each release. The first release is V1, the second modification is V2, and so on.
- Submitter: Operator who submits and releases the node.
- Submission Time: Version release time. If a version is submitted and then released, the release time covers the submission time. By default, the last release time of the operation is recorded.
- Change Type: Operation history of the current node. It is set to Added if the node is first released, and set to Modified if the node is modified.
- Status: Operation status record of the current node.

- Remarks: Change description of the current node when it is submitted. It facilitates other personnel to locate the related version when operating the node.
- Action: You can select Code and Roll Back in this column.
 - View Code: Click it to view the version code and precisely search for a record version to be rolled back.
 - Roll Back: Click it to roll back the current node to a previous version as required. You must submit the node for release again after rolling back.
- Compare: Click it to compare the code and parameters of two versions.



Click **View Details** to go to the details page and compare the code and scheduling attribute changes.



Note:

Only two versions can be compared. One or more than three (including three) nodes cannot be compared.

3.3.3 Structure

The structure is based on the current Code, which parses the process diagram that runs under SQL, help users quickly review the edited SQL situation, so that it can be easily modified and viewed.

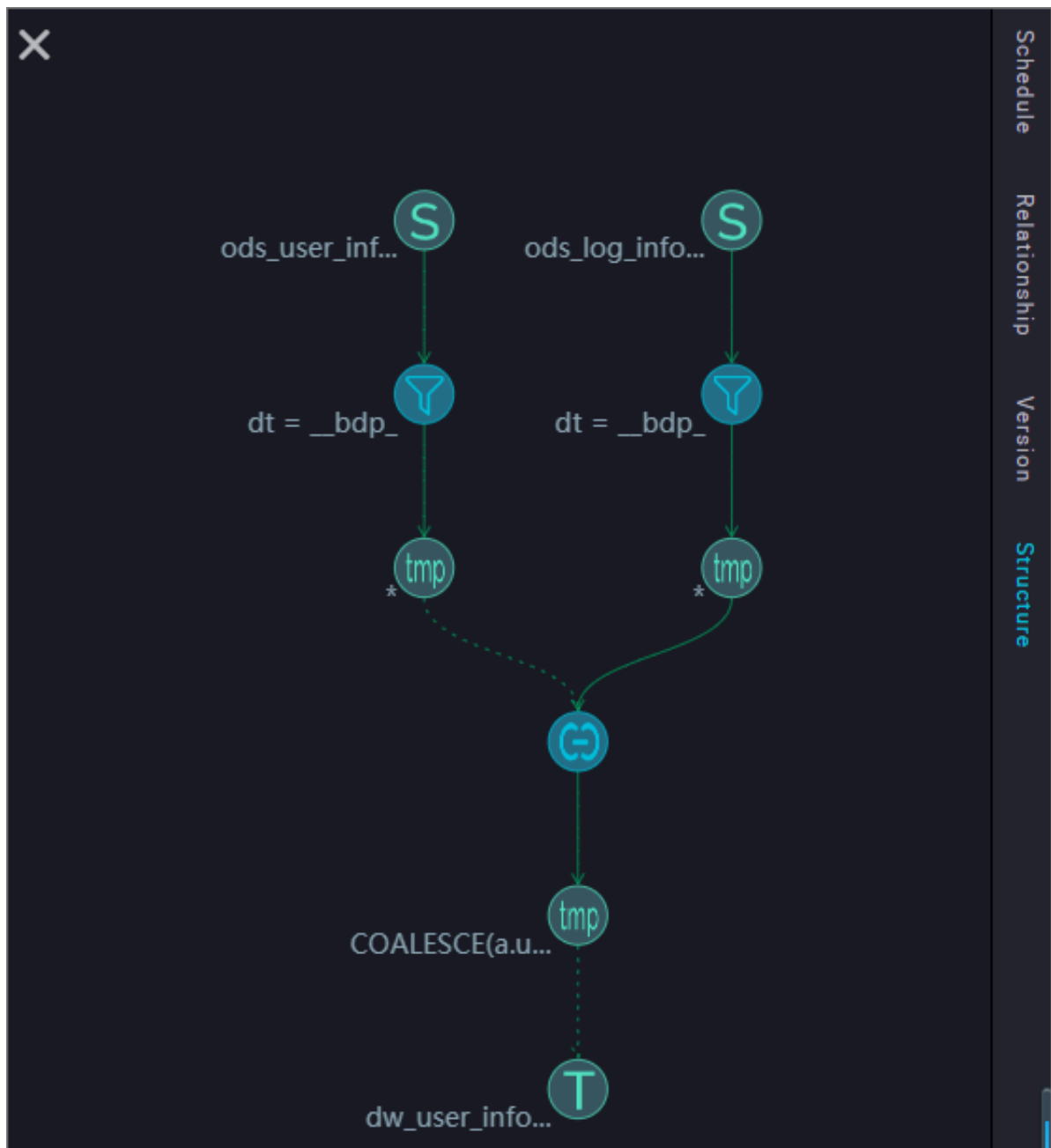
Structure

As shown in SQL:

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.
bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
, b.gender
```

```
    , b.age_range
    , B. flavdiac
    , a.region
    , a.device
    , a.identity
    , a.method
    , a.url
    , a.referer
    , a.time
FROM (
    VALUES
    From fig
    WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
    VALUES
    FROM ods_user_info_d
    WHERE dt = ${bdp.system.bizdate}
) b
on a.uid = b.uid ;
```

According to this Code, the structure is parsed:



When the mouse is placed in a circle, the corresponding explanation appears:

1. Source table: the target table for the SELECT query.
2. Filter: filters the specific partitions in the table that you want to query.
3. In the first part of the intermediate table (query view): place the results of the query data into a temporary table.
4. Join: mosaic the results of the two-part query through join.
5. In the second section, the intermediate table (the query view): summarize the results of the join into a temporary table, this temporary table exists for three days and is automatically cleared three days later.

6. Target table (insert): inserts the data obtained in the second part into the table in insert override

3.3.4 Relationship

kinship relations show the relationships between the current node and other nodes. This relationship shows two parts: the dependency diagram and the internal relationship map.

Dependency Graph

Depending on the dependency of the node, the dependency graph shows whether the dependency of the current node is what it expects, if not, you can return to the Schedule configuration interface to reset.



Internal relationship Map

The internal relationship map is parsed Based on the node's code, for example:

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
  , b.gender
  , b.age_range
  , B.flavdiac
  , a.region
  , a.device
  , a.identity
  , a.method
  , a.url
  , a.referer
  , a.time
FROM (
  VALUES
  From fig
```

```

WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
  VALUES
  FROM ods_user_info_d
  WHERE dt = ${bdp.system.bizdate}
) b
on a.uid = b.uid ;

```

According to this SQL, the following internal relationship map is resolved, parses an output table that will be used as a join mosaic to show the relationship relationship between the tables:



3.4 Business flow

3.4.1 Business flow

A business flow integrates different types of node tasks by business type. Such a structure better facilitates code development by business. The system organizes data development centered by the business flow and provides container dashboards of various types of development nodes. In this way, tools, optimization operations, and management operations are arranged based on objects on the data dashboards, making development and management more convenient and intelligent.

DataWorks code structure

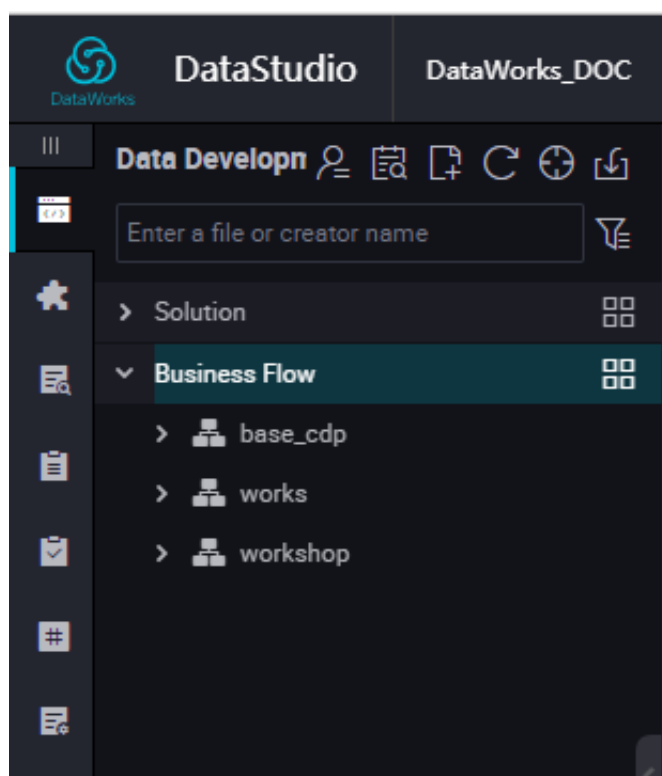
A work project supports computing engines of multiple types. A work project contains multiple business flows, each of which is a collection of various types of objects that are systematically associated with each other. You can view each business flow in the automatically generated

flowcharts. Objects in a process can be of the types such as data integration task, data development task, table, resource, function, algorithm, and operation flow.

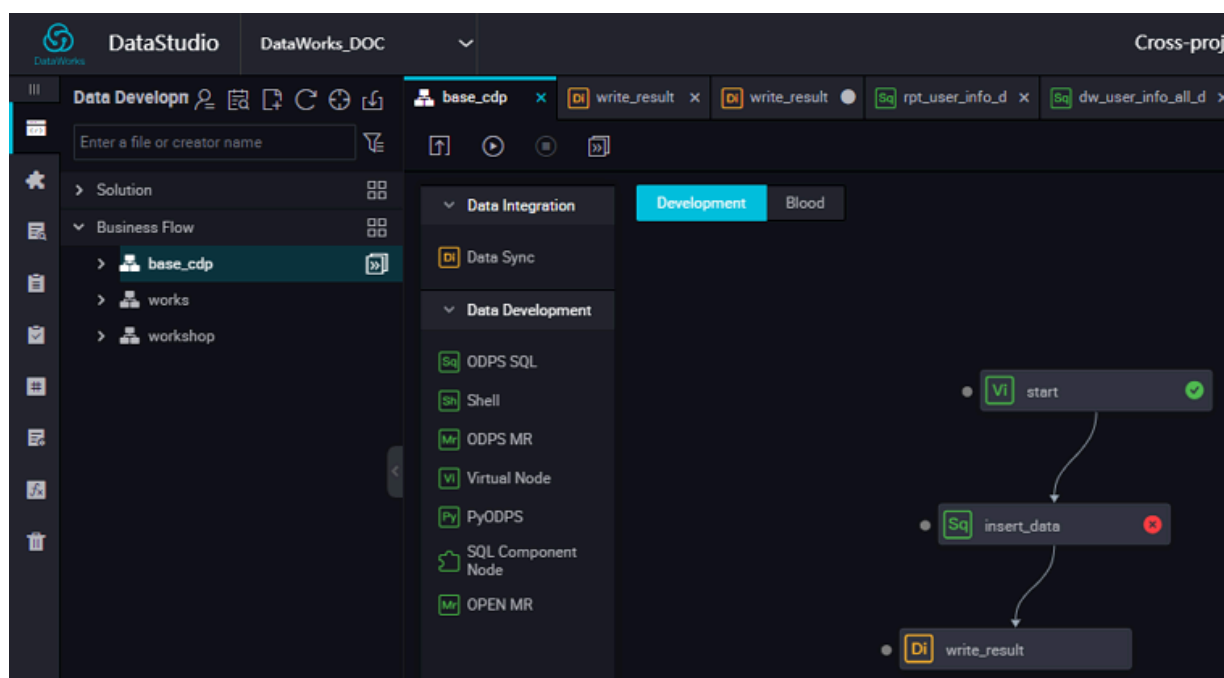
Each object type corresponds to an independent folder, under which sub-folders can be created. To facilitate management, we recommend that you create a maximum of four layers of sub-folders. If more than four layers of sub-folders are created, the planned business flow structure is too complex. We recommend that you split the business flow to one or more business flows and manage the related business flows in one solution. This code organization method is more efficient.

Business flow composition

1. Data Integration: see [Data integration node](#).
2. Data Development: see [Node type overview](#).
3. Table: see [Table Management](#).
4. Resources: see [Introduction to resources](#).
5. Functions: see [Introduction to functions](#).

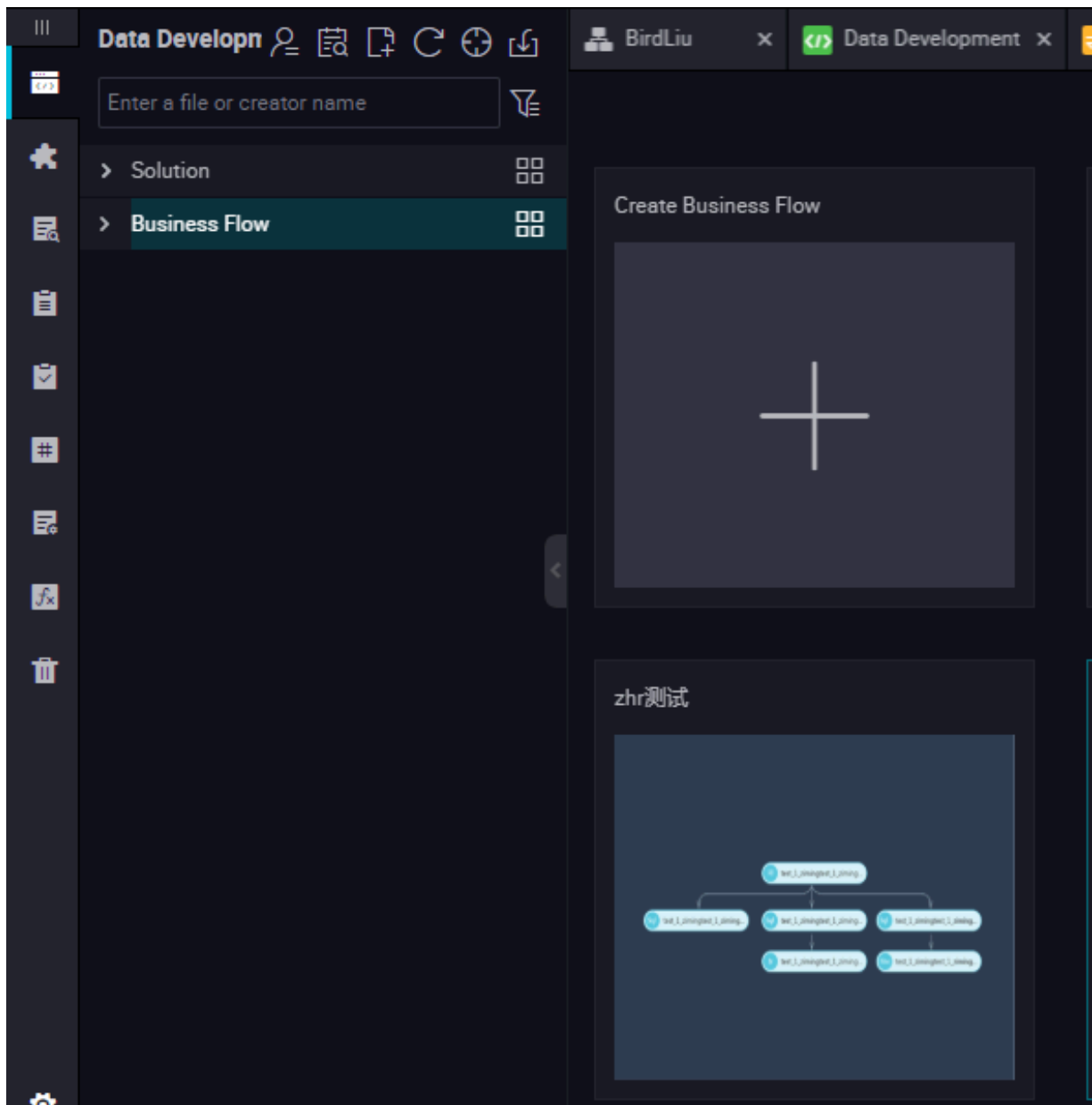


Double-click the name of a business flow node to view the relationship between nodes in the business flow in a workflow chart.



Business flow dashboard

You can check all business flows under a project on the business flow dashboard.

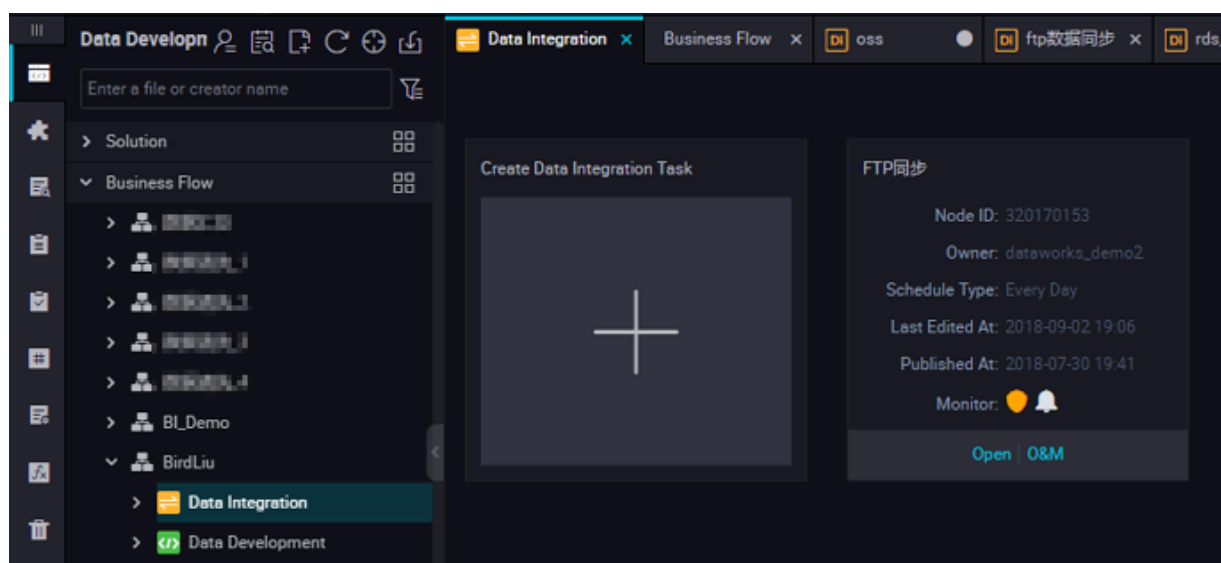


Business flow object dashboard

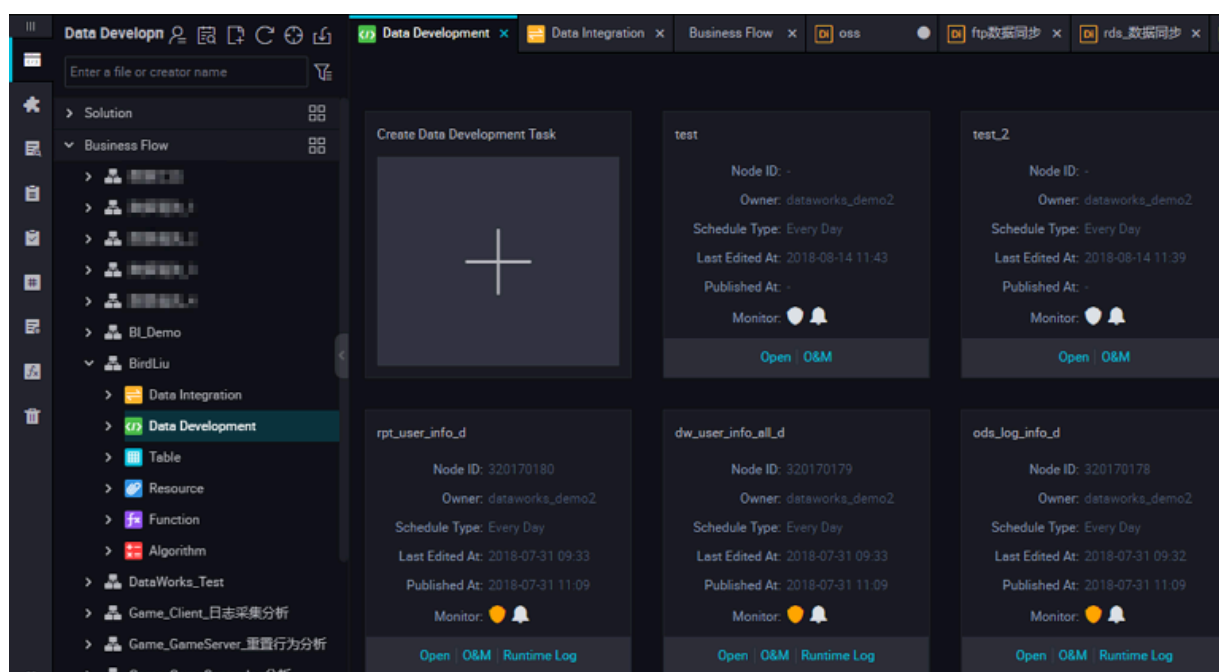
An object set dashboard is created for each type of objects in a business flow, and each object corresponds to an object card on the dashboard. You can attach the operation and optimization suggestions to the corresponding object so that the object management is intelligent and convenient.

For example, on the object card of the data development task, the baseline strong protection and custom reminder icons are displayed, facilitating you to understand the current protection status of the task. You can double-click the icon of each object under Business Flow to open the dashboard of the object type.

Data Integration task dashboard



Data Development task dashboard

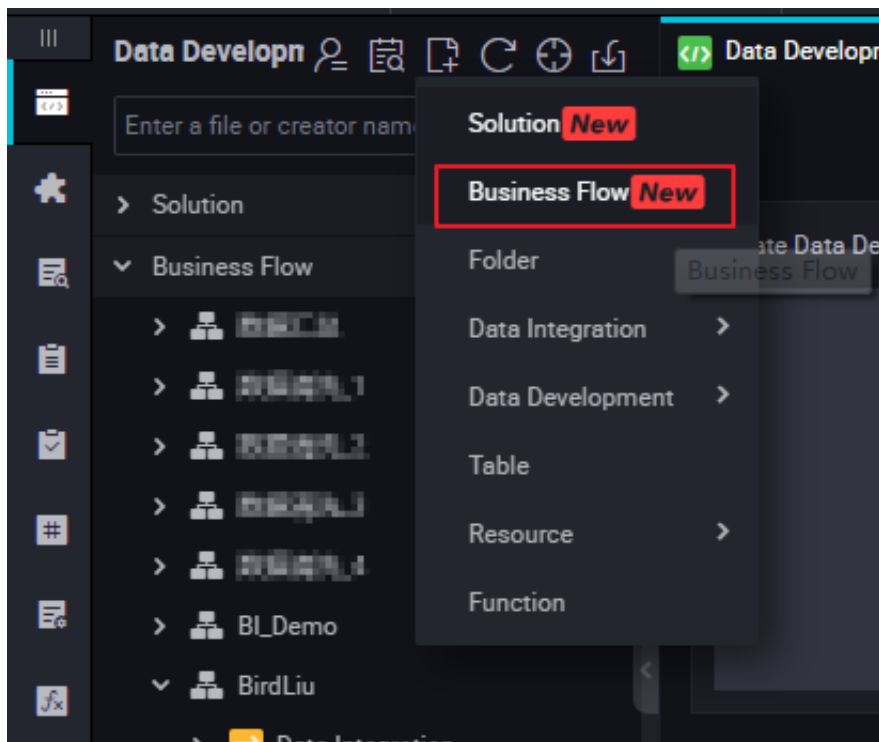


Note:

The number of nodes in a single business flow may not exceed 100.

Create a business flow

Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



3.4.2 Resource

If you want to use .jar, you need to upload it to the project's resource.

You can upload text files, ODPS tables, and various compressed packages (such as .zip, .tgz, .tar.gz, .tar, and .jar) as different types of resources to ODPS. Then, you can read or use these resources when running UDFs or MapReduce.

ODPS provides APIs for reading and using resources. The following types of ODPS resources are available:

- File
- Archive: The compression type is identified by the extension in the resource name. The following compressed file types are supported: .zip, .tgz, .tar.gz, .tar, and .jar.
- Jar: compiled Java jar packages.

On DataWorks, the process of creating a resource is a process of adding a resource. Currently, DataWorks supports addition of three types of resources in a visual manner, including the jar, file resources. The newly created entries are the same. The differences are as follows:

- Jar resource: You need to compile the Java code in the offline Java environment, compress the code into a jar package, and upload the package as the jar resource to ODPS.
- Small files: These resources are directly edited on DataWorks.

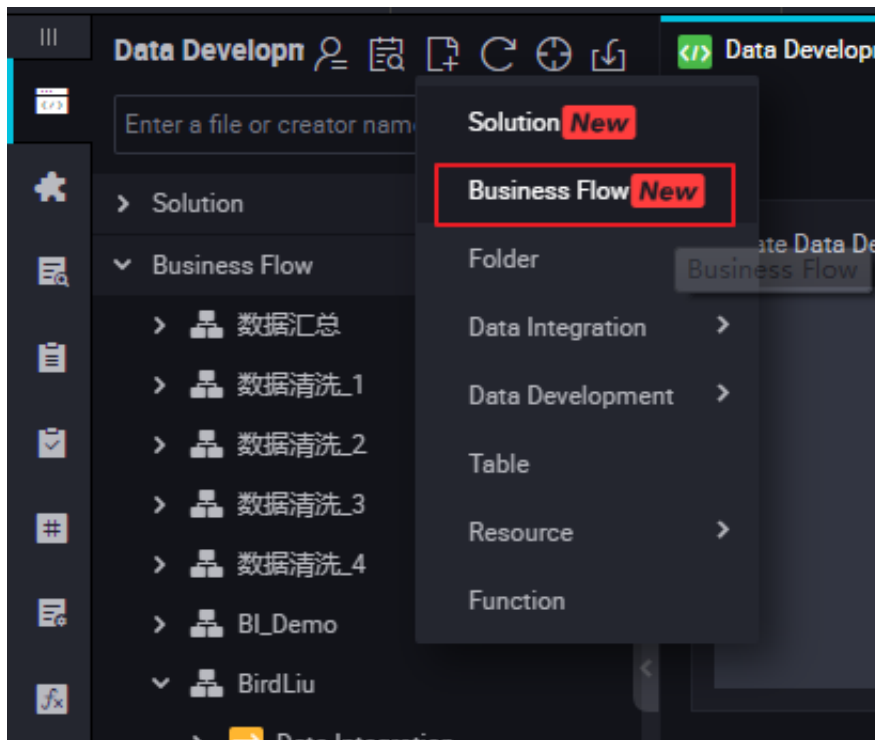
- File resource: When creating file resources, you need to select big files. You can also upload local resource files.

**Note:**

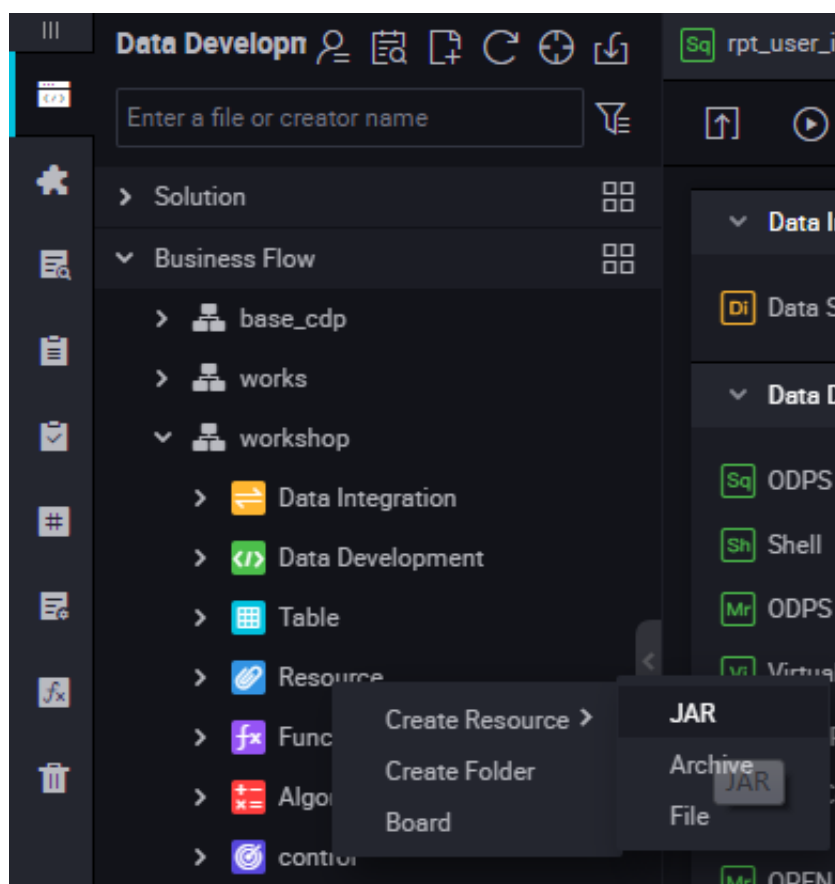
The resource package to be uploaded can not exceed 30 MB.

Create a resource instance

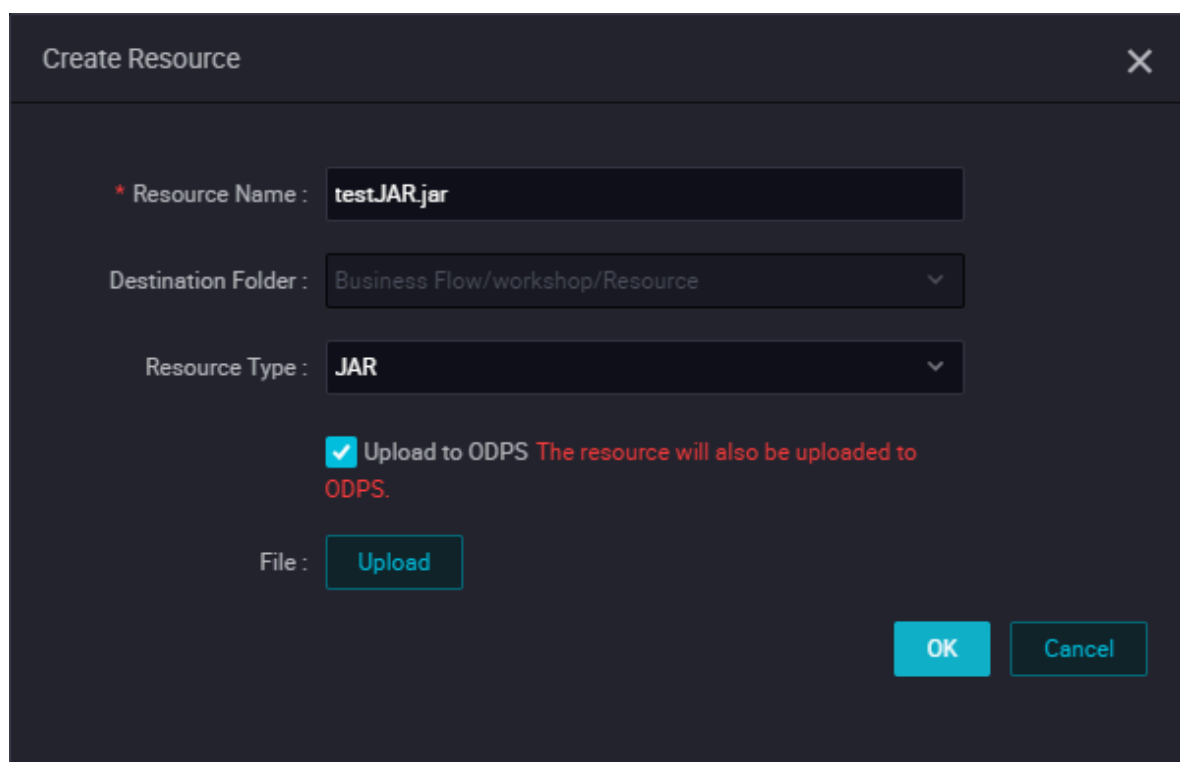
1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



2. Right-click **Resource**, and select **Create Resource > jar**.



3. The **Create Resource** dialog box is displayed. Enter the resource name according to the naming convention, set the resource type to jar, select a local jar package to the uploaded, and click **OK** to submit the package in the development environment.



**Note:**

- If this jar package has been uploaded on the ODPS client, you must deselect **Uploaded as the ODPS resource**. In this upload, the resource will also be uploaded to ODPS. Otherwise, an error will be reported during the upload process.
- The resource name is not necessarily the same as the name of the uploaded file.
- Naming convention for a resource name: a string of 1 to 128 characters, including letters, numbers, underlines, and dots. The name is case insensitive. If the resource is a jar resource, the extension is .jar.

4. Click **OK** to submit the resource to the development scheduling server.

Upload Resource

Saved Files : ip2region.jar

Unique Resource Identifier : OSS-KEY-I60u5o1g7t3g9uuim6j6polz

☒ Upload to ODPS The resource will also be uploaded to ODPS.

Re-upload :

5. Release a node task.

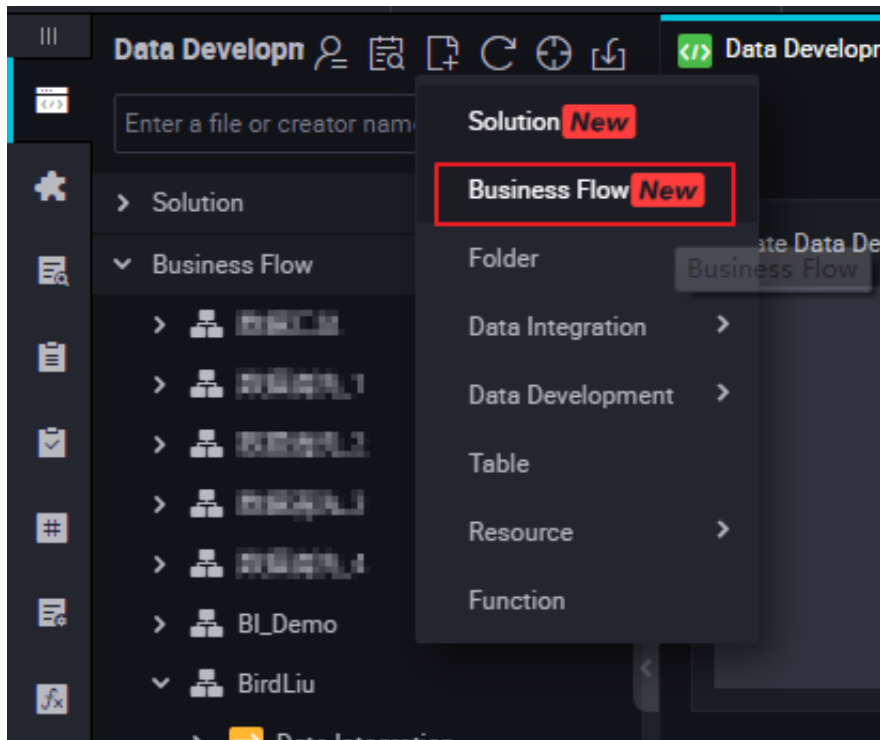
For more information about the operation, see [Publish a task](#).

3.4.3 Register the UDFs

Currently, the Python and Java APIs support implementation of UDFs. To compile a UDF program, you can upload the UDF code by [Adding resources](#) and then register the UDF.

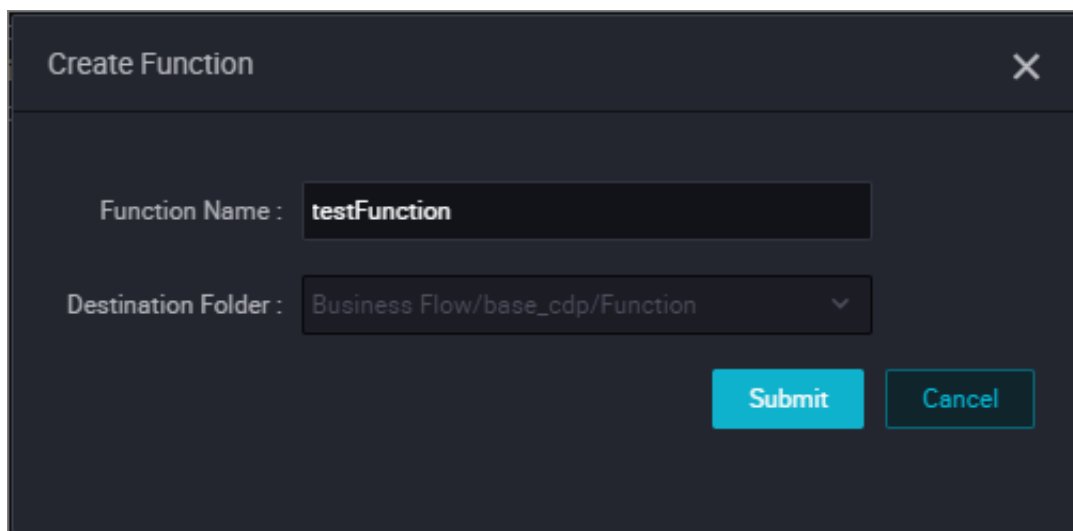
UDF registration procedure:

1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



2. In the offline Java environment, edit the program, compress the program into a jar package, create a jar resource, and submit and release the program. For more information, see [Create resources](#).
3. Create a function.

Right-click **Function**, select **Create Function**, and enter the configuration of the new function.



4. Edit the function configuration

Registry Function

Function Name : testFunction

* Class Name : test

* Resources : test.JAR.jar

Description :

Command Format :

Parameters :

- Class Name: name of the main class that implements the UDF.
- Resource List: Name of the resource in the second step. If there are multiple resources, separate them using commas.
- Description: UDF description. It is optional.

5. Submit the task.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Release a task

For more information about the operation, see [Publish a task](#).

3.5 Node type

3.5.1 Node type overview

Seven types of nodes are provided in DataWorks, which are applicable to different use cases.

Virtual node task

A virtual node is a control node that does not generate any data. Generally, it is used as the root node for overall planning of nodes in the workflow. For more information about the virtual node task, see [Virtual node](#).



Note:

The final output table of a workflow contains multiple branch input tables. Virtual nodes are usually used if these input tables do not have dependency between them.

ODPS SQL task

An ODPS SQL task enables you to edit and maintain SQL code directly on the Web, and easily implement running, debugging, and collaborative development. DataWorks also provides code version management, automatic resolution of upstream and downstream dependencies, and other capabilities. For more information about the examples, see [ODPS SQL node](#).

DataWorks uses the project of MaxCompute by default as the space for development and production, so that the code content of the ODPS SQL node follows the syntax of MaxCompute SQL. MaxCompute SQL adopts the syntax like that of Hive, which can be considered as a subset of standard SQL. However, MaxCompute SQL cannot be equated with a database, because it does not possess many features that a database does, such as transactions, primary key constraints, and indexes.

For more information about the specific MaxCompute SQL syntax, see [SQL overview](#).

ODPS MR task

MaxCompute supports the MapReduce programming APIs, whose Java APIs can be used to compile MapReduce program for data processing in MaxCompute. You can create ODPS MR nodes and use them for task scheduling. For more information about the examples, see [ODPS MR node](#).

PyODPS task

MaxCompute provides the [Python SDK](#), which can be used to operate MaxCompute.

DataWorks also provides the PyODPS task type and integrates the Python SDK of MaxCompute. You can directly edit the Python code to operate MaxCompute on a PyODPS node of DataWorks. For more information, see [PyODPS node](#).

SQL component node

An SQL component is an SQL code process template containing multiple input and output parameters. To handle an SQL code process, one or more source data tables are imported, filtered, joined, and aggregated to form a target table required for new business. For more information, see [SQL Component node](#).

Data synchronization task

A data synchronization node task is a stable, efficient, and automatically scalable external data synchronization cloud service provided by the Alibaba Cloud DTplus platform. With the data synchronization node, you can easily synchronize data in the business system to MaxCompute. For more information, see [Data integration node](#).

3.5.2 Data integration node

Currently, the data integration task supports the following data sources: MaxCompute, MySQL, DRDS, SQL Server, PostgreSQL, Oracle, MongoDB, DB2, OTS, OTS Stream, OSS, FTP, Hbase, LogHub, HDFS, and Stream. For details about more supported data sources, see [Supported data sources](#).

Configure a integration task

For more information, see [Create a synchronization task and the reader](#)

Node scheduling configuration.

Click the **Scheduling Configuration** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

Submit the node

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl+S to submit (and unlock) the node to the development environment.

Publish a node task

For more information about the operation, see Release management.

Test in the production environment.

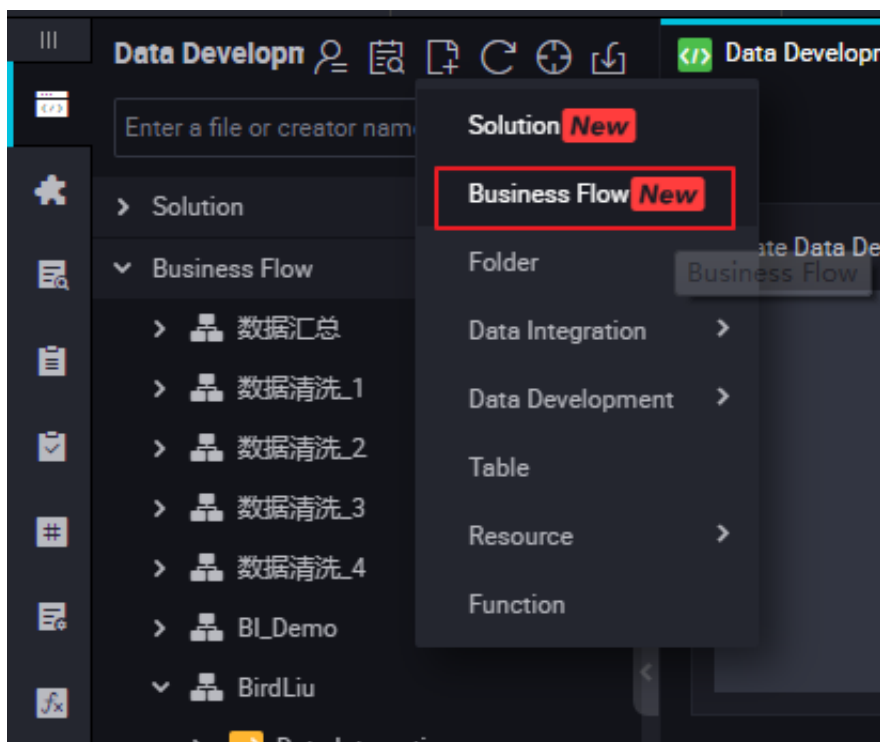
For more information about the operation, see [Cyclic task](#).

3.5.3 ODPS SQL node

ODPS SQL adopts the syntax similar to that of SQL, and is applicable to the distributed scenario in which the amount of data is massive (TB-level) but the real-time requirement is not high. It is an OLAP application oriented to throughput. Because it takes a long time to complete the process from preparation to submission of a job, ODPS SQL is recommended if a business needs to handle tens of thousands of transactions.

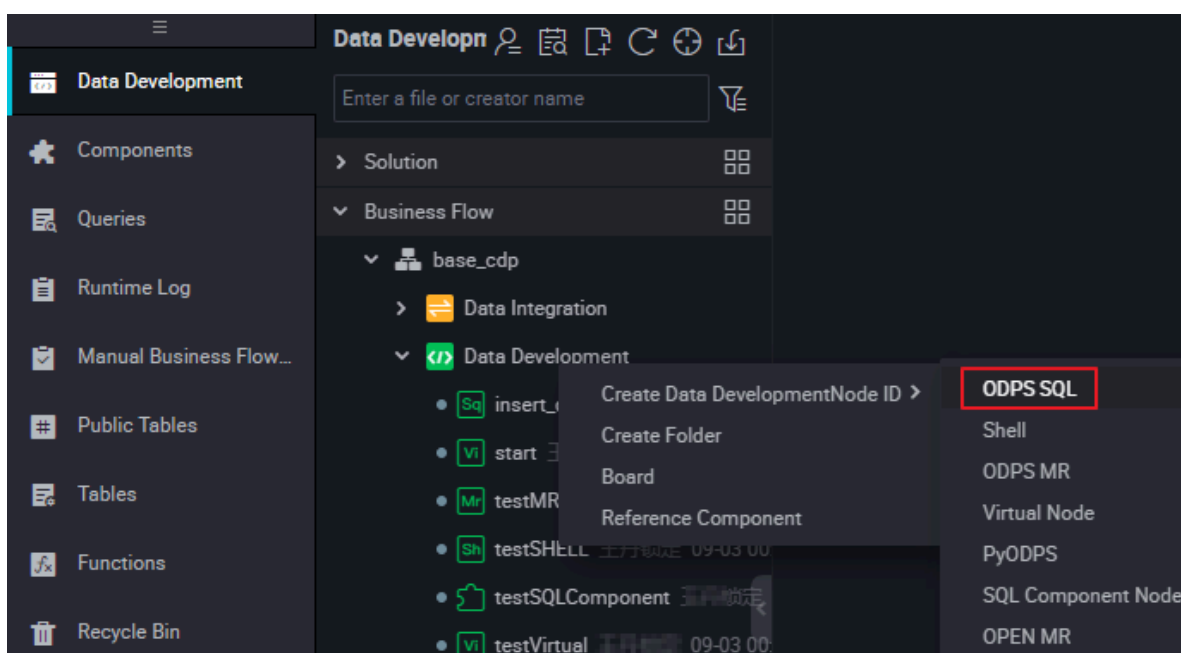
1. Create a business flow.

Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



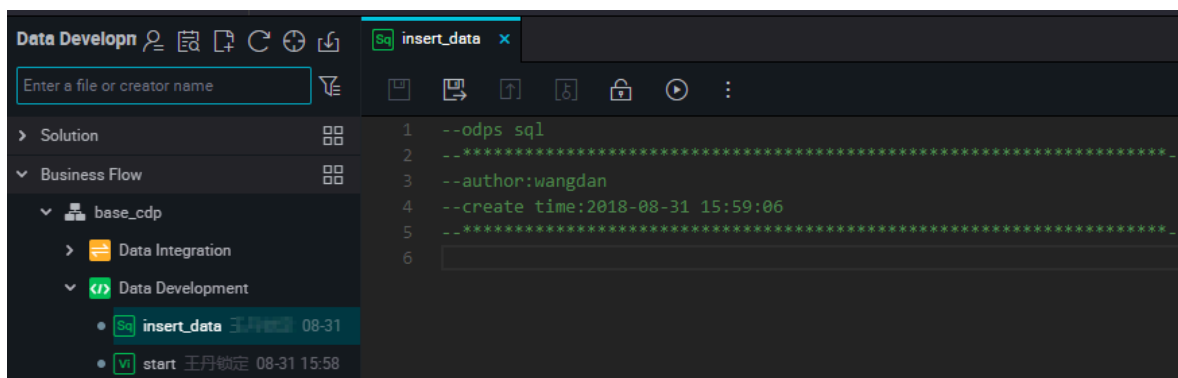
2. Create ODPS SQL node.

Right-click **Data Development**, and select **Create Data Development Node > ODPS SQL**.



3. Edit the node code.

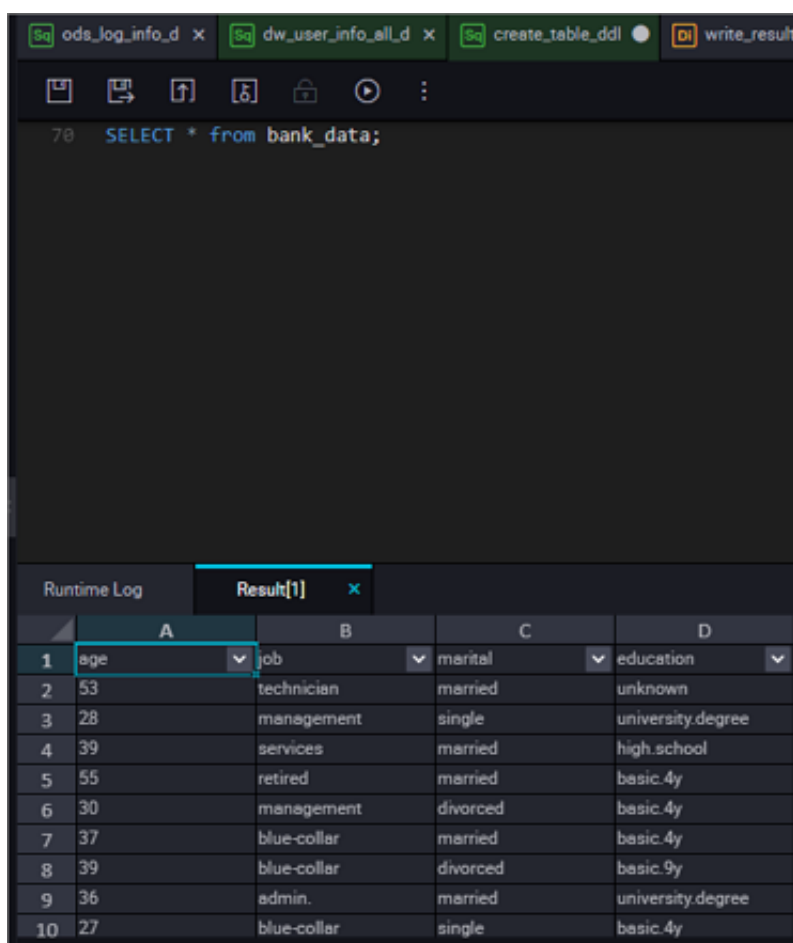
For more information about the syntax of the SQL statements, see [MaxCompute SQL statements](#).



4. Query result display

DataWorks query results are connected to the spreadsheet function, making it easier for users to operate on data results.

The query results are displayed directly in the style of a spreadsheet. Users can perform operations in DataWorks, open them in a spreadsheet, or freely copy content stations in local excels.



- Hidden column: select one or more columns hidden to hide the column.

- Copy the row: select one or more rows that need to be copied on the left side and click Copy the row.
- Copy the column: the column at the top selects a column or more points that need to be copied to copy the column.
- Copy: you can freely copy the selected content.
- Search: the search box will appear in the upper right corner of the query results to facilitate searching the data in the table.

5. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

6. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

7. Publish a node task.

For more information about the operation, see Release management.

8. Test in the production environment.

For more information about the operation, see [Cyclic task](#).

3.5.4 ODPS MR node

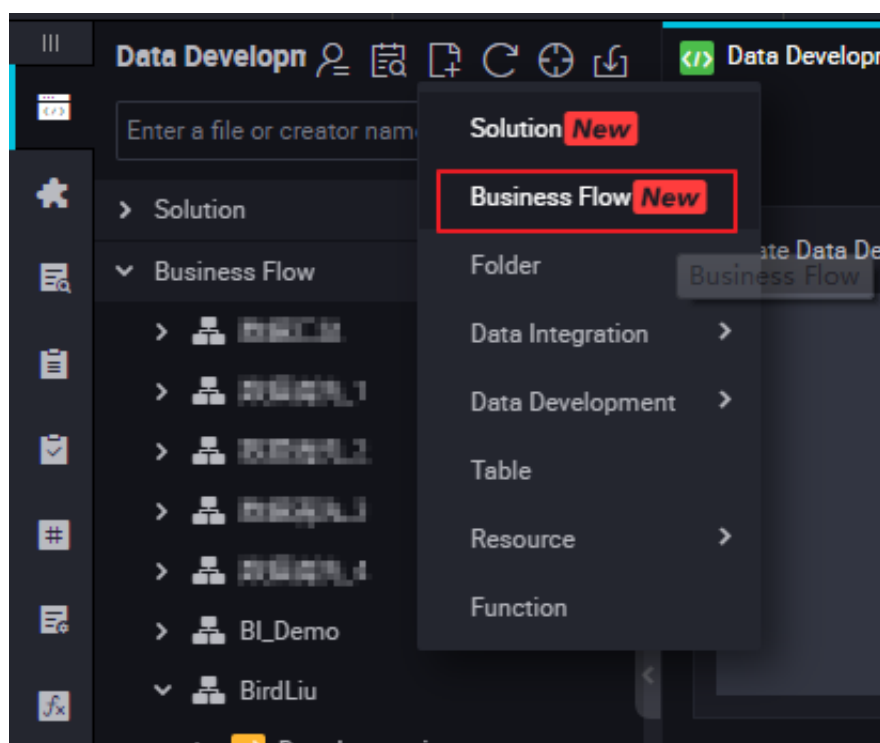
MaxCompute supports MapReduce programming APIs. You can use the Java API provided by MapReduce to write MapReduce programs for processing data in MaxCompute. You can create ODPS MR nodes and use them in Task Scheduling.

For how to edit and use the ODPS MR, see the examples in the MaxCompute documentation [WordCount examples](#).

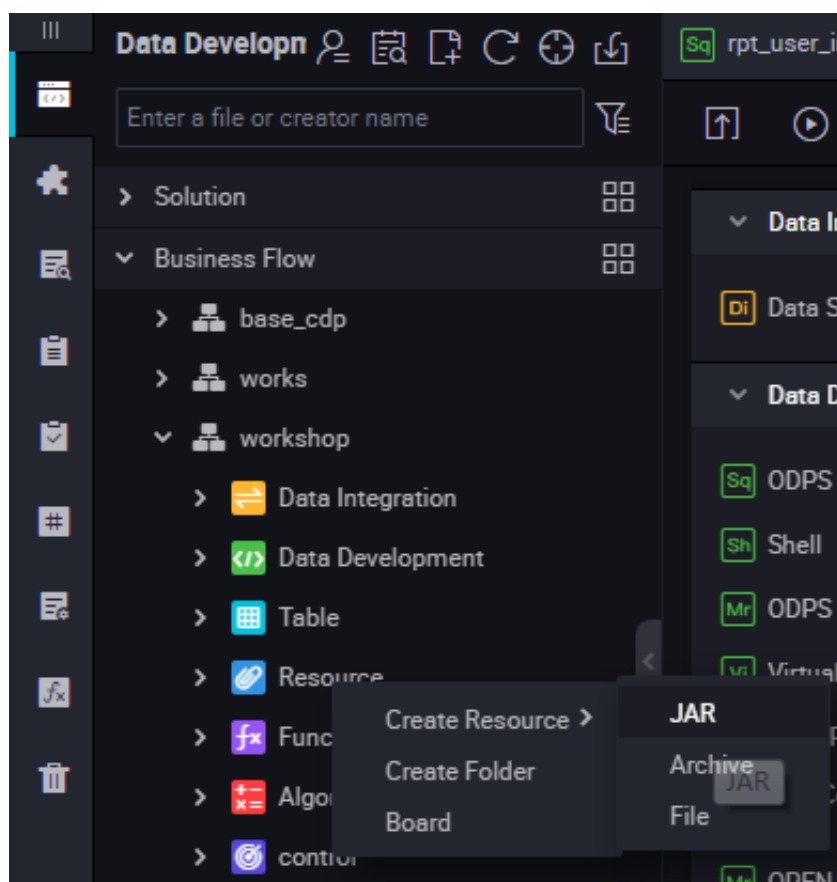
To use an ODPSMR node, you must first upload and release the resource to be used, and then create the ODPS MR node.

Create a resource instance

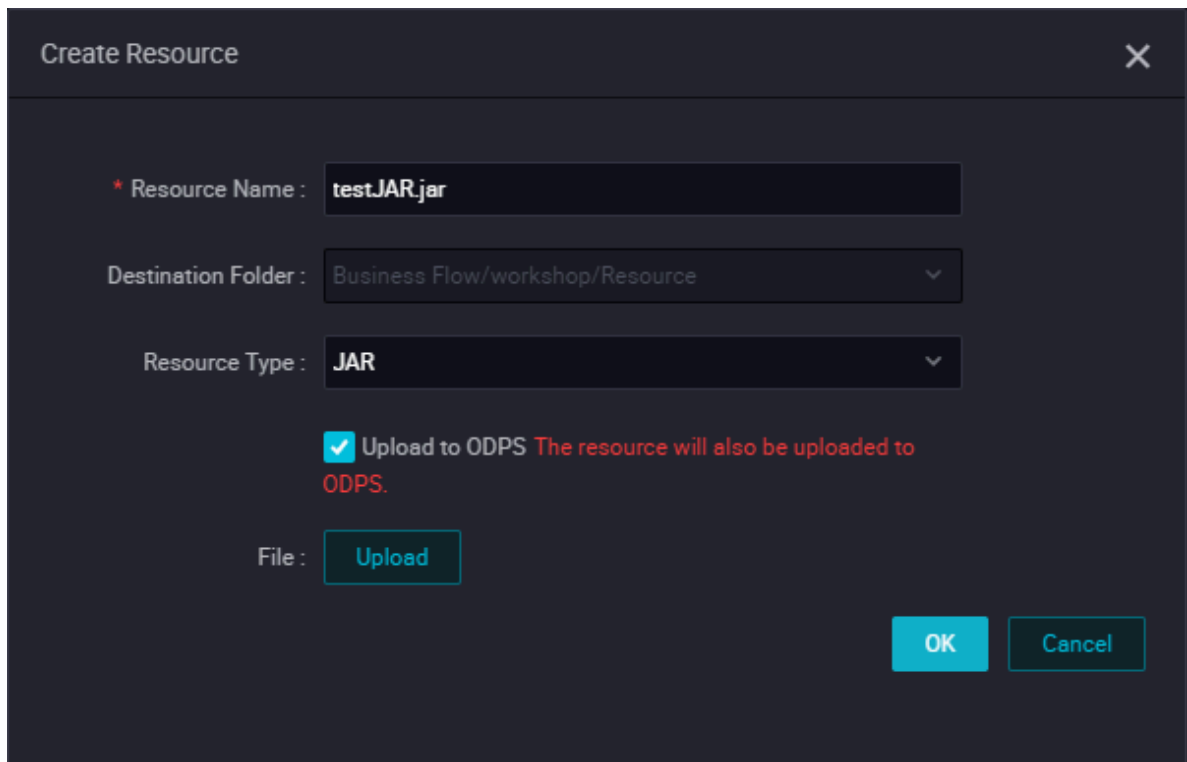
1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



2. Right-click **Resource**, and select **Create Resource > jar**.



3. Enter the resource name in the **Create Resource** according to the naming convention, set the resource type to jar and select a local jar package.



Create Resource [X]

* Resource Name :

Destination Folder :

Resource Type :

☒ Upload to ODPS The resource will also be uploaded to ODPS.

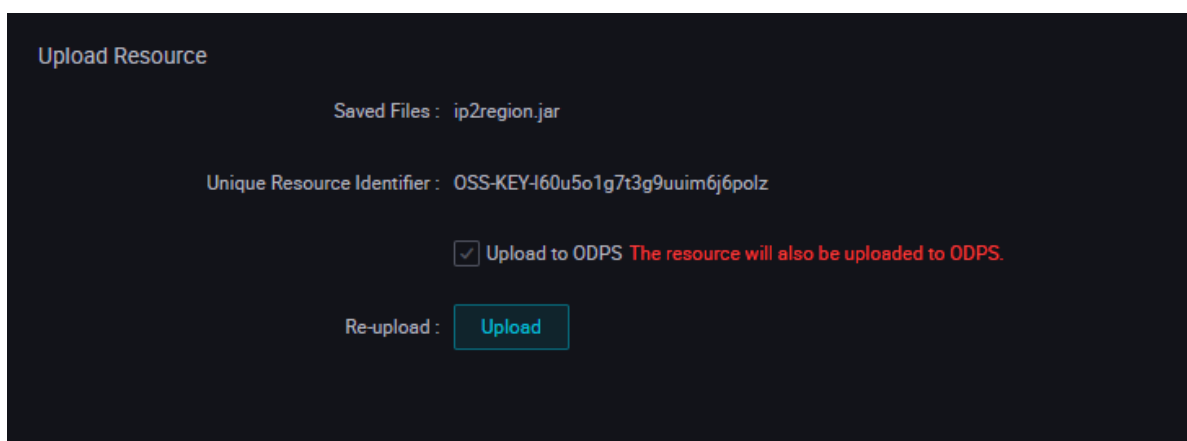
File :



Note:

- If this jar package has been uploaded on the ODPS client, you must deselect **Uploaded to ODPS**. Otherwise, an error will be reported during the upload process.
- The resource name is not necessarily the same as the name of the uploaded file.
- Naming convention for a resource name: a string of 1 to 128 characters, including letters, numbers, underlines, and dots. The name is case insensitive. If the resource is a jar resource, the extension is .jar. If the resource is a Python resource, the extension is .py.

4. Click **Submit** to submit the resource to the development scheduling server.



Upload Resource

Saved Files : ip2region.jar

Unique Resource Identifier : OSS-KEY-I60u5o1g7t3g9uuim6j6polz

☒ Upload to ODPS The resource will also be uploaded to ODPS.

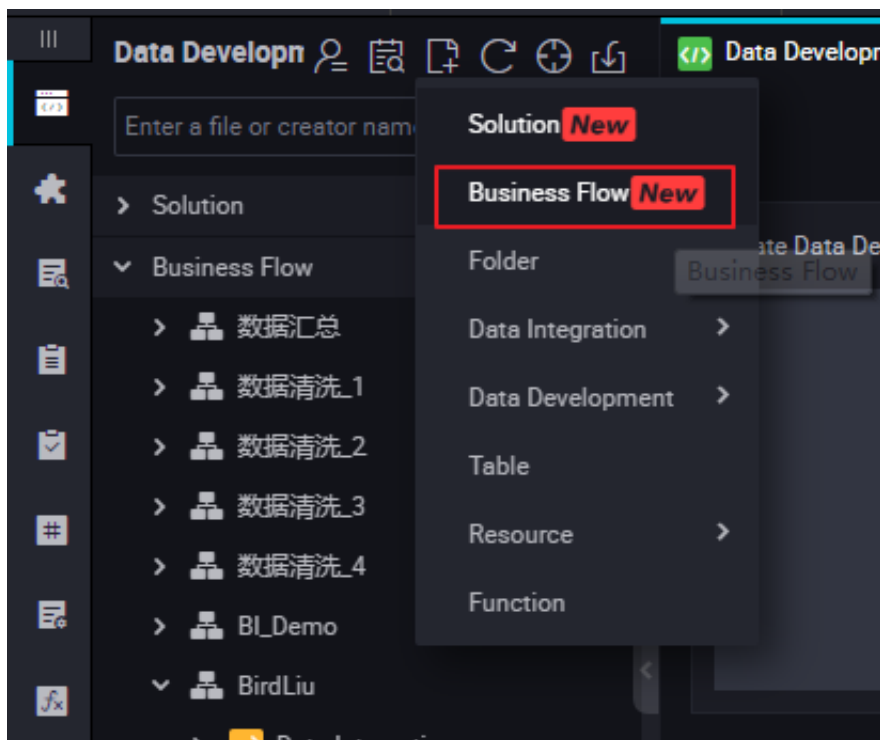
Re-upload :

5. Publish a node task.

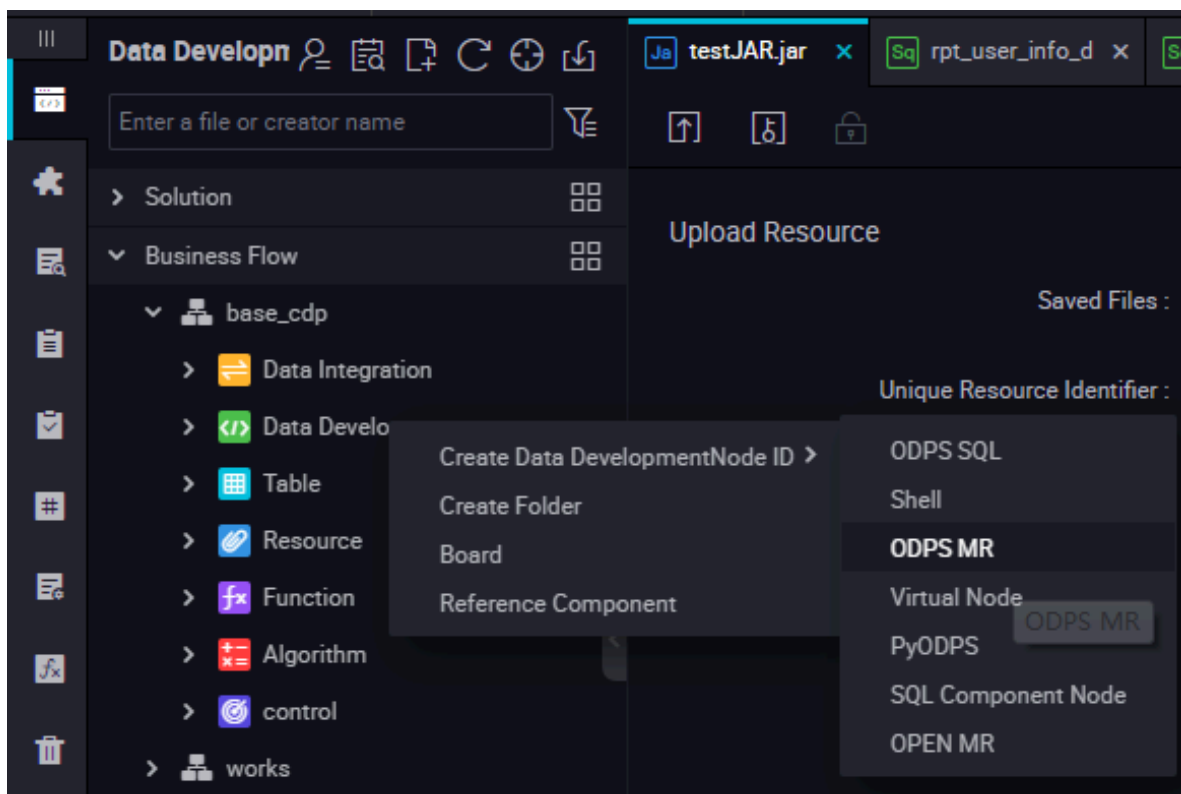
For more information about the operation, see Release management.

Create an ODPS MR node

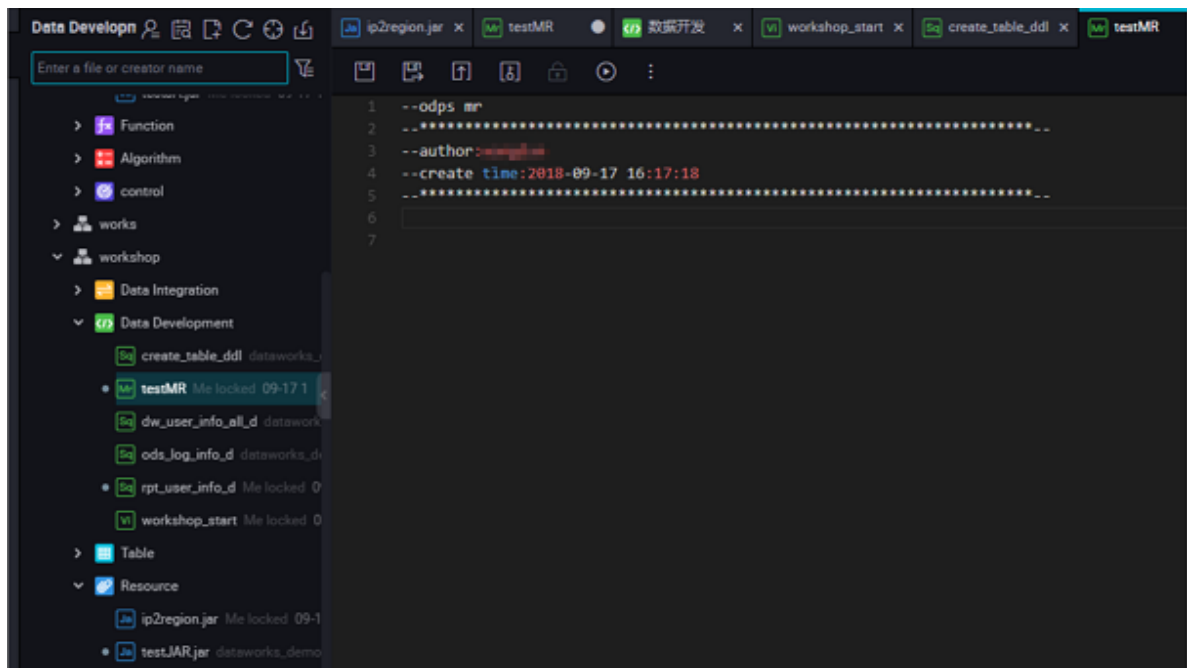
1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



2. Right-click **Data Development**, and select **Create Data Development Node > ODPS MR**.



3. Edit the node code. Double click the new ODPS MR node and enter the following interface:



Node code editing example:

```
jar -resources base_test.jar -classpath ./base_test.jar com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll
```

The code is described below:

- `-resources base_test.jar`: indicates the file name of the referenced jar resource.
- `-classpath`: jar package path.
- `com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll`: indicates the main class in the jar package that is called during execution. It must be consistent with the main class name in the jar package.

When one MR calls multiple jar resources, classpath must be written as follows: `-classpath ./xxxx1.jar, ./xxxx2.jar`, that is, two paths must be separated by a comma.

4. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see Release management.

7. Test in the production environment.

For more information about the operation, see [Cyclic task](#).

3.5.5 PyODPS node

DataWorks also provides the PyODPS task type and integrates the Python SDK of MaxCompute. You can directly edit the Python code to operate MaxCompute on a PyODPS node of DataWorks.

MaxCompute provides the [Python SDK](#), which can be used to operate MaxCompute.

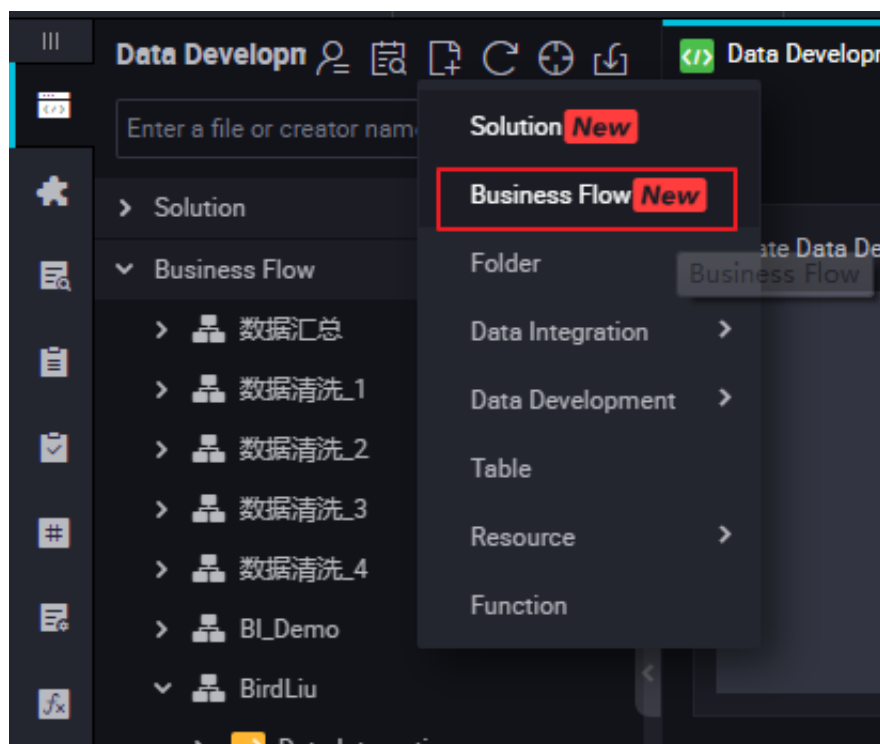


Note:

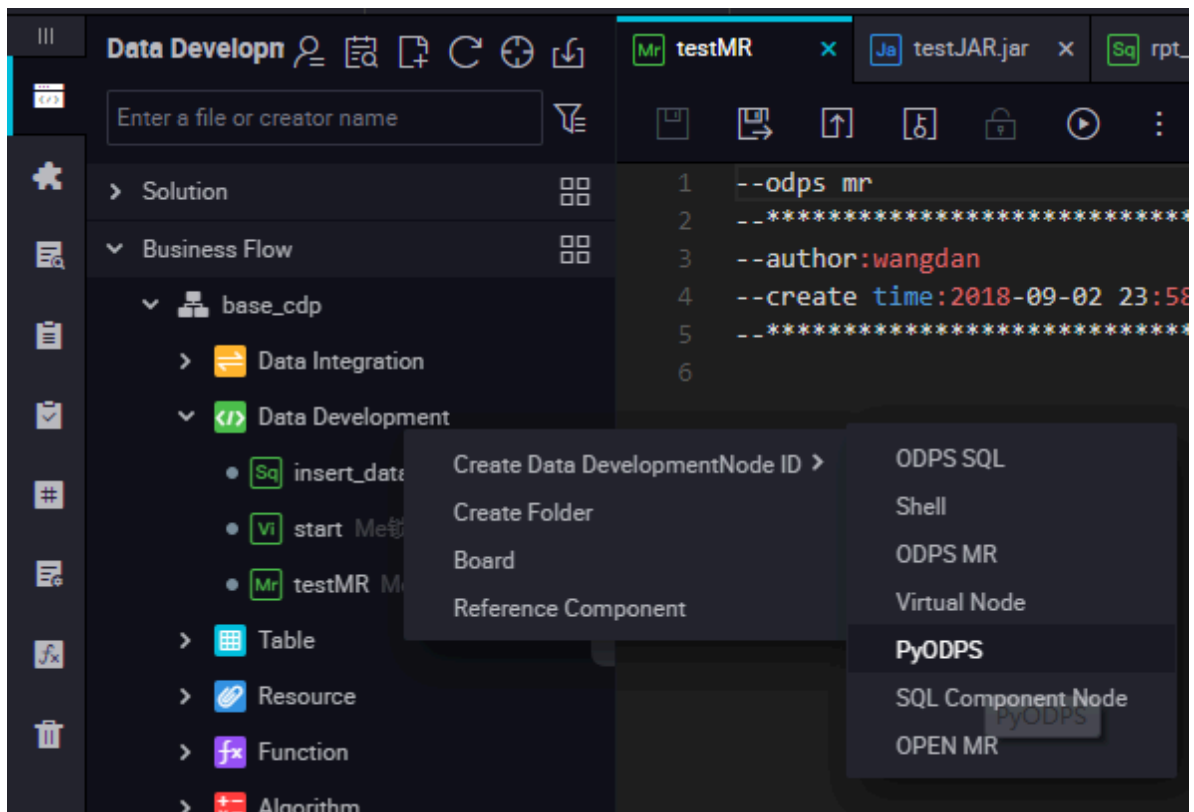
Python 2.7 is used at the underlying layer. The size of data that PyODPS nodes process should not exceed 50 MB, while the memory they occupy should not exceed 1 GB.

Create a PyODPS node

1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



2. Right-click **Data Development**, and select **Create Data Development Node > PyODPS**.



3. Edit the PyODPS node.

a. ODPS portal

On DataWorks, the PyODPS node contains a global variable `odps` or `o`, which is the ODPS entry. You do not need to manually define an ODPS entry.

```
print(odps.exist_table('PyODPS_iris'))
```

b. Run the SQL statements

PyODPS supports ODPS SQL query and can read the execution result. The return value of the `execute_sql` or `run_sql` method is the running instance.



Note:

Not all commands that can be executed on the ODPS console are SQL statements that are accepted by ODPS. You need to use other methods to call non DDL/DML statements. For example, use the `run_security_query` method to call the GRANT or REVOKE statements, and use the `run_xflow` or `execute_xflow` method to call PAI commands.

```
o.execute_sql('select * from dual') # Run the SQL statements in
synchronous mode. Blocking continues until execution of the SQL
statement is completed.
instance = o.runsql('select * from dual') # Run the SQL
statements in asynchronous mode.
print(instance.getlogview_address()) # Obtain the logview address
.
```

```
instance.waitforsuccess() # Blocking continues until execution of  
the SQL statement is completed.
```

c. Configure the runtime parameters

The runtime parameters must be set sometimes. You can set the hints parameter with the parameter type of dict.

```
o.execute_sql('select * from PyODPS_iris', hints={'odps.sql.mapper  
.split.size': 16})
```

After you add sql.settings to the global configuration, related runtime parameters are added upon each running.python.

```
from odps import options  
options.sql.settings = {'odps.sql.mapper.split.size': 16}  
o.execute_sql('select * from PyODPS_iris') # "hints" is added  
based on the global configuration.
```

d. Read the SQL statement execution results

The instance that runs the SQL statement can directly perform the open_reader operation. In one case, the structured data is returned as the SQL statement execution result.

```
with o.execute_sql('select * from dual').open_reader() as reader:  
for record in reader: # Process each record.
```

In another case, desc may be executed in an SQL statement. In this case, the original SQL statement execution result is obtained through the reader.raw attribute.

```
with o.execute_sql('desc dual').open_reader() as reader:  
print(reader.raw)
```



Note:

User-defined scheduling parameters are used in data development. If a PyODPS node is directly triggered on the page, the time must be clearly specified. The time of a PyODPS node cannot be directly replaced like that of an SQL node.

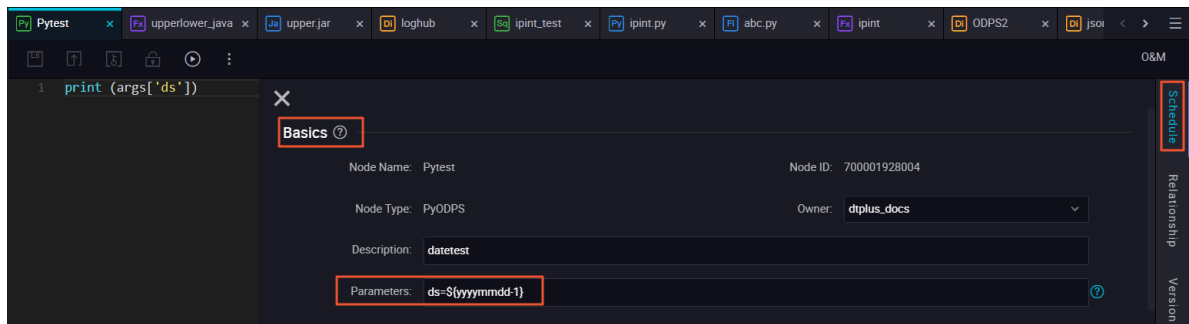
You can configure system parameters like this.

```

1 a = '${bdp.system.bizdate}'
2 print ((format(a)))
3 b = '${bdp.system.cyctime}'
4 print (format(b))

```

You can configure user-defined parameters like this.



4. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see Release management.

7. Test in the production environment.

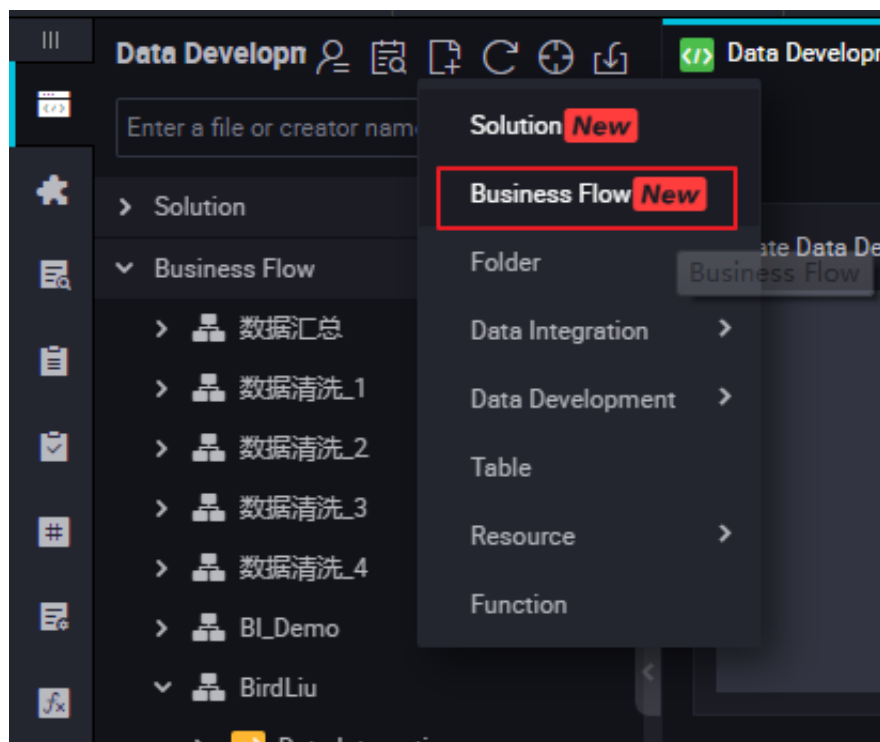
For more information about the operation, see [Cyclic task](#).

3.5.6 SHELL node

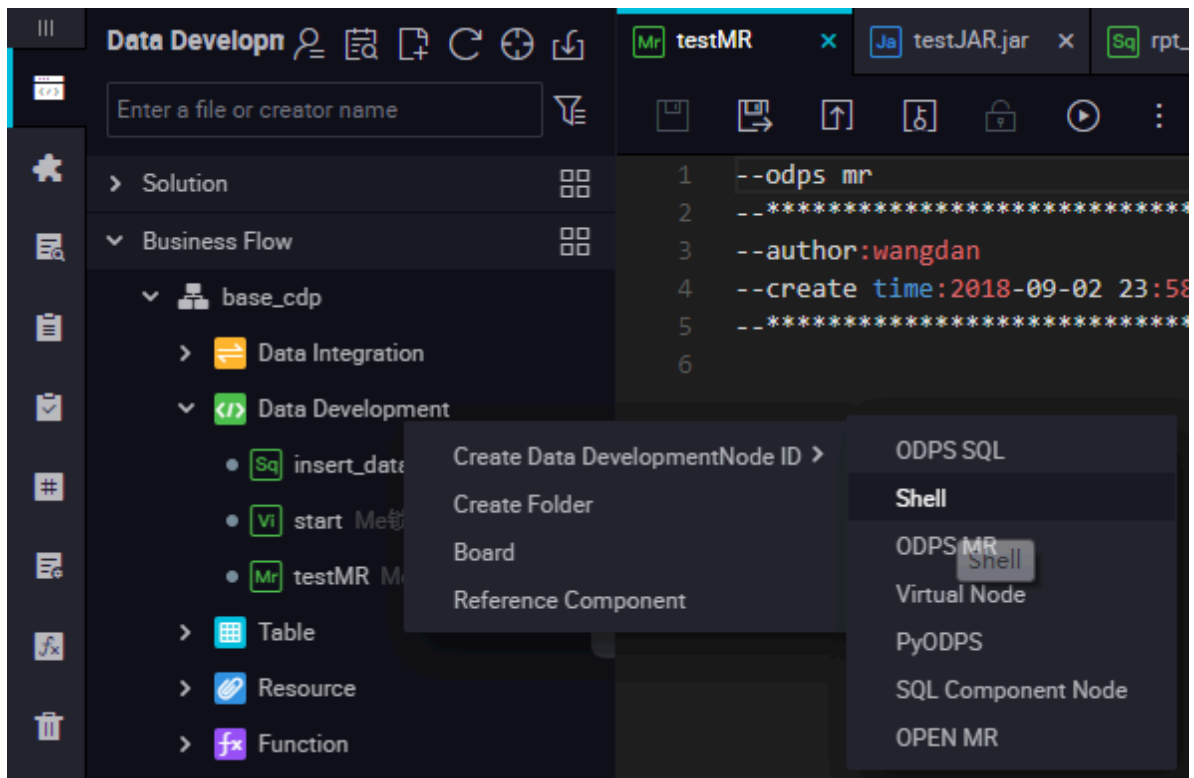
SHELL tasks support standard SHELL syntax but not interactive syntax. SHELL task can run on the default resource group. If you want to access an IP address or a domain name, add the IP address or domain name to the whitelist by choosing Project Configuration.

Procedure

1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.

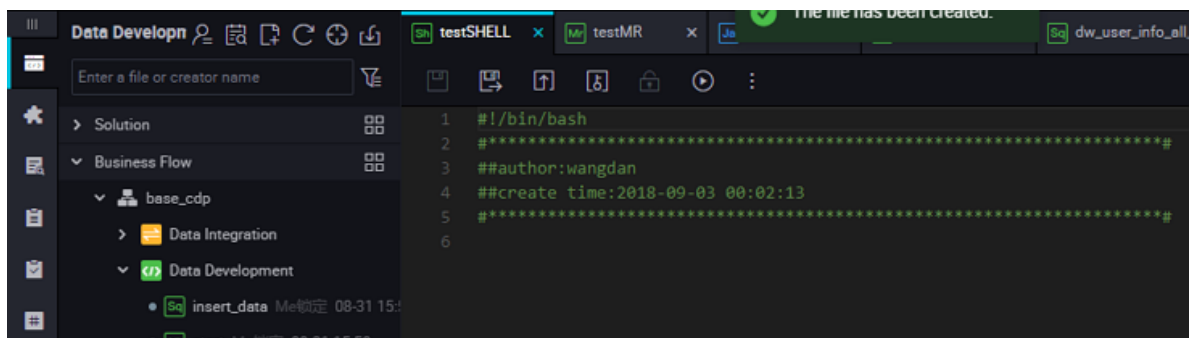


2. Right-click **Data Development**, and select **Create Data Development Node > SHELL**.



3. Set the node type to SHELL, enter the node name, select the target folder, and click **Submit**.
4. Edit the node code.

Go to the SHELL node code editing page and edit the code.



If you want to call the System Scheduling Parameters in a SHELL statement, compile the SHELL statement as follows:

```
echo "$1 $2 $3"
```



Note:

Parameter 1 Parameter 2... Multiple parameters are separated by spaces. For more information on the usage of system scheduling parameters, see [Parameter configuration](#).

5. Node scheduling configuration.

Click the **Scheduling Configuration** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

6. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

7. Release a node task.

For more information about the operation, see Release management.

8. Test in the production environment.

For more information about the operation, see [Cyclic task](#).

Use cases

Connect to a database using SHELL

- If the database is built on Alibaba Cloud and the region is China (Shanghai), you must open the database to the following whitelisted IP addresses to connect to the database.

10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8



Note:

If the database is built on Alibaba Cloud but the region is not China (Shanghai), we recommend that you use the Internet or buy an ECS instance in the same region of the database as the scheduling resource to run the SHELL task on a custom resource group.

- If the database is built locally, we recommend that you use the Internet connection and open the database to the preceding whitelisted IP addresses.



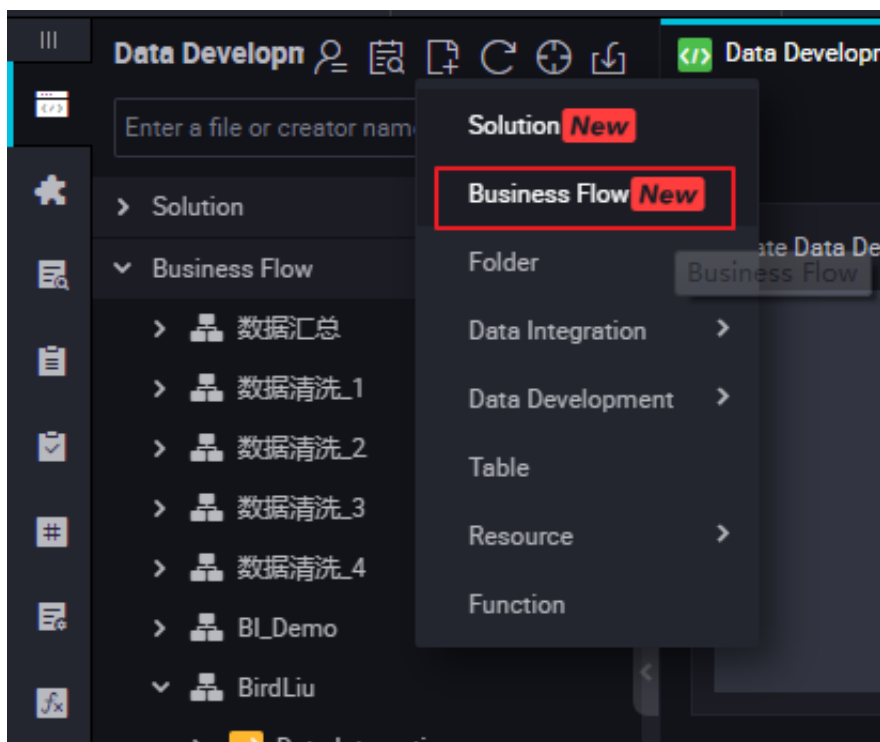
Note:

If you are using a custom resource group to run the SHELL task, you must add the IP addresses of machines in the custom resource group to the preceding whitelist.

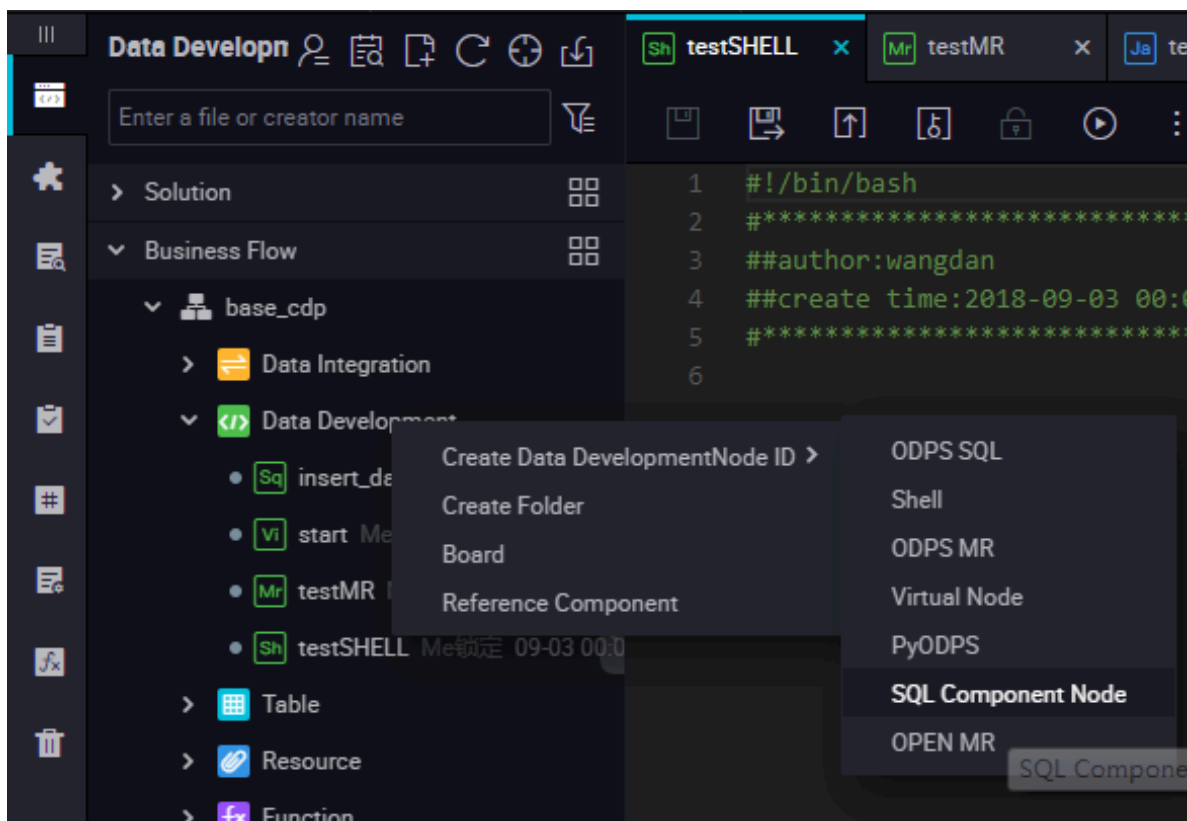
3.5.7 SQL Component node

Procedure

- Right-click **Business Flow** under **Data Development**, select **Create Business Flow**



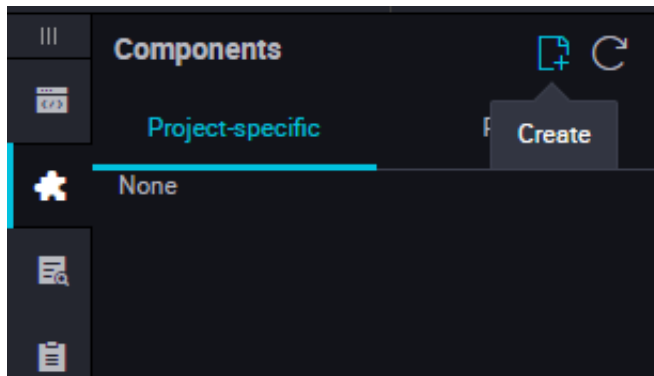
2. Right-click **Data Development**, and select **Create Data Development Node > SQL Component Node**.



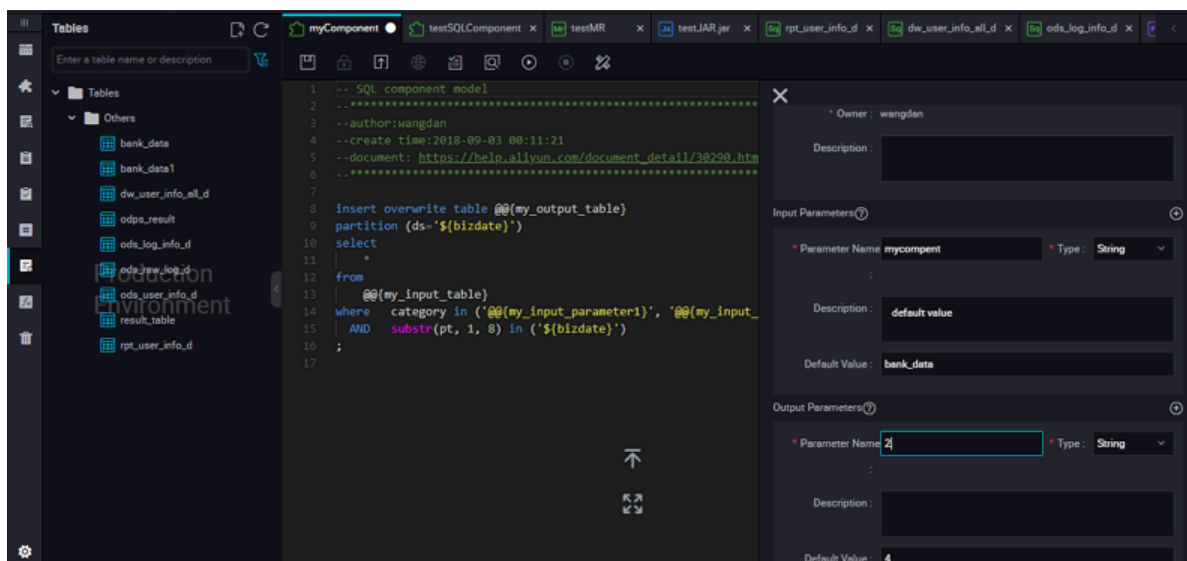
3. To improve the development efficiency, data task developers can use components contributed by project members and tenant members to create data processing nodes.

- Components created by members of the local project are located under Project Components.
- Components created by tenant members are located under Public Components.

When create a node, set the node type to the **SQL Component node** type, and specify the name of the node.



Specify parameters for the selected component.



Enter the parameter name, and set the parameter type to Table or String.

Specify three get_top_n parameters in sequence.

Specify the following input table for the parameters of the Table type: test_project.test_table.

4. Node scheduling configuration.

Click the **Scheduling Configuration** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit a node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see Release management.

7. Test in a production environment.

For more information about the operation, see [Cyclic task](#).

Upgrade the version of an SQL Component node.

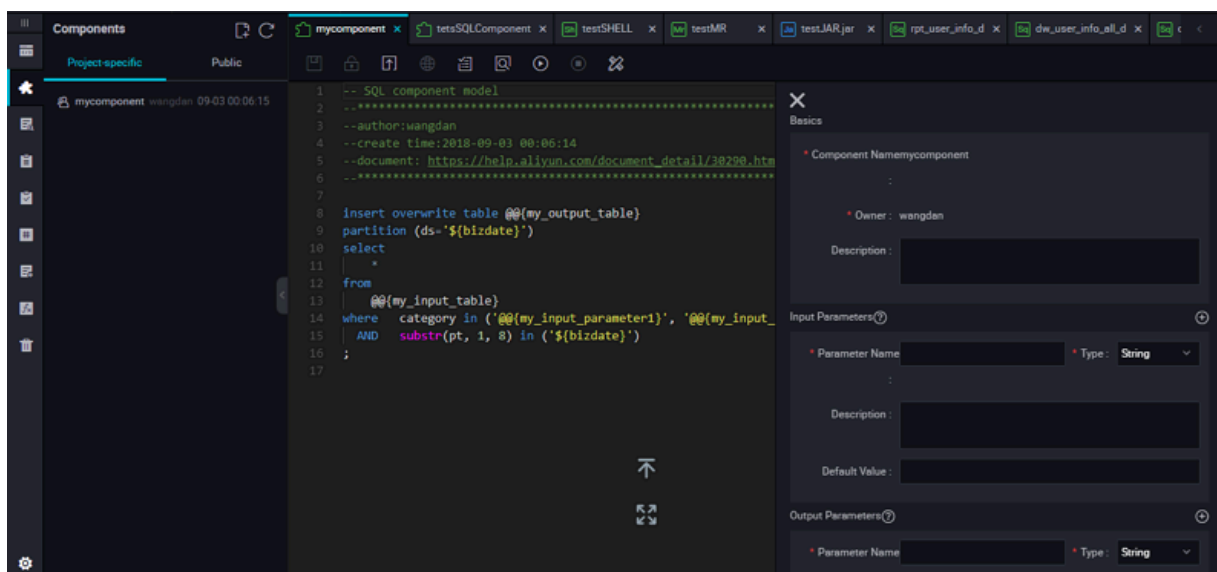
After the component developer release a new version, the component users can choose whether to upgrade the use instance of the existing component to the latest version of the used component .

With the component version mechanism, developers can continuously upgrade components and component users can continuously enjoy the improved process execution efficiency and optimized business effects after upgrade of components.

For example, user A uses the v1.0 component developed by user C, and the component owner C upgrades the component to V.2.0. After the upgrade, user A can still use the v1.0 component, but will receive the upgrade reminder. After comparing the new code with the old code, user A finds that the business effects of the new version are better than those of the old version, and therefore can determine whether to upgrade the component to the latest version.

To upgrade an SQL Component node developed based on the component template, you only need to select Upgrade, check whether parameter settings of the SQL Component node are still effective in the new version, make some adjustments based on the instructions of the new version component, and then submit and release the node like a common SQL Component node.

Interface functions



The interface features are described below:

No.	Feature	Description
1	Save	Click it to save settings of the current component.
2	Steal lock Edit	Click it to steal lock edit the node if you are not the owner of the current component.
3	Submit	Click it to submit the current component to the development environment.
4	Publish Component	Click it to publish a universal global component to the entire tenant, so that all users in the tenant can view and use the public component.
5	Resolve Input and Output Parameters	Click it to resolve the input and output parameters of the current code.
6	Precompilation	Click it to edit custom and component parameters of the current component.
7	Run	Click it to run the component locally in the development environment.
8	Stop Run	Click it to stop a running component.
9	Format	Click it to sort the current component code by keyword.
10	Parameter Settings	Click it to view the component information, input parameter settings, and output parameter settings.
11	Version	Click it to view the submission and release records of the current component.

No.	Feature	Description
12	Reference Records	Click it to view the use record of the component.

3.5.8 Virtual node

A virtual node is a control node that does not generate any data. Generally, it is used as the root node for overall planning of nodes in the workflow.

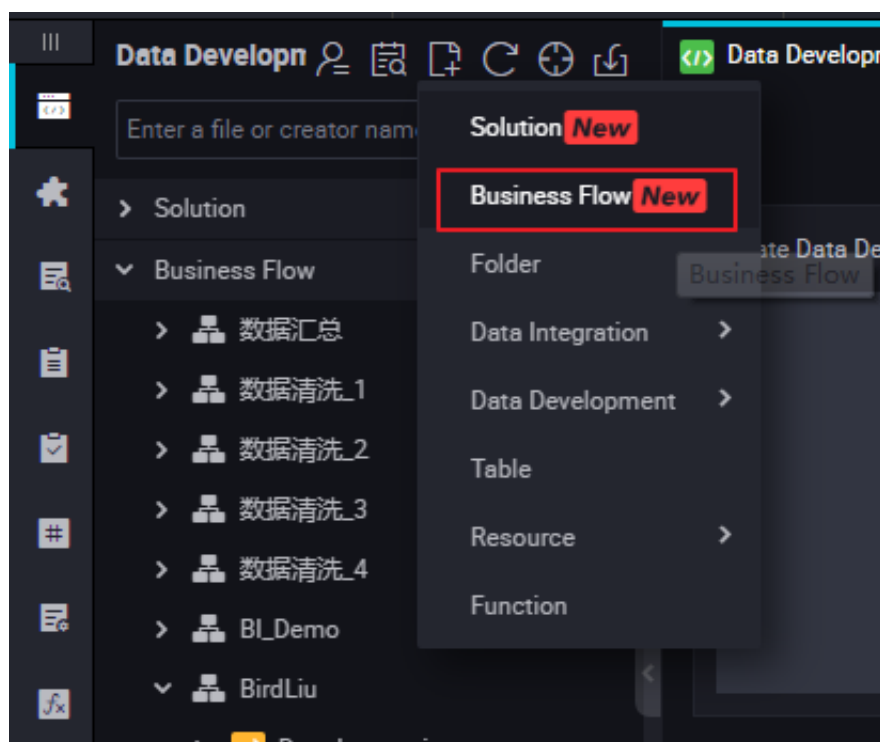


Note:

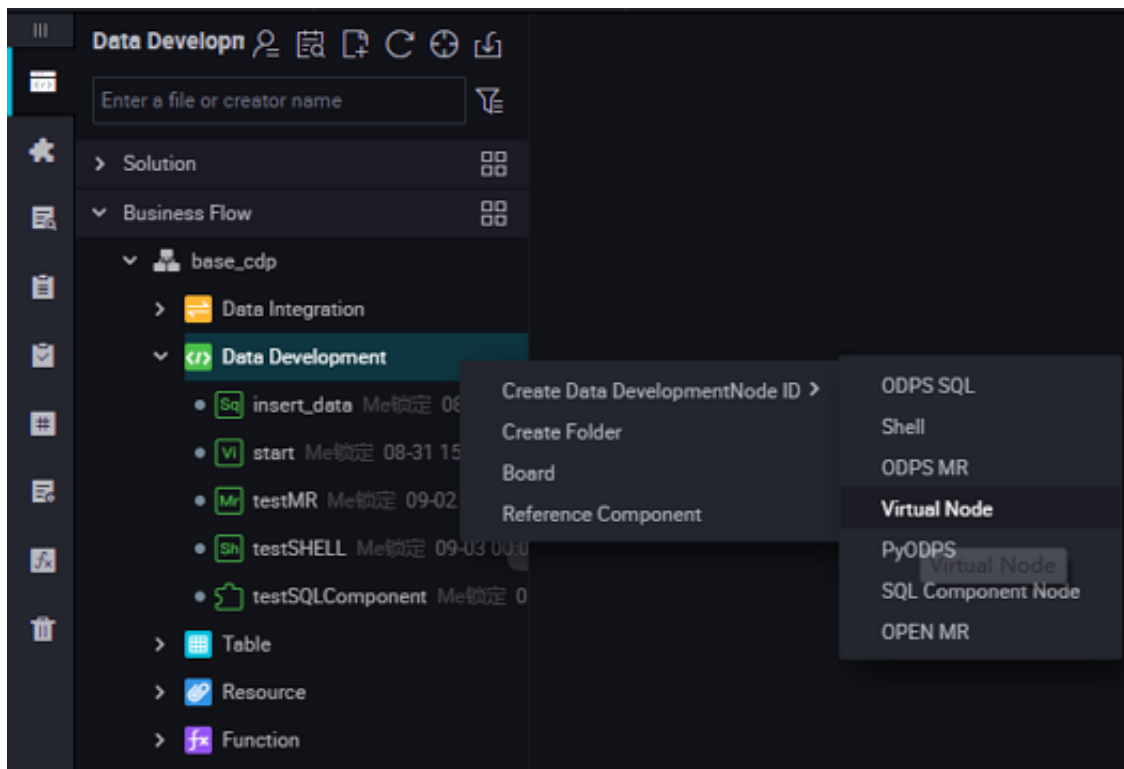
The final output table of a workflow contains multiple branch input tables. Virtual nodes are usually used if these input tables do not have dependency between them.

Create a virtual node task

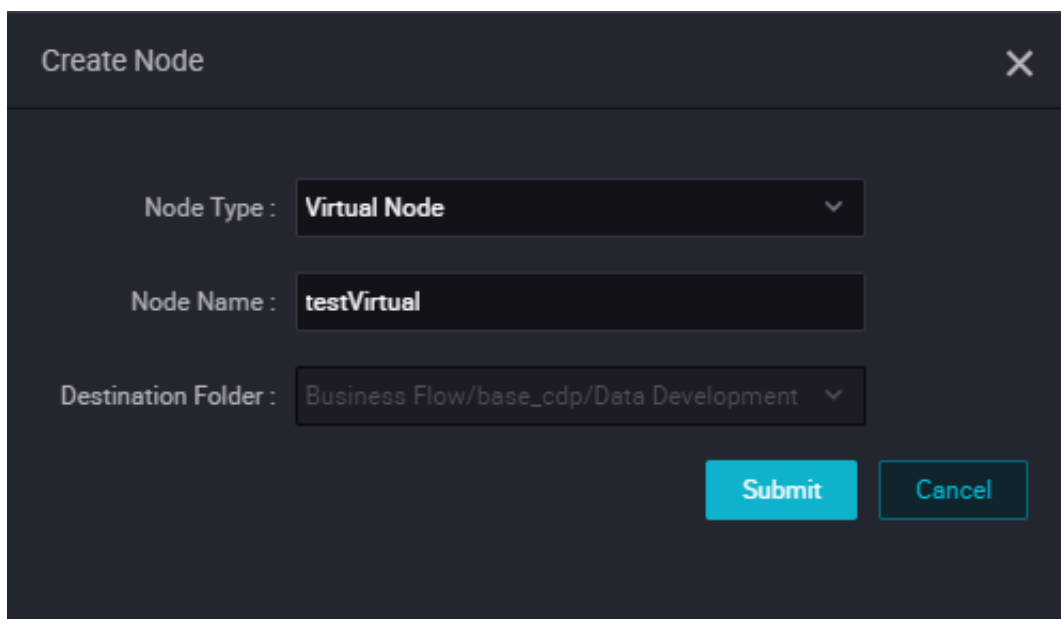
1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.



2. Right-click **Data Development**, and select **Create Data Development Node > Virtual Node**.



3. Set the node type to **Virtual Node**, enter the node name, select the target folder, and click **Submit**.



4. Edit the node code: You do not need to edit the code of a virtual node.
5. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

6. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

7. Publish a node task.

For more information about the operation, see Release management.

8. Test in the production environment.

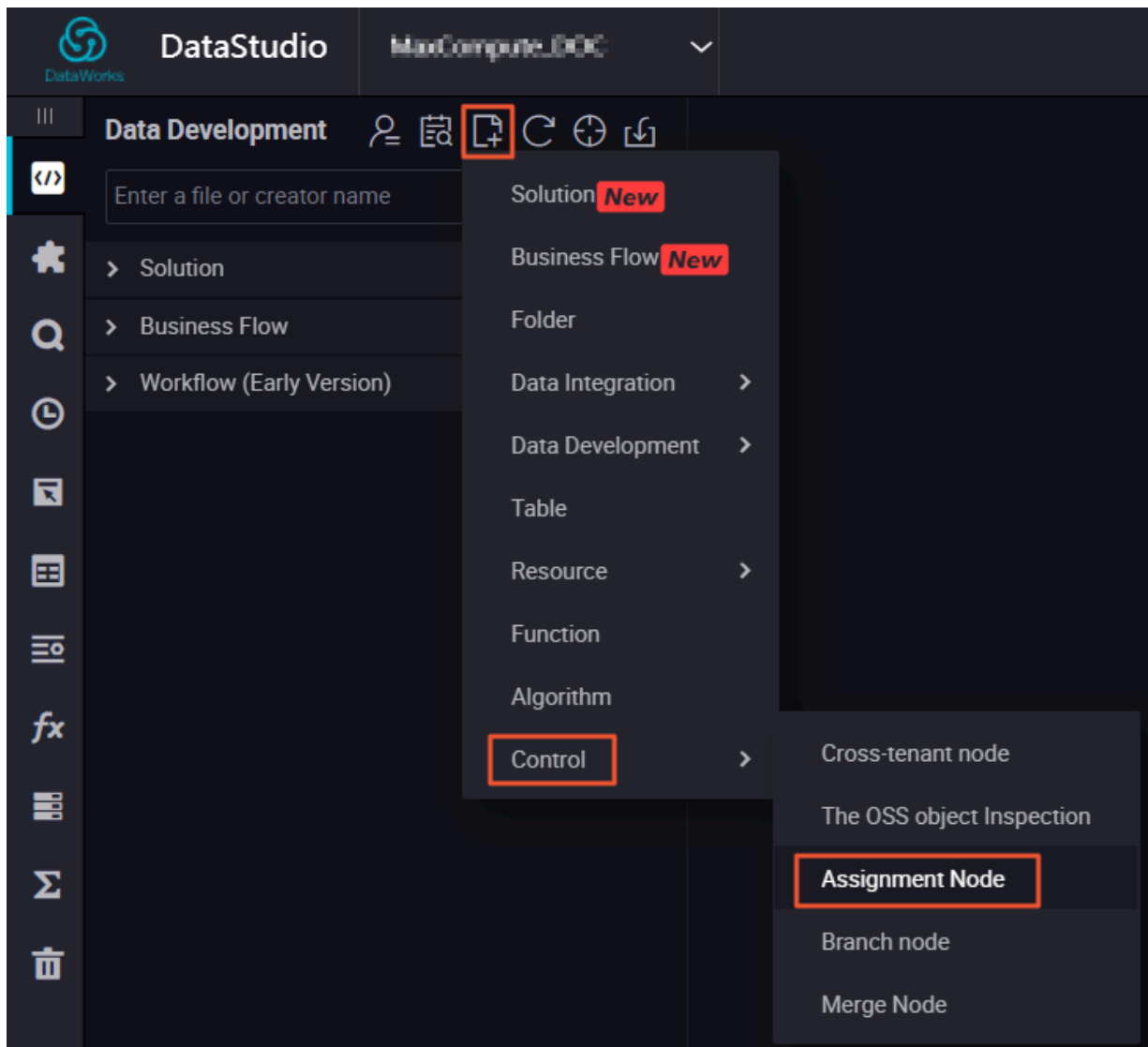
For more information about the operation, see [Cyclic task](#).

3.5.9 Assignment node

Assignment node is a special type of node. It supports assignment of output parameters by writing code in the node, and transfers them in combination with the node context, for downstream nodes to reference and use their values.

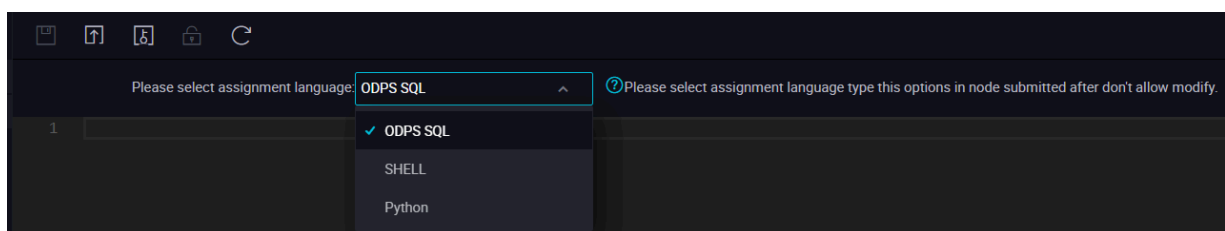
Create an assignment node

Assignment Node is located in the **Control** class directory of the new node menu, as shown in the following figure.



Write the value logic of assignment node

The assignment node has a fixed output parameter named **outputs** in the **Node Context**. It supports the use of MaxCompute, Shell and Python to write code to assign parameters, whose values are the operation and calculation results of node code. Only one language can be selected for a single assignment node.



Note:

- The value of the **outputs** parameter takes only the output from the last line of code, that is:

- The output of the SELECT statement on the last line of MaxCompute SQL .
- Data from the ECHO statement on the last line of shell.
- The output of the PRINT statement on the last line of Python.
- There is a certain limit to the value of the outputs parameter, with a maximum transfer value of 2M. If the output of the assignment statement exceeds this limit, the assignment node will fail to run.

The Node Output Parameters Add

No.	Parameter Name	Type	Value	Description	Source	Actions
1	outputs	Variable	\${outputs}	MaxCompute SQL, Shell, Python	Added by Default	Edit Delete

Use the output of the assignment node on the downstream Node

In the downstream node, after adding an assignment node as an upstream dependency, define the output of the assignment node as an input parameter for the node by the way of node context, and reference it in the code, the specific values for the output parameters of the upstream assignment node can be obtained. For more information, see [Node context](#).

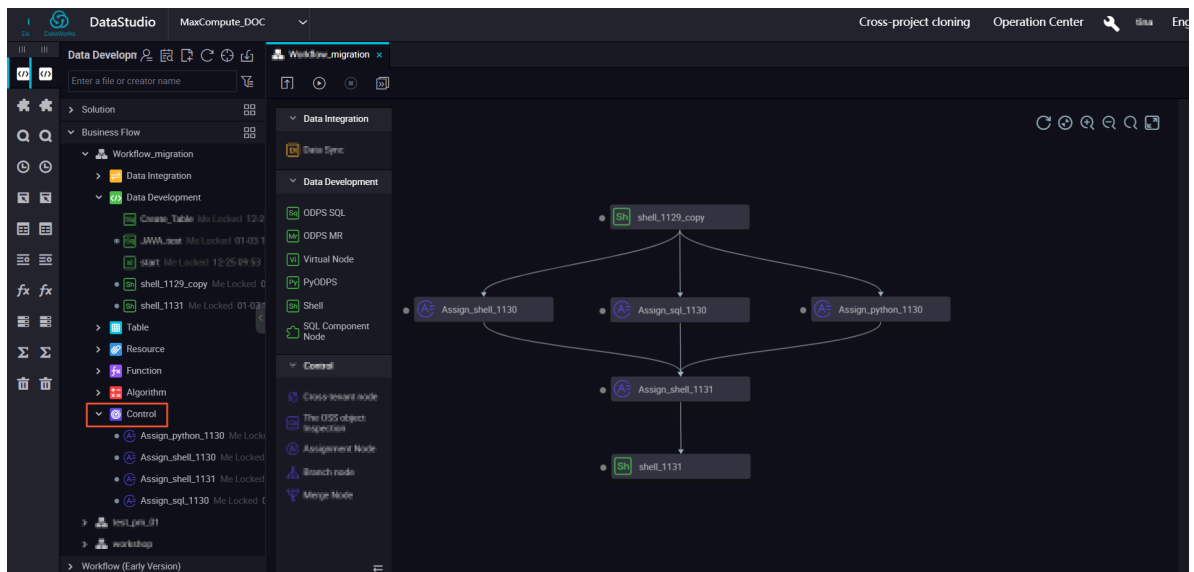
Node Context ?

The Node Input Parameters Add

No.	Parameter Name	Value Of The Source	Description	Parent Node ID	Source	Actions
1	input	MaxCompute_DDC 213:outputs	MaxCompute SQL, Shell, Python	70000015963	Added by Default	Edit Delete

An example of assignment node

1. Create the business process, and then create the following nodes respectively, as shown in the following figure.



2. When configuring the assignment node, the system will display a **outputs** parameter by default. After running, you can find the relevant parameter results in the related **Operation Center > Properties > Context** page.

No.	Parameter Name	Type	Value	Description	Source	Actions
1	outputs	Variable	\$outputs		Added by Default	Edit Delete

3. The upstream **outputs** parameter is used as the downstream input parameter, as shown in the figure below.

The screenshot shows the DataStudio interface for configuring a node named 'Assign_shell_1131'. The left sidebar displays the project structure, including 'Workflow_migration'. The main area shows the node configuration details. The 'Upstream Node Output Name' is 'MaxCompute_DOC_Assign_Shell_1131'. The 'Output' section lists two outputs: 'MaxCompute_DOC_500152926_out' and 'MaxCompute_DOC_Assign_Shell_1131'. The 'Node Context' section shows the 'Input' parameter with the value 'MaxCompute_DOC_Assign_Shell_1131 outputs'.

Run the assignment node task



Note:

In general operation and maintenance, the above configuration parameters can be validated by patch data operation, but the test operation parameters can not be validated.

1. When the task is configured and scheduled, a run instance is generally generated the next day.
2. At runtime, you can view the input and output parameters of the context, and click the following link to see your input or output results.
3. In the **Running Log**, you can view the final output of the code through 'finalResult'.

```

#####
echo $1;
echo 'this is name,ok';
echo 'this is password';
shell output: shell
shell output: this is name,ok
shell output: this is password
2018-12-19 17:12:25.897 [main] INFO c.a.d.a.w.handler.AssignmentHandler - ...
2018-12-19 17:12:26.897 [main] INFO c.a.d.a.w.handler.AssignmentHandler - result: this is password
2018-12-19 17:12:26.925 [main] INFO c.a.d.a.w.handler.AssignmentHandler - ---finalResult: [{"this is password"}]
2018-12-19 17:12:27.363 [main] INFO c.a.d.a.w.handler.AssignmentHandler - cost Time: 1
2018-12-19 17:12:27.363 [main] INFO c.a.d.w.alisa.wrapper.ControllerWrapper - job finished!
2018-12-19 17:12:27.363 [Thread-2] INFO s.c.a.AnnotationConfigApplicationContext - Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@48cf768c: startup da
te [Wed Dec 19 17:12:24 CST 2018]; root of context hierarchy
2018-12-19 17:12:27.365 [Thread-2] INFO o.s.j.e.a.AnnotationBeanExporter - Unregistering JMX-exposed beans on shutdown
2018-12-19 17:12:27 INFO =====
2018-12-19 17:12:27 INFO Exit code of the Shell command 0
2018-12-19 17:12:27 INFO --- Invocation of Shell command completed ---
2018-12-19 17:12:27 INFO Shell run successfully!
2018-12-19 17:12:27 INFO Current task status: FINISH
2018-12-19 17:12:27 INFO Cost time is: 4.131s
/home/admin/aliyun/tasksnode/taskInfo/20181219/Phoenix/Expired/17/12/22/Log/run54u21650634e3c7c9e/T3_1629174701.log-END-EOF

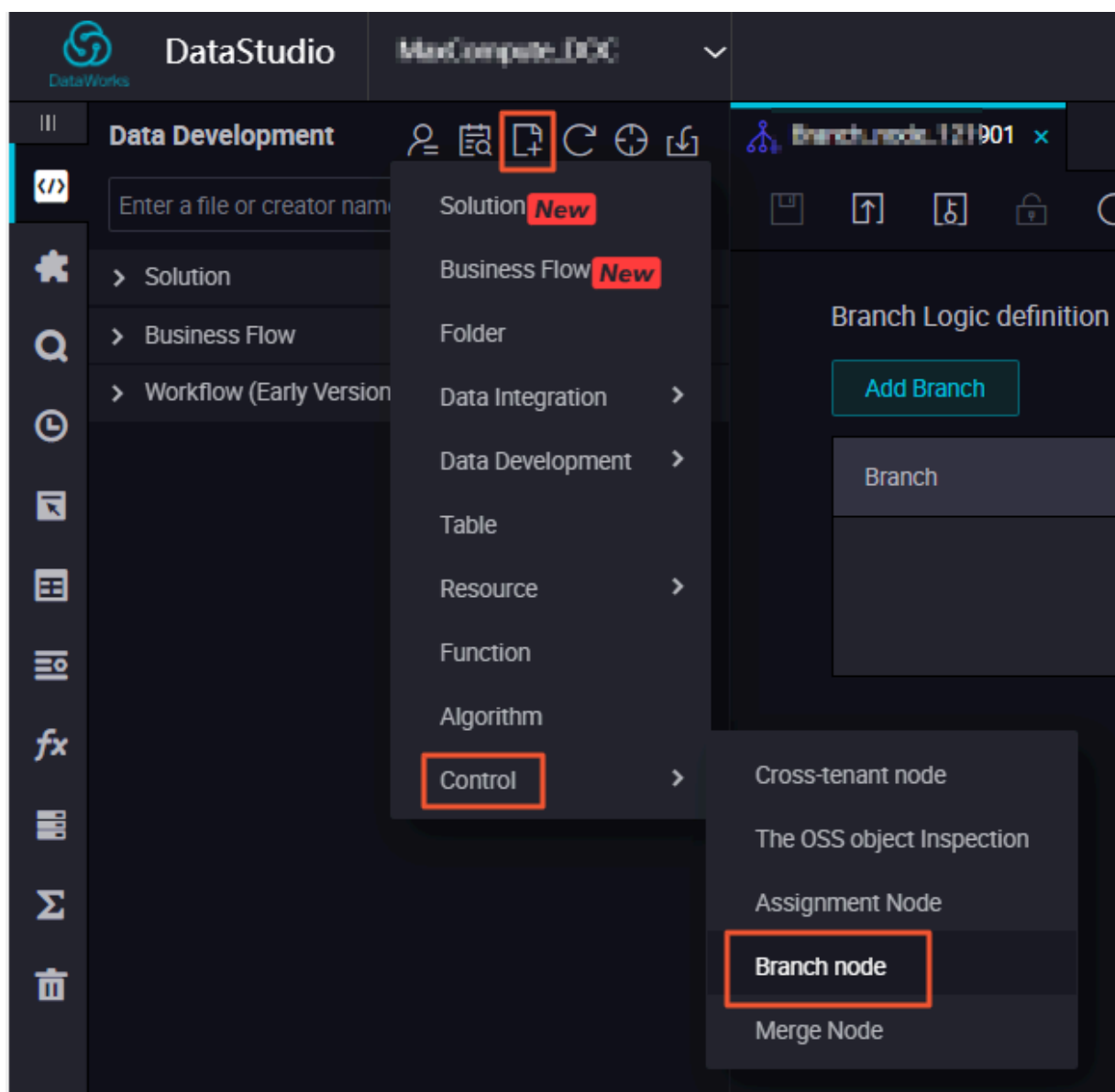
```

3.5.10 Branch node

Branch node is one of the logical control family nodes provided in DataStudio. The branch node can define the **branch logic** and the direction of downstream branches under **different logical conditions**.

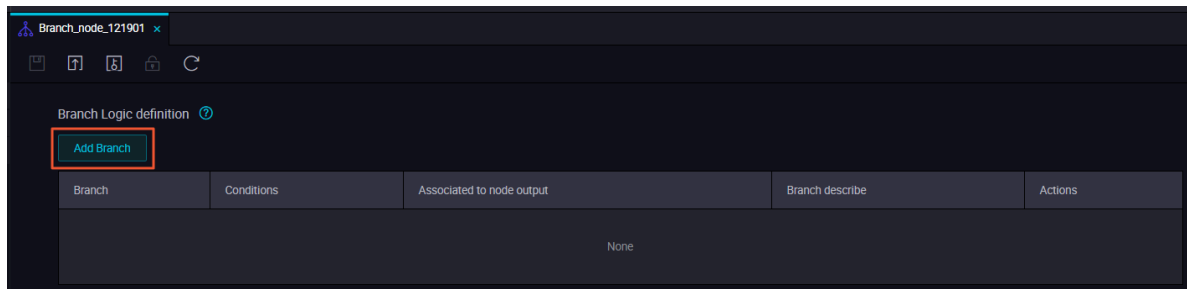
Create a branch node

Branch node is located in the **Control** class directory of new node menu, as shown in the following figure.

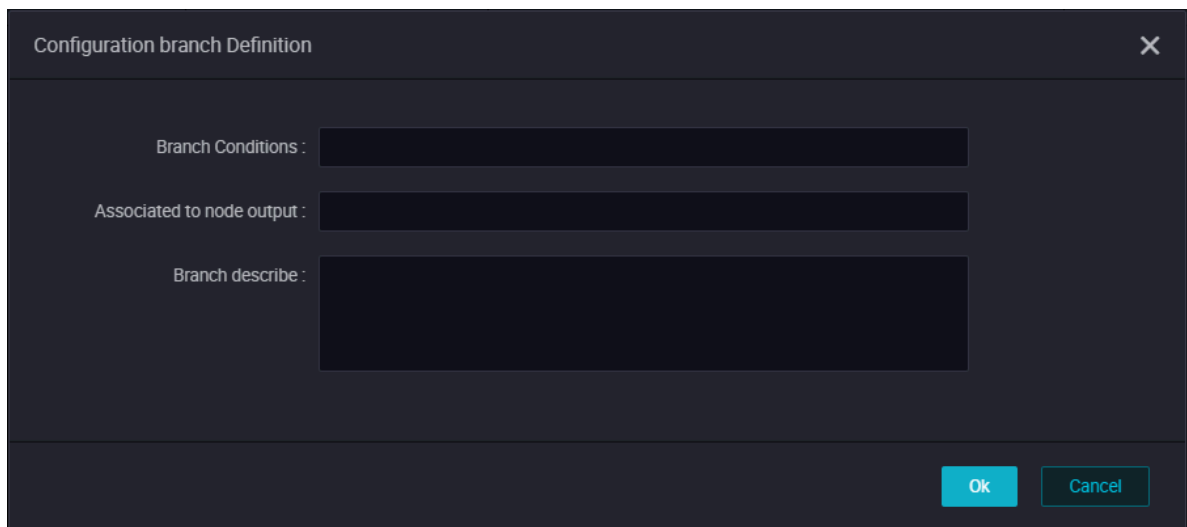


Define the branch logic

1. After creating the branch node, jump to the **Branch Logic definition** page, as shown in the following figure.



2. In the **Branch Logic definition** page, you can use **Add Branch** button to define the **Branch Conditions**, **Associated to node output**, and the **Branch describe**, as shown in the following figure.

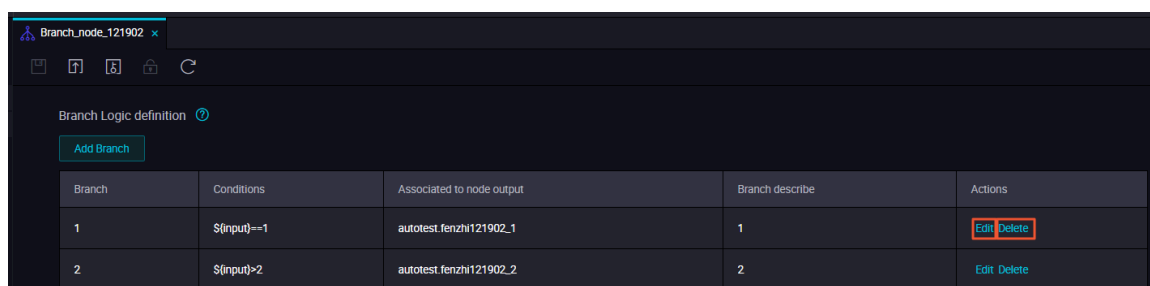


The parameters are as follows:

- **Branch Conditions**
 - The branch condition only supports defining logical judgment condition according to the Python comparison operators.
 - If the value of the running state expression is true, it means that the corresponding branching condition is satisfied, otherwise it is unsatisfactory.
 - If the parsing error of the running state expression is reported, the running state of the whole branch node will be set to failure.
 - The branching conditions support using global variables and parameters defined in node context, such as `${Input}` in the figure, which can be a node input parameter defined in the branching node.
- **Associated to node output**
 - Node output is used to mount dependencies for downstream node of branch node.

- When the branch condition is satisfied, the downstream node mounted on the corresponding associated with the node output is selected to run (also refer to the status of other upstream nodes that the node depends on).
- When the branch condition is not satisfied, the downstream node mounted on the corresponding associated with the node output is not selected to execute, the downstream node is placed in a state that is not running because the branch condition is not satisfied.
- **Branch describe**: refer to the description of the branch definition.

Defining two branches: $\$ \{ Input \} == 1$ and $\$ \{ Input \} > 2$, as shown in the following figure.

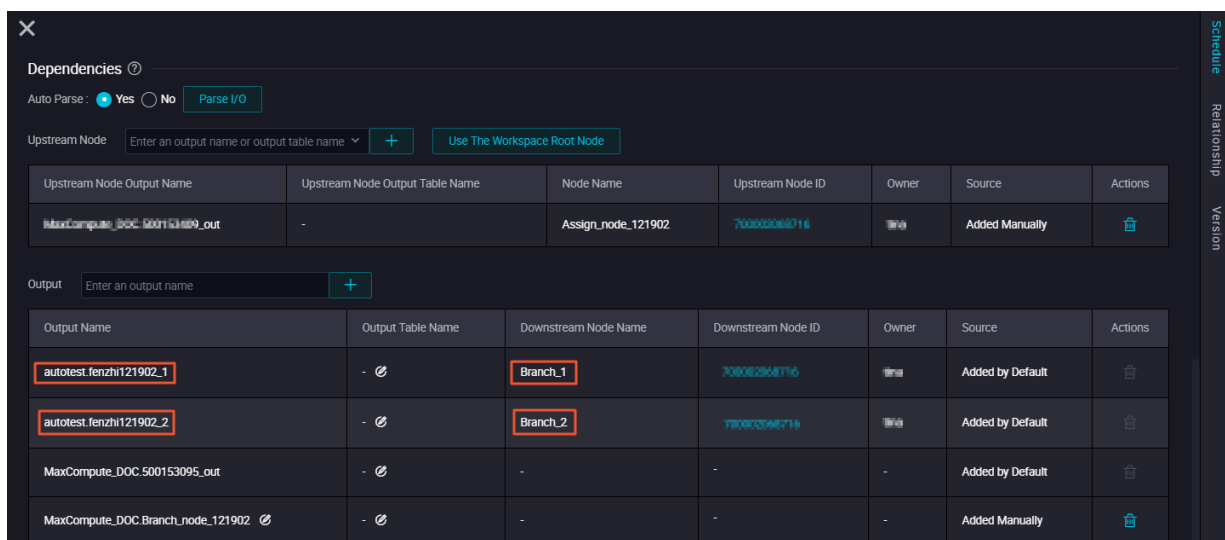


Branch	Conditions	Associated to node output	Branch describe	Actions
1	$\$ (input) == 1$	autotest.fenzhi121902_1	1	Edit Delete
2	$\$ (input) > 2$	autotest.fenzhi121902_2	2	Edit Delete

- Edit: Click **Edit** button, you can modify the setting branches and the relevant dependencies will also change.
- Delete: Click **Delete** button, you can delete the setting branches and the related dependencies will also change.

Scheduling configuration

After defining the branch condition, the output name is automatically added to the node **Output** of the **Schedule**, and the downstream node can rely on the output name to mount. As shown in the following figure:



Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500153095_out	-	Assign_node_121902	7000000000716		Added Manually	

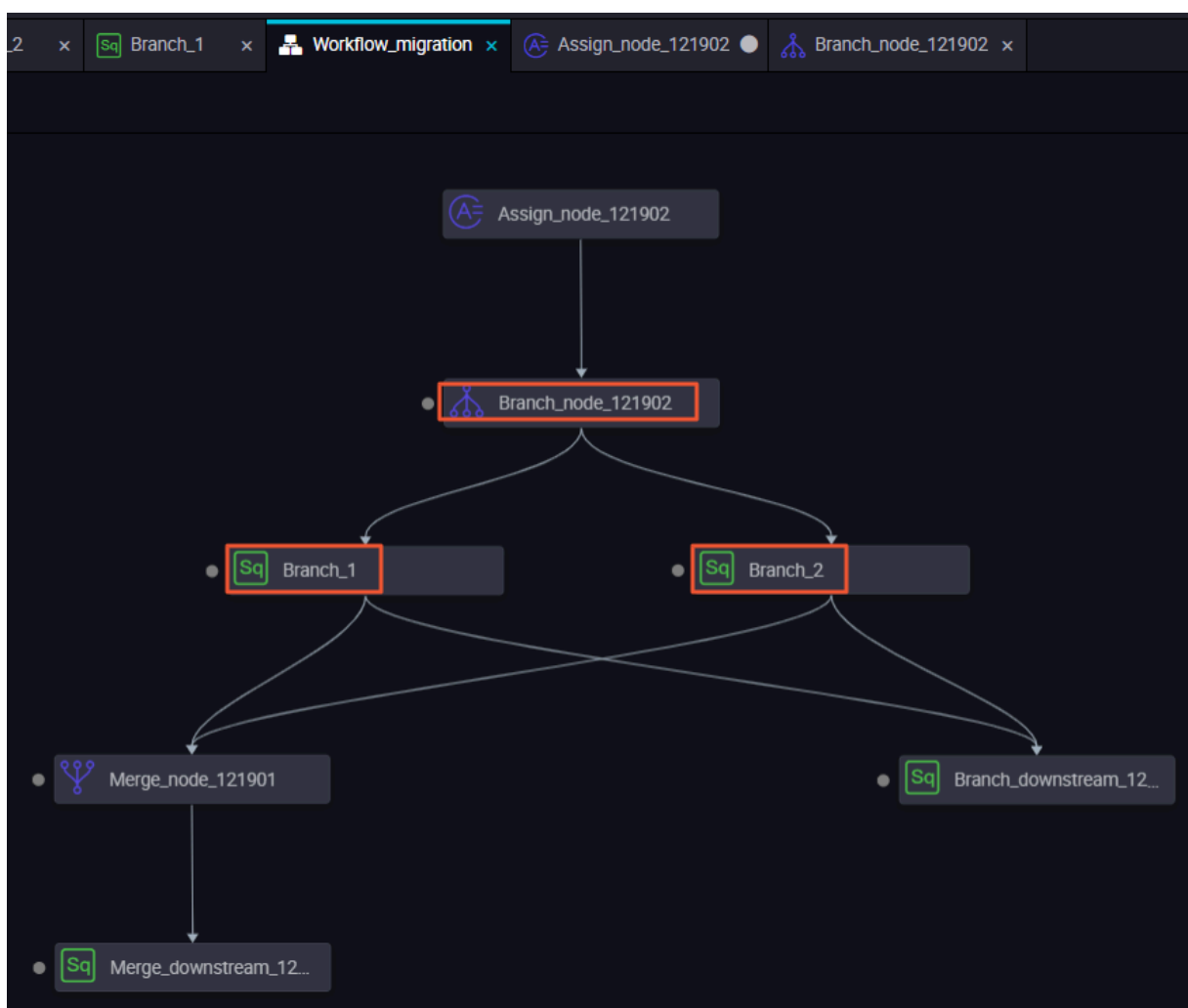
Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
autotest.fenzhi121902_1	-	Branch_1	7000000000716		Added by Default	
autotest.fenzhi121902_2	-	Branch_2	7000000000716		Added by Default	
MaxCompute_DOC.500153095_out	-	-	-	-	Added by Default	
MaxCompute_DOC.Branch_node_121902	-	-	-	-	Added Manually	

**Note:**

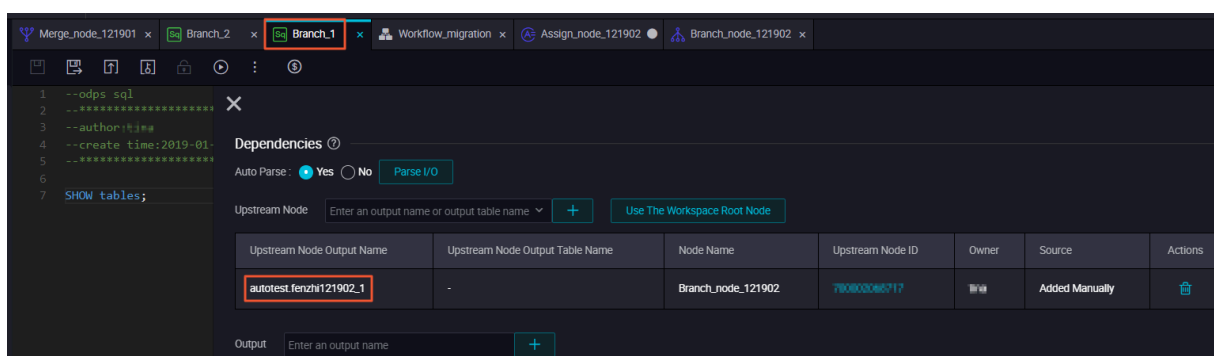
If there is no output record in the scheduling configuration for context dependencies established by wiring, enter it manually.

Output case - downstream node mounted to branch node

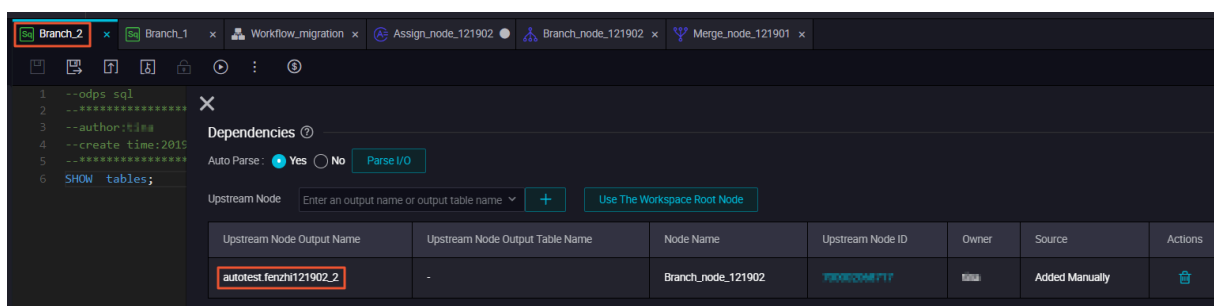
In the downstream node, after adding the branch node as the upstream node, you can define the branch direction under different conditions by selecting the corresponding branch node output. For example, in the business process shown in the figure below, **Branch_1** and **Branch_2** are both downstream nodes of the branch node.



Branch_1 depends on the output of 'autotest.fenzhi121902_1', as shown in the following figure.



Branch_2 depends on the output of 'autotest.fenzhi121902_2', as shown in the following figure.



Submit scheduling operation

Submit the dispatch to the operation center to run, and the branch node satisfies the condition (that is depending on 'autotest.fenzhi121902_1').Therefore, the print result of its log is as follows.

- When the branch condition is satisfied and select the downstream node of the branch to run. You can see the details of the run in **Running Log**.
- When the branch condition is not satisfied and do not select the downstream node of the branch to run. You can see that the node is set to 'skip' in **Running Log**.

Addition: supported Python comparison operators

In the table below, we assume that variable a is 10 and variable b is 20.

Comparison operators	Description	Example
==	Equal - compare objects for equality	(a==b) return 'false'
!=	Not equal - compare whether two objects are not equal.	(a!=b) return 'true'
<>	Not equal - compare whether two objects are not equal.	(a<>b) return 'true'. This operator is similar to '!='.

Comparison operators	Description	Example
>	Greater than - return whether x is greater than y.	(a>b) return 'false'
<	Less than - return whether x is less than y. All comparison operators return 1 for true and 0 for false. This is equivalent to the special variables True and False, respectively.	(a<b) return 'true'
>=	Greater than or equal to - return whether x is greater than or equal to y.	(a>=b) return 'false'
<=	Less than or equal to - return whether x is less than or equal to y.	(a<=b) return 'true'

3.5.11 Merge node

This article introduces the concept of merge node, how to create merge node and define merging logic. It also shows you the scheduling configuration and operation details of the merge node through a practical case.

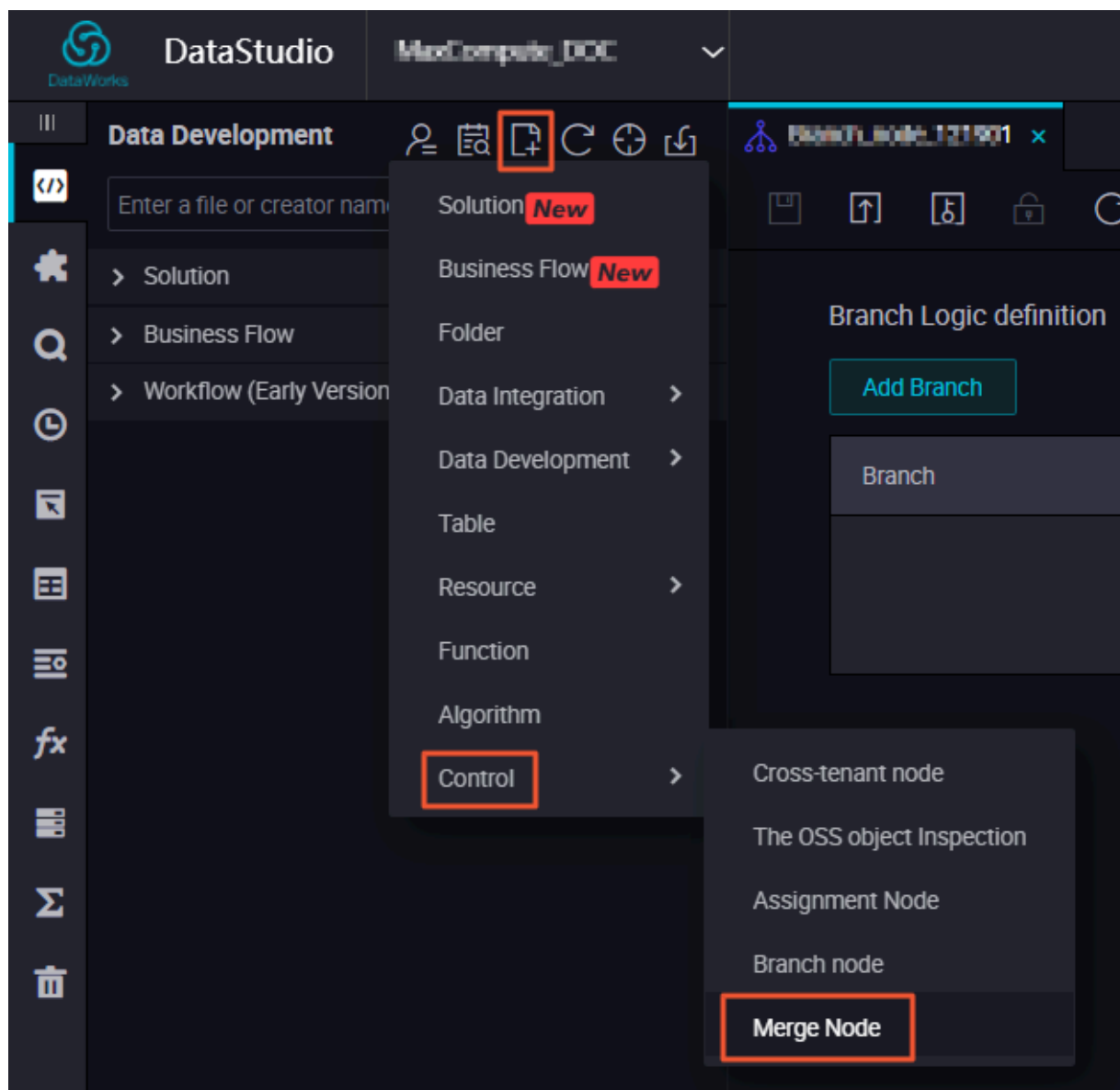
Concept

- The merge node is one of the logical control family nodes provided in DataStudio.
- The Merge node can merge the running states of upstream nodes, aiming to solve the problem of dependency mounting and running triggering of downstream nodes of branch nodes.
- The current logical definition of merge node does not support selecting the running state of the node, but only supports merging multiple downstream nodes of branch nodes into a successful merge, so that the more downstream nodes can directly mount the merge node as a dependency.

For example, branch node C defines two logically exclusive branches C1 and C2. Different branches use different logic to write to the same MaxCompute table. If downstream node B depends on the output of this MaxCompute table, it must use merge node J to merge branches first, then add merge node J as the upstream dependency of B. If B is mounted directly under C1 and C2, at any time, C1 and C2, one of them will always fail to run because of unsatisfactory branching conditions, and B can not be triggered by the schedule to run.

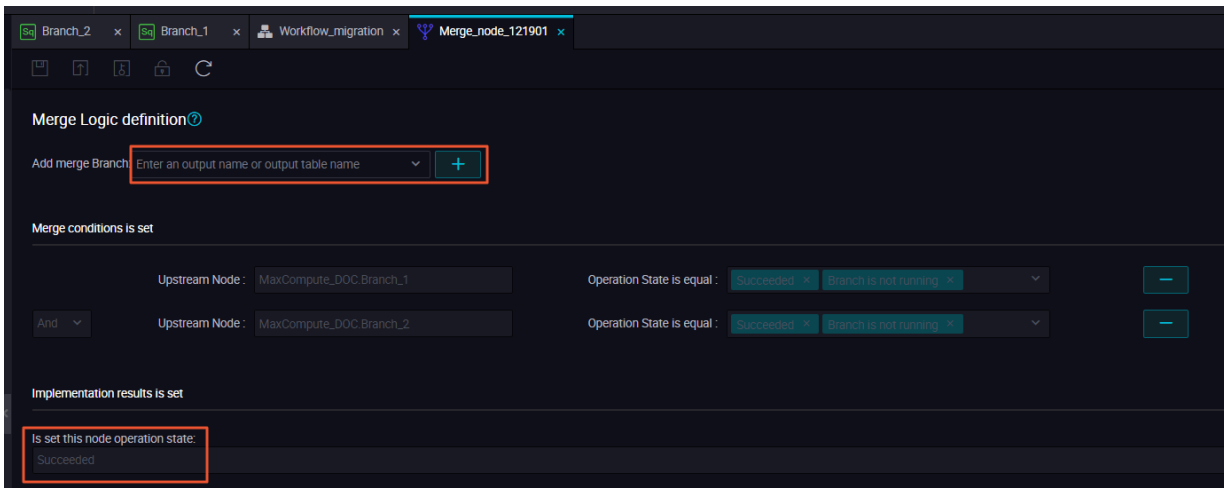
Create a merge Node

Merge Node is located in the **Control** class directory of the new node menu, as shown in the following figure.

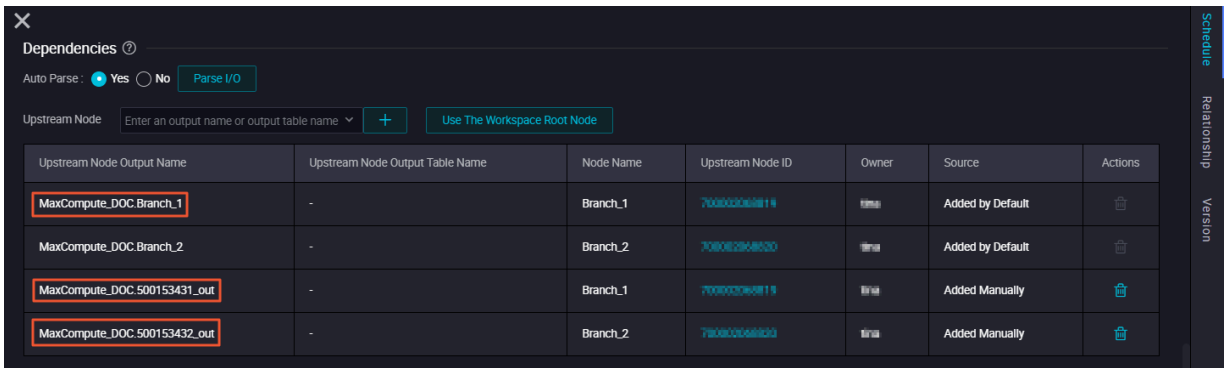


Define the merge Logic

Add merge branch. You can input the output name or output table name of the parent node, click add, you can see records under merge condition, and the execution results will show you the running status, currently there are only two running states: Successful, Branch not running, as shown in the following figure.

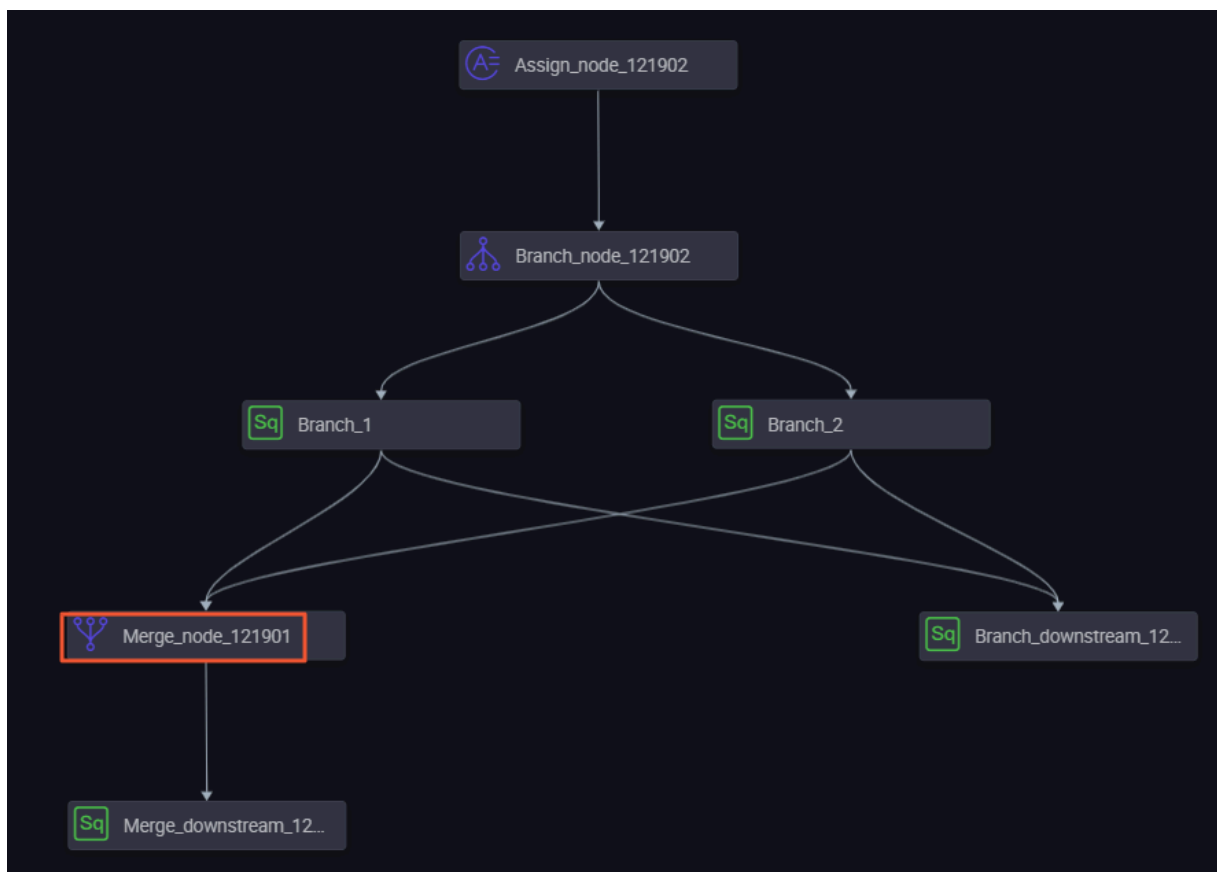


The scheduling attribute of the merge node is shown in the following figure.

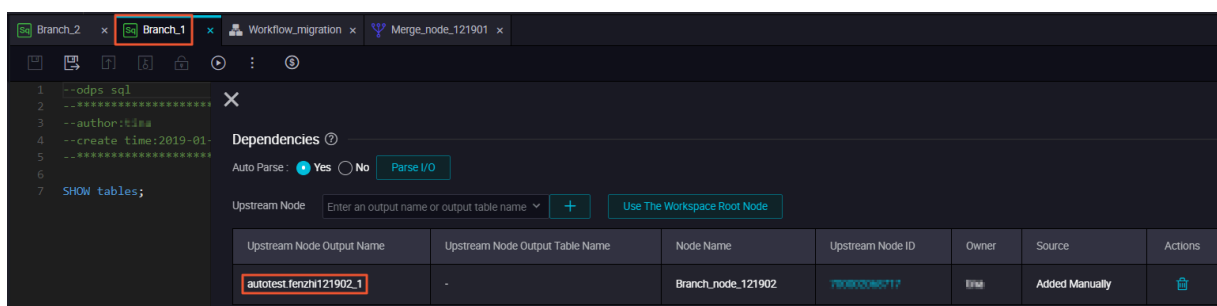


An example of merge node

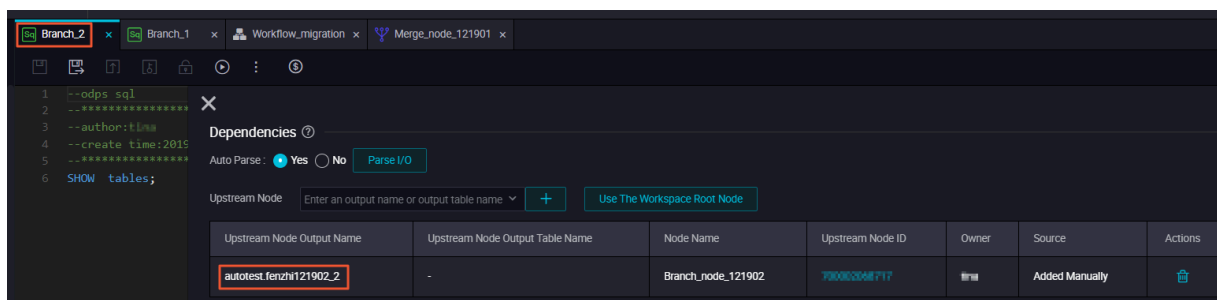
In the downstream node, after adding the branch node as the upstream node, you can define the branch direction under different conditions by selecting the corresponding branch node output. For example, in the business process shown in the figure below, **Branch_1** and **Branch_2** are both downstream nodes of the branch node.



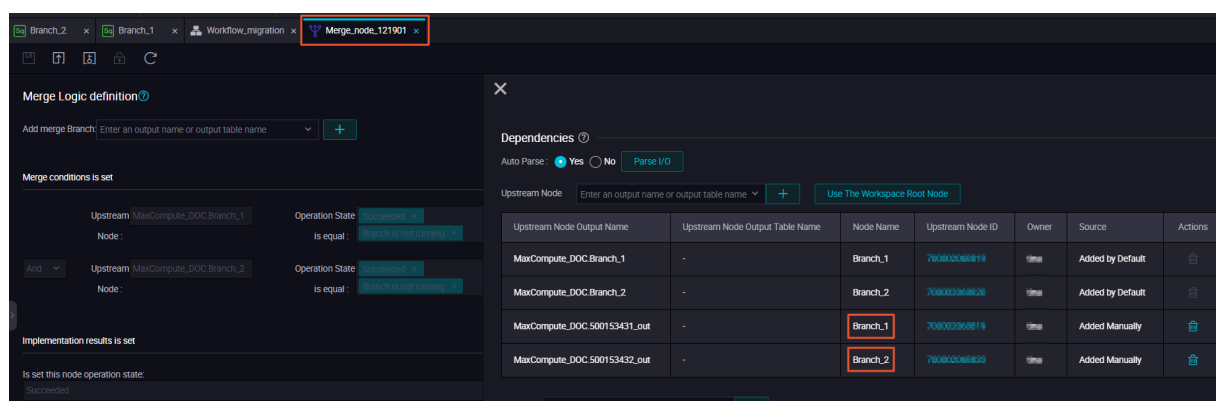
Branch_1 depends on the output of 'autotest.fenzhi121902_1', as shown in the following figure.



Branch_2 depends on the output of 'autotest.fenzhi121902_2', as shown in the following figure.



The scheduling attribute of the merge node is shown in the following figure.



Run the task

When the branch condition is satisfied and select the downstream node of the branch to run. You can see the details of the run in the **Running Log**.

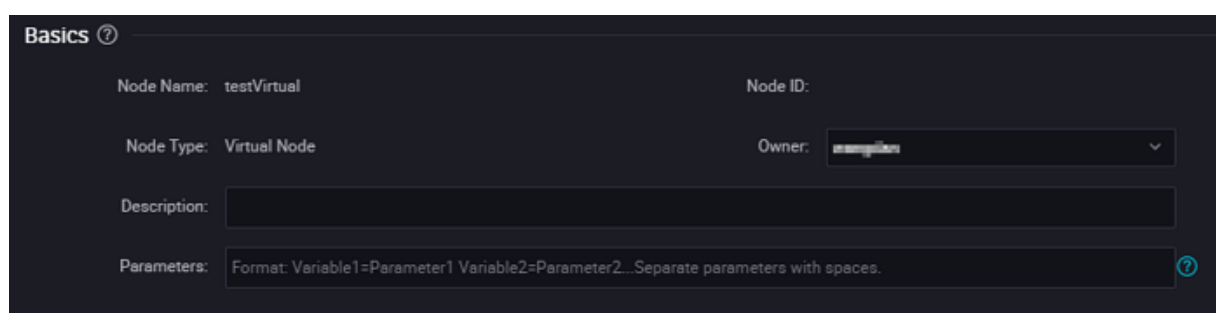
When the branch condition is not satisfied and do not select the downstream node of the branch to run. You can see that the node is set to 'skip' in the **Running Log**.

The downstream node of the merge node is running normally.

3.6 Scheduling Configuration

3.6.1 Basic attributes

The figure below shows the basic attribute configuration interface:



- **Node Name:** It is the node name that you enter when creating a workflow node. To modify a node name, right-click the node on the directory tree and choose **Rename** from the short-cut menu.
- **Node ID:** It is the unique node ID generated when a task is submitted, and cannot be modified.
- **Node Type:** It is the node type that you select when creating a workflow node, and cannot be modified.

- **Owner:** It is the node owner. The owner of a newly created node is the current logon user by default. To modify the owner, click the input box, and enter the owner name or directly select another user.


Note:

When you select another user, the user must be a member of the current project.

- **Description:** It is generally used to describe the business and purpose of the node.
- **Parameter:** It is used to assign value to a variable in the code during task scheduling.

For example, when a variable "pt=\${datetime}" is used to indicate the time in the code, you can assign a value to the variable here. The assigned value can use the scheduling built-in time parameter "datetime=\$bizdate".

Parameter value assignment formats for various node types

- ODPS SQL, ODPS PL, ODPS MR types: Variable name 1=Parameter 1 Variable name 2=Parameter 2..., Multiple parameters are separated by space.
- SHELL type: Parameter 1 Parameter 2..., Multiple parameters are separated by space.

Some frequently-used time parameters are provided as built-in scheduling parameters. For more information about these parameters, see [Parameter configuration](#).

3.6.2 Parameter configuration

To ensure that tasks can dynamically adapt to environment changes when running automatically at the scheduled time, DataWorks provides the parameter configuration feature. Pay special attention to the following two issues before configuring parameters:

- No space can be added on either side of the equation mark "=" of a parameter. Correct: bizdate=\$bizdate

The screenshot shows the 'Basics' configuration tab for a node. The 'Parameters' field is highlighted, showing the text 'bizdate=\$bizdate'. A red arrow points to the '=' sign, and a red text annotation above it says 'no space is added on both sides of the equal sign'.

- Multiple parameters (if any) must be separated by spaces.

Basics ?

Node Name: testVirtual Node ID:

Node Type: Virtual Node Owner: [dropdown]

Description: **if there are multiple parameters, each parameter is separated by spaces.**

Parameters: bizdate=\${bizdate} datetime=\${yyyymmdd} ?

System parameters

DataWorks provides two system parameters, which are defined as follows:

- `${bdp.system.cyctime}`: It is defined as the scheduled run time of an instance. Default format: `yyyymmddhh24miss`.
- `${bdp.system.bizdate}`: It is defined as the business date on which an instance is calculated. Default business data is one day before the running date, which is displayed in default format: `yyyymmdd`.

According to the definitions, the formula for calculating the runtime and business date is as follows: `Runtime = Business date + 1`.

To use the system parameters, directly reference `'${bizdate}'` in the code without setting system parameters in the editing box, and the system will automatically replace the reference fields of system parameters in the code.



Note:

The scheduling attribute of a periodic task is configured with a scheduled runtime. Therefore, you can backtrack the business date based on the scheduled runtime of an instance and retrieve the values of system parameters for the instance.

Example

Set an ODPS_SQL task that runs every hour between 00:00 and 23:59 every day. To use system parameters in the code, perform the following statement.

```
insert overwrite table tbl partition(ds = '20180606') select
c1,c2,c3
from (
select * from tb2
where ds = '${bizdate}');
```

Configure scheduling parameters for a non-Shell node



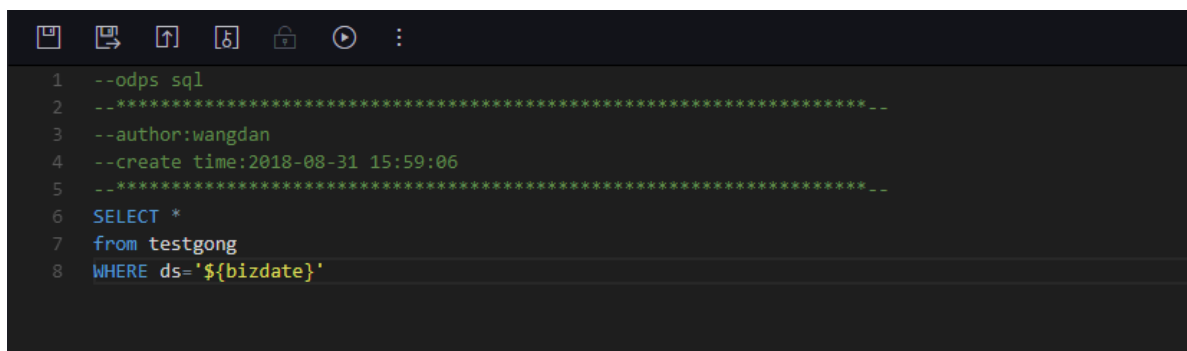
Note:

The name of a variable in the SQL code can contain only a-z, A-Z, numbers, and underlines. If the variable name is "date", the value "\$bizdate" is automatically assigned to this variable, and you do not need to assign the value in the scheduling parameter configuration. Even if another value is assigned, this value is not used in the code because the value "\$bizdate" is automatically assigned in the code by default.

For a non-Shell node, you need to first add \${variable name} (indicating that the function is referenced) in the code, then input a specific value to assign the value to the scheduling parameter.

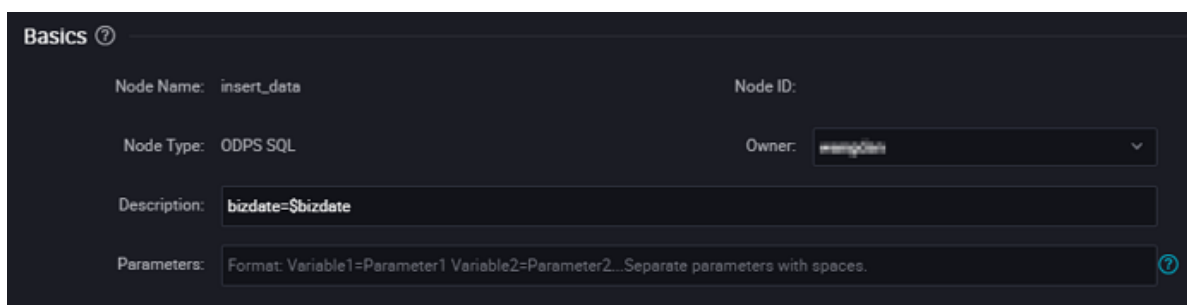
For example, for an ODPS SQL node, add \${variable name} in the code, and then configure the parameter item "variable name=built-in scheduling parameter" for the node.

1. For a parameter referenced in the code, you must add the resolved value during scheduling.



```
1  --odps sql
2  --*****
3  --author:wangdan
4  --create time:2018-08-31 15:59:06
5  --*****
6  SELECT *
7  from testgong
8  WHERE ds='${bizdate}'
```

2. Values must be assigned to variables referenced in the code. The value assignment rule is variable name=parameter.



Basics ⓘ

Node Name:	insert_data	Node ID:	
Node Type:	ODPS SQL	Owner:	wangdan
Description:	bizdate=\$bizdate		
Parameters:	Format: Variable1=Parameter1 Variable2=Parameter2... Separate parameters with spaces. ⓘ		

Configure scheduling parameters for a Shell node

The parameter configuration procedure of a Shell node is similar to that of a non-Shell node except that rules are different. For a Shell node, variable names cannot be customized and must be named '\$1,\$2,\$3...'.

For example, for a Shell node, the Shell syntax declaration in the code is: \$1, and the node parameter configuration in scheduling is: \$xxx (built-in scheduling parameter). That is, the value of \$xxx is used to replace \$1 in the code.

1. For a parameter referenced in the code, you must add the resolved value during scheduling.

```

1 #!/bin/bash
2 #####
3 ##author: [redacted]
4 ##create time:2018-06-16 17:27:47
5 #####
6
7 echo $1

```



Note:

For a Shell node, when the number of parameters reaches 10, \${10} should be used to declare the variable.

2. Values must be assigned to variables referenced in the code. The value assignment rule is parameter 1 parameter 2 parameter 3....(Replaced variables are resolved based on the parameter location, for example, \$1 is resolved to parameter 1).

Basics ?

Node Name: testSHELL Node ID: [empty]

Node Type: Shell Owner: [dropdown menu]

Description: [empty text area]

Parameters: \$bizdate ?

The variable value is a fixed value

Take an SQL node for example. For \${variable name} in the code, configure the parameter item "variable name="fixed value"" for the node.

Code: select xxxxxx type='\${type}'

Value assigned to the scheduling variable: type="aaa"

During scheduling, the variable in the code is replaced by type='aaa'.

The variable value is a built-in scheduling parameter

Take an SQL node for example. For \${variable name} in the code, configure the parameter item "variable name=scheduling parameter"" for the node.

Code: select xxxxxx dt=\${datetime}

Value assigned to the scheduling variable: datetime=\$bizdate

During scheduling, if today is July 22, 2017, the variable in the code is replaced by dt=20170721.

Built-in scheduling parameter list

\$bizdate: business date in the format of `yyyymmdd` NOTE: This parameter is widely used, and is the date of the previous day by default during routine scheduling.

For example, In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$bizdate`. Today is July 22, 2017. When the node is executed today, `$bizdate` is replaced by `pt=20170721`.

For example, In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$gmtdate`. Today is July 22, 2017. When the node is executed today, `$gmtdate` is replaced by `pt=20170722`.

\$cyctime: scheduled time of the task. If no scheduled time is configured for a daily task, `cyctime` is 00:00 of the current day. The time is accurate to hour, minute, and second, and is generally used for a hour-level or minute-level scheduling task. Example: `cyctime=$cyctime`.



Note:

Pay attention to the difference between the time parameters configured using `[]` and `{}`.

\$bizdate: business date, which is one day before the current time by default. **\$cyctime:** It is the scheduled time of the task. If no scheduled time is configured for a daily task, the task is executed on 00:00 of the current day. The time is accurate to hour, minute, and second, and is generally used for an hour-level or minute-level scheduling task. If a task is scheduled to run on 00:30, for example, on the current day, the scheduled time is `yyyy-mm-dd 00:30:00`. If the time parameter is configured using `[]`, `cyctime` is used as the benchmark for running. For more information about the usage, see the instructions below. The time calculation method is the same with that of Oracle. During data population, the parameter is replaced by the selected business date plus 1 day. For example, if the business date 20140510 is selected during data population, `cyctime` will be replaced by 20140511.

\$jobid: ID of the workflow to which a task belongs. Example: `jobid=$jobid`.

\$nodeid: ID of a node. Example: `nodeid=$nodeid`

\$taskid: ID of a task, that is, ID of a node instance. Example: `taskid=$taskid`.

\$bizmonth: business month in the format of `yyyymm`.

- If the month of a business date is equal to the current month, `$bizmonth` = Month of the business date - 1; otherwise, `$bizmonth` = Month of the business date.

- For example: In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$bizmonth`. Today is July 22, 2017. When the node is executed today, `$bizmonth` is replaced by `pt=201706`.

`$gmtdate`: current date in the format of `yyyymmdd`. The value of this parameter is the current date by default. During data population, `gmtdate` that is input is the business date plus 1.

Custom parameter `${...}` Parameter description:

- Time format customized based on `$bizdate`, where `yyyy` indicates the 4-digit year, `yy` indicates the 2-digit month, `mm` indicates the month, and `dd` indicates the day. The parameter can be combined as expected, for example, `${yyyy}`, `${yyyymm}`, `${yyyymmdd}`, and `${yyyy-mm-dd}`.
- `$bizdate` is accurate to year, month, and day. Therefore, the custom parameter `${.....}` can only represent the year, month, or day.
- Methods for obtaining the period before or after a certain duration:

Next N years: `${yyyy+N}`

Previous N years: `${yyyy-N}`

Next N months: `${yyyymm+N}`

Previous N months: `${yyyymm-N}`

Next N weeks: `${yyyymmdd+7*N}`

Previous N weeks: `${yyyymmdd-7*N}`

Next N days: `${yyyymmdd+N}`

Previous N days: `${yyyymmdd-N}`

`${yyyymmdd}`: business date in the format of `yyyymmdd`. The value is consistent with that of `$bizdate`.

- This parameter is widely used, and is the date of the previous day by default during routine scheduling. The format of this parameter can be customized, for example, the format of `${yyyy-mm-dd}` is `yyyy-mm-dd`.
- For example: In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyymmdd}`. Today is July 22, 2013. When the node is executed today, `${yyyymmdd}` is replaced by `pt=20130721`.

`${yyyymmdd-/+N}`: `yyyymmdd` plus or minus N days

`${yyyymm-/+N}`: `yyyymm` plus or minus N month

`${yyyy-/N}`: year (yyyy) plus or minus N years

`${yy-/N}`: year (yy) plus or minus N years

yyyymmdd indicates the business date and supports any separator, such as yyyy-mm-dd. The preceding parameters are derived from the year, month, and day of the business date.

Example:

- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyy-mm-dd}`. Today is July 22, 2018. When the node is executed today, `${yyyy-mm-dd}` is replaced by `pt=2018-07-21`.
- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyymmdd-2}`. Today is July 22, 2018. When the node is executed today, `${yyyymmdd-2}` is replaced by `pt=20180719`.
- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyymm-2}`. Today is July 22, 2018. When the node is executed today, `${yyyymm-2}` is replaced by `pt=201805`.
- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyy-2}`. Today is July 22, 2018. When the node is executed today, `${yyyy-2}` is replaced by `pt=2018`.

In the ODPS SQL node configuration, multiple parameters are assigned values, for example, `startdatetime=$bizdate enddatetime=${yyyymmdd+1} starttime=${yyyy-mm-dd} endtime=${yyyy-mm-dd+1}`.

Example: (Assume `$cyctime=20140515103000`)

- `${yyyy}` = 2014, `${yy}` = 14, `${mm}` = 05, `${dd}` = 15, `${yyyy-mm-dd}` = 2014-05-15, `${hh24:mi:ss}` = 10:30:00, `${yyyy-mm-dd hh24:mi:ss}` = 2014-05-1510:30:00
- `${hh24:mi:ss - 1/24}` = 09:30:00
- `${yyyy-mm-dd hh24:mi:ss -1/24/60}` = 2014-05-1510:29:00
- `${yyyy-mm-dd hh24:mi:ss -1/24}` = 2014-05-1509:30:00
- `[add_months(yyyymmdd,-1)]` = 2014-04-15
- `[add_months(yyyymmdd,-12*1)]` = 2013-05-15
- `[hh24]` =10
- `[mi]` =30

Method for testing the parameter `$cyctime`:

After the instance runs, right-click the node to **check the node attribute**. Check whether the scheduled time is the time at which the instance runs periodically.

Result after the parameter value is replaced by the scheduled time minus one hour.

FAQ

- Q: The table partition format is `pt=yyyy-mm-dd hh24:mi:ss`, but spaces are not allowed in scheduling parameters. How should I configure the format of `${yyyy-mm-dd hh24:mi:ss}`?

A: Use the custom variable parameters `datetime=${yyyy-mm-dd}` and `hour=${hh24:mi:ss}` to acquire the date and time, respectively. Then, join them together to form `pt="${datetime} ${hour}"` in code. (The two custom parameters are separated by space).

- Q: The table partition is `pt="${datetime} ${hour}"` in code. To acquire the data for the last hour during execution, the custom variable parameters `datetime=${yyyymmdd}` and `hour=${hh24-1/24}` can be used to acquire the date and time, respectively. However, for an instance running at 0:00, the calculation result is 23:00 of the current day, instead of 23:00 of the previous day. What measures should be taken in this case?

A: Modify the formula of `datetime` to `${yyyymmdd-1/24}` and remain the formula of `hour` `${hh24-1/24}`. The calculation result is as follows:

- For an instance with the scheduled time of 2015-10-27 00:00:00, the values of `${yyyymmdd-1/24}` and `${hh24-1/24}` are 20151026 and 23, respectively, because the scheduled time minus one hour is a time value belonging to yesterday.
- For an instance with the scheduled time of 2015-10-27 01:00:00, the values of `${yyyymmdd-1/24}` and `${hh24-1/24}` are 20151027 and 00, respectively, because the scheduled time minus one hour is a time value belonging to the current day.

Dataworks provides four ways to run.

- Running on data development pages: Temporary value assignment is needed on the parameter configuration page to ensure the proper running. However, the assignment is not saved as the task attribute, and does not take effect in other three running modes.
- Automatic run at an interval: No configuration is needed in the parameter editing box, and the scheduling system automatically replaces the parameters with the scheduled runtime of the current instance.
- Test run/data supplement run: A business date needs to be specified when the run is triggered, and the scheduled runtime is derived from the formula described earlier to get the two system parameter values of each instance.

3.6.3 Time attributes

The time attribute configuration page is shown in the following figure:

Node states

- Normal: Nodes are normally scheduled based on the following scheduling cycle. This option is selected by default.
- Zero-load: After this option is selected, nodes are configured and scheduled based on the following scheduling cycle. However, once this task is scheduled, a success is directly returned without executing the task.
- Error retries: the node has encountered an error, and the node can be rerun. Default error automatically retries 3 times, time interval 2 minutes.
- Suspend scheduling: After this check box is selected, nodes are configured and scheduled based on the following scheduling cycle. However, once this task is scheduled, a failure is directly returned without executing the task. It is used when a task is suspended but will be executed later.

Scheduling interval

In DataWorks, when a task is successfully submitted, the underlying scheduling system generates an instance every day starting from the next day based on the time attributes of the task, and runs the instances based on the running results and time points of the depended upstream instances. For a task that is successfully submitted after 23:30, the instances are generated starting from the third day.

**Note:**

If a task needs to run on every Monday, the task runs only when the runtime is Monday. If the runtime is not Monday, the task (which is directly set to successful) runs pretendedly. For this reason, select Business date = Runtime -1 for weekly scheduled tasks during test or data supplement run.


For a task that runs cyclically, the priority of its dependency is higher than that of its time attribute. This means that, when the time specified by its time attribute reaches, the task instance does not run immediately but first checks whether all the upstream instances have run successfully.

- If not all the depended upstream instances run successfully and the scheduled runtime is reached, the instance remains in the not running status.
- If not all the depended upstream instances run successfully and the scheduled runtime is reached, the instance remains in the not running status.
- If all the depended upstream instances run successfully and the scheduled runtime is reached, the instance enters the waiting for resource status to be ready for running.

Daily scheduling


Daily scheduled tasks run automatically once every day. When you create a cyclic task, the task is set to run at 00:00 every day by default. You can specify another runtime as needed. For example, you can specify the runtime as 13:00 every day, as shown in the following figure.

1. If Regular Scheduling is deselected, the scheduled time of instances of the daily task is the date of the current day in YYYY-MM-DD and the default scheduling time that is randomly generated between 0:00 and 0:30.
2. If Regular Scheduling is selected, the scheduled time of instances of the daily task is the date of the current day in YYYY-MM-DD and the scheduled time in HH:MM:SS. A scheduled task can run only when the upstream task successfully runs, and the scheduled time is reached. If either condition is not met, the task cannot run. The conditions do not have the order.


Validity Period : 1970-01-01 - 9999-01-01 

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Day 

Plan Time : ☒

Planned Time : 13:00 

Note: The default planned time is randomly selected from 0.00 to 0.30.

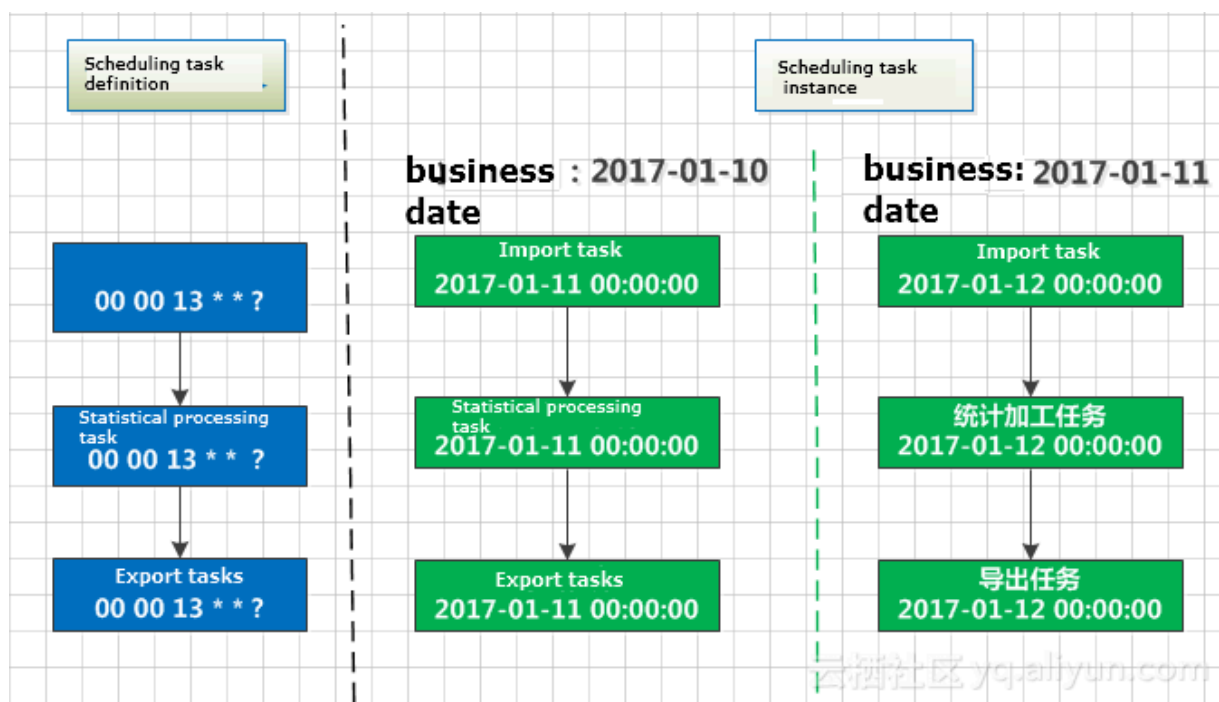
CRON Expression : 00 00 13 * * ?

Depend on Last Interval : ☐

Use cases:

Import, statistical processing, and export tasks are all daily tasks with the runtime of 13:00, as shown in the preceding figure. Statistical processing tasks depend on import tasks, and export tasks depend on statistical processing tasks. The following figure shows the configuration of their dependencies(In the dependency attribute configuration for the statistical processing tasks, the upstream task is set to import task).

Based on the configuration in the preceding figure, the scheduling system automatically generates instances for the tasks and runs them as follows:



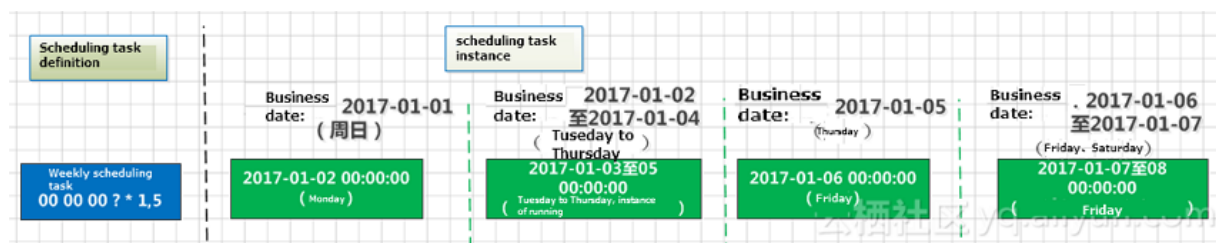
Weekly scheduling

Weekly scheduled tasks automatically run at specific time points of specific days each week.

When an unspecified date reaches, the system also generates instances and directly sets them as successfully running without running any logic or consuming any resource to ensure the proper running of downstream instances.

As shown in the preceding figure, instances generated on every Monday and Friday run as scheduled, and other instances generated on every Tuesday, Wednesday, Thursday, Saturday, and Sunday are directly set as successfully running.

Based on the configuration in the preceding figure, the scheduling system automatically generates instances for the tasks and runs them as follows:



Monthly scheduling

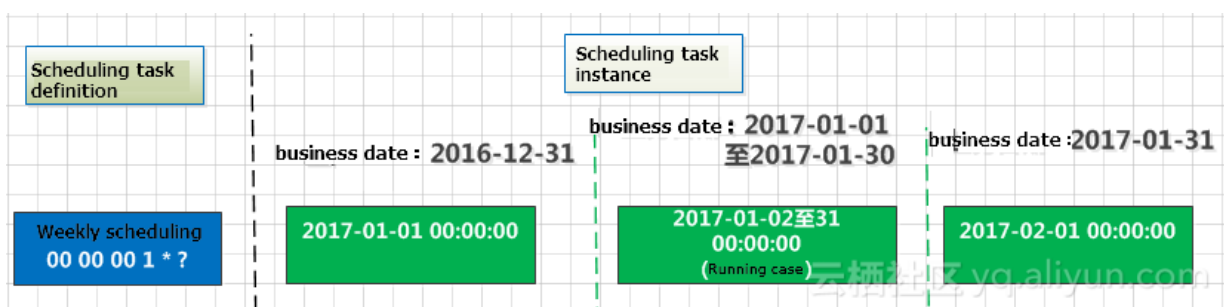
Monthly scheduled tasks run automatically at specific time points of specific days each month.

When an unspecified date reaches, the system also generates instances every day and directly

sets them as successfully running without running any logic or consuming any resource to ensure the proper running of downstream instances.

As shown in the preceding figure, instances generated on the first day of each month run as scheduled, and instances generated every day for the rest days of the month are directly set as successfully running.

Based on the configuration in the preceding figure, the scheduling system automatically generates instances for the tasks and runs them as follows:



Hourly scheduling

Hourly scheduled tasks run every $N \times 1$ hours in a specific period each day, such as running every one hour every day from 1:00 to 4:00.



Note:

The running interval is calculated based on the left-close and right-close principle. For example, if an hourly scheduled task is configured to run every one hour between 0:00 and 3:00, it indicates that the time period is [00:00, 03:00], and the interval is one hour. The scheduling system generates four instances every day, which run at 0:00, 1:00, 2:00 and 3:00.

Error Rate this product : ☐ ?

Validity Period : 1970-01-01 - 9999-01-01

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Hour

Plan Time : ☒

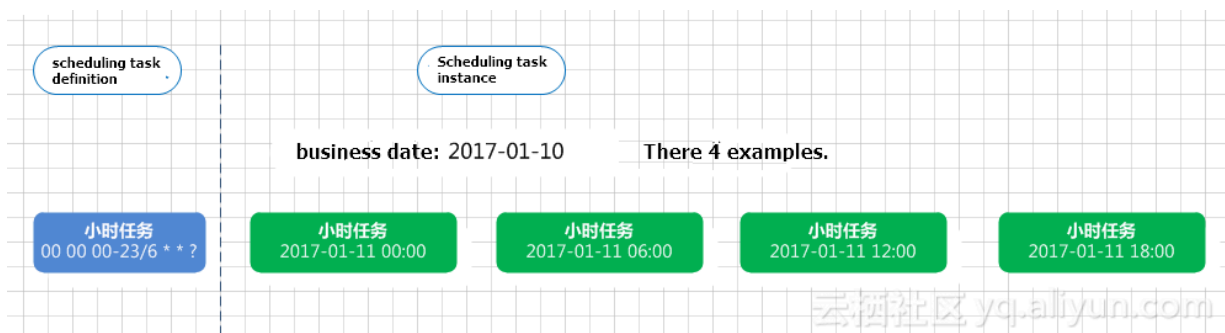
Start Time : 00:00 Interval : 1 h End Time : 23:59

Specified Time : 0:00

CRON Expression : 00 00 00-23/1 * * ?

Depend on Last Interval : ☐

As shown in the preceding figure, an automatic scheduling is triggered every six hours every day from 00:00 to 23:59. Therefore, the scheduling system automatically generates instances for the task and runs them as follows:



By-minute scheduling

By-minute scheduled tasks run every $N \times 1$ minutes in a specific period each day, as shown in the following figure:

The task is scheduled every 30 minutes from 00:00 to 23:00 each day.

Schedule ?

Schedule : ☒ Normal ☐ Zero-load

Error Rate this product : ☐ ?

Validity Period : 1970-01-01 - 9999-01-01

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval : Minute

Plan Time : ☒

Start Time : 00:00

Interval : 30 min

End Time : 23:00

CRON Expression : 00 */30 00-23 ** ?

Currently, by-minute scheduling supports the granularity of at least five minutes. The time expression must be selected and cannot be manually modified.

Schedule ?

Schedule : ☒ Normal ☐ Zero-load

Error Rate this product : ☐ ?

Validity Period : 1970-01-01 - 9999-01-01

Note: The schedule will be effective date effect and automatic scheduling, on the other hand, validity Period of the task will not be automatic scheduling, manual scheduling.

Pause Scheduling : ☐

Schedule Interval :

Plan Time : ☐

Start Time :

Interval : 30 min

End Time : 23:59

CRON Expression : 00 */30 00-23 ** ?

FAQ

Q: If my upstream task A is an hourly scheduled task and downstream task B is a daily scheduled task, and task B needs to be executed once each day after task A is completed, can tasks A and B be mutually dependent?

A: A daily task can depend on an hourly task. If task A is configured as an hourly scheduled task, task B is configured as a daily task that is irregularly scheduled, and tasks A and B are mutually dependent, task B can run after task A successfully runs instances for 24 hours each day. (For more information about the dependency configuration, see the scheduling dependency description). Therefore, tasks of each cycle can depend on each other, and the scheduling cycle of each task is determined by the time attribute of the task.

Q: I want my task A to run once each hour and task B to run once each day, and task B starts to run after the first time that task A successfully runs. How can I configure it?

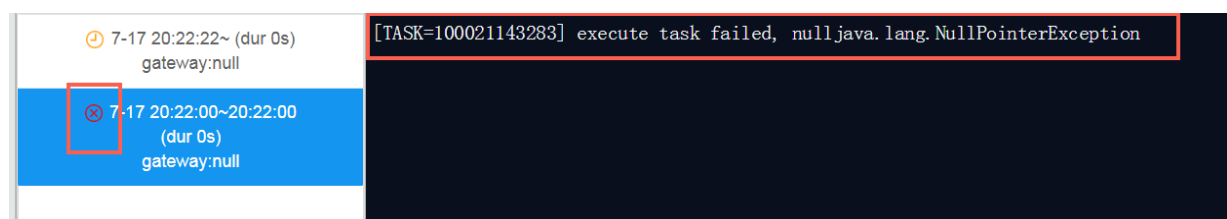
A: When configuring task A, you need to select Previous Cycle Dependent and Current Node, and set the scheduled time of task B to 0:00. In this way, instances of task B in the automatically scheduled instances each day only depend on the 0:00 instance of task A, that is, the first instance of task A.

Q: If task A runs on every Monday and task B depends on task A, how can I configure to enable task B to run on every Monday?

A: You can set the time attribute of task B the same as that of task A, that is, you need to set the scheduling cycle to Weekly Scheduling and Monday.

Q: Are the instances of a task affected when the task is deleted?

A: When a task is deleted after running for a period, its instances are remained because the scheduling system still generates one or more instances for the task according to the time attribute . For this reason, when the instances are triggered after the task is deleted, the following error message is displayed because the required code cannot be found:



Q: What can I do if I want to calculate monthly data on the last day of each month?

A: Currently, the system does not support setting the runtime as the last day of each month.

Therefore, if the task is set to run on the 31st day of each month, scheduling is triggered on one day for the month having 31 days, and instances are generated and directly set as successfully running on other days.

For monthly statistics, we recommend that you calculate the data for the previous month on the first day of each month.

3.6.4 Dependencies

Scheduling dependency is the foundation of constructing orderly business process. Only by correctly configuring dependencies between tasks, business data can be produced effectively and timely.

DataWorks V2.0 provides three dependency configuration modes: automatic recommendation, automatic parsing and custom configuration. See [Best practices for setting scheduling dependencies](#) for an example of the actual operation of dependencies.



Note:

You can watch videos to learn more about dependencies: [DataWorks V2.0 FAQs and Difficulty Analysis](#).

The scheduling dependency configuration page is shown in the following figure:

Overall scheduling logic: The downstream scheduling can be started only when the upstream scheduling is successfully implemented. Therefore, all workflow nodes must have at least one parent node. Scheduling dependency is used to set the parent-child relationship. The principle and configuration of scheduling dependency configuration are described in detail as follows.

**Note:**

If there is a need for interdependence between standard mode and simple mode projects, please apply for a bill of lading.

Introduction to standardized R&D scenarios

- Concept preparation
 - DataWorks Task: See [Concepts](#) for details.
 - Output Name: See [Concepts](#) for details. The system will assign a default output name ending with '.out' for each node, and you can also add a custom output name, but note that the node output name is not allowed to repeat within the tenant.
 - Output table: refers to the table after the INSERT or CREATE in the SQL statement of a node.
 - Input table: refers to the table after the FROM in the SQL statement of a node.
 - SQL statement: refers to [MaxCompute SQL](#).

In practice, a DataWorks task can contain a single SQL statement or multiple SQL statements.

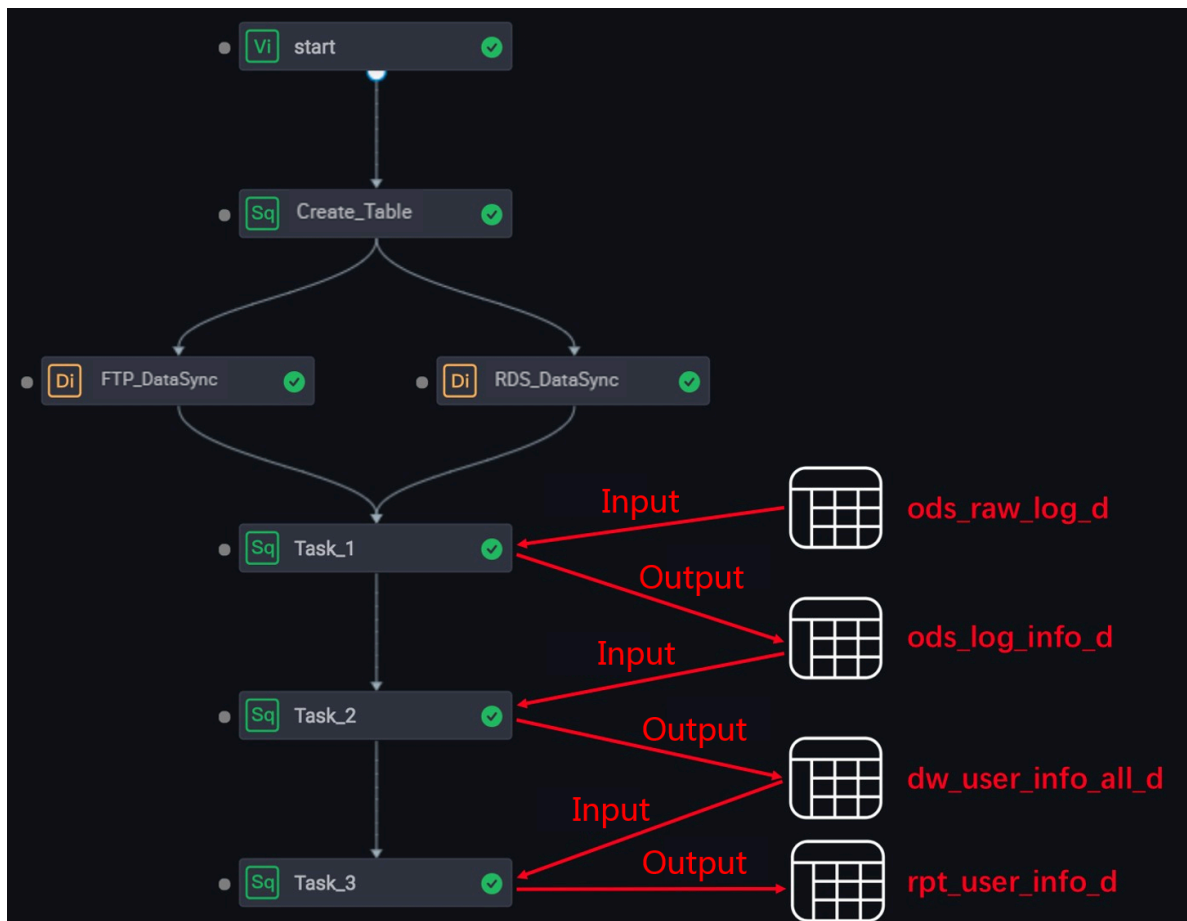
Each task that forms an upstream and downstream relationship is associated by an output name, and the root node of the project (node name: projectname_root) can be configured as the upstream node of the most upstream node created.

- Introduction to the standard development process

In the normalized development process, multiple SQL tasks are established to form a dependency between upstream and downstream, and we recommended to follow:

- The input table for the downstream task must be the output table for the upstream task.
- The same table is output by only one task.

The purpose is to quickly configure complex dependencies through "Auto Parse" when business processes are inflated.



In the figure above, each task and its code are as follows.

- The task code of Task_1 is as follows. The input data of this task comes from the table "ods_raw_log_d", and the data is output to the table "ods_log_info_d".

```
INSERT OVERWRITE TABLE ods_log_info_d PARTITION (dt=${bdp.system.bizdate})
SELECT ..... //Refers to your select operation
FROM (
    SELECT ..... //Refers to your select operation
    FROM ods_raw_log_d
    WHERE dt = ${bdp.system.bizdate}
) a;
```

- The task code of Task_2 is as follows. The input data of this task comes from the table "ods_user_info_d" and table "ods_log_info_d", and the data is output to the table "dw_user_info_all_d".

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.bizdate}')
SELECT ..... //Refers to your select operation
FROM (
    SELECT *
    FROM ods_log_info_d
    WHERE dt = ${bdp.system.bizdate}
) a
```

```

LEFT OUTER JOIN (
  SELECT *
  FROM ods_user_info_d
  WHERE dt = ${bdp.system.bizdate}
) b
ON a.uid = b.uid;

```

- The task code of Task_3 is as follows. The input data of this task comes from the table "dw_user_info_all_d", and the data is output to the table "rpt_user_info_d".

```

INSERT OVERWRITE TABLE rpt_user_info_d PARTITION (dt='${bdp.system.bizdate}')
SELECT ..... //Refers to your select operation
FROM dw_user_info_all_d
WHERE dt = ${bdp.system.bizdate}
GROUP BY uid;

```

Depended upstream node

Upstream node: Specifies the parent node that the current node depends on.

Here, it is required to enter the output name of upstream node (one node may have multiple output names at the same time, only enter one), rather than the upstream node name. You can manually search for the output name of upstream node to add, or you can parse it through the SQL blood code.

Dependencies ⓘ

Auto Parse: ☒ Yes ☐ No [Parse I/O](#)

Upstream Node [+](#) [Use The Workspace Root Node](#)

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DDC_root	-	maxcompute_ddc_root	70000002794	digital_dba	Added Manually	🗑️
MaxCompute_DDC_jd	-	-	-	-	Auto Parse	🗑️

Output [+](#)

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DDC_500117448_out	- 🗑️	-	-	-	Added by Default	🗑️
MaxCompute_DDC_task_B 🗑️	- 🗑️	-	-	-	Added Manually	🗑️



Note:

If added by search, the searcher searches according to the output name of the node that has been submitted to the scheduling system.

- Search by entering output name of the parent node

You can construct a dependency by searching for the output name of a node and configuring it as the upstream dependency of the current node.

Upstream Node

Auto Parse: ☒ Yes ☐ No [Parse I/O](#)

Upstream Node: [+](#) [Use The Workspace Root Node](#)

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC_id	-	-	-	-	Auto Parse	
maxcompute_doc_root	-	maxcompute_doc_root	760006833799	dtplus_docs	Added Manually	

Output: [+](#)

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC_500117440_ven	-	-	-	-	Added by Default	

Downstream Node

Auto Parse: ☒ Yes ☐ No [Parse I/O](#)

Upstream Node: [^](#) [+](#) [Use The Workspace Root Node](#)

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC_id	-	-	-	-	Auto Parse	
maxcompute_doc_root	-	-	-	-	-	-

Output: [+](#)

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC_500117440_ven	-	-	-	-	Added by Default	

- Search by entering the table name of the parent node's output name

This method must ensure that one of the output names of the parent node is the table name after INSERT or CREATE in the SQL code of the node, such as "projectname.tablename" (such output name can generally be obtained through automatic parsing).

task_2 task_1 task_3 ipint_test test

```
--odps.sql
--author:dtplus_docs
--create time:2018-11-27 19:48:48
INSERT OVERWRITE TABLE ipint_test
select * from ipresource
WHERE ipint('1.2.24.2') >= start_ip
AND ipint('1.2.24.2') <= end_ip
```

Dependencies

Auto Parse: ☒ Yes ☐ No [Parse I/O](#)

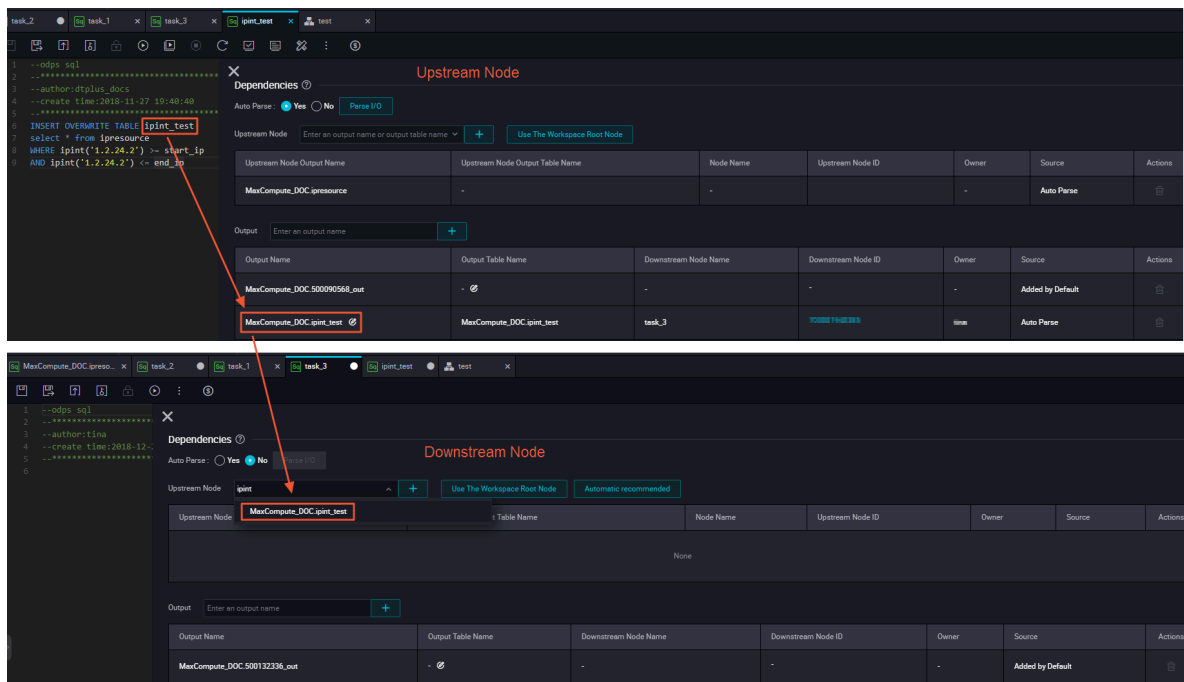
Upstream Node: [+](#) [Use The Workspace Root Node](#)

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC_ipresource	-	-	-	-	Auto Parse	

Output: [+](#)

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC_500090368_out	-	-	-	-	Added by Default	
MaxCompute_DOC_ipint_test	MaxCompute_DOC_ipint_test	task_3	752881768288	lms	Auto Parse	

After the submission is executed, the output name can be searched by other nodes by searching the table name.



Current node output

Output: Specifies output of the current node.

Each node is assigned a default output name ending with ".out", and you can also add a custom output name or get an output name through automatic parsing.



Note:

The name of the output node is globally unique and no duplication is allowed in the entire Alibaba Cloud account system.

Auto-parsing dependencies

DataWorks can parse different dependencies according to the actual SQL content in the task node. The output names of the parent node and the current node that obtained by parsing are as follows.

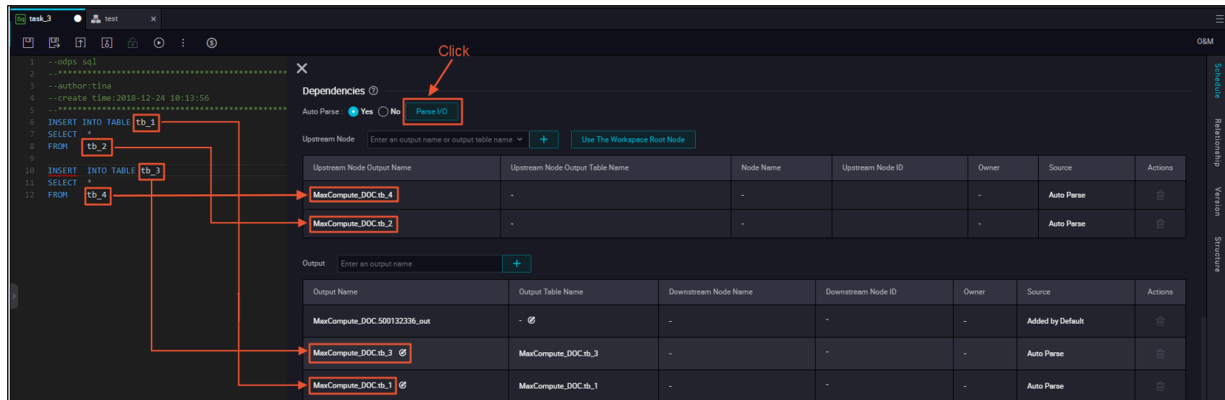
- Output name of the parent node: the table name after projectname.INSERT.
- Output names of the current node:
 - the table name after projectname.INSERT.
 - the table name after projectname.CREATE (Generally used for temporary tables).



Note:

If you upgrade from DataWorks V1.0 to DataWorks V2.0, the output name of the current node is "projectname.nodename".

If multiple INSERT and FROM statements are displayed, multiple output and input names will be parsed.



If you construct multiple tasks with dependencies, and these tasks satisfy the condition that all input tables of downstream tasks come from the output tables of upstream tasks, the fast configuration of full workflow dependencies can be achieved by automatic parsing.

Upstream Node

Dependencies

Auto Parse: ☒ Yes ☐ No

Upstream Node: Enter an output name or output table name

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC_root	-	maxcompute_doc_root	700000822799	dpplus_aliem	Added Manually	
MaxCompute_DOC.tb.2	-	-	-	-	Auto Parse	

Output: Enter an output name

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC_500132330_out	-	task_2	700001940384	dpplus_aliem	Added by Default	
MaxCompute_DOC.tb.2	MaxCompute_DOC.tb.2	-	-	-	Auto Parse	

Primary sub-node

Dependencies

Auto Parse: ☒ Yes ☐ No

Upstream Node: Enter an output name or output table name

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC_root	-	maxcompute_doc_root	700000822799	dpplus_aliem	Added Manually	
MaxCompute_DOC.tb.2	-	-	-	-	Auto Parse	

Output: Enter an output name

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC_500132335_out	-	task_3	700001940385	dpplus_aliem	Added by Default	
MaxCompute_DOC.tb.3	MaxCompute_DOC.tb.3	-	-	-	Auto Parse	

Secondary sub-node

Dependencies

Auto Parse: ☒ Yes ☐ No

Upstream Node: Enter an output name or output table name

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC_root	-	maxcompute_doc_root	700000822799	dpplus_docs	Added Manually	
MaxCompute_DOC.tb.3	-	-	-	-	Auto Parse	

Output: Enter an output name

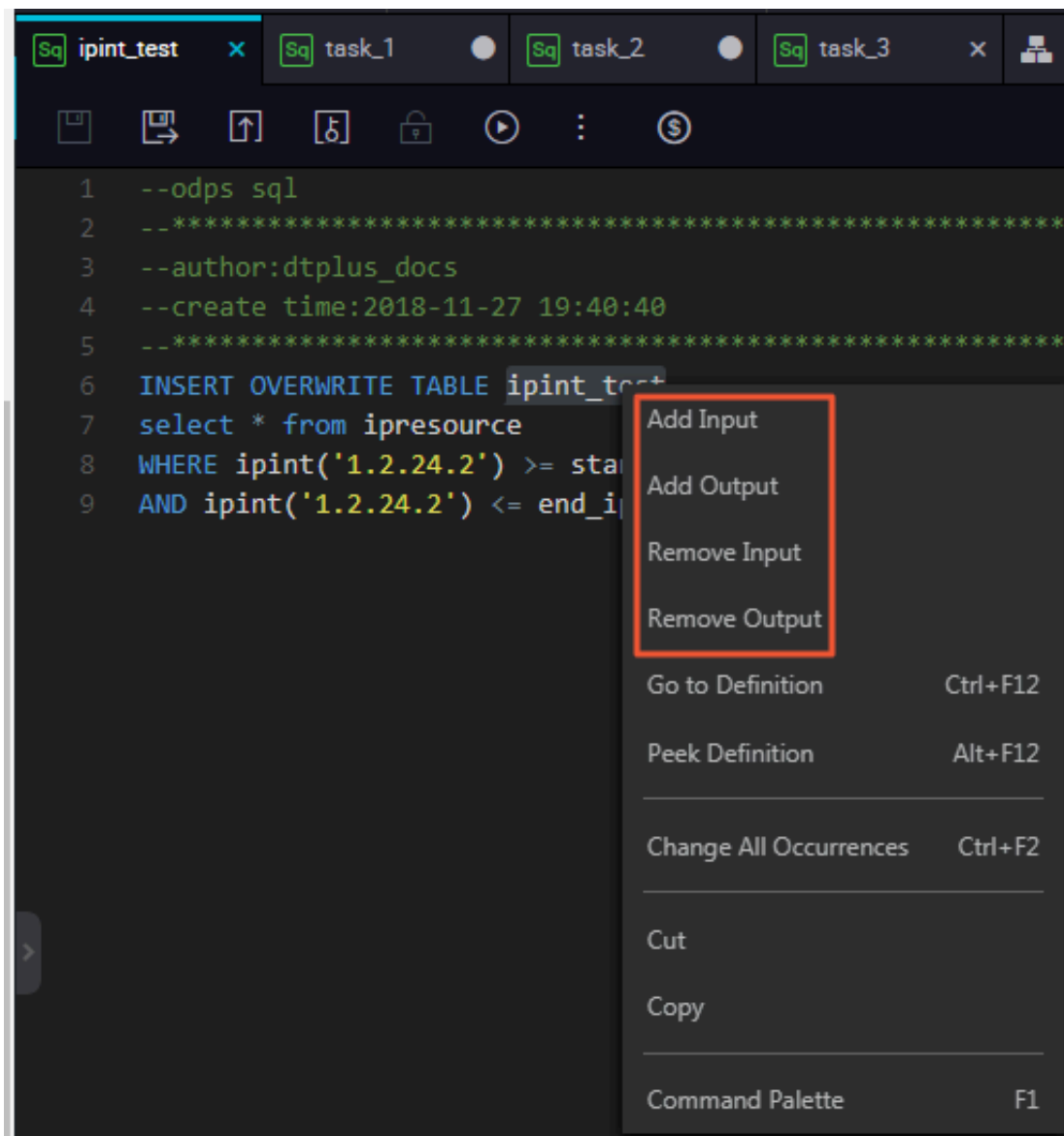
Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC_500132336_out	-	-	-	-	Added by Default	
MaxCompute_DOC.tb.4	MaxCompute_DOC.tb.4	-	-	-	Auto Parse	



Note:

- To increase the flexibility of task, we recommended that a task contain only one output point, so that you can flexibly assemble SQL business processes for decoupling purpose.
- If a table name in an SQL statement is both an output table and a referenced table (a dependent table), it will only be parsed as an output table.
- If a table name in an SQL statement is referenced or output many times, only one scheduling dependency is parsed.
- If there is a temporary table in the SQL code (for example, a table beginning with "t_" is specified as a temporary table in the [Project configuration](#)), the table will not be resolved to a scheduling dependency.

Under the premise of automatic parsing, you can avoid/increase the characters in some SQL statements to be automatically parsed into output name/input name by manually setting add/delete and input/output.



Selecting the table name and right-clicking, you can add or delete the output and input of all the table names that appear in the SQL statement. After the operation, the characters added to be input will be parsed as the output name of parent node, and the characters added to be output will be parsed as the output of the current node, otherwise the deletion of the operation will not be resolved.

**Note:**

In addition to right-clicking the characters in the SQL statement, you can also modify the dependencies by adding comments. The specific code is as follows.

```
--@extra_input=table name --Add an input
--@extra_output=table name --Add an output
--@exclude_input=table name --Delete an input
--@exclude_output=table name --Delete an output
```

Customize Adding Dependencies

When the dependencies between nodes cannot be accurately resolved through the SQL blood relationship, you can choose "no" in the following figure to self-configure dependencies.

Dependencies ⓘ

Auto Parse: ☐ Yes ☒ No

Parse I/O

Upstream Node

Enter an output name or output table name

+

Use The Workspace Root Node

Automatic recommended

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
mlsCompute_SVC_root	-	maxcompute_doc_root	756880632799	@plata.plate	Added Manually	<div>🗑</div>

Output

Enter an output name

+

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
mlsCompute_SVC_SINK_WGSSE_out	-	-	-	-	Added by Default	<div>🗑</div>

When auto-parsing column is set to "No", you can click **Automatic recommended** to enable the auto-recommended upstream dependency function. The system will recommend all other SQL node tasks that output the current node input table for you based on the SQL blood relationship of the project. You can select one or more tasks in the recommended list on demand and configure as the current node's upstream dependency tasks.



Note:

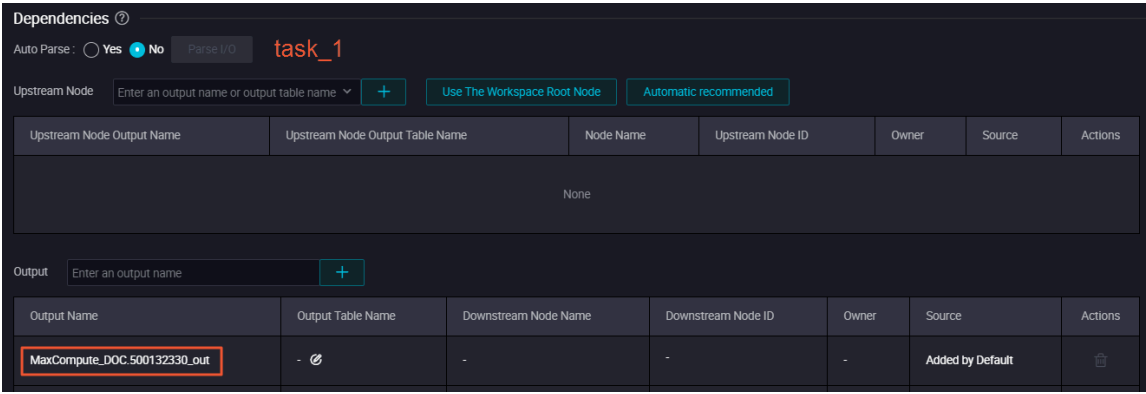
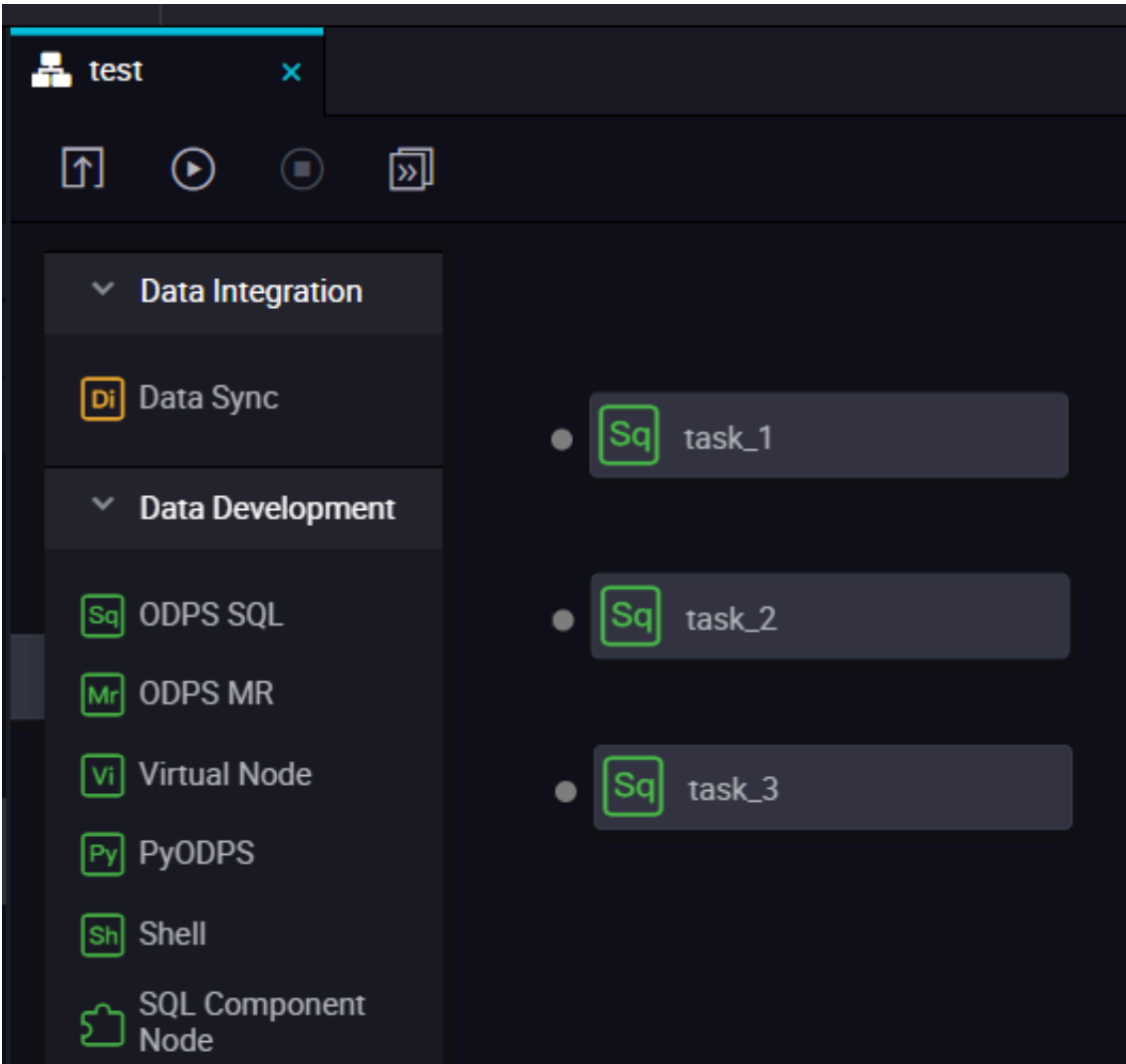
The recommended nodes need to be submitted to the scheduling system the day before, and can be recognized by the automatic recommendation function after the data output on the second day.

Common scenarios:

- The current task's input table is not equal to the upstream task's output table.
- The current task's output table is not equal to the downstream task's input table.

In custom mode, you can configure dependencies in two ways.

- Manually add dependent upstream nodes
 1. Create three new nodes and the system will configure one output name for each of them by default.



Dependencies ⓘ

Auto Parse: ☒ Yes ☐ No Parse I/O **task_2**

Upstream Node: + Use The Workspace Root Node

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
None						

Output: +

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132335_out	-	-	-	-	Added by Default	

Dependencies ⓘ

Auto Parse: ☒ Yes ☐ No Parse I/O **task_3**

Upstream Node: + Use The Workspace Root Node

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
None						

Output: +

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132336_out	-	-	-	-	Added by Default	

2. Configure the upstream node task_1 to depend on the root node of the project, and click Save.

Dependencies ⓘ

Auto Parse: ☐ Yes ☒ No Parse I/O **task_1**

Upstream Node: + Use The Workspace Root Node Automatic recommended

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC.root	-	maxcompute_doc_root	700000822799	dtplus_docs	Added Manually	

Output: +

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132330_out	-	-	-	-	Added by Default	

3. Configure task_2 to depend on the output name of task_1, and click Save.

Dependencies ⓘ

Auto Parse: ☒ Yes ☐ No Parse I/O **task_2**

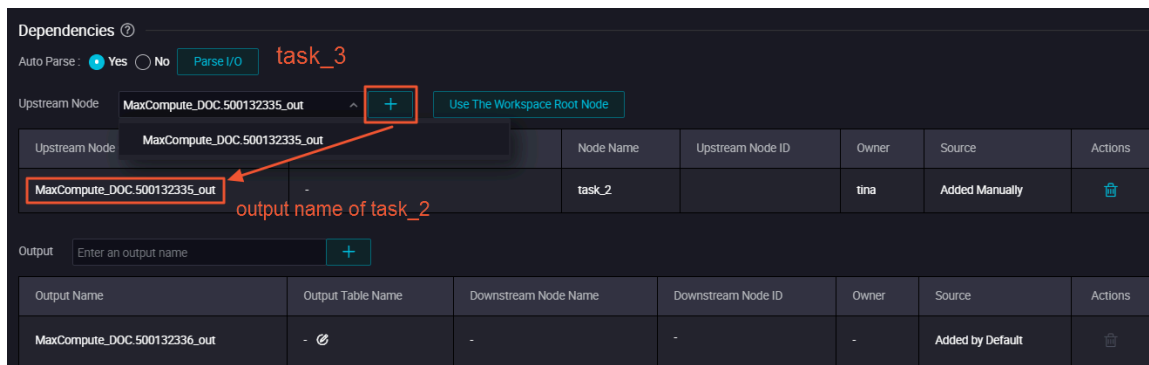
Upstream Node: + Use The Workspace Root Node

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132330_out	-	task_1		tina	Added Manually	

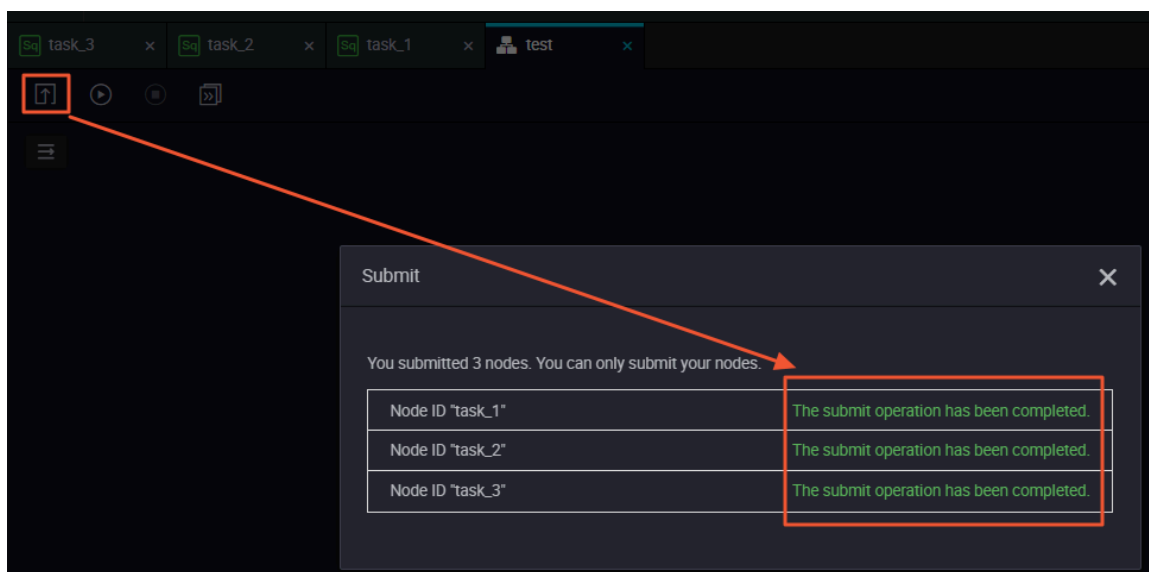
Output: +

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132335_out	-	-	-	-	Added by Default	

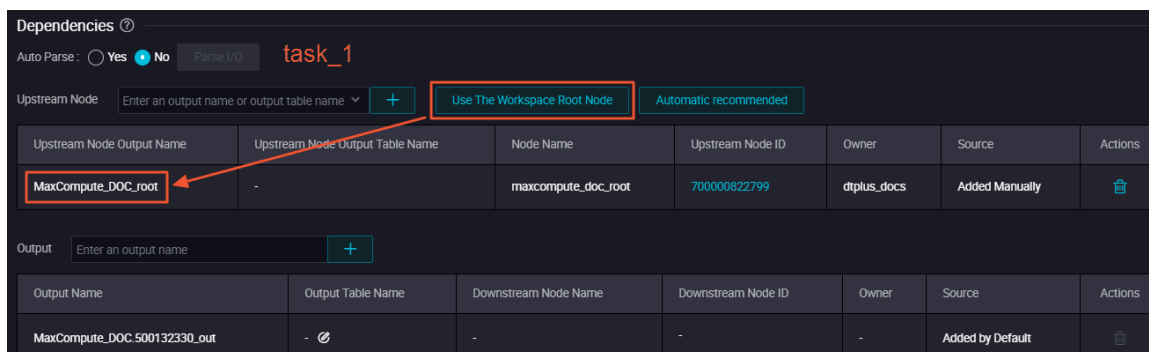
4. Configure task_3 to depend on the output name of task_2, click Save.



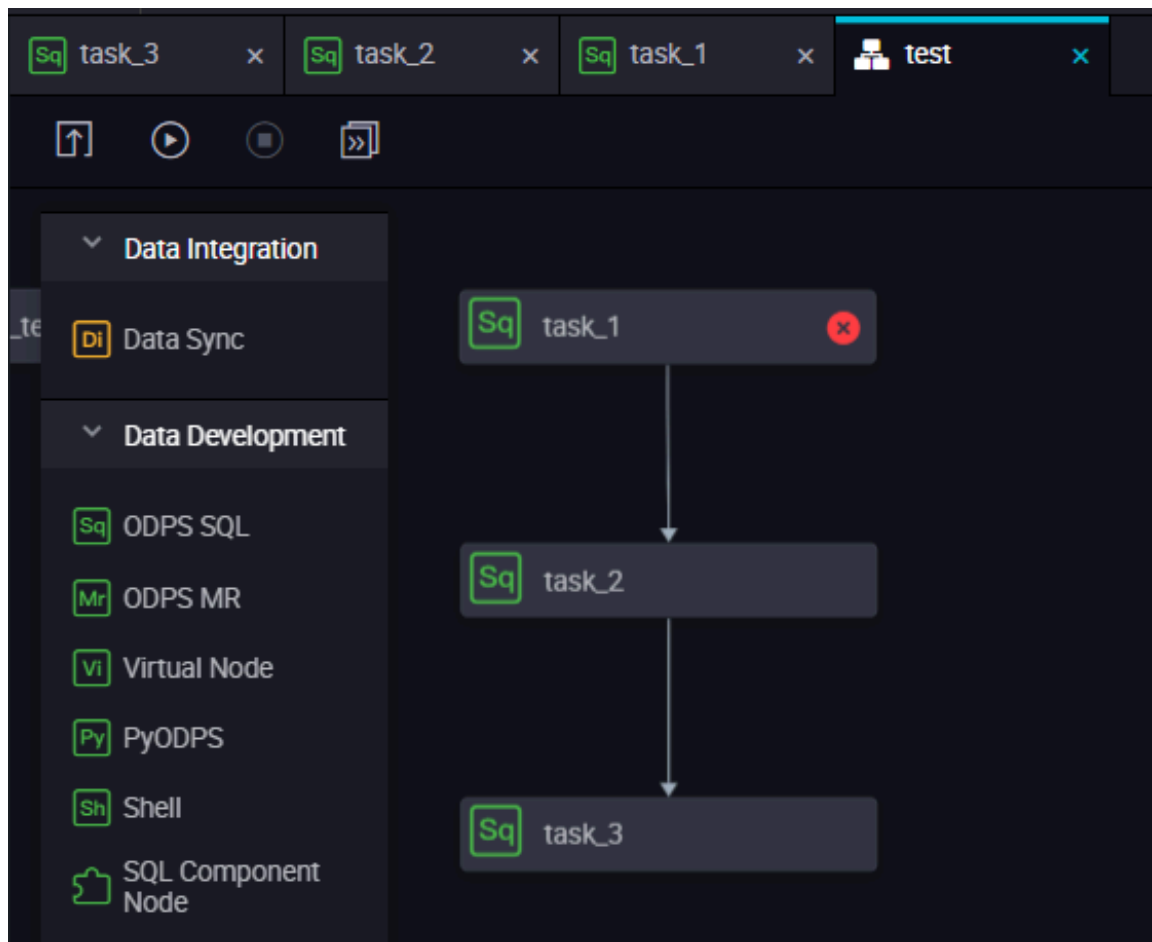
5. After the configuration is complete, click Submit to determine whether the dependency relationship is correct. If the submission is successful, the dependency configuration is correct.



- Construct dependencies by dragging and dropping
1. Create three nodes: task_1, task_2, task_3, and configure the upstream task_1 to depend on the root node, then click Save.



2. Connect the three tasks by dragging and pulling.



3. Check the dependency configuration of task_2 and task_3, you can see the dependent parent node output name that has been automatically generated.

Dependencies ②

Auto Parse: ☐ Yes ☒ No Parse I/O task_2

Upstream Node

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132330_out	-	task_1	700001940385	tina	Added Manually	<input type="button" value="🗑"/>

The system adds the output name of task_1 automatically.

Output

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132335_out	-	task_3	700001940385	tina	Added by Default	<input type="button" value="🗑"/>

Dependencies ②

Auto Parse: ☐ Yes ☒ No Parse I/O task_3

Upstream Node

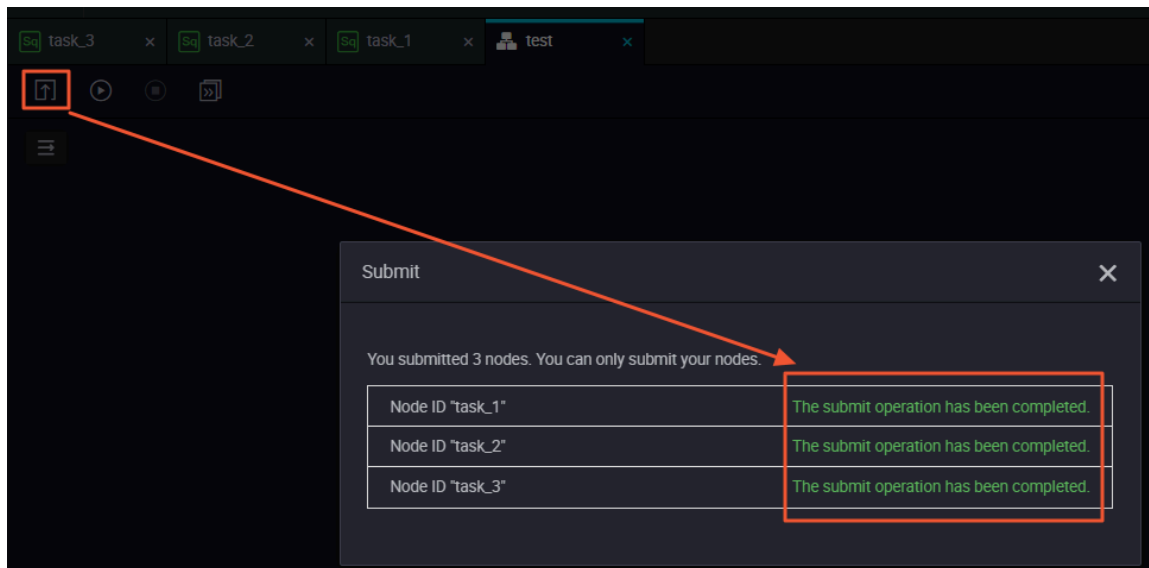
Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132335_out	-	task_2	700001940384	tina	Added Manually	<input type="button" value="🗑"/>

The system adds the output name of task_2 automatically.

Output

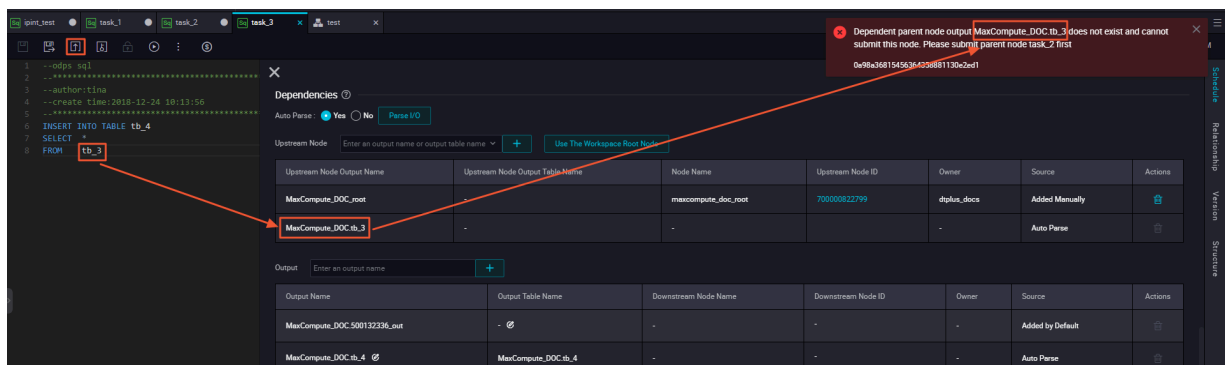
Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
MaxCompute_DOC.500132336_out	-	-	-	-	Added by Default	<input type="button" value="🗑"/>

4. After the configuration is complete, click Submit to determine whether the dependency relationship is correct. If the submission is successful, the dependency configuration is correct.



FAQ

Q: After automatic parsing, the submission fails. Error: Dependent parent node output MaxCompute_DOC.tb_3 does not exist and cannot submit this node. Please submit parent node task_2 first.



A: This can be caused by the following two reasons for this.

- The upstream node is not submitted, and you can try again after submission.
- The upstream node has been submitted, but the output name of the upstream node is not MaxCompute_DOC.tb_3.



Note:

Usually, the parent node output name and the current node output name that automatically parsed are obtained according to the table name after INSERT/CREATE/FROM. Make

sure that the configuration is consistent with the way described in the section "Auto-parsing dependencies".

Q: In the output of the current node, the downstream node name and downstream node ID are all empty and cannot be entered.

A: If there is no sub-node for downstream of the current node, there is no content. After the sub-node is configured for downstream of the current node, the content is automatically parsed.

Q: What is the node's output name used for?

A: The node's "output name" is used to establish dependencies between nodes. For example, If the output name of node A is "ABC" and node B takes "ABC" as its input, the upstream and downstream relationship is established between nodes A and B.

Q: Can a node have multiple "output names"?

A: Yes. If a downstream node references an output name from the current node (as the "parent node output name" of the downstream node), it establishes a dependency with the current node.

Q: Can multiple nodes have the same "output name"?

A: No. The "output name" of each node must be unique in Alibaba Cloud account system. If multiple nodes output data to the same MaxCompute table, we recommend that you use "table name_partition ID" as the output of these nodes.

Q: How do I not parse to an middle table when using auto-parsing dependencies?

A: Select the middle table name in the SQL code and right-click the **Remove Input** or **Remove Output**, and then perform the automatic parsing of the input and output again.

Q: How do I configure dependencise of the most upstream task?

A: In general, you can choose to depend on the root node of this project.

Q: Why did I search for the output name of the node B that does not exist when searching for the upstream node output name on the node A?

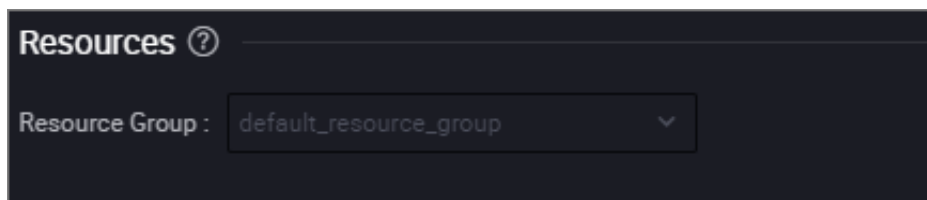
A: Because the search function is based on the submitted node information. If the output name of node B is deleted after the successful submission of node B and not submitted to the scheduling system, then the deleted output name of node B can still be found on node A.

Q: If I have three tasks A, B, and C, how do I implement the task flow of A->B->C once an hour (After A is completed, execute B, after B is completed, execute C)?

A: The dependency of A, B, and C is set to the output of A as the input of B, the output of B is the input of C, also the scheduling periods of A, B, and C are set to hours.

3.6.5 Resource type

The resource attribute configuration page is shown in the following figure:



Resource Group: Machine resources bound to task scheduling. The system contains a resource group by default. Other resource groups are added only when custom machines are required in special cases.

3.6.6 Node Context

Node Context is used to transfer parameter between upstream and downstream nodes. The basic way to use Node Context function is that first define output parameters and their values on the upstream node, then defined input parameter on the downstream node (the value references the output parameters of the upstream node). You can use this parameter in the downstream node to get the values which is transferred from the upstream node.

Node context parameter can be configured at **Schedule > Node Context** in a specific node, as shown in the following figure.

Output: +

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
bigdata_doc.test	-	-	-	-	Added Manually	
bigdata_DOC.30135300_output	-	-	-	-	Added by Default	

Node Context ?

The Node Input Parameters Add

No.	Parameter Name	Value Of The Source	Description	Parent Node ID	Source	Actions
None						

The Node Output Parameters Add

No.	Parameter Name	Type	Value	Description	Source	Actions
None						

Output Parameters

The **Node Output Parameters** can be defined on **Node Context**. There are two types of Output Parameter value which are **Constant** and **Variable**. **Constant** is a fixed string. **Variable** are global variables supported by the system. The output parameter can be reused at downstream node as input parameter value, after upstream node submitted with output parameter.



Note:

It is not supported that assigning value to the defined Output parameter on current node (like PyODPS node) by internal code writing.

Node Context ?

The Node Input Parameters Add

No.	Parameter Name	Value Of The Source	Description	Parent Node ID	Source	Actions
None						

The Node Output Parameters Add

No.	Parameter Name	Type	Value	Description	Source	Actions
1	output_const	Constant	abc	example of constant v	Added Manually	Save Cancel

The fields are described as follows.

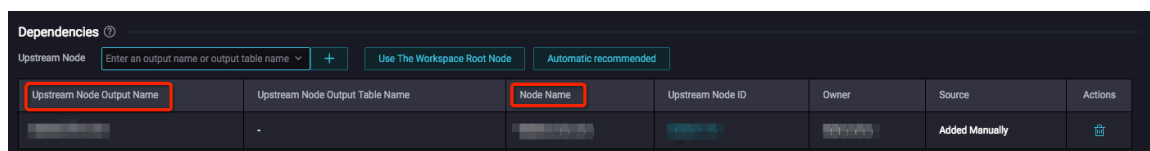
Field	Description	Note
No.	The value of No. is generated by system and it is automatic increased.	N/A
Parameter name	Defined output parameter name	N/A
Type	Parameter Type	There are two types of Output Parameter value which are Constant and Variable .
Value	Value Of the Source	<ol style="list-style-type: none"> String can be input directly when Type is selected as Constant. System variables, Schedule built-in parameters, Customized parameters \$ {...} and \$ [...] are supported when Type is selected as Variable.
Description	A brief description of the parameters	N/A
Action	Edit and Delete can be selected	Edit and Delete are not supported when there is a downstream node dependence. Before adding references to upstream nodes, please make sure that the upstream output is defined correctly.

Input Parameters

The Node Input Parameters are used to define a reference to the output of upstream node which it is dependent on , and it can be used inside the node similar as other parameters.

- Definition of **The Node Input Parameters**

- Add a dependent upstream node on **Dependencies**.



- Add an input parameter definition with value , which references the upstream node, in **Node Context > The Node Input Parameters**.

The fields are described as follows.

Field	Description	Note
No.	The value of No. is generated by system and it is automatic increased.	N/A
Parameter name	Defined input parameter names	N/A
Value Of the Source	Parameter's value source, reference to upstream node's Value	The specific parameter value when upstream node is running
Description	A brief description of the parameters	Automatically parsed from the upstream node.
Parent Node ID.	Parent Node ID	Automatically parsed from the upstream node.
Action	Edit and Delete can be selected	N/A

- Use of input parameters

The format of reuse defined input parameter is similar as other system. The format is `${input parameter name}`. For example, a reference in a shell node is shown in the following figure.

```
echo 'input_from_up_const:' ${input_from_up_const}
echo 'input_from_up_var:' ${input_from_up_var}
```

Global variables supported by the system

- System variable

```
$ {projectid}: Project ID
$ {project name}: MaxCompute project name
$ {nodeid}: Node ID
$ {gmtdate}: 00:00:00 at the instance date,format: 'yyyy-mm-dd 00:00:00'.
$ {taskid}: Instance Task ID
$ {seq}: Task instance sequence number,represents the instance's sequence number in the same node on current day.
$ {cyctime}: instance time
$ {status}: Status of instance-Success, Failure
```

```

$ {bizdate}: Business Date
$ {finishtime}: Instance End Time
$ {taskType}: Instance Status—NORMAL, MANUAL, PAUSE, SKIP, UNCHOOSE,
SKIP_CYCLE
$ {nodeName}: Node name

```

- See additional parameter settings [Parameter configuration](#).

Examples

Node test22 is the upstream node of node test223. Please configure **Node Context > The Node Output Parameters** on Node test22. In this example, the parameter name is `date1` and the value is `${yyyymmdd}`, click **Run** as shown in the following figure.

The screenshot displays the configuration interface for a node in DataWorks. The 'Dependencies' section shows the 'Upstream Node' as 'Enter an output name or output table name'. The 'Output' section shows the 'Output Name' as 'date1'. The 'Node Context' section shows 'The Node Input Parameters' and 'The Node Output Parameters'. The 'The Node Output Parameters' table has one row with 'Parameter Name' 'date1', 'Type' 'Variable', 'Value' '\${yyyymmdd}', 'Description' 'date', and 'Source' 'Added Manually'.

No.	Parameter Name	Type	Value	Description	Source	Actions
1	date1	Variable	\${yyyymmdd}	date	Added Manually	Save Cancel

After node test22 submitted successfully, configure the downstream node test223.



Note:

Please make sure **Dependencies > Upstream Node Output Name** in test223 is same as **Dependencies > Output Name** in test22.

Enter the parameter name of test22 `date1` in **Node Context > The Node Input Parameter > Parameter Name**, then there will be options available in **Value Of The Source** dropdown . Choose specific source then click **Save**.

Upstream Node Output Name	Upstream Node Output Table Name	Node Name	Upstream Node ID	Owner	Source	Actions
	-	Test22	700001940205	alidocs	Added Manually	

Output +

Output Name	Output Table Name	Downstream Node Name	Downstream Node ID	Owner	Source	Actions
	-	-	-	-	Added by Default	

Node Context

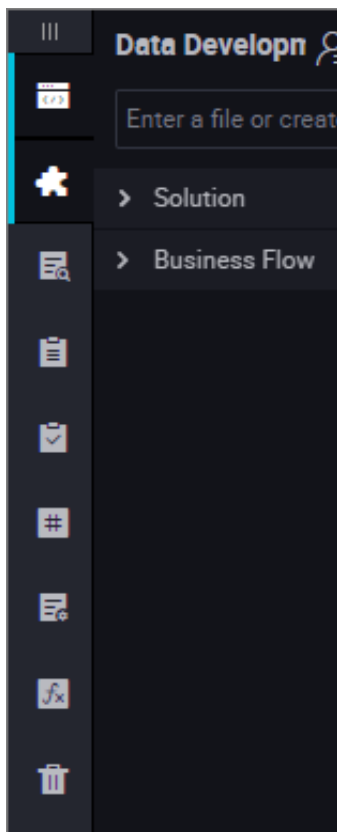
The Node Input Parameters +

No.	Parameter Name	Value Of The Source	Description	Parent Node ID	Source	Actions
1	date1	<input type="text" value="Please select"/>			Added Manually	Save Cancel

3.7 Configuration management

3.7.1 Overview of configuration management

Configuration management is the configuration of the DataStudio interface, including code, folder, theme, add and delete modules, and so on. You can enter the configuration management page by clicking the pinon in the lower left corner of the data development.

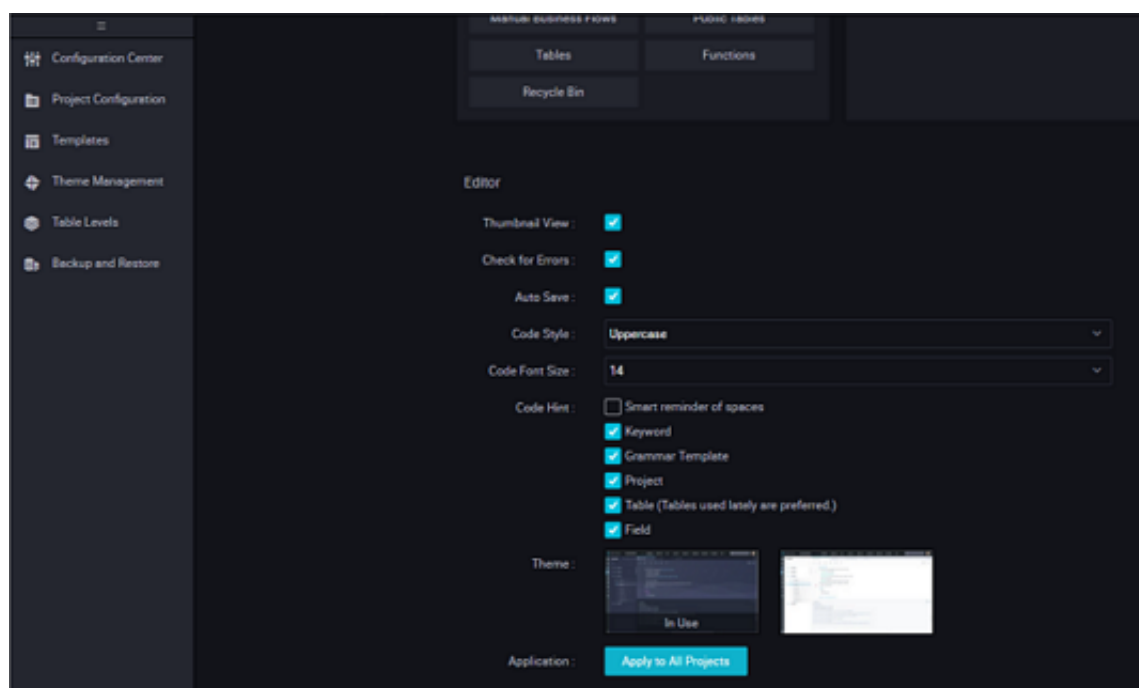


Configuration management is divided into five modules. For more information, see the following documents.

- [Configuration center](#)
- [Project configuration](#)
- [Templates](#)
- [Theme management](#)
- [Table Levels](#)

3.7.2 Configuration center

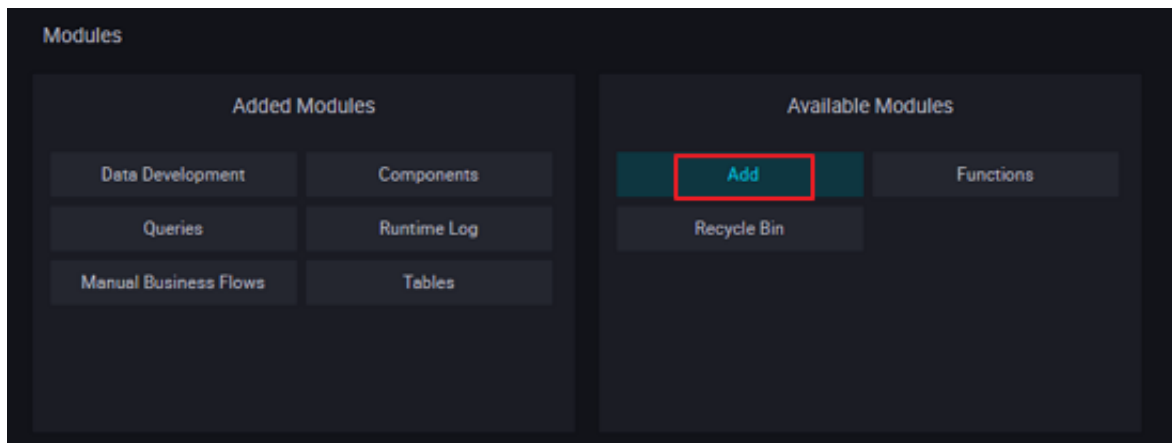
The configuration center is the setting for common features, including module management and editor management.



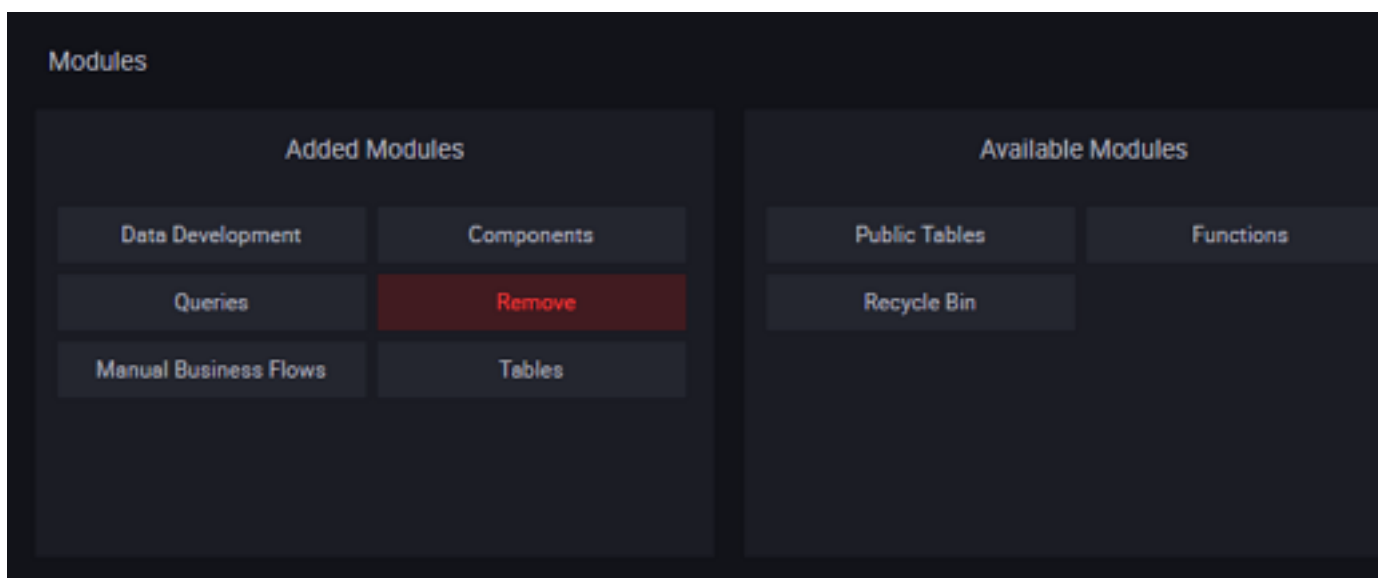
Module management

Module management is the operation of adding and deleting modules to the left-side column function module of the DataStudio interface, you can click to filter the functional modules that need to be displayed in the left side, you can also sort the functions of a module by dragging and dropping.

When the mouse is over the module you want to add, the module turns blue and displays **Add**.



When the mouse is over the module that needs to be removed, the module turns red and displays **Remove**.

**Note:**

Template management filtering takes effect immediately and takes effect for the current project, if you want to take effect for all projects, click **the above settings to apply to all projects**.

Editor management

The editor is the setting for code and keywords, the setting takes effect in real-time without refreshing the interface.

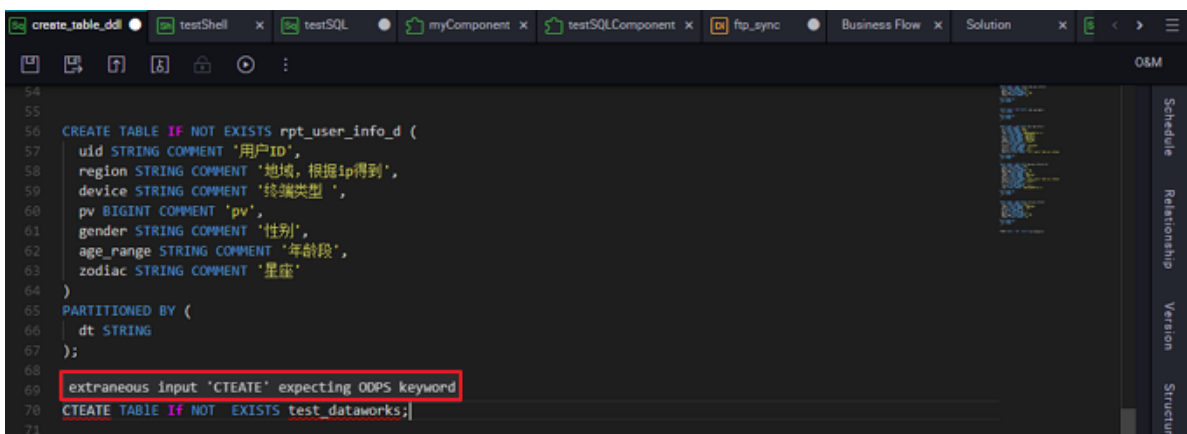
- Thumbnail View

The display of the current interface code is displayed on the right side of the code, the shaded area in the figure is the area currently being displayed, and when the code is longer, you can move the mouse up and down to switch the displayed code area.



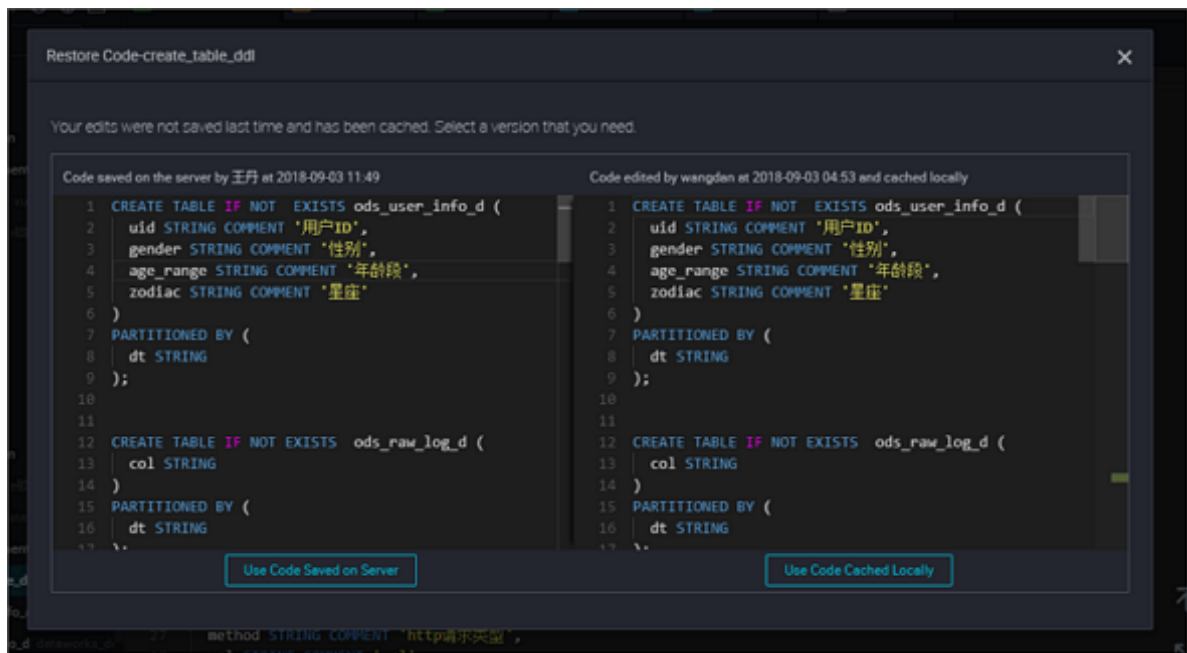
- Check for errors

Check the error statement in the current code. When the mouse is placed in the red error code area, an error-specific field condition is displayed.



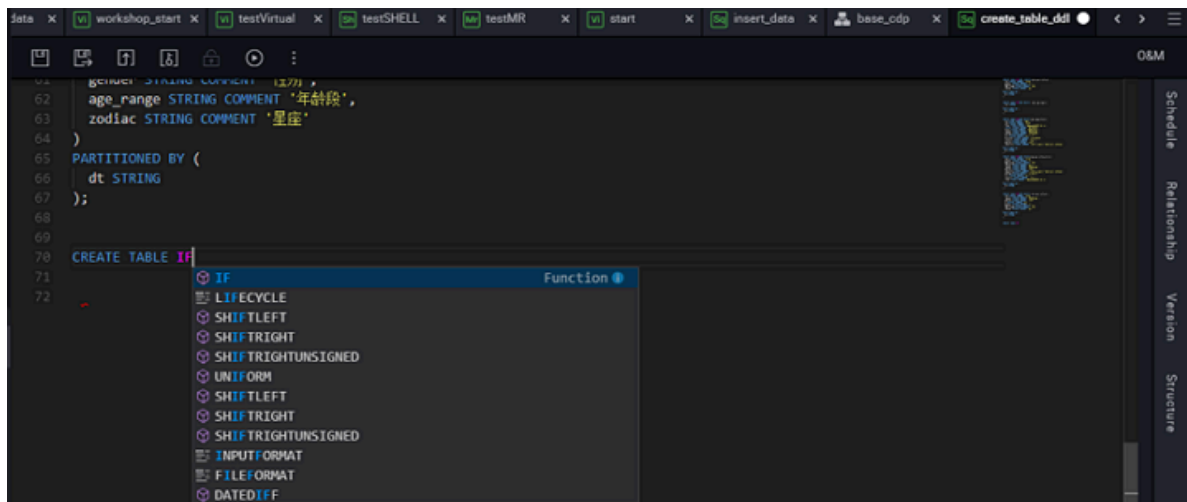
- Auto save

Automatically cache the currently edited code to avoid the page crashing and causing the code to not be saved during the editing process. You can choose **Use server-saved code** in the left-side or **Use locally cached code** in the right-side.



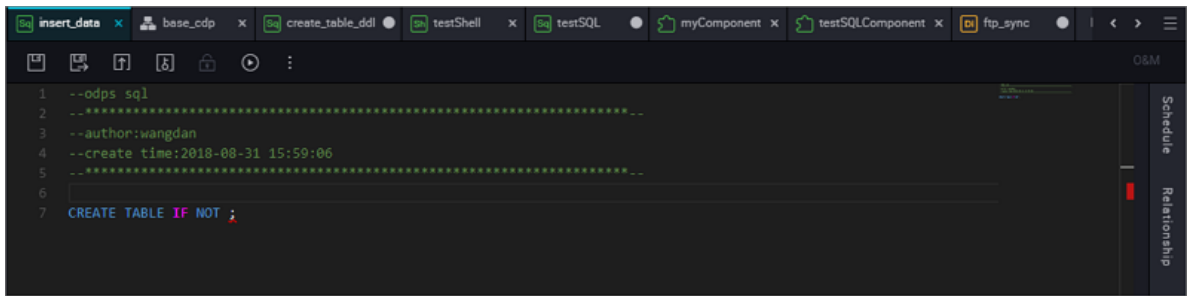
- Code style

The code style can be set to uppercase or lowercase, select your favorite style. Enter the keyword and press Enter to enter the required keywords through Lenovo shortcut.



- Code font size

The code font size supports a minimum of 12 and a maximum of 18 fonts, change the setting according to your code writing habits and quantity.



- Code Hint

Code prompts are used during code entry, and the display of intelligent prompts is divided into the following sections.

- Space Smart Tip: Add a space after selecting Lenovo's keywords, tables, and fields.
- keywords: the prompt code supports the keywords entered.
- Syntax templates: supported syntax templates.
- Project: enter the project name of the Lenovo.
- Table: The table that Lenovo needs to enter.
- Field: Smart prompt for fields in this table.

- Theme

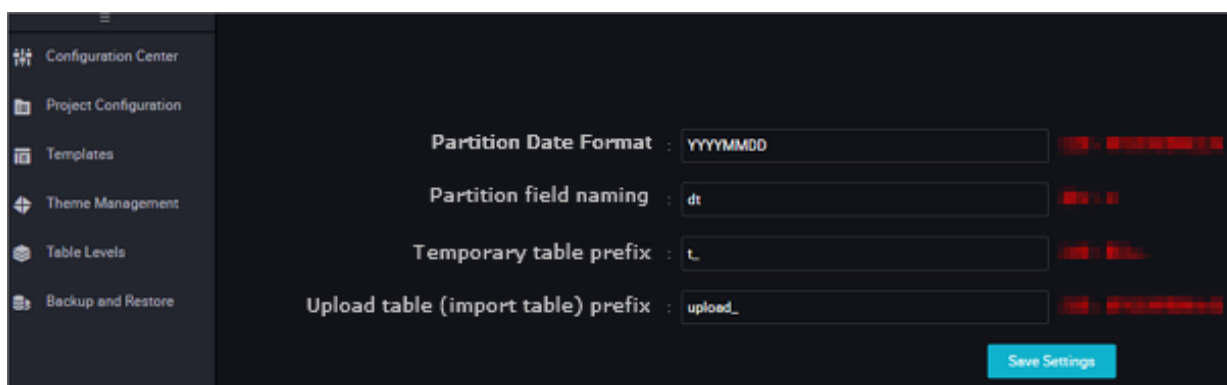
The theme style is the setting of the DataStudio interface style, currently supports both black and white.

- Application

Apply the above template management and editor management settings to all currently existing projects.

3.7.3 Project configuration

Project configuration includes partition date format, partition field naming, temporary table prefix, and upload table (import table) prefix four configuration items.

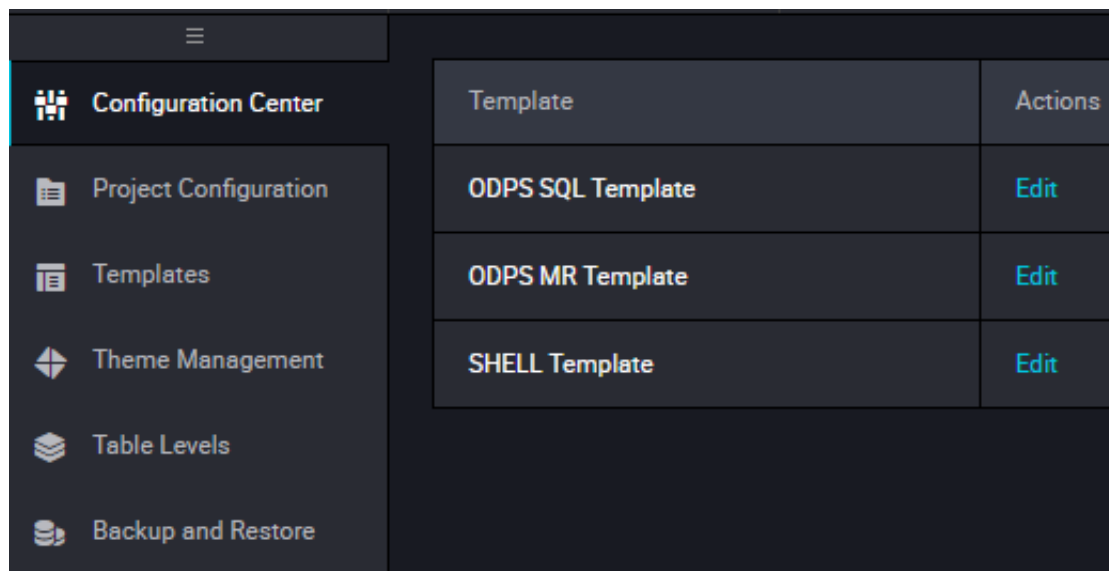


- Partition Date Format: the default parameter, the display format of the parameters in the code, you can also modify the format of the parameters according to your own requirements.
- Partition field naming: The partition default field name.
- Temporary table prefix: fields that begin with "t_" are identified as temporary tables by default.
- Upload table (import table) prefix: The name prefix of the table when the DataStudio interface uploads the table.

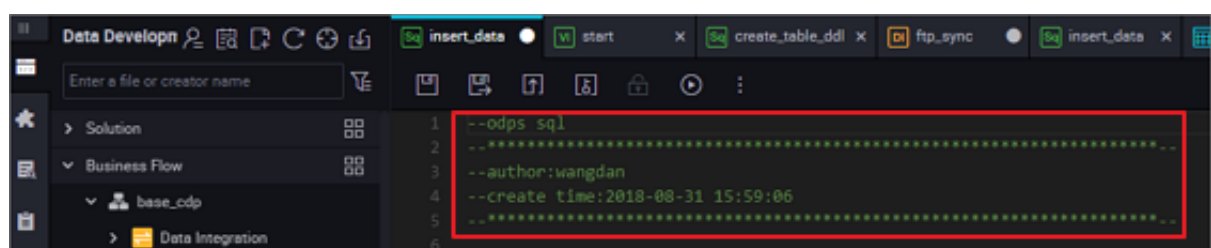
3.7.4 Templates

Template management is the content that is displayed at the front of the code by default after the node is created, the project administrator can modify the display style of the template as required.

Currently, the title is set for the ODPS SQL template, the ODPS MR template, the ODPS PL template, the PERL template, and the SHELL template.



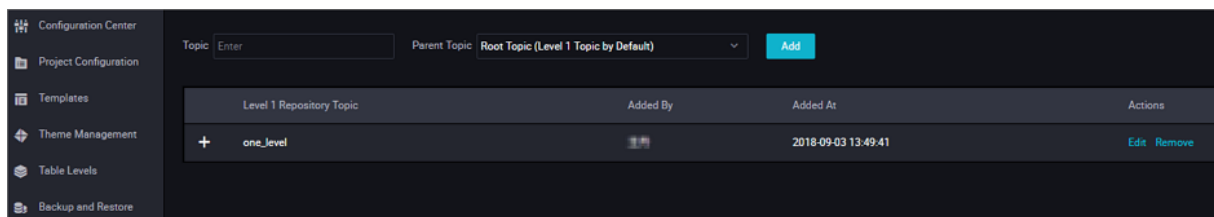
Take the SQL node as an example, the template display style:



3.7.5 Theme management

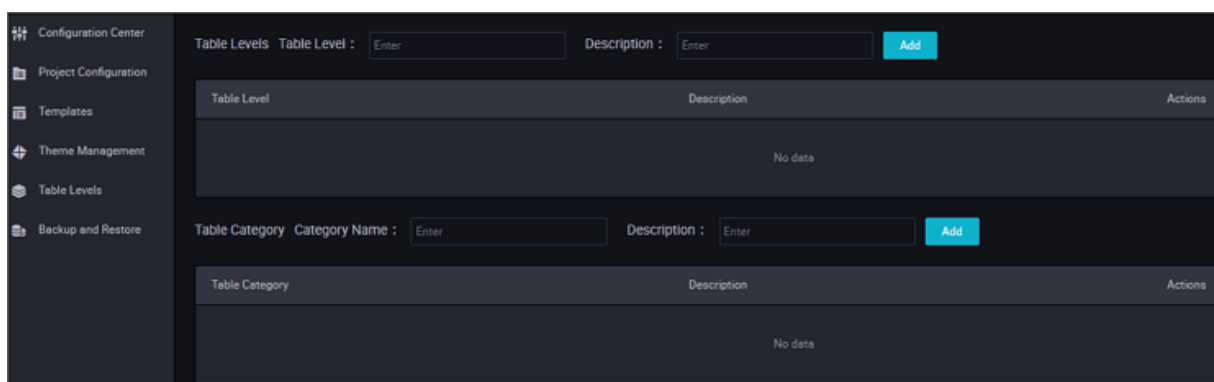
There are many tables in table management, the table is stored under the second-level sub-Folder according to the selected topics. These folders are summarized in the table, which is the theme.

The administrator can add multiple themes based on project requirements, classify and organize the tables according to their purpose and name.



3.7.6 Table Levels

Table Levels is the physical level design of a table. According to the importance of the table to the project, the table is divided to avoid the problem that when a problem occurs in a table, the impact on the project cannot be accurately located, which leads to the normal operation of the online operation.



There is no default hierarchy for the project, and the project owner or administrator needs to be added manually according to the purpose and needs of the project.

3.8 Publish management

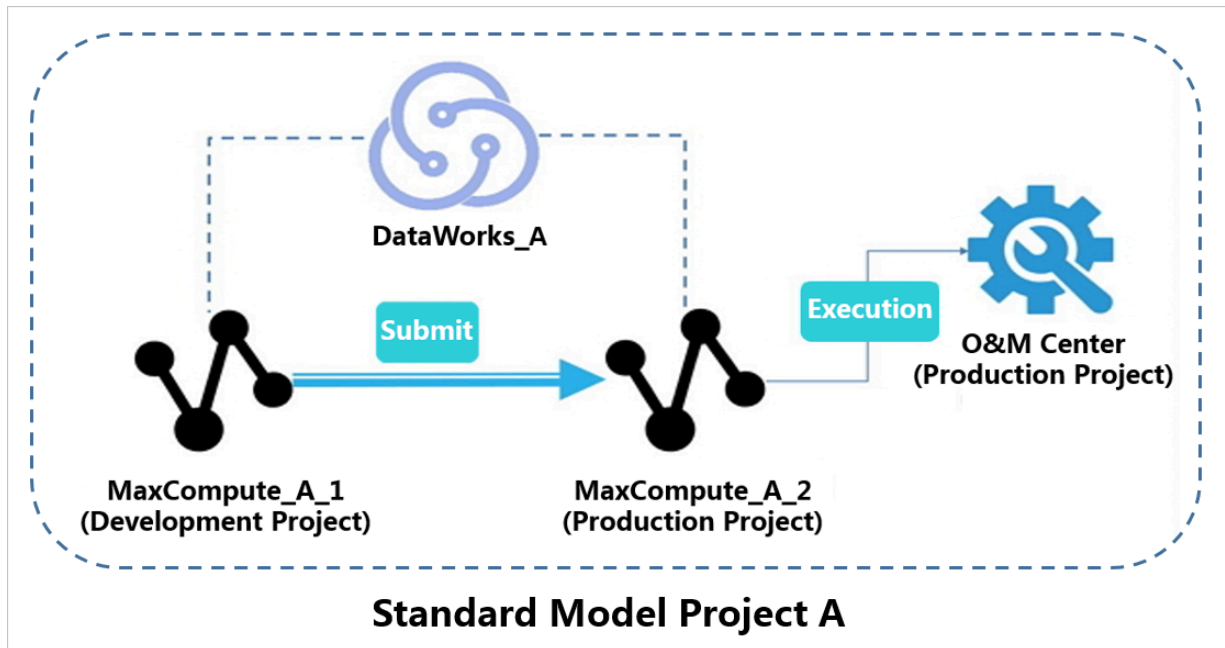
3.8.1 Publish a task

In a complete data development process, developers develop code, debug processes, configure dependencies, configure scheduled tasks, and then submit the tasks to the production environment for execution.

The *standard mode* of DataWorks can process data seamlessly from the development to production stages in a project. We recommend that you use this mode for data development, production, and publishing.

Publish a task in the standard mode

Each DataWorks project in standard mode corresponds to two MaxCompute projects that are associated with one another, one for the development environment and the other for the testing environment. You can directly submit and release a project to the production environment from the development environment.



The procedure is as follows:

1. Click **Submit** after the code and task are debugged and configured. The system will automatically check the dependencies between code objects.
2. When the submission is complete, click **Publish**.
3. Navigate to the **For Publish** page and select the target objects. Click **Add For Publish** and the **Publish List** page appears.

On the Publish List page that appears, you can filter the objects by publisher, node type, change type, publish date, and task name or ID. If you click **Publish Selected Items**, the objects are released to the production environment for scheduling immediately.

4. Click **Open For Publish > Publish All** to release the objects to the production environment.



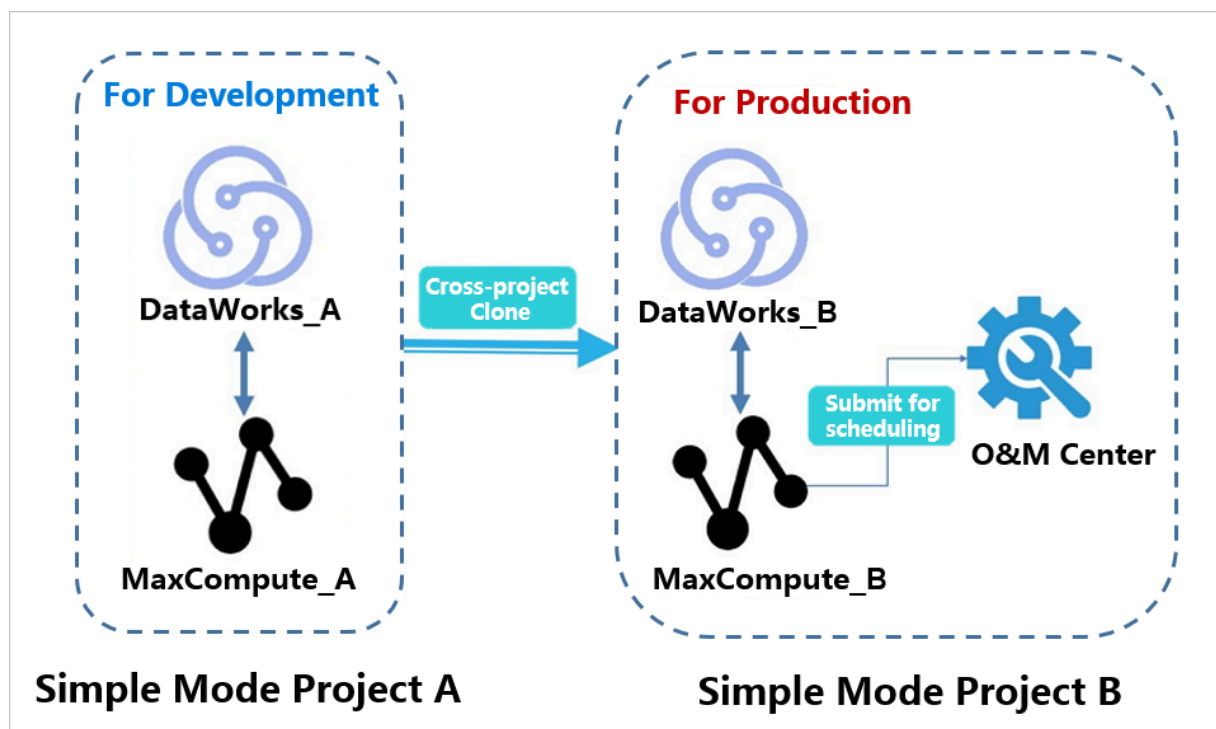
Note:

The standard mode strictly prohibits direct operation on table data in the production environment. You can obtain a stable, secure, and reliable production environment. We strongly recommend that you use this mode to publish and schedule a task.

Cross-project clone in the simple mode

A simple mode project (for development) cannot publish tasks. To develop data and isolate the production environment, you must clone and then submit a task to a production project. This creates a simple mode project (for production).

As shown in the following figure, Simple Mode Project A is created for development and Simple Mode Project B for production. You can use **cross-project cloning** to clone a task of Project A to Project B, and submit the task to the scheduling engine for scheduling.



Note:

- Permission requirements: a RAM user that is not the project owner requires administrative permissions, such as creating a clone package and publishing a clone task, to run the operation and complete the process.
- Supported subject types: Only tasks of a simple mode project can be cloned to other projects. Standard mode projects do not support this operation.
- Prerequisites: source project A (a simple mode project) and target project B (a standard mode project).

1. Submit a task

Select and submit the source task after it is edited.

2. Click **Cross-project Cloning**.

3. Select the source task name in the list of submitted tasks and the target project name, click **Add For Clone**.

4. Run a clone operation

Click **For Cloning** . Check whether the information of the source task is correct and click **Clone All**. Click **Confirm** to run the operation and complete the process.

5. View a cloned task

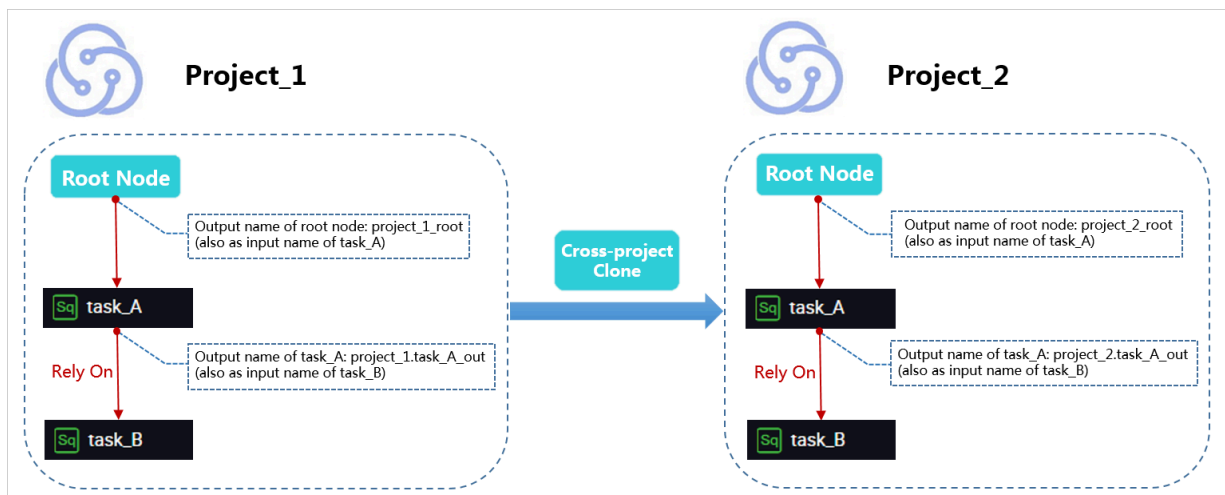
You can view the successful tasks on the **Clone List** of source project A. View target project B to check whether the source task is cloned to the business flow.

3.8.2 Cross-project cloning

After you successfully clone a task using the **cross-project cloning** feature, the system will automatically alter the output name of each task to replicate or maintain the dependencies between two nodes. This allows the system to distinguish different projects under the same Alibaba Cloud account.

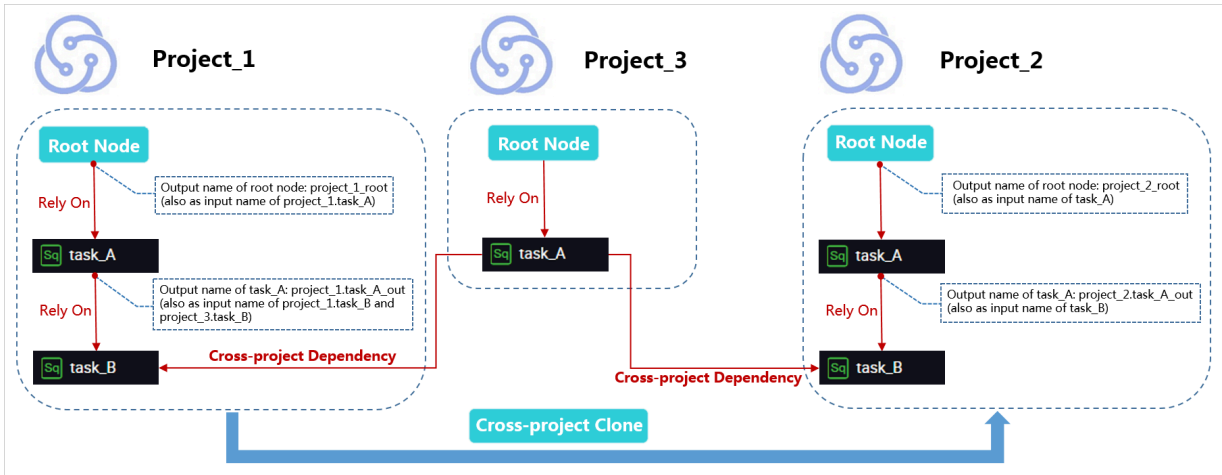
A complete business flow cloning process

The output name project_1.task_1_out of task_A in Project_1 will be renamed as project_2.task_out after it is cloned to Project_2.



Cross-project dependencies cloning

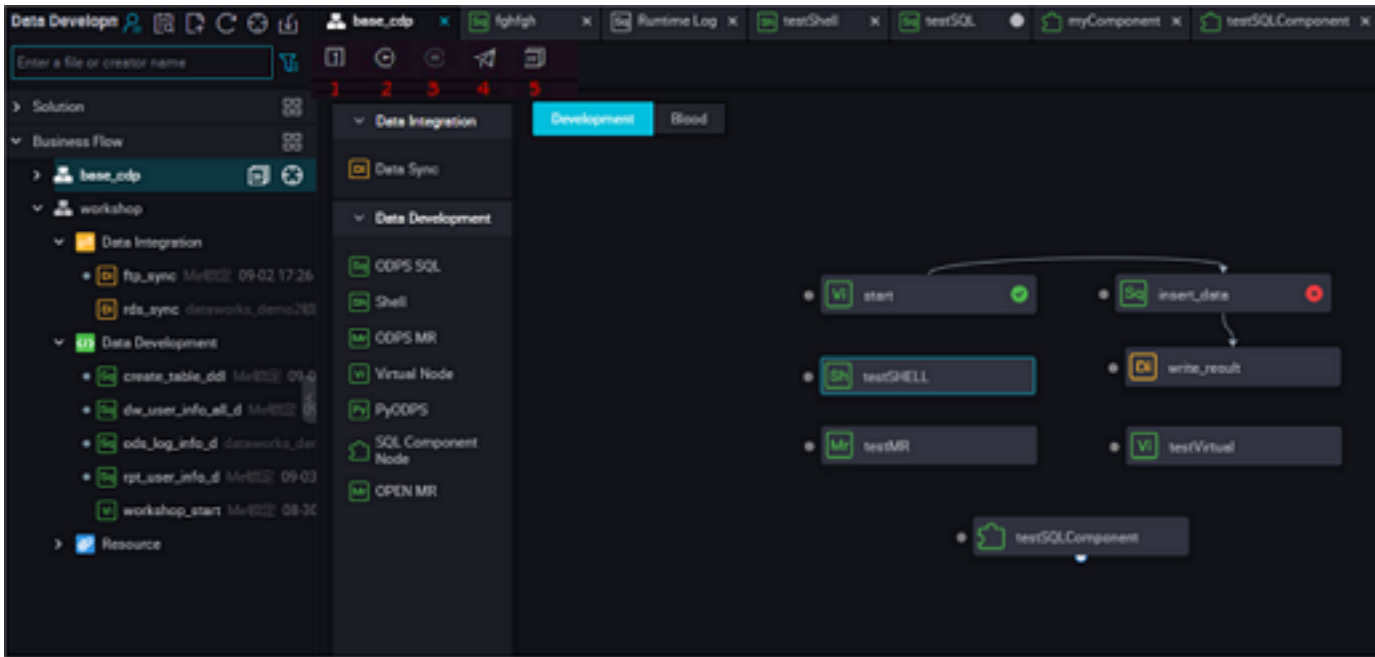
By default, task_B in Project_1 is dependent on task_A in Project_3. After you clone task_B in Project_1 to Project_2, the dependencies between task_B in Project_1 and task_A in Project_3 are also cloned, which means task_B in Project_2 is still dependent on task_A in Project_3.



3.9 Manual business flow

3.9.1 Manual Business Flow Introduction

In a Manual Business Flow, all created nodes must be manually triggered and cannot be executed by means of scheduling. Therefore, it is unnecessary to configure the parent node dependency and local node output for nodes in a manual business flow.



The functions of the manual business flow interface are described below:

No.	Function	Description
1	Submit	Click it to submit all nodes in the current manual business flow.

No.	Function	Description
2	Run	Click it to run all nodes in the current manual business flow. As dependency does not exist among manual tasks , these tasks will run concurrently.
3	Stop Run	Click it to stop a node that is running.
4	Publish	Click it to go to the task publishe interface, where you can publishe some or all of the nodes that have been submitted but not published to the production environment.
5	Go to O&M	Click it to go to the O&M center.
6	Reload	Click it to reload the current manual business flow interface.
7	Auto Layout	Click it to automatically sequence the nodes in the current manual business flow.
8	Zoom-in	Click it to zoom in the interface.
9	Zoom-out	Click it to zoom out the interface.
10	Query	Click it to query a node in the current manual business flow.
11	Full Screen	Click it to show the nodes in the current manual business flow in full screen mode.
12	Parameters	Click it to configure parameters. The priority of a flow parameter is higher than that of a node parameter. If a parameter key matches a parameter, the business flow parameter is configured preferentially.
13	Operation Records	Click it to view the operation history of all nodes in the current manual business flow.
14	Version	Click it to view the submission and publishe records of all nodes in the current manual business flow.

3.9.2 Resource

Resource is a concept unique in ODPS. Resources must be available if you want to use ODPS UDFs or ODPS MR.

- ODPS SQL UDF: After compiling a UDF, you must upload the compiled jar package to the ODPS. When running this UDF, ODPS automatically downloads the jar package, extracts the user code, and runs the UDF. The process of uploading the jar package is the process that a resource is created in ODPS. The jar package is a type of ODPS resource.

- ODPS MapReduce: After compiling a MapReduce program, you must upload the compiled jar package as a resource to ODPS. When running a MapReduce job, the MapReduce framework automatically downloads this jar resource and extracts the user code.

Similarly, you can upload text files, ODPS tables, and various compressed packages (such as .zip, .tgz, .tar.gz, .tar, and .jar) as different types of resources to ODPS. Then, you can read or use these resources when running UDFs or MapReduce.

ODPS provides APIs for reading and using resources. The following types of ODPS resources are available:

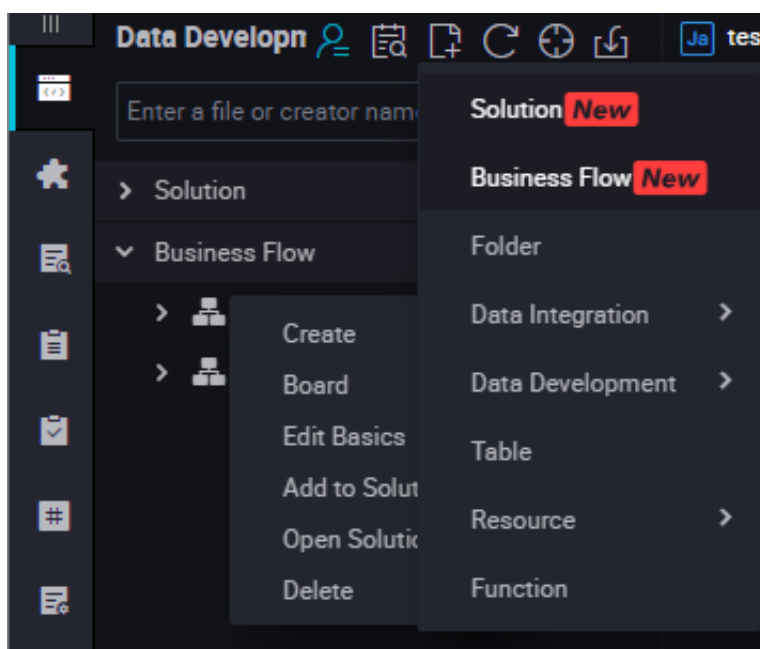
- File
- Archive: The compression type is identified by the extension in the resource name. The following compressed file types are supported: .zip, .tgz, .tar.gz, .tar, and .jar.
- Jar: compiled Java jar packages.

In DataWorks, the process of creating a resource is a process of adding a resource. Currently, DataWorks supports addition of three types of resources in a visual manner, including the jar, Python, file resources. The newly created entries are the same, the differences are as follows:

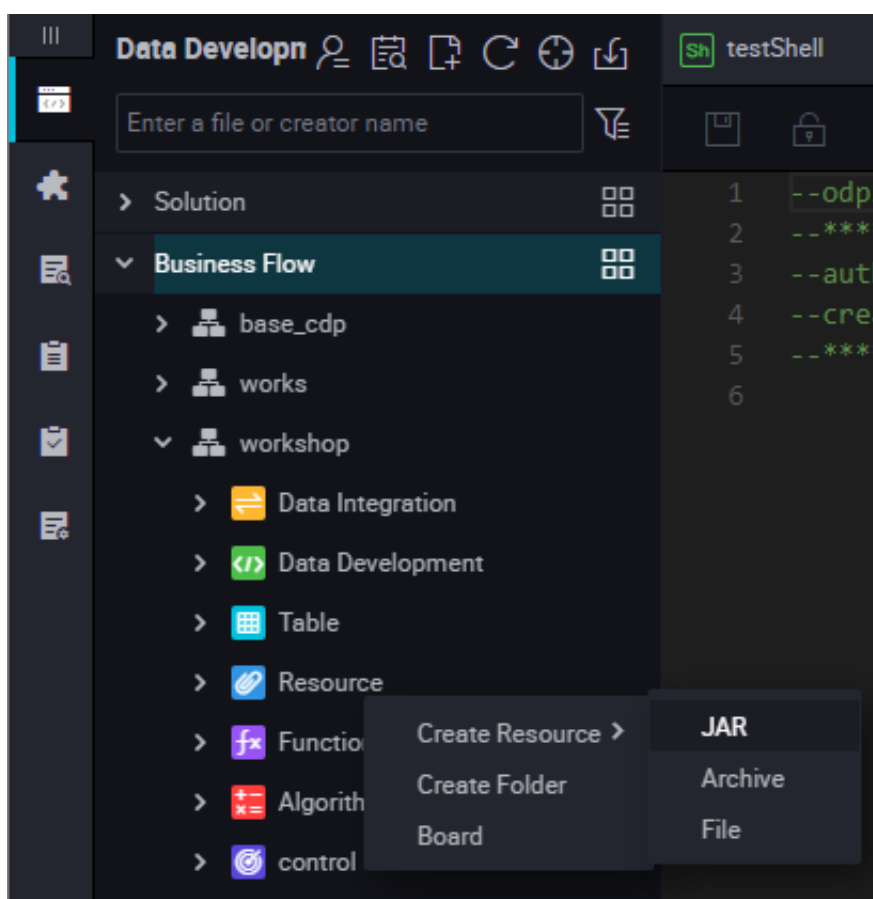
- Jar resource: You need to compile the Java code in the offline Java environment, compress the code into a jar package, and upload the package as the jar resource to ODPS.
- Small files: These resources are directly edited on DataWorks.
- File resource: When creating file resources, you need to select big files. You can also upload local resource files.

Create a resource instance

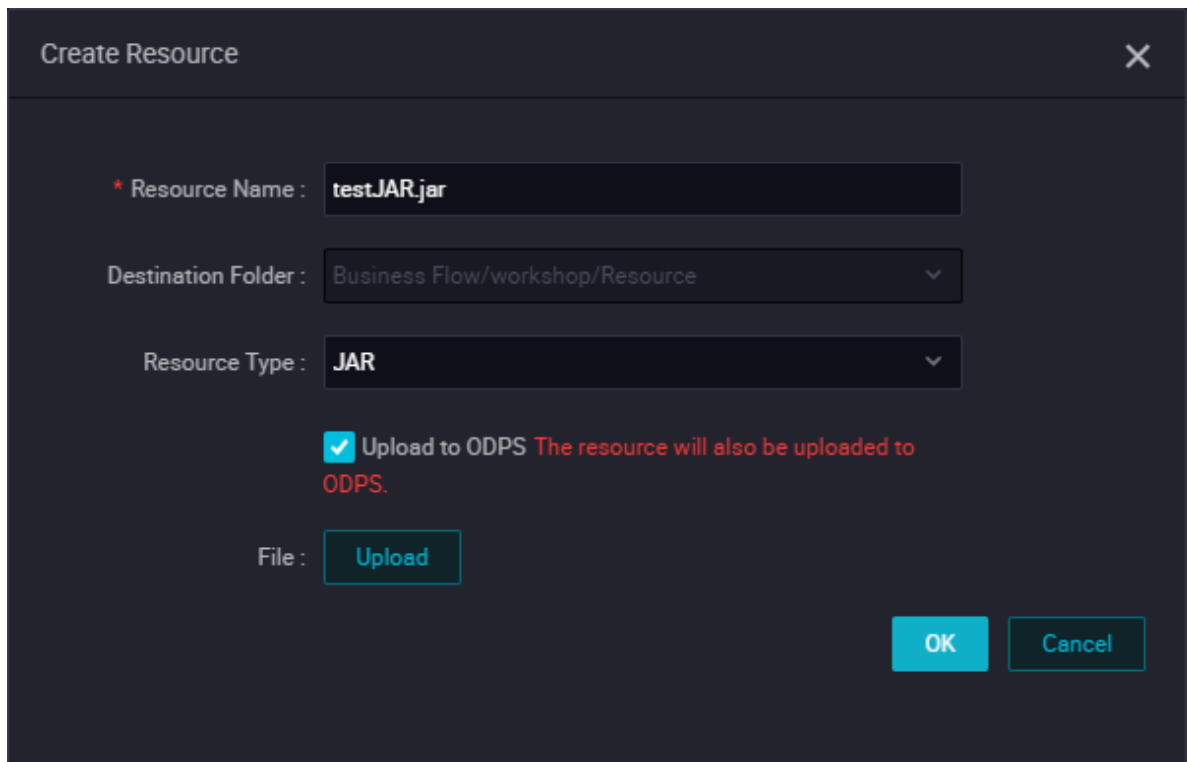
1. Click **Manual Business Flow** in the left-hand navigation bar, select **Create Business Flow**.



2. Right-click **Resource**, select **Create Resource > jar**.



3. The **Create Resource** dialog box is displayed. Enter the resource name according to the naming convention, set the resource type to jar, select a local jar package to the uploaded, and click Submit to submit the package in the development environment.



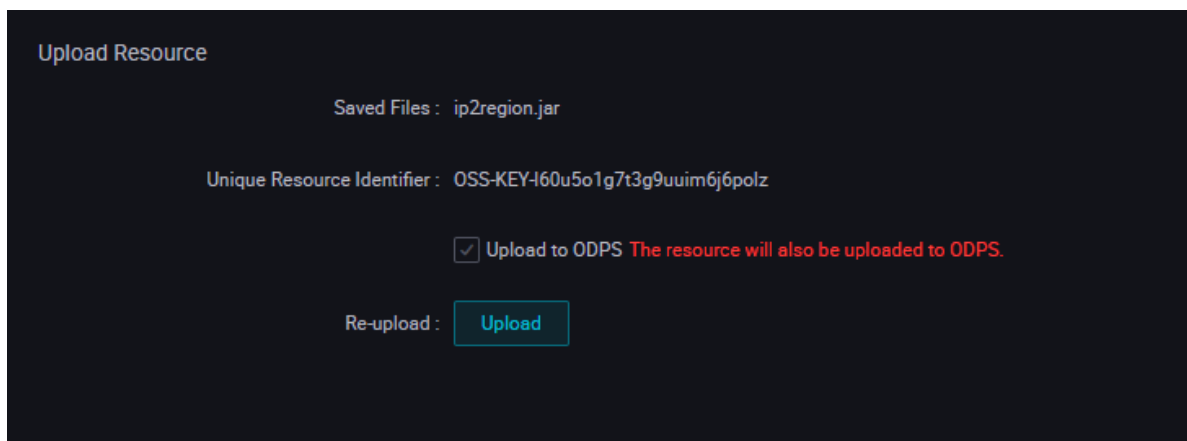
The 'Create Resource' dialog box is shown with a dark background. It contains the following fields and controls:

- Resource Name:** A text input field containing 'testJAR.jar'.
- Destination Folder:** A dropdown menu showing 'Business Flow/workshop/Resource'.
- Resource Type:** A dropdown menu showing 'JAR'.
- Upload to ODPS:** A checked checkbox with the text 'The resource will also be uploaded to ODPS.' in red.
- File:** A button labeled 'Upload'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

**Note:**

- If this jar package has been uploaded on the ODPS client, you must deselect **Uploaded as the ODPS resource**. In this upload, the resource will also be uploaded to ODPS. Otherwise, an error will be reported during the upload process.
- The resource name is not necessarily the same as the name of the uploaded file.
- Naming convention for a resource name: a string of 1 to 128 characters, including letters, numbers, underlines, and dots. The name is case insensitive. If the resource is a jar resource, the extension is .jar.

4. Click **Submit** to submit the resource to the development scheduling server.



The 'Upload Resource' dialog box is shown with a dark background. It contains the following fields and controls:

- Saved Files:** A text input field containing 'ip2region.jar'.
- Unique Resource Identifier:** A text input field containing 'OSS-KEY-I60u5o1g7t3g9uuim6j6polz'.
- Upload to ODPS:** A checked checkbox with the text 'The resource will also be uploaded to ODPS.' in red.
- Re-upload:** A button labeled 'Upload'.

5. Release a node task

For more information about the operation, see [Publish a task](#).

3.9.3 Function

Register the UDF

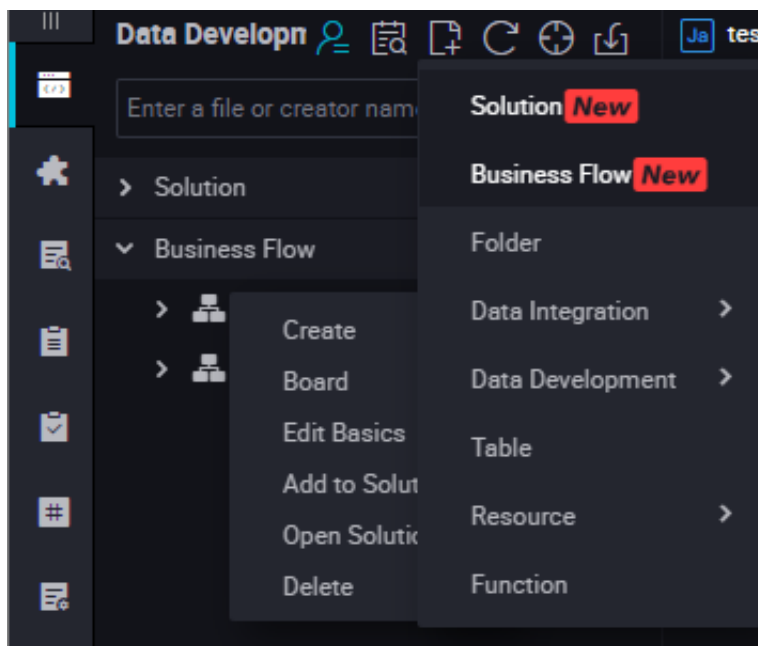
MaxCompute supports the UDFs. For more information, see [UDF overview](#).

DataWorks provides the visual GUI to register functions for replacing the ODPS command line `add function`.

Currently, the Python and Java APIs support implementation of UDFs. To compile a UDF program, you can upload the UDF code by [Adding resources](#) and then register the UDF.

UDF registration procedure

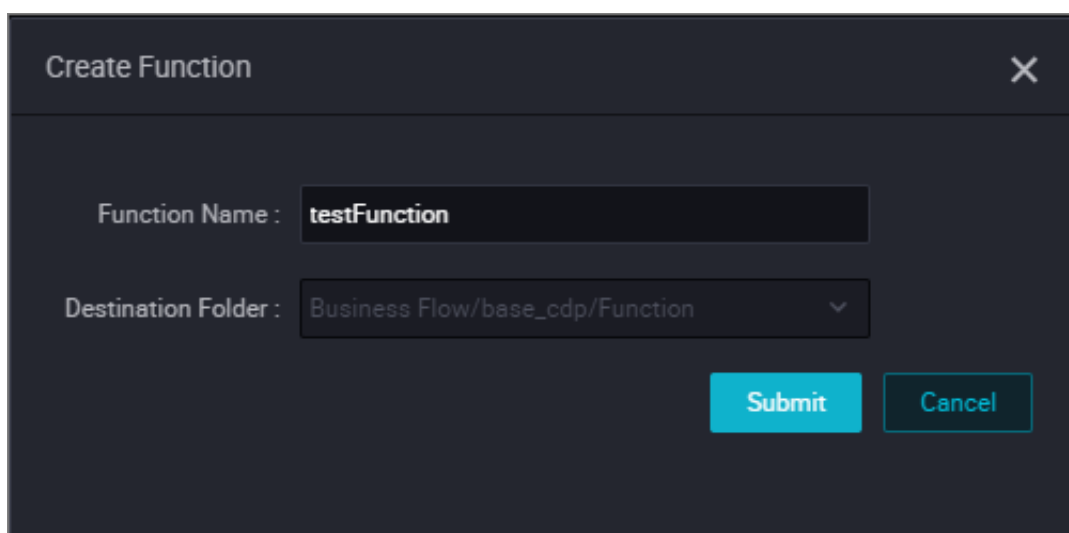
1. Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. In the offline Java environment, edit the program, compress the program into a jar package, create a jar resource, and submit and release the program.

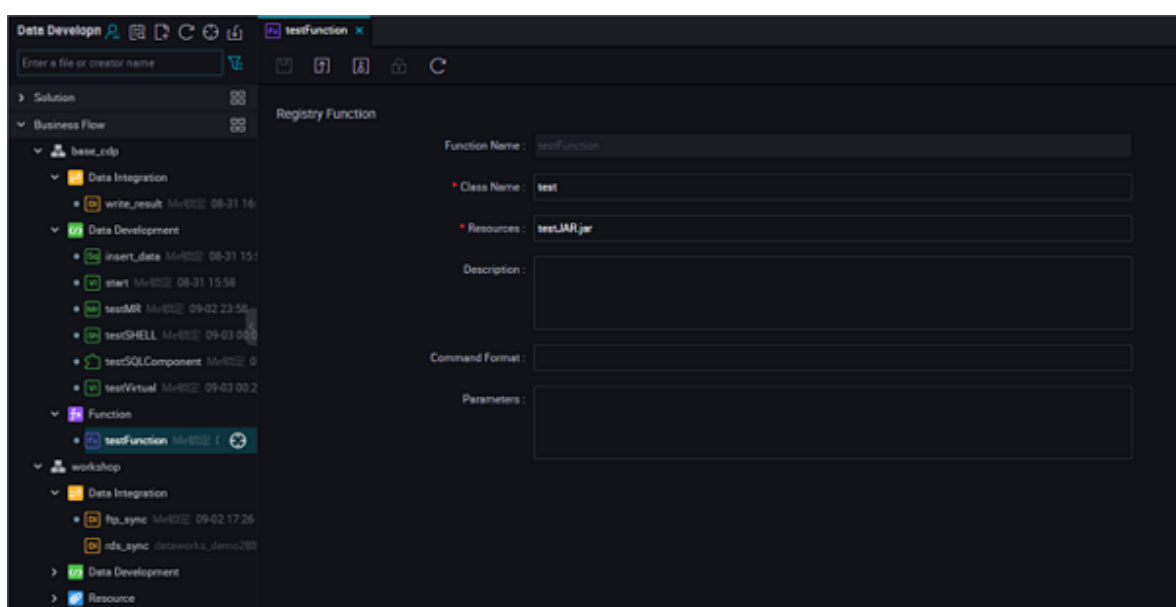
Alternatively, create a Python resource, compile and save the Python code, and then submit and release the code. For more information, see [Create Resources](#).

3. Select **Function** > **Create Function**, enter the configuration of the new function, click **Submit**.



The 'Create Function' dialog box is shown with a dark background. It has a title bar with 'Create Function' and a close button (X). Inside, there are two input fields: 'Function Name' with the value 'testFunction' and 'Destination Folder' with the value 'Business Flow/base_cdp/Function'. At the bottom right, there are two buttons: 'Submit' (blue) and 'Cancel' (gray).

4. Edit the function configuration.



- **Class Name:** name of the main class that implements the UDF. When the resource is Python, the typical style of writing is: Python resource name.Class name ('.py' is not needed in the resource name).
- **Resources:** Name of the resource in the second step, if there are multiple resources, separate them using commas.
- **Description:** UDF description. It is optional.

5. Submit the job.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

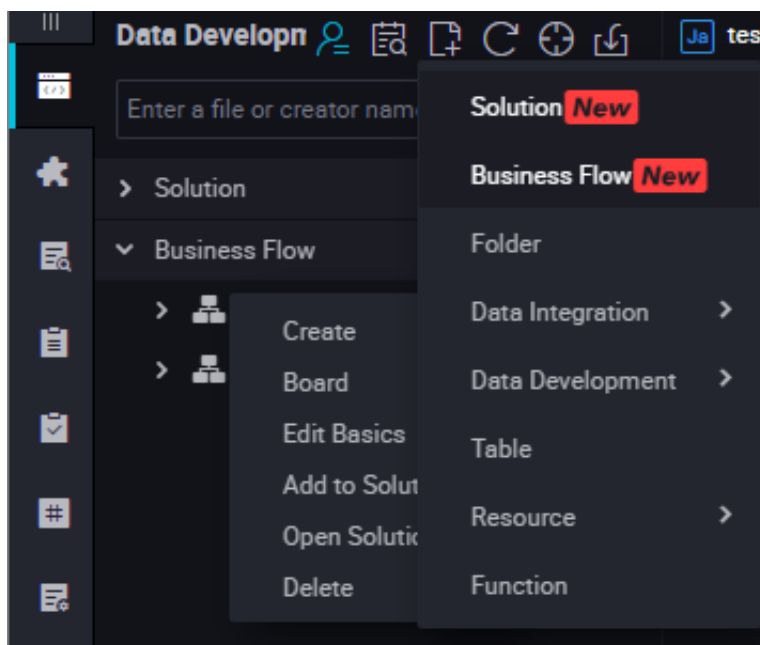
6. Release a node task

For more information about the operation, see [Publish a task](#).

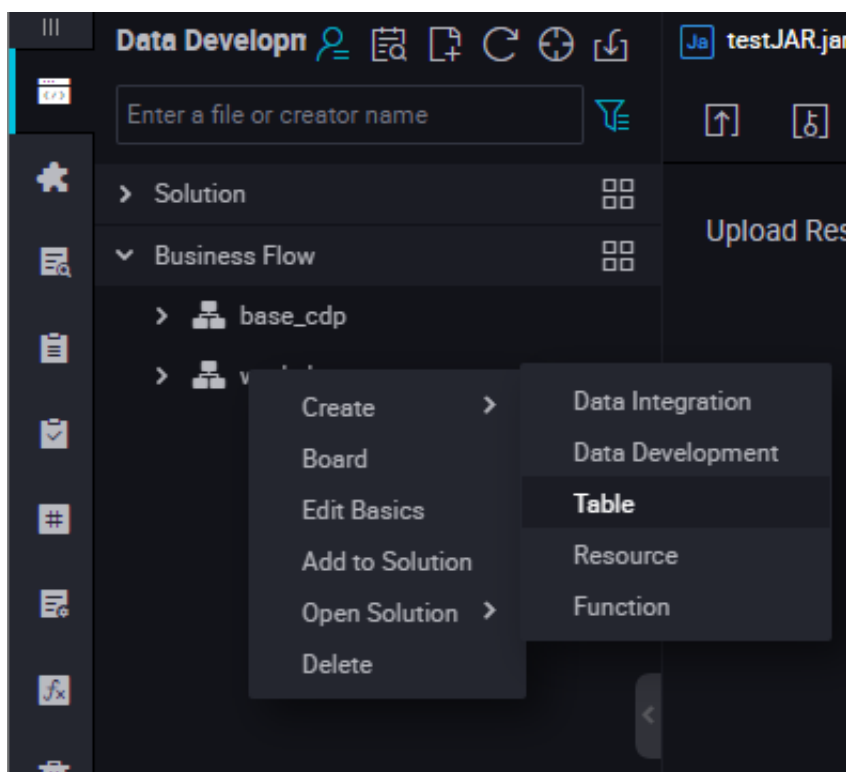
3.9.4 Table

Create a table

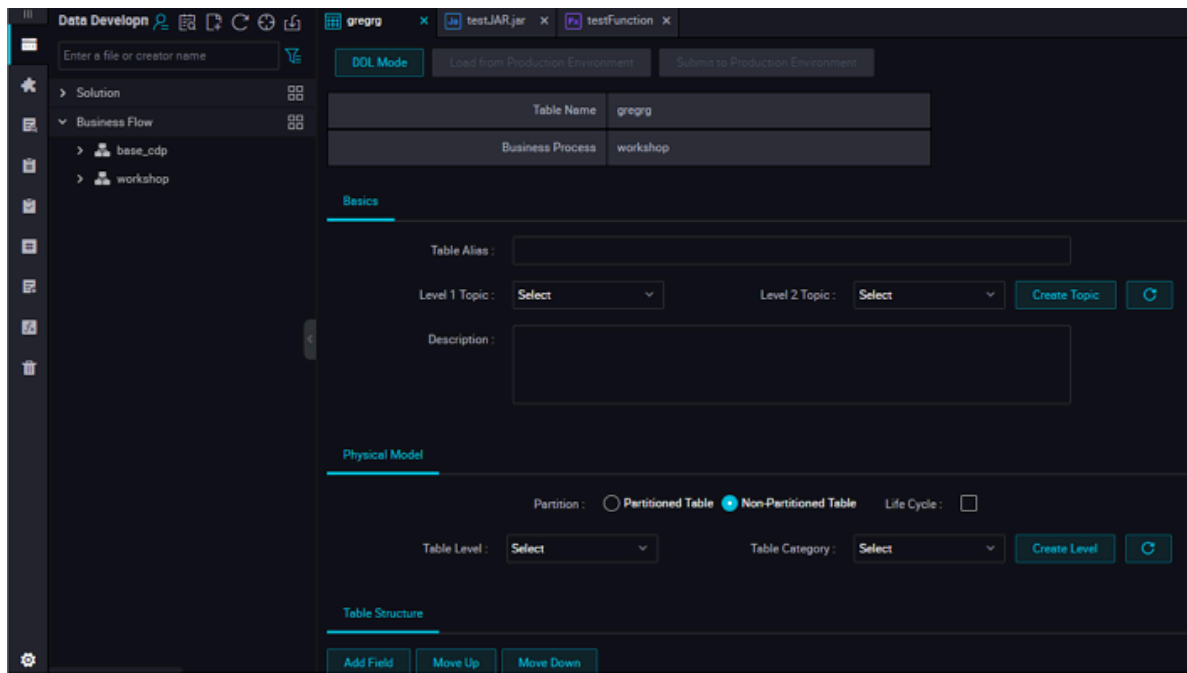
1. Click **Manual Business Flow**, select **Create Business Flow**.



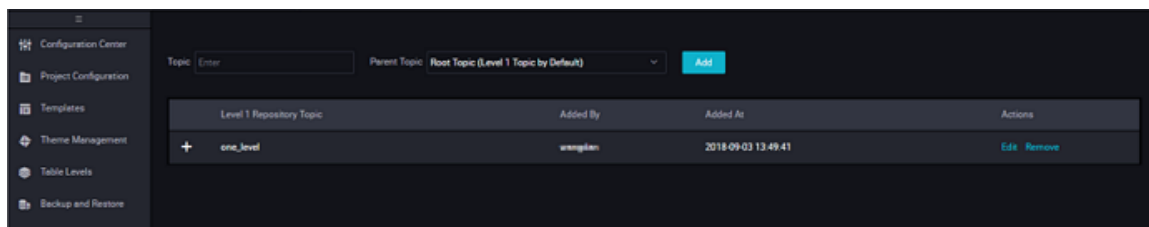
2. Right-click **Table** and select **Create Table**.



3. Set basic attributes.

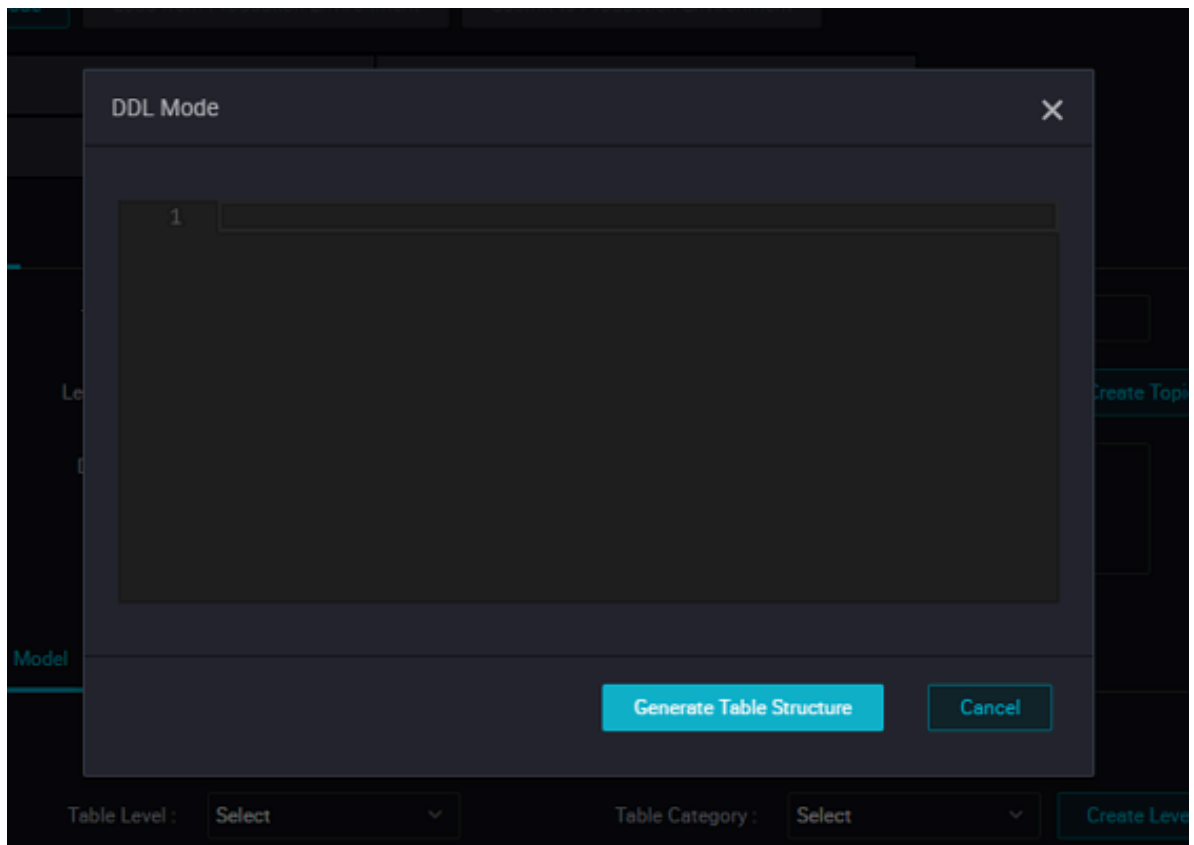


- Chinese Name: Chinese name of the table to be created.
- Level-1 Topic: Name of the level-1 target folder of the table to be created.
- Level-2 Topic: Name of the level-2 target folder of the table to be created.
- Description: Description of the table to be created.
- Click **Create Topic**. On the displayed Topic Management page, create level-1 and level-2 topics.



4. Create a table in DDL mode

Click **DDL Mode**. In the displayed dialog box, enter the standard table creation statements.



After editing the table creation SQL statements, click **Generate Table Structure**. Information in the Basic Attributes, Physical Model Design, and Table Structure Design areas is automatically entered.

5. Create a table on the GUI

If creating a table in DDL mode is not applicable, you can create the table on the GUI by performing the following settings.

- Physical model design
 - Partition Type: It can be set to Partitioned Table or Non-partitioned Table.
 - Life Cycle: Life cycle function of MaxCompute. Data in the table (or partition) that is not updated within a period specified by Life Cycle (unit: day) will be cleared.
 - Level: It can be set to DW, ODS, or RPT.
 - Physical Category: It can be set to Basic Business Layer, Advanced Business Layer, or Other. Click **Create Level**. On the displayed Level Management page, create a level.
- Table structure design
 - English Field Name: English name of a field, which may contain letters, digits, and underscores (_).
 - Chinese Name: Abbreviated Chinese name of a field.

- Field Type: MaxCompute data type, which can only be String, Bigint, Double, Datetime, or Boolean.
- Description: Detailed description of a field.
- Primary Key: Select it to indicate the field is the primary key or a field in the joint primary key.
- Click **Add Field** to add a column for a new field.
- Click **Delete Field** to delete a created field.

**Note:**

If you delete a field from a created table and submit the table again, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.

- Click **Move Up** to adjust the field order of the table to be created. However, to adjust the field order of a created table, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.
- Click **Move Down**, the operation is the same as that of **Move Up**.
- Click **Add Partition** to create a partition for the current table. To add a partition to a created table, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.
- Click **Delete Partition** to delete a partition. To delete a partition from a created table, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.
- Action: You can confirm to submit a new field, delete a field, and edit more attributes.

More attributes include information related to the data quality, which is provided for the system to generate the verification logic. They are used in scenarios such as data profiling, SQL scan, and test rule generation.

- **0 Allowed**: If it is selected, the field value can be zero. This option is applicable only to Bigint and Double fields.
- **Negative Value Allowed**: If it is selected, the field value can be a negative number. This option is applicable only to Bigint and Double fields.
- **Security Level**: It can be set to Non-sensitive, Sensitive, or Confidential.

C: Customer data, B: Company data, S: Business data.
C1–C2, B1, and S1 are non-sensitive data.
C3, B2–B4, S2, and S3 are sensitive data.

C4, S4, and B4 are confidential data.

- **Unit:** Unit of the amount, which can be dollar or cent. This option is not required for fields unrelated to the amount.
- **Lookup Table Name/Kay Value:** It is applicable to enumerated value-type fields, such as the member type and status. You can enter the name of the dictionary table (or dimension table) corresponding to the field. For example, the name of the dictionary table corresponding to the member status is `dim_user_status`. If you use a globally unique dictionary table, enter the corresponding `key_type` of the field in the dictionary table. For example, the corresponding key value of the member status is `TAOBAO_USER_STATUS`.
- **Value Range:** The maximum and minimum values of the current field. It is applicable only to bigint and double fields.
- **Regular Expression Verification:** Regular expression used by the current field. For example, if a field is a mobile phone number, its value can be limited to an 11-digit number by regular expression (or more strict limitation).
- **Maximum Length:** Maximum number of characters of the field value. It is applicable only to string fields.
- **Date Precision:** Precision of the date, which can be set to Hour, Day, Month, or others. For example, the precision of `month_id` in the monthly summary table is Month, although the field value is 2014-08-01 (it seems that the precision is Day). It is applicable to date values of the datetime or string type.
- **Date Format:** It is applicable only to date values of the string type. The format of the date value actually stored in the field is similar to `yyyy-mm-dd hh:mm:ss`.
- **KV Primary Separator/Secondary Separator:** It is applicable to a large field (of the string type) combined by KV pairs. For example, if a product expansion attribute has a value similar to `"key1:value1;key2:value2;key3:value3;..."`, the semicolon (;) is the primary separator of the field that separates the KV pairs, and the colon (:) is the secondary separator that separates the key and value in a KV pair.
- **Partition Field Design:** This option is displayed only when Partition Type in the Physical Model Design area is set to Partitioned Table.
- **Field Type:** We recommend that you use the string type for all fields.
- **Date Partition Format:** If a partition field is a date (although its data type may be string), select or enter a date format, such as `yyyymmdd`.
- **Date Partition Granularity:** For example, Day, Month, or Hour.

Submit a table

After editing the table structure information, submit the new table to the development environment and production environment.

- Click **Load from Development Environment**. If the table has been submitted to the development environment, this button is highlighted. After you click the button, the information of the created table in the development environment overwrites the information on the current page.
- Click **Submit to Development Environment**, the system checks whether all required items on the current editing page are completely set. If any omission exists, an alarm is reported, forbidding you to submit the table.
- Click **Load from Production Environment**, the detailed information of the table submitted to the production environment overwrites the information on the current page.
- Click **Create in Production Environment**, the table is created in the project of the production environment.

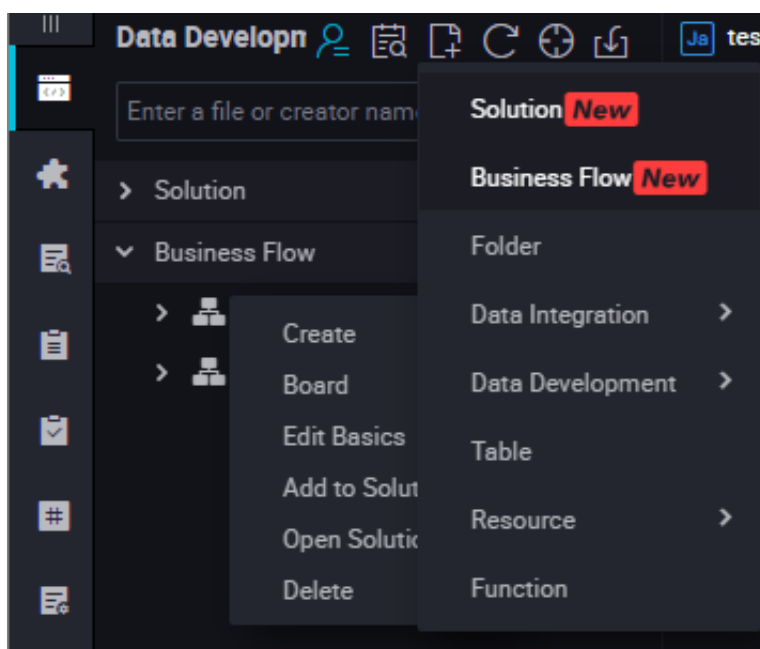
3.10 Manual task node type

3.10.1 ODPS SQL node

ODPS SQL adopts the syntax similar to that of SQL, and is applicable to the distributed scenario in which the amount of data is massive (TB-level) but the real-time requirement is not high. It is an OLAP application oriented to throughput. Because it takes a long time to complete the process from preparation to submission of a job, ODPS SQL is recommended if a business needs to handle thousands or tens of thousands of transactions.

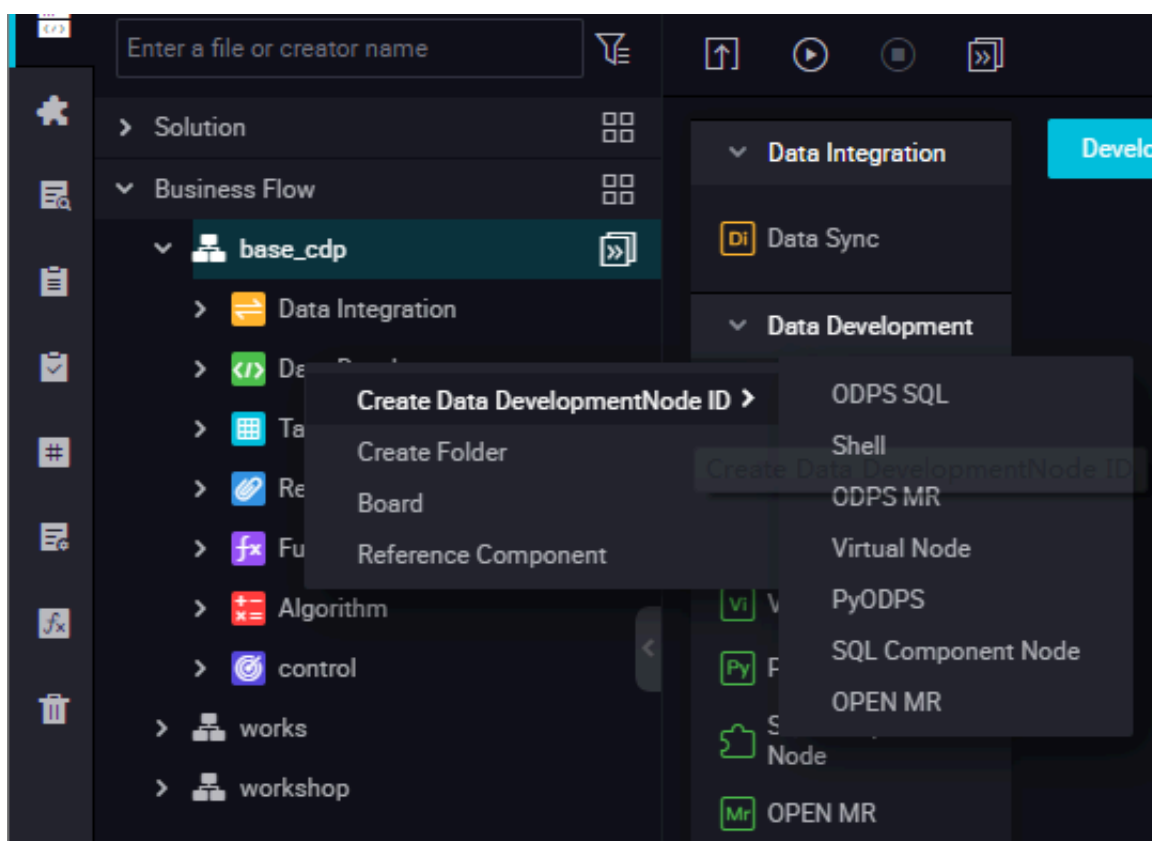
1. Create a business flow.

Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. Create ODPS SQL node.

Right-click **Data Development**, and select **Create Data Development Node > ODPS SQL**.



3. Edit the node code.

For more information about the syntax of the SQL statements, see [MaxCompute SQL statements](#).

4. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see [Release management](#).

7. Test in the production environment.

For more information about the operation, see [Manual task](#).

3.10.2 PyODPS node

DataWorks also provides the PyODPS task type and integrates the Python SDK of MaxCompute. You can directly edit the Python code to operate MaxCompute on a PyODPS node of DataWorks.

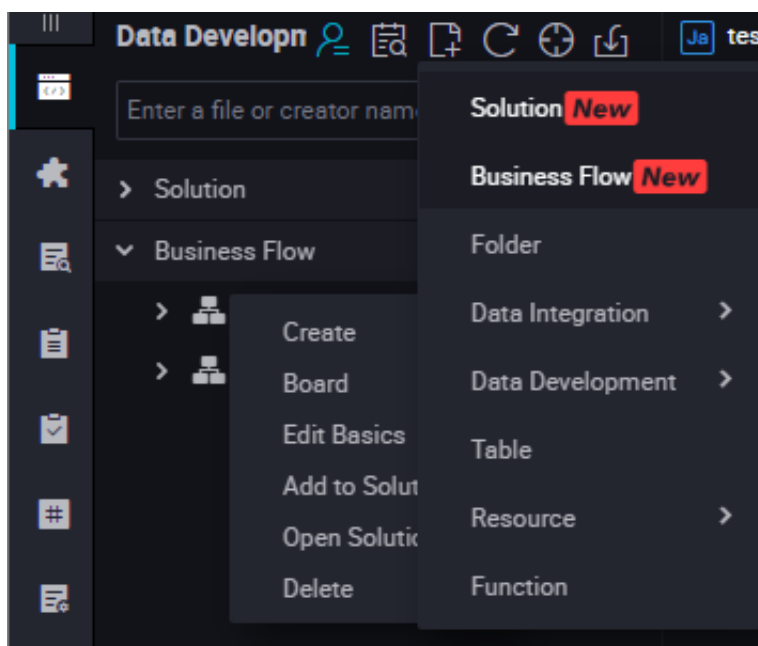
Create a PyODPS Node

MaxCompute provides the [Python SDK](#), which can be used to operate MaxCompute.

To create a PyODPS node, perform the following steps:

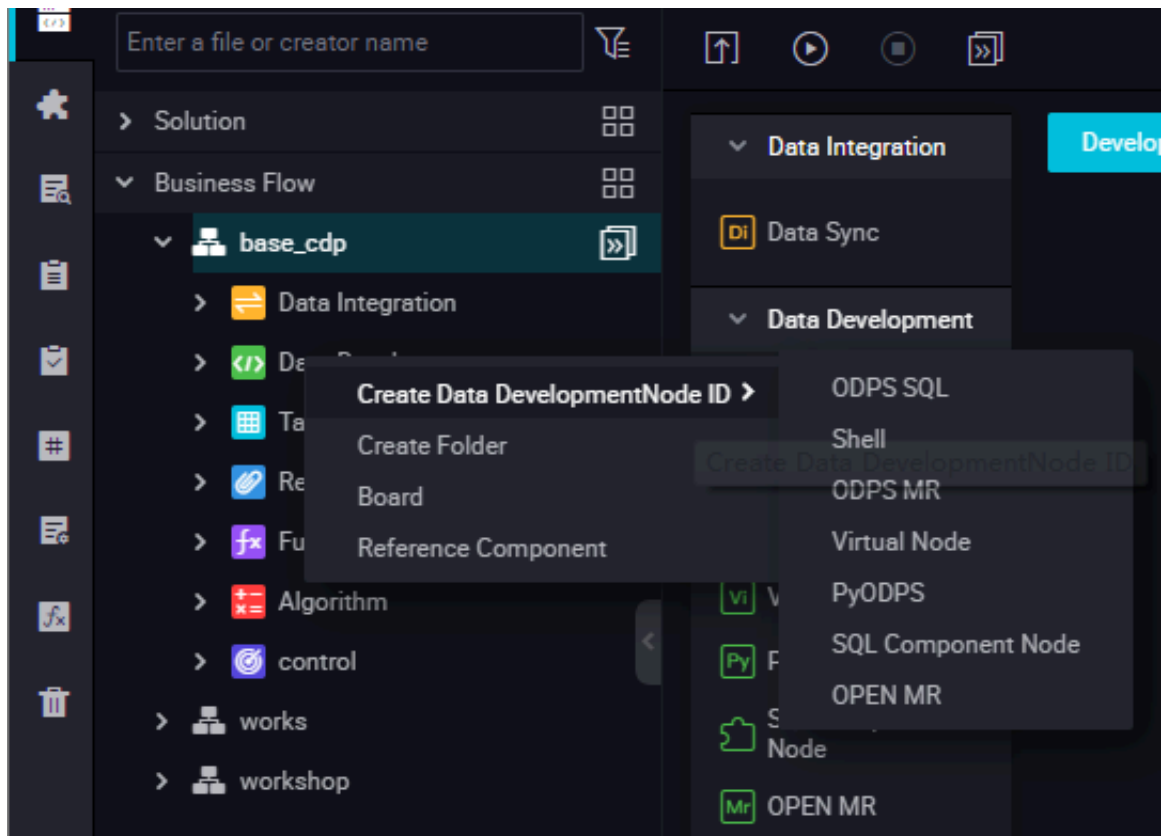
1. Create a business flow

Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. Create a PyODPS node.

Right-click **Data Development**, and select **Create Data Development Node > PyODPS**.



3. Edit the PyODPS node.

a. ODPS portal

On DataWorks, the PyODPS node contains a global variable `odps` or `o`, which is the ODPS entry. You do not need to manually define an ODPS entry.

```
print(odps.exist_table('PyODPS_iris'))
```

b. Run the SQL statements

PyODPS supports ODPS SQL query and can read the execution result. The return value of the `execute_sql` or `run_sql` method is the running instance.



Note:

Not all commands that can be executed on the ODPS console are SQL statements that are accepted by ODPS. You need to use other methods to call non DDL/DML statements. For example, use the `run_security_query` method to call the GRANT or REVOKE statements, and use the `run_xflow` or `execute_xflow` method to call PAI commands.

```
o.execute_sql('select * from dual') # Run the SQL statements in
synchronous mode. Blocking continues until execution of the SQL
statement is completed.
```

```
instance = o.runsql('select * from dual') # Run the SQL
statements in asynchronous mode.
print(instance.getlogview_address()) # Obtain the logview address
instance.waitforsuccess() # Blocking continues until execution of
the SQL statement is completed.
```

c. Configure the runtime parameters

The runtime parameters must be set sometimes. You can set the hints parameter with the parameter type of dict.

```
o.execute_sql('select * from PyODPS_iris', hints={'odps.sql.mapper
.split.size': 16})
```

After you add sql.settings to the global configuration, related runtime parameters are added upon each running.python.

```
from odps import options
options.sql.settings = {'odps.sql.mapper.split.size': 16}
o.execute_sql('select * from PyODPS_iris') # "hints" is added
based on the global configuration.
```

d. Read the SQL statement execution results

The instance that runs the SQL statement can directly perform the open_reader operation. In one case, the structured data is returned as the SQL statement execution result.

```
with odps.execute_sql('select * from dual').open_reader() as
reader:
    for record in reader: # Process each record.
```

In another case, desc may be executed in an SQL statement. In this case, the original SQL statement execution result is obtained through the reader.raw attribute.

```
with odps.execute_sql('desc dual').open_reader() as reader:
    print(reader.raw)
```



Note:

User-defined scheduling parameters are used in data development. If a PyODPS node is directly triggered on the page, the time must be clearly specified. The time of a PyODPS node cannot be directly replaced like that of an SQL node.

4. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see [Release management](#).

7. Test in the production environment.

For more information about the operation, see [Manual task](#).

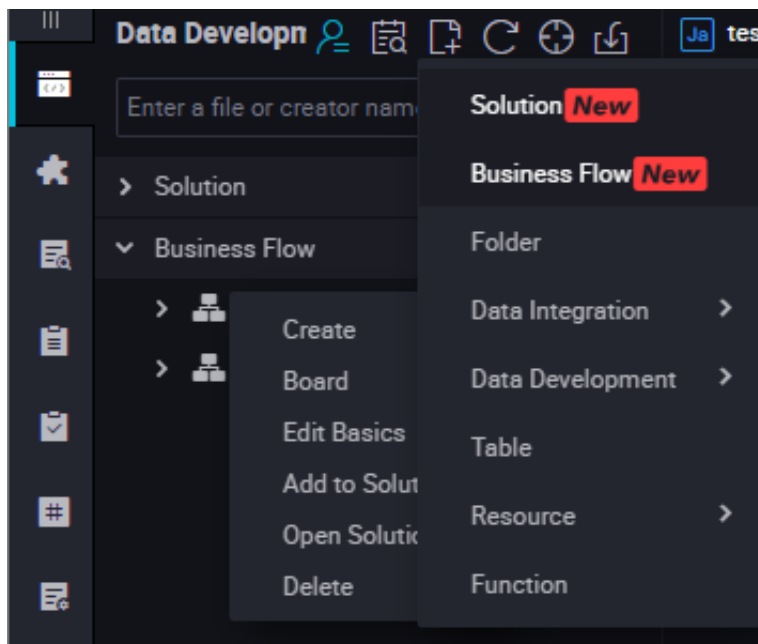
3.10.3 Manual data integration node

Currently, the data integration task supports the following data sources: MaxCompute, MySQL, DRDS, SQL Server, PostgreSQL, Oracle, MongoDB, DB2, Table Store, OTSStream, OSS, FTP, Hbase, LogHub, HDFS, and Stream. For details about more supported data sources, see [Supported data sources](#).



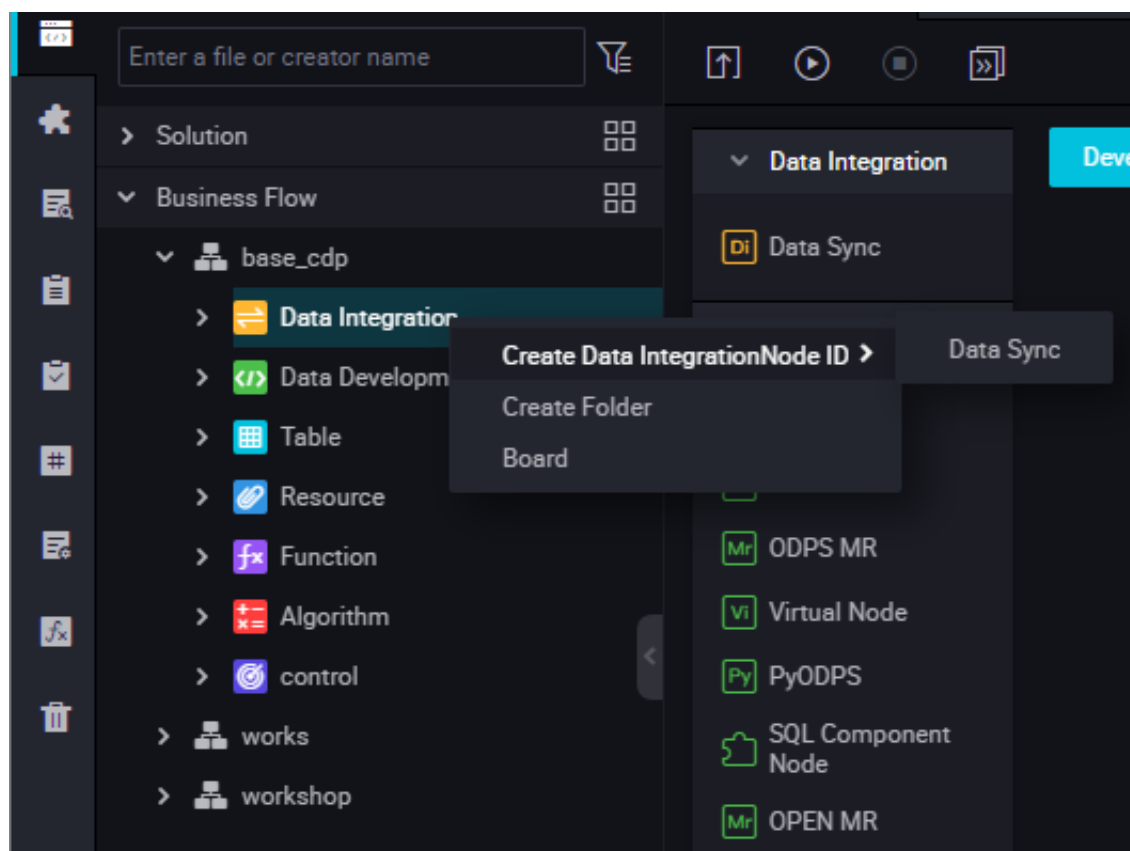
1. Create a business flow

Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. Create a data integration node

Right-click **Data Integration**, and select **Create Data Data Integration Node > Data Integration**.

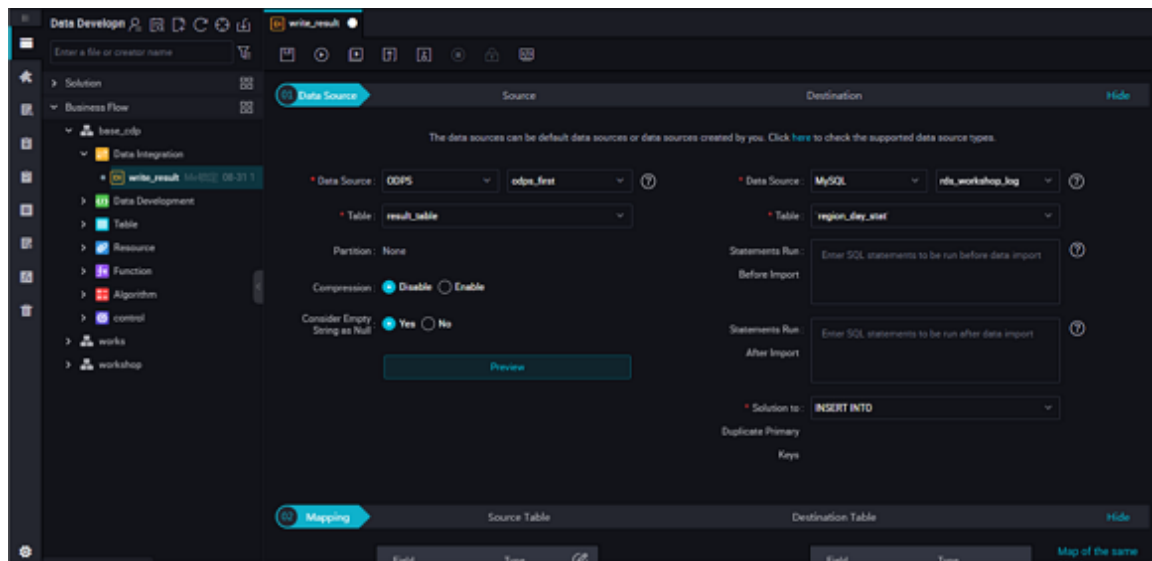


3. Configure a intergration task

You can enter the source table name and target table name to complete a simple task configuration.

After you enter a table name, a list of objects that match the table name is automatically displayed (Currently, only exact match is supported. Therefore, you must enter the correct and complete table name). Some objects are not supported by the current intergration center and are marked **Not supported**. You can move the mouse over an object. The detailed information about the object, such as the database, IP address, and owner of the table, is automatically displayed. The information helps you select an appropriate table object. After selecting an object, click the object. The column information is automatically filled in. You can edit columns, for example, moving, deleting, or adding column.

a. Configure intergration tables.



b. Edit the data source.

Generally, you do not need to edit the content of the source table unless necessary.

- Click **Insert** on the right of a column to insert a new column.
- Click **Delete** on the right of a column to delete the column.

c. Edit the data destination.

Generally, you do not need to edit the field information of the destination table unless necessary (for example, you need to import data of only some columns).



Note:

If the destination is an ODPS table, columns cannot be deleted. In configuration of a intergration task, the field settings of the source table matches those of the destination table in one-to-one relationship by page instead of by field name.

d. Incremental intergration and full intergration.

- Shard format for incremental intergration: `ds=${bizdate}`
- Shard format for full intergration: `ds=*`



Note:

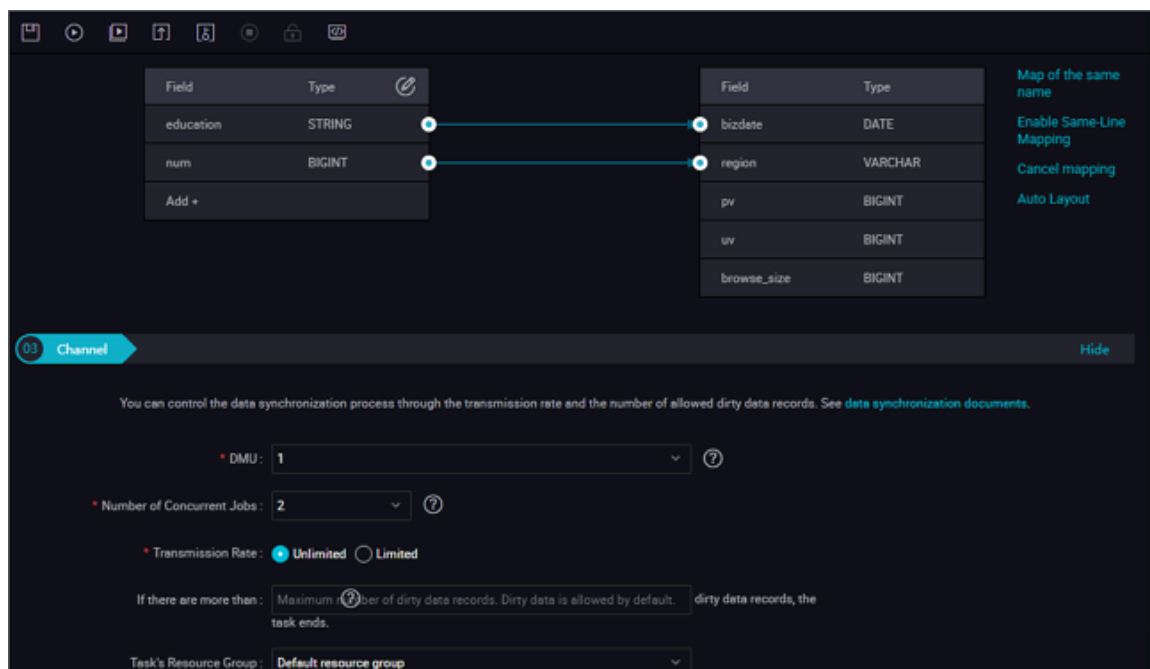
If multiple shards need to be synchronized, the intergration center supports simple regular expressions.

- For example, if you need to synchronize multiple shards, but it is difficult to write regular expressions, use the following method: `ds=20180312 | ds=20180313 | ds=20180314;`

- If you need to synchronize shards in the same range, the intergration center supports an extended syntax similar to the following: `/*query*/ds>=20180313 and ds<20180315`; If this method is used, you must add `/query/`.
- The variable `bizdate` must be defined in the following parameter: `-p"-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour"`. If you need to customize a variable, for example, `pt=${selfVar}`, also define the variable in the parameter, for example, `-p"-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour -DselfVar=xxxx"`.

e. Field mapping.

Fields are mapped based on the locations of fields in the source table and destination table, instead of based on the field names and types.



Note:

If the source table is an ODPS table, fields cannot be added during data intergration. If the source table is not an ODPS table, fields can be added during data intergration.

f. Tunnel control.

Tunnel control is used to control the speed and error rate when you select a intergration task.

- DMU: Data migration unit, which measures the resources (including the CPU, memory, and network) consumed during data integration.

- Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data integration task.
- integration speed: Maximum speed of the integration task.
- Maximum error count: It is used to control the amount of dirty data, and is set by yourself based on the amount of synchronized data when the field types of the source table do not match those of the destination table. It indicates the maximum dirty data count allowed. If it is set to 0, no dirty data is allowed; if it is not specified, dirty data is allowed.
- Task resource group: To select a resource group where the current integration node is located, you can add or modify the resource group on the data integration page.

4. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit a node task.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see Release management.

7. Test in the production environment.

For more information about the operation, see [Cyclic task](#).

3.10.4 ODPS MR node

MaxCompute supports MapReduce programming APIs. You can use the Java API provided by MapReduce to write MapReduce programs for processing data in MaxCompute. You can create ODPS MR nodes and use them in Task Scheduling.

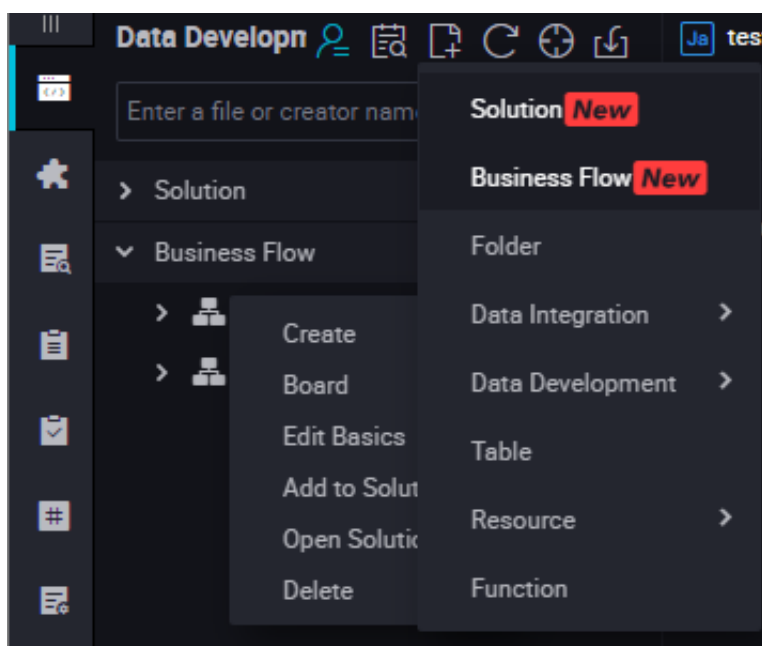
For how to edit and use the ODPS MR, see the examples in the MaxCompute documentation [WordCount examples](#).

To use an ODPS MR node, you must first upload and release the resource to be used, and then create the ODPS MR node.

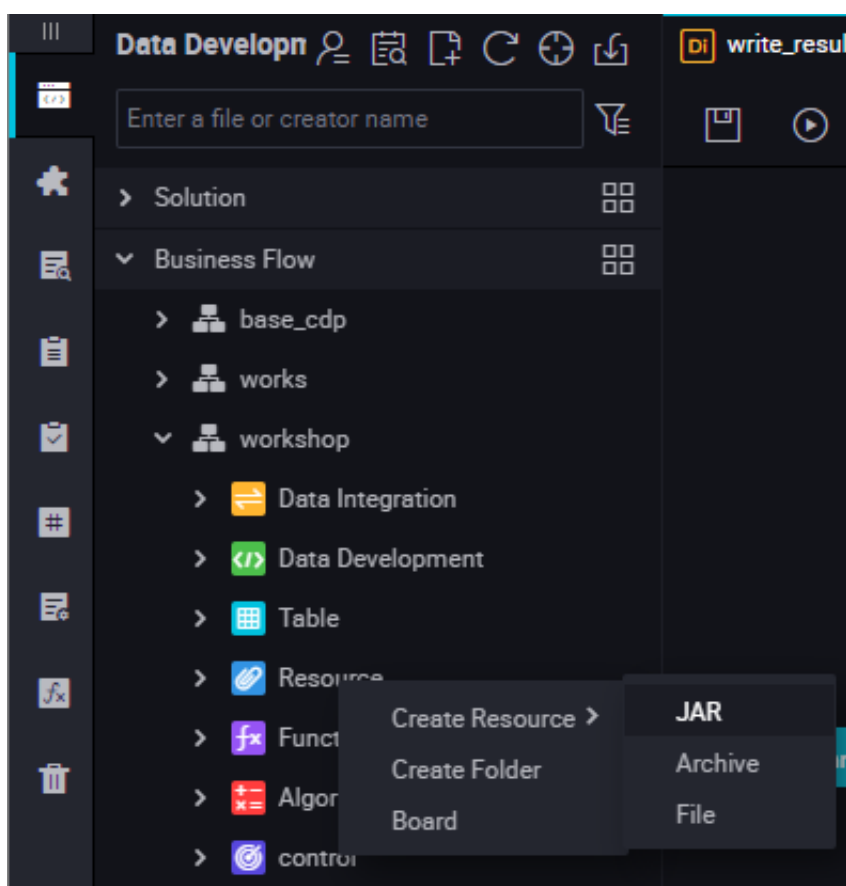
Create a resource instance

1. Create a business flow

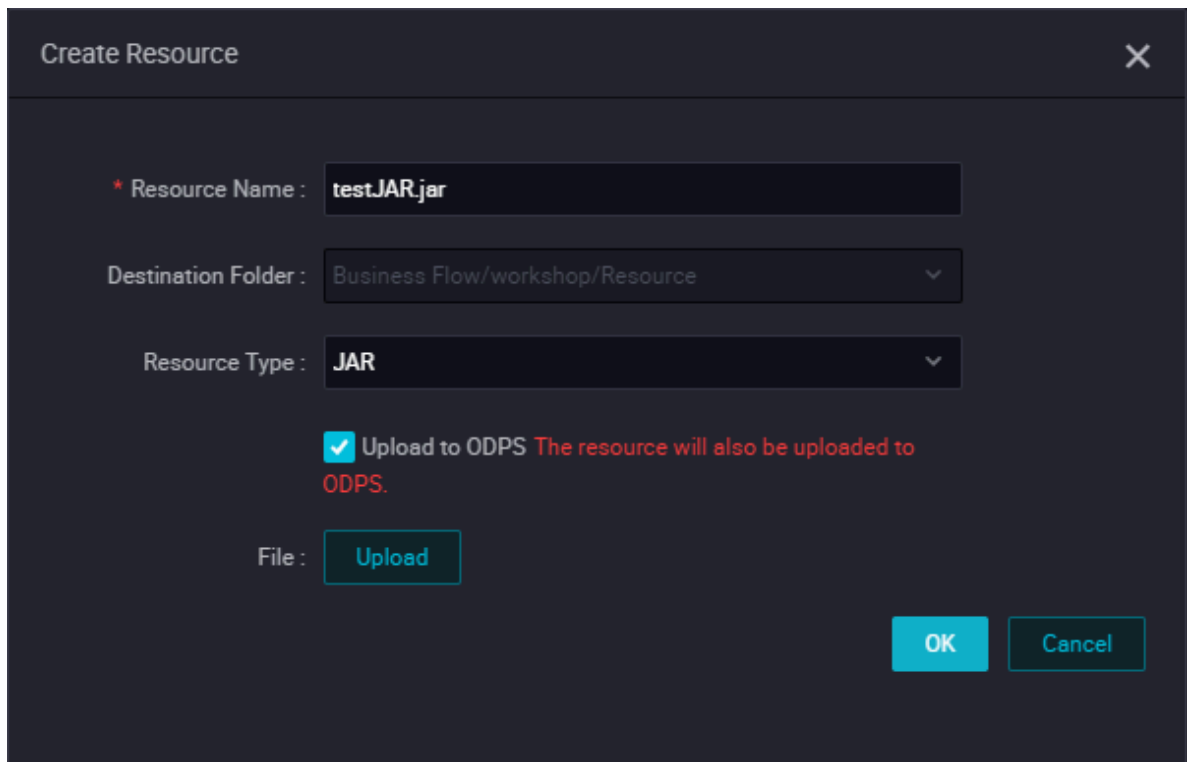
Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. Right-click **Resource**, and select **Create Resource > jar**.



3. Enter the resource name in the **Create Resource** according to the naming convention, set the resource type to jar, select a local jar package to the uploaded.



Create Resource

* Resource Name : testJAR.jar

Destination Folder : Business Flow/workshop/Resource

Resource Type : JAR

☒ Upload to ODPS The resource will also be uploaded to ODPS.

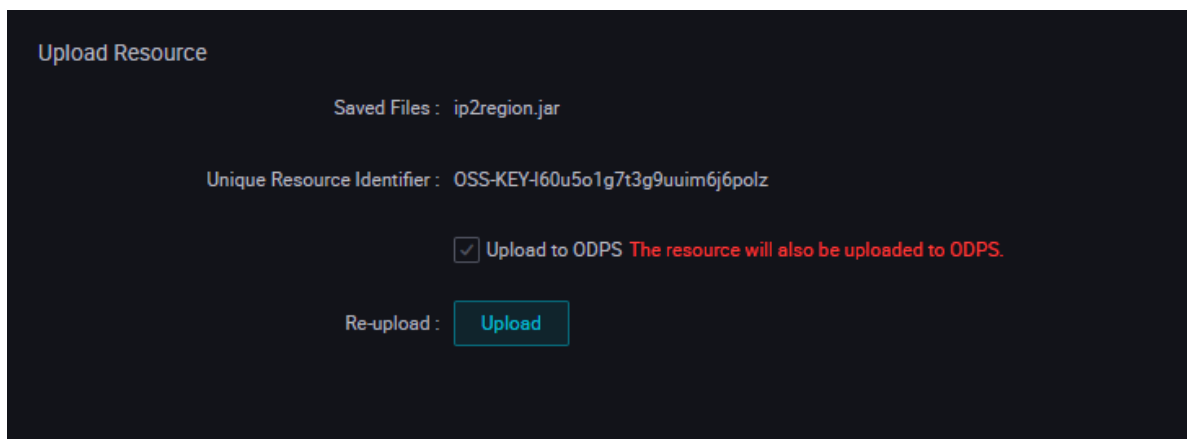
File :



Note:

- If this jar package has been uploaded on the ODPS client, you must deselect **Uploaded as the ODPS resource**. In this upload, the resource will also be uploaded to ODPS. Otherwise, an error will be reported during the upload process.
- The resource name is not necessarily the same as the name of the uploaded file.
- Naming convention for a resource name: a string of 1 to 128 characters, including letters, numbers, underlines, and dots. The name is case insensitive. If the resource is a jar resource, the extension is .jar. If the resource is a Python resource, the extension is .py.

4. Click **Submit** to submit the resource to the development scheduling server.



Upload Resource

Saved Files : ip2region.jar

Unique Resource Identifier : OSS-KEY-I60u5o1g7t3g9uuim6j6polz

☒ Upload to ODPS The resource will also be uploaded to ODPS.

Re-upload :

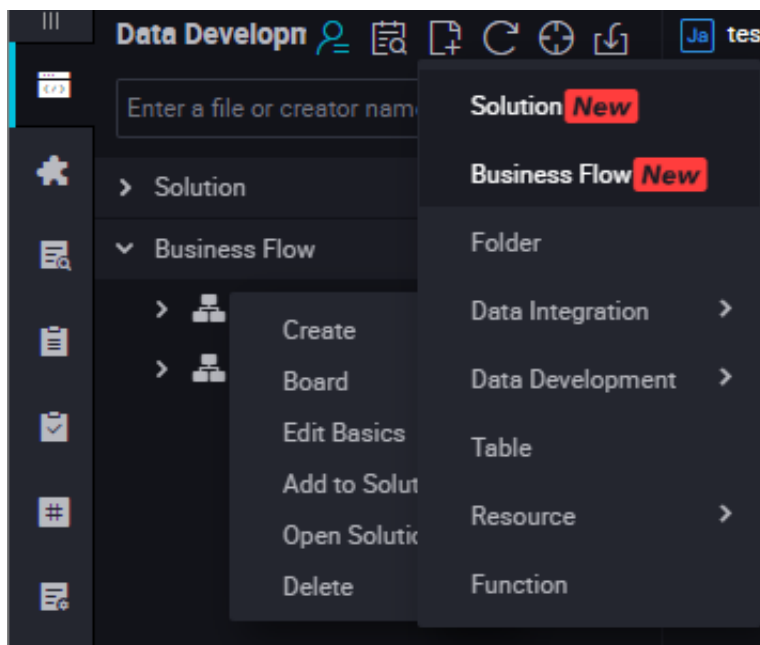
5. Publish a node task.

For more information about the operation, see Release management.

Create an ODPS MR node

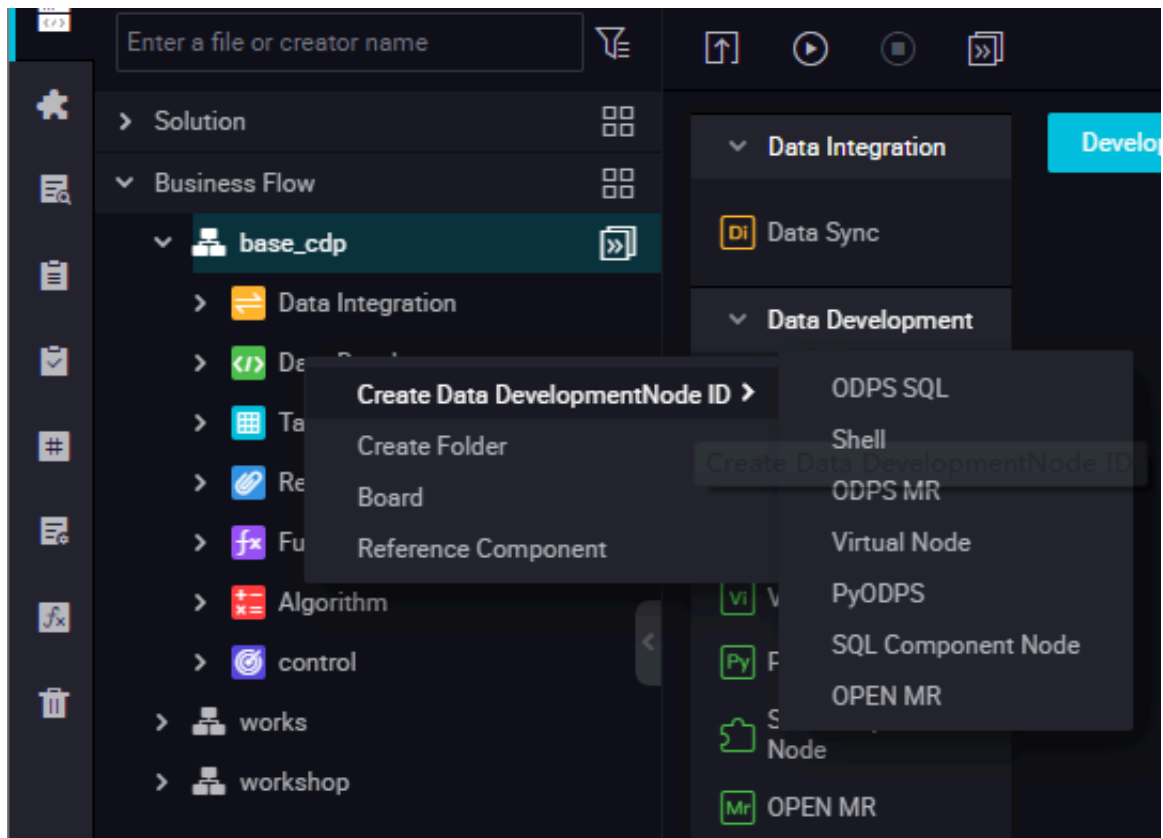
1. Create a business flow

Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.

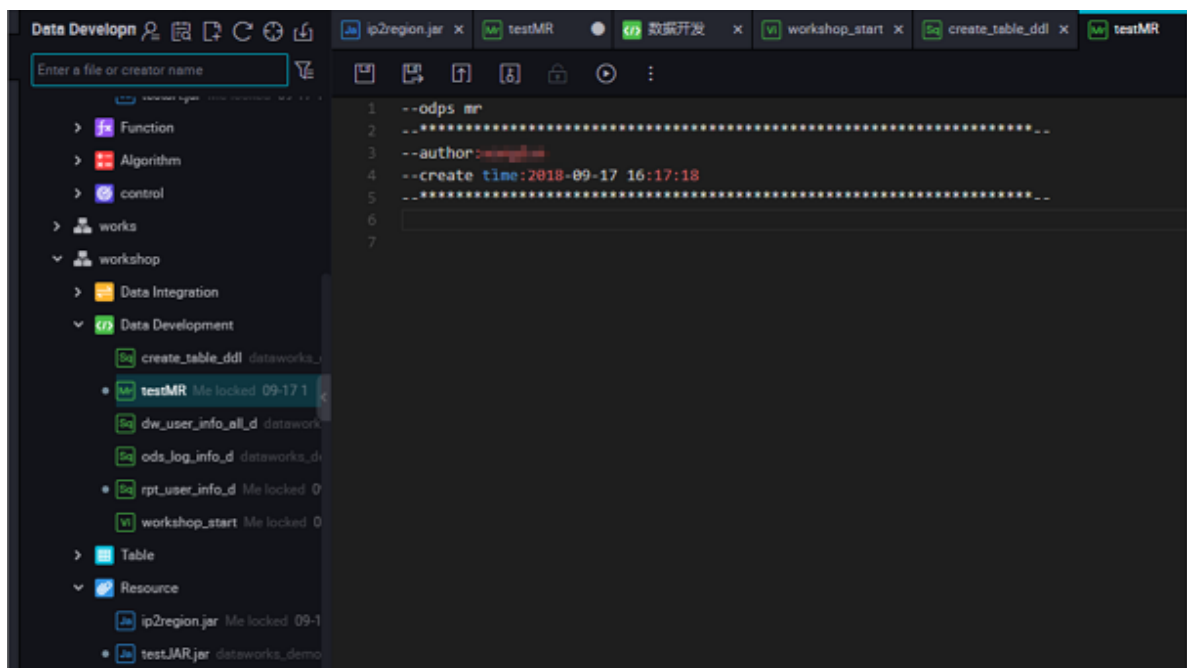


2. Create an ODPS MR node.

Right-click **Data Development**, and select **Create Data Development Node > ODPS MR**.



3. Edit the node code. Double click the new ODPS MR node and enter the following interface.



Node code editing example:

```
jar -resources base_test.jar -classpath ./base_test.jar com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll
```

The code is described below:

- `-resources base_test.jar`: indicates the file name of the referenced jar resource.
- `-classpath`: jar package path, you can right-click the Reference resource and obtain this path.

**Note:**

Double click the new ODPS MR node and enter the jar resource after entering the ODPS MR node interface.

- `com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll`: indicates the main class in the jar package that is called during execution. It must be consistent with the main class name in the jar package.

When one MR calls multiple jar resources, classpath must be written as follows: `-classpath ./xxxx1.jar,./xxxx2.jar`, that is, two paths must be separated by a comma.

4. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see Release management.

7. Test in the production environment.

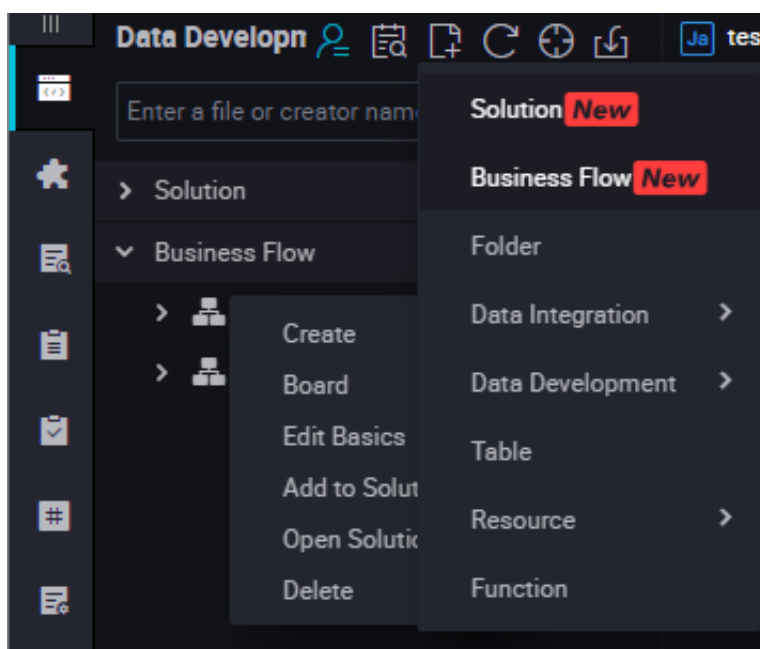
For more information about the operation, see [Manual task](#).

3.10.5 SQL component node

Procedure

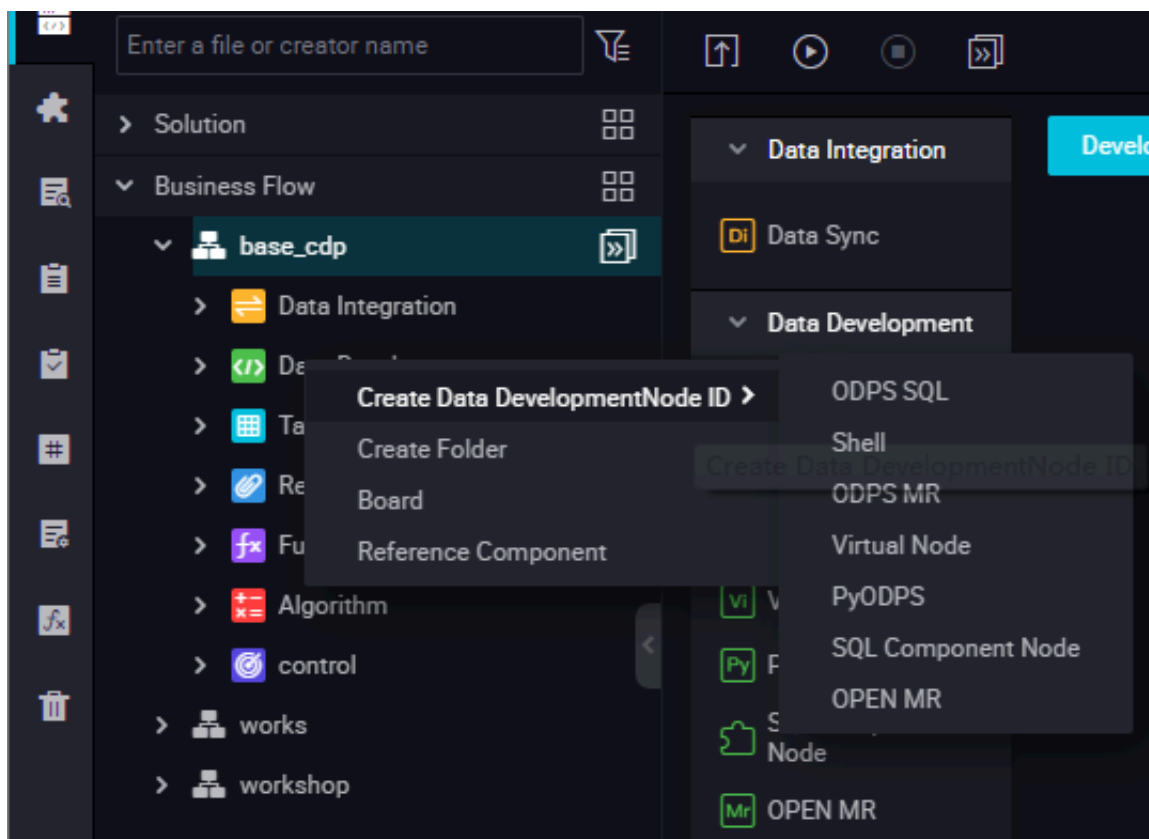
1. Create Business Flow

Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. Create an SQL component node

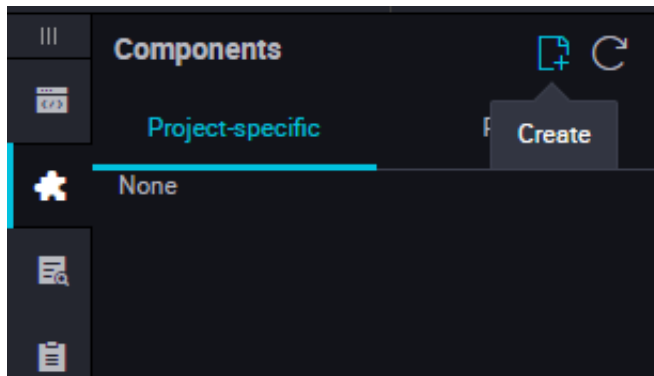
Right-click **Data Development**, and select **Create Data Development Node > SQL Component Node**.



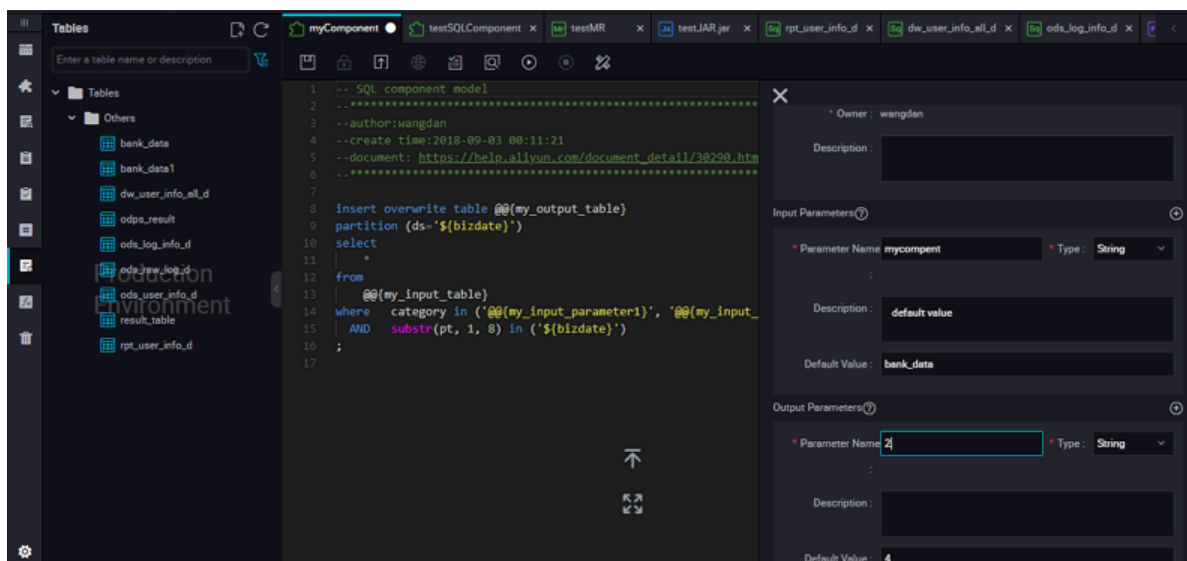
3. To improve the development efficiency, data task developers can use components contributed by project members and tenant members to create data processing nodes.

- Components created by members of the local project are located under Project Components.
- Components created by tenant members are located under Public Components.

When create a node, set the node type to the **SQL component node** type, and specify the name of the node.



Specify parameters for the selected component.



Enter the parameter name, and set the parameter type to Table or String.

Specify three get_top_n parameters in sequence.

Specify the following input table for the parameters of the Table type: test_project.test_table.

4. Node scheduling configuration.

Click the **Schedule Configuration** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

5. Submit a node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

6. Publish a node task.

For more information about the operation, see Release management.

7. Test in a production environment.

For more information about the operation, see [Manual task](#).

Upgrade the version of an SQL component node.

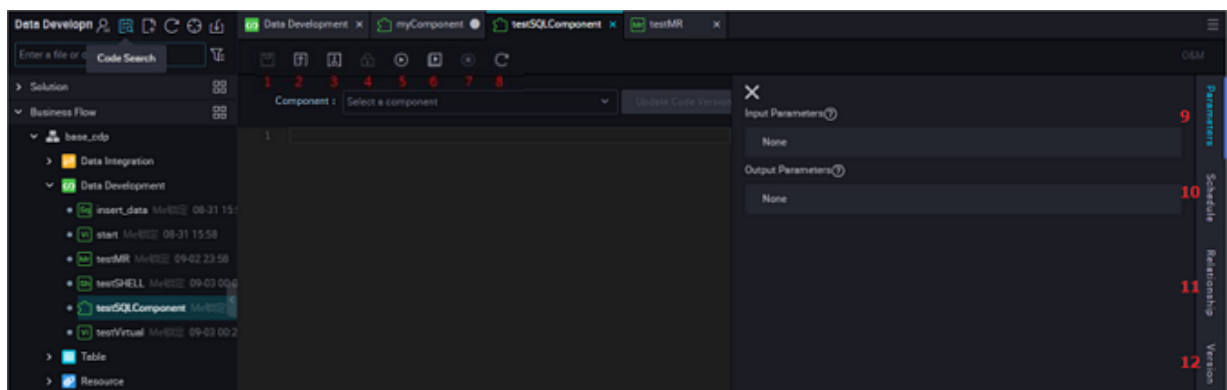
After the component developer release a new version, the component users can choose whether to upgrade the use instance of the existing component to the latest version of the used component .

With the component version mechanism, developers can continuously upgrade components and component users can continuously enjoy the improved process execution efficiency and optimized business effects after upgrade of components.



For example, user A uses the v1.0 component developed by user C, and the component owner C upgrades the component to V.2.0. After the upgrade, user A can still use the v1.0 component, but will receive the upgrade reminder. After comparing the new code with the old code, user A finds that the business effects of the new version are better than those of the old version, and therefore can determine whether to upgrade the component to the latest version.

To upgrade an SQL component node developed based on the component template, you only need to select Upgrade, check whether parameter settings of the SQL component node are still effective in the new version, make some adjustments based on the instructions of the new version component, and then submit and release the node like a common SQL component node.

Interface functions



The interface features are described below:

No.	Feature	Description
1	Save	Click it to save settings of the current component.
2	Submit	Click it to submit the current component to the development environment.
3	Submit and Unlock	Click it to submit the current node and unlock the node to edit the code.
4	Steallock Edit	Click it to steallock edit the node if you are not the owner of the current component.
5	Run	Click it to run the component locally in the development environment.
6	Advanced Run (with Parameters)	<p>Click it to run the code of the current node using the parameters configured for the code.</p> <div>  Note: Advanced Run is unavailable to a Shell node. </div>
7	Stop Run	Click it to stop a running component.
8	Re-load	<p>Click it to refresh the interface and restore the last saved status. Unsaved content will be lost.</p> <div>  Note: If cache is enabled in the configuration center, after the interface is refreshed, you are notified of the code that is cached but not saved. In this case, select the version that you need. </div>
9	Parameter Settings	Click it to view the component information, input parameter settings, and output parameter settings.
10	Attributes	Click it set the owner, description, parameters, and resource group of the node.
11	Kinship	Click it to view the map of kinship between SQL component nodes and the internal kinship map of each SQL component node.
12	Version	Click it to view the submission and release records of the current component.

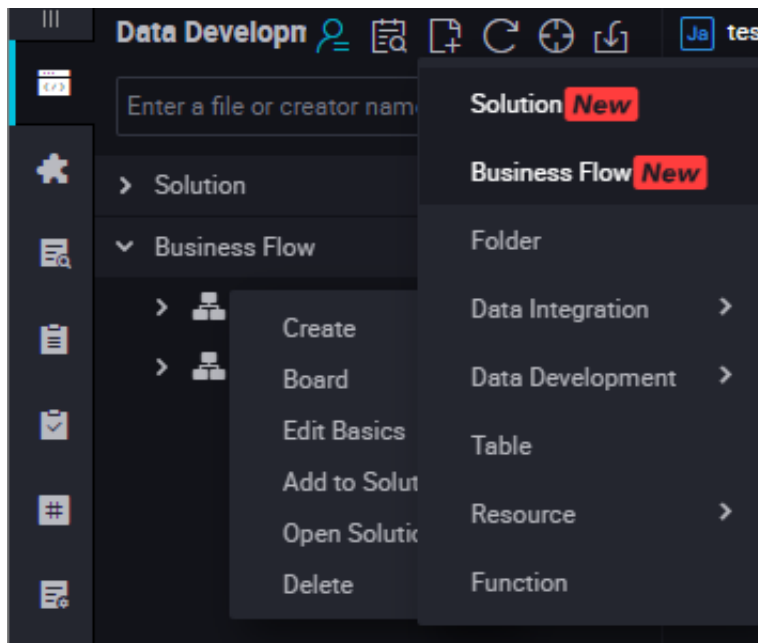
3.10.6 Virtual node

A virtual node is a control node that does not generate any data. Generally, it is used as the root node for overall planning of nodes in the workflow.

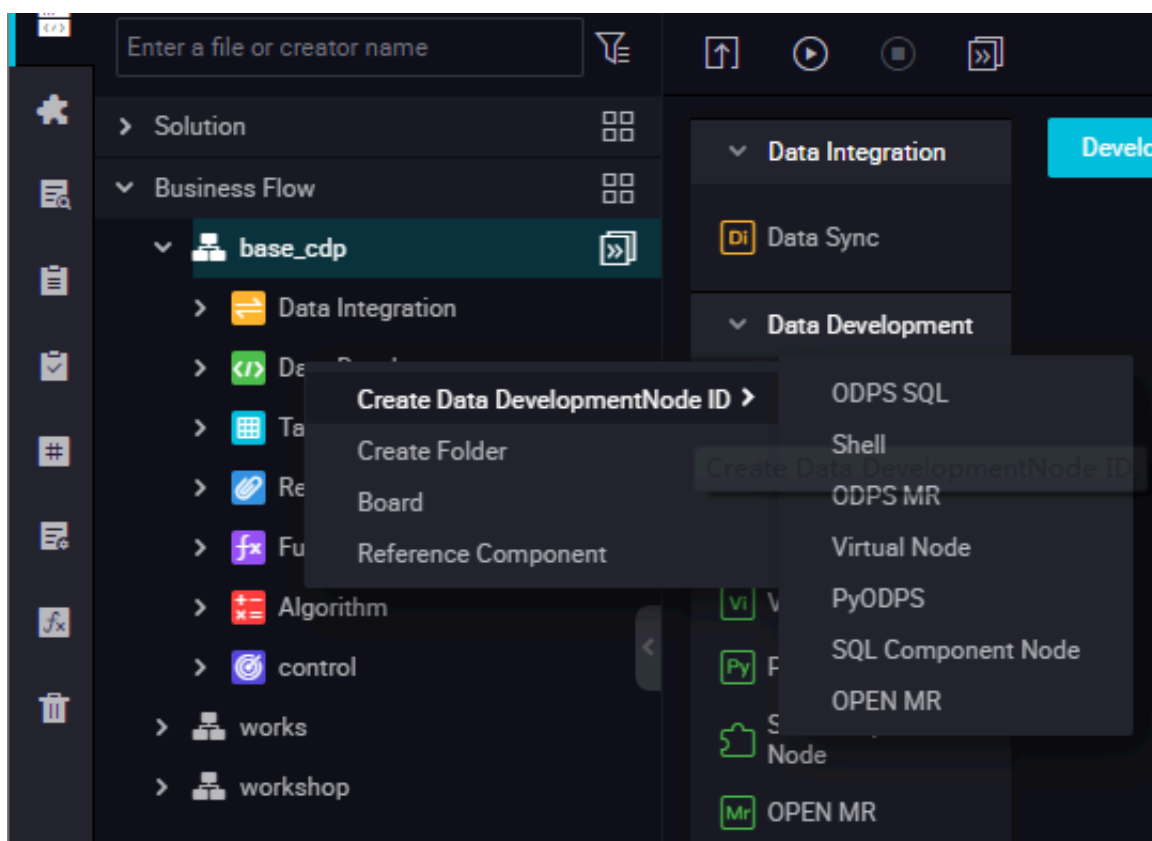
Create a virtual node task

1. Create a business flow

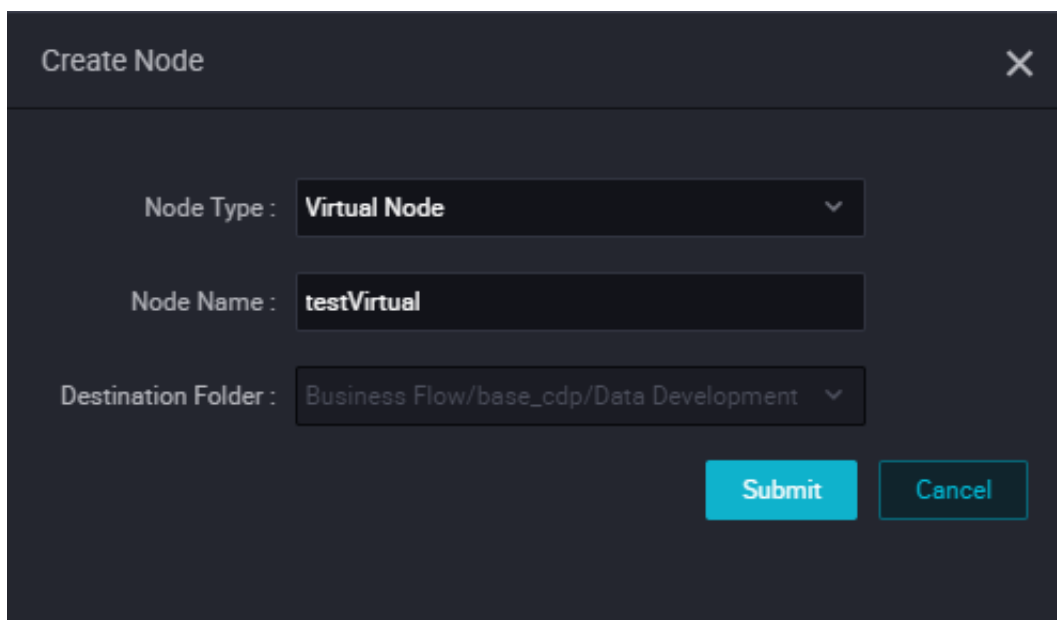
Click **Manual Business Flow** in the left-side navigation pane, select **Create Business Flow**.



2. Create a virtual node. Right-click **Data Development**, and select **Create Data Development Node > Virtual Node**.



3. Set the node type to **Virtual Node**, enter the node name, select the target folder, and click **Submit**.



4. Edit the node code: You do not need to edit the code of a virtual node.
5. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

6. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

7. Publish a node task.

For more information about the operation, see Release management.

8. Test in the production environment.

For more information about the operation, see [Manual task](#).

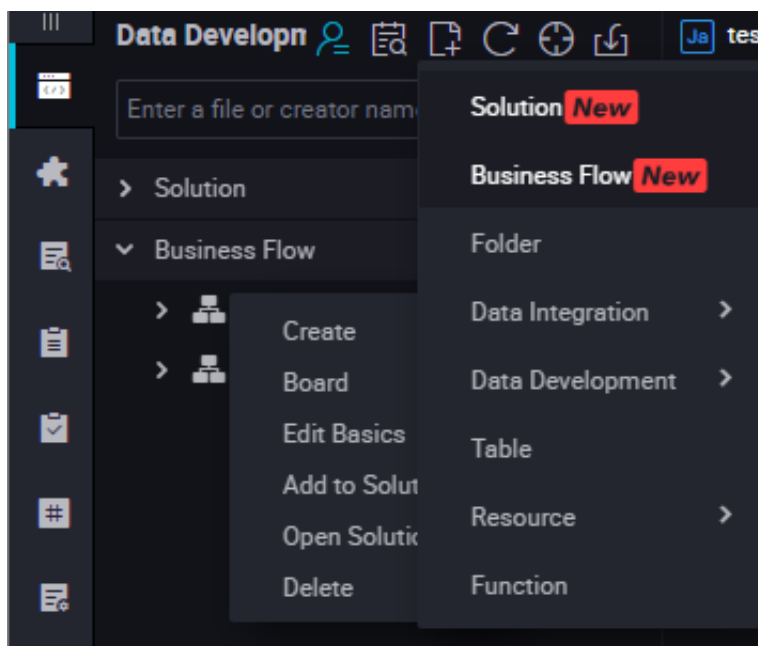
3.10.7 SHELL Node

SHELL tasks support standard SHELL syntax but not interactive syntax. SHELL task can run on the default resource group. If you want to access an IP address or a domain name, add the IP address or domain name to the whitelist by choosing Project Configuration.

Procedure

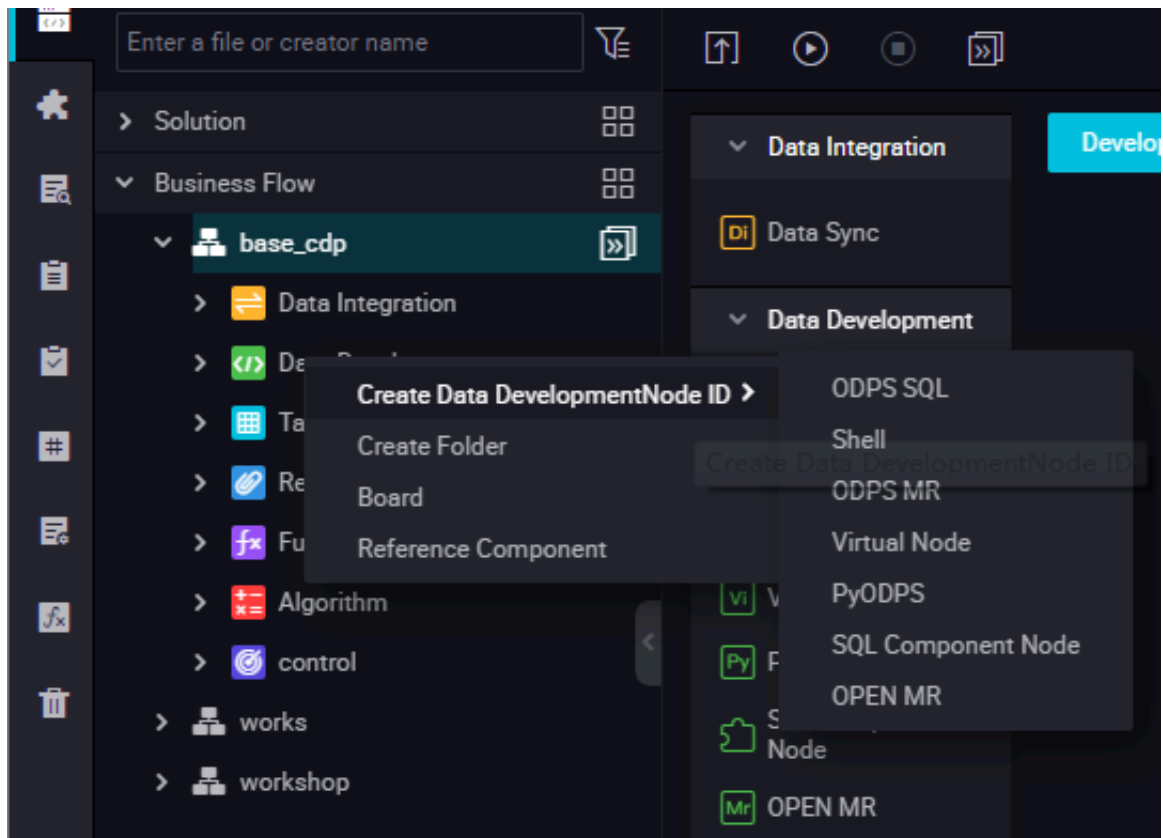
1. Create Business Flow

Click **Manual Business Flow** in the left-side navigation pane, select **Manual Business Flow**.



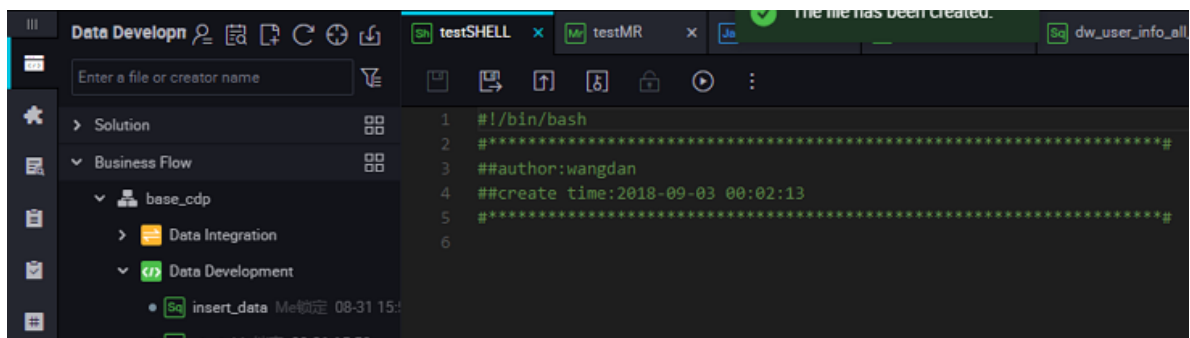
2. Create a SHELL node.

Right-click **Data Development**, and select **Create Data Development Node > SHELL**.



3. Set the node type to SHELL, enter the node name, select the target folder, and click **Submit**.
4. Edit the node code.

Go to the SHELL node code editing page and edit the code.



If you want to call the System Scheduling Parameters in a SHELL statement, compile the SHELL statement as follows:

```
echo "$1 $2 $3"
```



Note:

Parameter 1 Parameter 2... Multiple parameters are separated by spaces. For more information on the usage of system scheduling parameters, see [Parameter configuration](#).

5. Node scheduling configuration.

Click the **Schedule** on the right of the node task editing area to go to the **node scheduling configuration** page. For more information, see [Scheduling configuration](#).

6. Submit the node.

After the configuration is completed, click **Save** in the upper left corner of the page or press Ctrl +S to submit (and unlock) the node to the development environment.

7. Release a node task.

For more information about the operation, see Release management.

8. Test in the production environment.

For more information about the operation, see [Manual task](#).

Use cases

Connect to a database using SHELL

- If the database is built on Alibaba Cloud and the region is China (Shanghai), you must open the database to the following whitelisted IP addresses to connect to the database.

10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8



Note:

If the database is built on Alibaba Cloud but the region is not China (Shanghai), we recommend that you use the Internet or buy an ECS instance in the same region of the database as the scheduling resource to run the SHELL task on a custom resource group.

- If the database is built locally, we recommend that you use the Internet connection and open the database to the preceding whitelisted IP addresses.



Note:

If you are using a custom resource group to run the SHELL task, you must add the IP addresses of machines in the custom resource group to the preceding whitelist.

3.11 Manual task parameter settings

3.11.1 Basic Attributes

The figure below shows the basic attribute configuration interface:

Basics ⓘ

Node Name: insert_data Node ID:

Node Type: ODPS SQL Owner: TA

Description:

Parameters: bizdate=\$bizdate datetime=\${yyyyymmdd} ⓘ

- **Node Name:** It is the node name that you enter when creating a workflow node. To modify a node name, right-click the node on the directory tree and choose **Rename** from the short-cut menu.
- **Node ID:** It is the unique node ID generated when a task is submitted, and cannot be modified.
- **Node ID:** It is the unique node ID generated when a task is submitted, and cannot be modified.
- **Owner:** It is the node owner. The owner of a newly created node is the current login user by default. To modify the owner, click the input box, and enter the owner name or directly select another user.



Note:

When you select another user, the user must be a member of the current project.

- **Description:** It is generally used to describe the business and purpose of the node.
- **Parameter:** It is used to assign value to a variable in the code during task scheduling.

For example, when a variable "pt=\${datetime}" is used to indicate the time in the code, you can assign a value to the variable here. The assigned value can use the scheduling built-in time parameter "datetime=\$bizdate".

- **Resource Group:** It specifies the resource group for running the node.

Parameter value assignment formats for various node types

- **ODPS SQL, ODPSPL, ODPS MR, and XLIB types:** Variable name 1=Parameter 1 Variable name 2=Parameter 2..., Multiple parameters are separated by spaces.
- **SHELL type:** Parameter 1 Parameter 2..., Multiple parameters are separated by spaces.

Some frequently-used time parameters are provided as built-in scheduling parameters. For more information about these parameters, see [Parameter configuration](#).

3.11.2 Configure manual node parameters

To ensure that tasks can dynamically adapt to environment changes when running automatically at the scheduled time, DataWorks provides the parameter configuration feature. Pay special attention to the following issues before configuring parameters.

- No space can be added on either side of the equation mark "=" of a parameter. Correct:
bizdate=\$bizdate

The screenshot shows the 'Basics' configuration panel for a node named 'insert_data'. The 'Node Type' is 'ODPS SQL'. The 'Parameters' field contains the text 'bizdate=\$datetime'.

- Multiple parameters (if any) must be separated by spaces.

The screenshot shows the 'Basics' configuration panel for a node named 'insert_data'. The 'Parameters' field contains the text 'bizdate=\$bizdate datetime=\$\${yyyymmdd}'. A red arrow points to the space between '\$bizdate' and 'datetime', with the text 'Add spaces between the two parameters.'

System parameters

DataWorks provides two system parameters, which are defined as follows:

- `${bdp.system.cyctime}`: It is defined as the scheduled run time of an instance. Default format: `yyyymmddhh24miss`.
- `${bdp.system.bizdate}`: It is defined as the business date on which an instance is calculated. Default business data is one day before the running date, which is displayed in default format: `yyyymmdd`.

According to the definitions, the formula for calculating the runtime and business date is as follows: `Runtime = Business date - 1`.

To use the system parameters, directly reference '`${bizdate}`' in the code without setting system parameters in the editing box, and the system will automatically replace the reference fields of system parameters in the code.



Note:

The scheduling attribute of a periodic task is configured with a scheduled runtime. Therefore, you can backtrack the business date based on the scheduled runtime of an instance and retrieve the values of system parameters for the instance.

Example

Set an ODPS_SQL task that runs every hour between 00:00 and 23:59 every day. To use system parameters in the code, perform the following statement.

```
insert overwrite table tbl partition(ds ='20180606') select
c1,c2,c3
from (
select * from tb2
where ds ='${bizdate}');
```

Configure scheduling parameters for a non-Shell node



Note:

The name of a variable in the SQL code can contain only a-z, A-Z, numbers, and underlines. If the variable name is "date", the value "\${bizdate}" is automatically assigned to this variable, and you do not need to assign the value in the scheduling parameter configuration. Even if another value is assigned, this value is not used in the code because the value "\${bizdate}" is automatically assigned in the code by default.

For a non-Shell node, you need to first add \${variable name} (indicating that the function is referenced) in the code, then input a specific value to assign the value to the scheduling parameter.

For example, for an ODPS SQL node, add \${variable name} in the code, and then configure the parameter item "variable name=built-in scheduling parameter" for the node.

For a parameter referenced in the code, you must add the parsed value during scheduling.

```
1  --odps sql
2  --_*****_
3  --author:wangdan
4  --create time:2018-08-31 15:59:06
5  --_*****_
6  SELECT *
7  from testgong
8  WHERE ds='${bizdate}'
```

Configure scheduling parameters for a Shell node

The parameter configuration procedure of a Shell node is similar to that of a non-Shell node except that rules are different. For a Shell node, variable names cannot be customized and must be named '\$1,\$2,\$3...'.

For example, for a Shell node, the Shell syntax declaration in the code is: \$1, and the node parameter configuration in scheduling is: \$xxx (built-in scheduling parameter). That is, the value of \$xxx is used to replace \$1 in the code.

For a parameter referenced in the code, you must add the parsed value during scheduling.



```

1  #!/bin/bash
2  #*****#
3  ##author:
4  ##create time:2018-06-16 17:27:47
5  #*****#
6
7  echo $1

```



Note:

For a Shell node, when the number of parameters reaches 10, \${10} should be used to declare the variable.

The variable value is a fixed value

Take an SQL node for example. For \${variable name} in the code, configure the parameter item "variable name="fixed value"" for the node.

Code: select xxxxxx type='\${type}'

Value assigned to the scheduling variable: type="aaa"

During scheduling, the variable in the code is replaced by type='aaa'.

The variable value is a built-in scheduling parameter

Take an SQL node for example. For \${variable name} in the code, configure the parameter item variable name=scheduling parameter for the node.

Code: select xxxxxx dt='\${datetime}'

Value assigned to the scheduling variable: datetime=\$bizdate

During scheduling, if today is July 22, 2017, the variable in the code is replaced by dt=20170721.

Built-in scheduling parameter list

\$bizdate: business date in the format of `yyyymmdd` NOTE: This parameter is widely used, and is the date of the previous day by default during routine scheduling.

For example: In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$bizdate`. Today is July 22, 2017. When the node is executed today, `$bizdate` is replaced by `pt=20170721`.

For example, In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$gmtdate`. Today is July 22, 2017. When the node is executed today, `$gmtdate` is replaced by `pt=20170722`.

For example, In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$bizdate`. Today is July 1, 2017. When the node is executed today, `$bizdate` is replaced by `pt=20130630`.

For example, In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$gmtdate`. Today is July 1, 2017. When the node is executed today, `$gmtdate` is replaced by `pt=20170701`.

\$cyctime: scheduled time of the task. If no scheduled time is configured for a daily task, `cyctime` is 00:00 of the current day. The time is accurate to hour, minute, and second, and is generally used for a hour-level or minute-level scheduling task. Example: `cyctime=$cyctime`.



Note:

Pay attention to the difference between the time parameters configured using `[]` and `{}`.

\$bizdate: business date, which is one day before the current time by default. **\$cyctime:** It is the scheduled time of the task. If no scheduled time is configured for a daily task, the task is executed on 00:00 of the current day. The time is accurate to hour, minute, and second, and is generally used for an hour-level or minute-level scheduling task. If a task is scheduled to run on 00:30, for example, on the current day, the scheduled time is `yyyy-mm-dd 00:30:00`. If the time parameter is configured using `[]`, `cyctime` is used as the benchmark for running. For more information about the usage, see the instructions below. The time calculation method is the same with that of Oracle. During data population, the parameter value after replacement will be the business date + 1 day. For example, if the date of 20140510 is selected as the business date, the `cyctime` will be replaced by 20140511.

\$jobid: ID of the workflow to which a task belongs. Example: `jobid=$jobid`.

`$nodeid`: ID of a node. Example: `nodeid=$nodeid`.

`$taskid`: ID of a task, that is, ID of a node instance. Example: `taskid=$taskid`.

`$bizmonth`: business month in the format of `yyyymm`.

- If the month of a business date is equal to the current month, `$bizmonth` = Month of the business date - 1; otherwise, `$bizmonth` = Month of the business date.
- For example: In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=$bizmonth`. Today is July 22, 2017. When the node is executed today, `$bizmonth` is replaced by `pt=201706`.

`$gmtdate`: current date in the format of `yyyymmdd`. The value of this parameter is the current date by default. During data population, `gmtdate` that is input is the business date plus 1.

Custom parameter `${...}` Parameter description:

- Time format customized based on `$bizdate`, where `yyyy` indicates the 4-digit year, `yy` indicates the 2-digit month, `mm` indicates the month, and `dd` indicates the day. The parameter can be combined as expected, for example, `${yyyy}`, `${yyyymm}`, `${yyyymmdd}`, `${yyyy-mm-dd}`.
- `$bizdate` is accurate to year, month, and day. Therefore, the custom parameter `${.....}` can only represent the year, month, or day.
- Methods for obtaining the period plus or minus certain duration:

Next N years: `${yyyy+N}`

Previous N years: `${yyyy-N}`

Next N months: `${yyyymm+N}`

Previous N months: `${yyyymm-N}`

Next N weeks: `${yyyymmdd+7*N}`

Previous N weeks: `${yyyymmdd-7*N}`

Next N days: `${yyyymmdd+N}`

Previous N days: `${yyyymmdd-N}`

`${yyyymmdd}`: business date in the format of `yyyymmdd`. The value is consistent with that of `$bizdate`.

- Note: The value is consistent with that of `$bizdate`. This parameter is widely used, and is the date of the previous day by default during routine scheduling. The format of this parameter can be customized, for example, the format of `${yyyy-mm-dd}` is `yyyy-mm-dd`.

- For example: In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyymmdd}`. Today is July 22, 2013. When the node is executed today, `${yyyymmdd}` is replaced by `pt=20130721`.

`${yyyymmdd-/+N}`: `yyyymmdd` plus or minus `N` days

`${yyyymm-/+N}`: `yyyymm` plus or minus `N` month

`${yyyy-/+N}`: year (`yyyy`) plus or minus `N` years

`${yy-/+N}`: year (`yy`) plus or minus `N` years

NOTE: `yyyymmdd` indicates the business date and supports any separator, such as `yyyy-mm-dd`. The preceding parameters are derived from the year, month, and day of the business date.

Example:

- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyy-mm-dd}`. Today is July 22, 2018. When the node is executed today, `${yyyy-mm-dd}` is replaced by `pt=2018-07-21`.
- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyymmdd-2}`. Today is July 22, 2018. When the node is executed today, `${yyyymmdd-2}` is replaced by `pt=20180719`.
- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyymm-2}`. Today is July 22, 2018. When the node is executed today, `${yyyymm-2}` is replaced by `pt=201805`.
- In the code of the ODPS SQL node, `pt=${datetime}`. In the parameter configuration of the node, `datetime=${yyyy-2}`. Today is July 22, 2018. When the node is executed today, `${yyyy-2}` is replaced by `pt=2018`.

In the ODPS SQL node configuration, multiple parameters are assigned values, for example, `startdatetime=$bizdate enddatetime=${yyyymmdd+1} starttime=${yyyy-mm-dd} endtime=${yyyy-mm-dd+1}`.

Example: (Assume `$cyctime=20140515103000`)

- `${yyyy}` = 2014, `${yy}` = 14, `${mm}` = 05, `${dd}` = 15, `${yyyy-mm-dd}` = 2014-05-15, `${hh24:mi:ss}` = 10:30:00, `${yyyy-mm-dd hh24:mi:ss}` = 2014-05-1510:30:00
- `${hh24:mi:ss - 1/24}` = 09:30:00
- `${yyyy-mm-dd hh24:mi:ss -1/24/60}` = 2014-05-1510:29:00
- `${yyyy-mm-dd hh24:mi:ss -1/24}` = 2014-05-1509:30:00
- `${add_months(yyyymmdd,-1)}` = 2014-04-15

- `$(add_months(yyymmdd,-12*1)) = 2013-05-15`
- `$(hh24) =10`
- `$(mi) =30`

Method for testing the parameter `$cycetime`:

After the instance runs, right-click the node to **check the node attribute**. Check whether the scheduled time is the time at which the instance runs periodically.

Result after the parameter value is replaced by the scheduled time minus one hour.

3.12 Component management

3.12.1 Create components

Definition of components

A component is an SQL code process template containing multiple input and output parameters. To handle an SQL code process, one or more source data tables are imported, filtered, joined, and aggregated to form a target table required for new business.

Value of components

In actual businesses, many SQL code processes are similar. The input and output tables in a process have the same or compatible structures but different names. In this case, component developers can abstract such SQL process to an SQL component node, and variable input and output tables in the SQL process to input and output parameters to reuse the SQL code.

When using SQL component nodes, component users only need to select components like their own business flows from the component list, configure specific input and output tables in their own businesses for these components, and generate new SQL component nodes without repeatedly copying the code. This greatly improves the development efficiency and avoids repeated development. Publishing and scheduling of the SQL component nodes after generation is the same as those of common SQL nodes.

Composition of components

Like a function definition, a component consists of the input parameters, output parameters, and component code processes.

Component input parameters

A component input parameter contains the attributes such as the name, type, description, and definition. The parameter type can be table or string.

- A table-type parameter specifies tables to be referenced in a component process. When using a component, the component user can set the parameter to the table required for the specific business.
- A string-type parameter specifies variable control parameters in a component process. For example, if a result table of a specific process only outputs the sales amount of top N cities in each region, the value of N can be specified by the string-type parameter.

If a result table of a specific process needs to output the total sales amount of a province, a province string-type parameter can be set to specify different provinces and obtain the sales amount of the specified province.

- Parameter description specifies the role of a parameter in a component process.
- Parameter definition is a text definition of the table structure, which is required only for table-type parameters. When this attribute is specified, the component user must provide an input table that is compatible with the field names and types defined by the table parameter so that the component process can run properly. Otherwise, an error is reported when the component process runs because the specified field in the input table cannot be found. The input table must contain the field names and types defined by the table parameter. The fields and types can be in different orders, and the input table can also contain other fields. The definition is for reference only. It provides guidance for users and does not need to be immediately and forcibly checked.
- The recommended definition format of the table parameter is as follows:

```
Field 1 name Field 1 type Field 1 comment  
Field 2 name Field 2 type Field 2 comment  
Field n name Field n type Field n comment
```

Example:

```
area_id string 'Region ID'  
city_id string 'City ID'  
order_amt double 'Order amount'
```

Component output parameters

- A component output parameter contains the attributes such as the name, type, description, and definition. The parameter type can only be table. A string-type output parameter does not have the logical meaning.

- A table-type parameter: specifies tables to be generated from a component process. When using a component, the component user can set the parameter to the result table that the component process generates for the specific business.
- Parameter description: specifies the role of a parameter in a component process.
- Parameter definition: it is a text definition of the table structure. When this attribute is specified, the component user must provide the parameter with an output table that has the same number of fields and compatible type as defined by the table parameter so that the component process can run properly. Otherwise, an error is reported when the component process runs because the number of fields does not match or the type is incompatible. The field names of the output table do not need to be consistent with those defined by the table parameter. The definition is for reference only. It provides guidance for users and does not need to be immediately and forcibly checked.
- The recommended definition format of the table parameter is as follows:

```
Field 1 name Field 1 type Field 1 comment  
Field 2 name Field 2 type Field 2 comment  
Field n name Field n type Field n comment
```

Example:

```
area_id string 'Region ID'  
city_id string 'City ID'  
order_amt double 'Order amount'  
rank bigint 'Rank'
```

Component process bodies

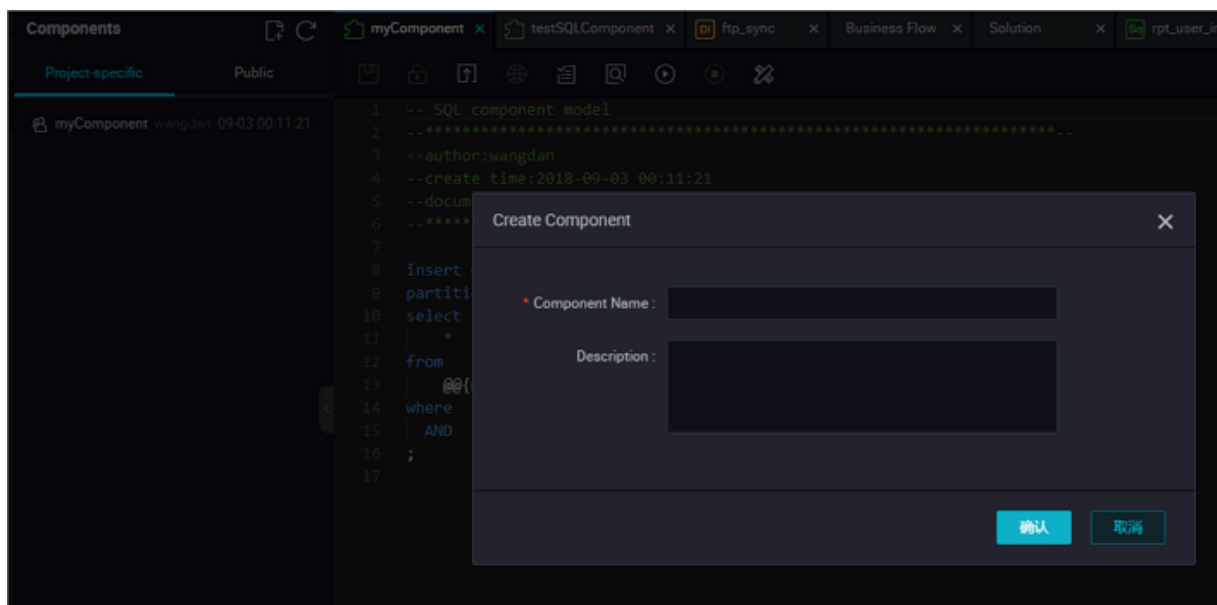
The reference format of the parameters in a process body is as follows: @@{parameter name}

By compiling an abstract SQL working process, the process body controls the specified input tables based on the input parameters and generates output tables with business value.

Certain skills are required for the development of a component process. Input parameters and output parameters must be well used for the component process code so that different values of input parameters and output parameters can generate correct and runnable SQL code.

Example of creating a component

You can create a component as shown in the following figure.



Source table schema definition

The source MySQL schema definition of the sales data is described in the following table:

Field Name	Field type	Field description
order_id	varchar	Order ID
report_date	datetime	Order date
customer_name	varchar	Customer Name
order_level	varchar	Order grade
order_number	double	Order quantity
order_amt	double	Order amount
back_point	double	Discount
shipping_type	varchar	Transportation mode
profit_amt	double	Profit amount
price	double	Unit price
shipping_cost	double	Transportation cost
area	varchar	Region
province	varchar	Province
city	varchar	City
product_type	varchar	Product Type
product_sub_type	varchar	Product subtype
product_name	varchar	Product Name

Field Name	Field type	Field description
product_box	varchar	Product packing box
shipping_date	Datetime	Transportation date

Business implication of components

Component name: get_top_n

Component description:

In the component process, the specified sales data table is used as the input parameter (table type), the number of the top cities is used as the input parameter (string type), and the cities are ranked by sales amount. In this way, the component user can easily obtain the rank of the specified top N cities in each region.

Definition of component parameters

Input parameter 1:

Parameter name: myinputtable type: table

Input parameter 2:

Parameter name: topn type: string

Input parameter 3:

Parameter name: myoutput type: table

Parameter definition:

area_id string

city_id string

order_amt double

rank bigint

Table creation statement:

```
CREATE TABLE IF NOT EXISTS company_sales_top_n
(
  area STRING COMMENT 'Region',
  city STRING COMMENT 'City',
  sales_amount DOUBLE COMMENT 'Sales amount',
  rank BIGINT COMMENT 'Rank'
)
COMMENT 'Company sales ranking'
PARTITIONED BY (pt STRING COMMENT '')
```

```
LIFECYCLE 365;
```

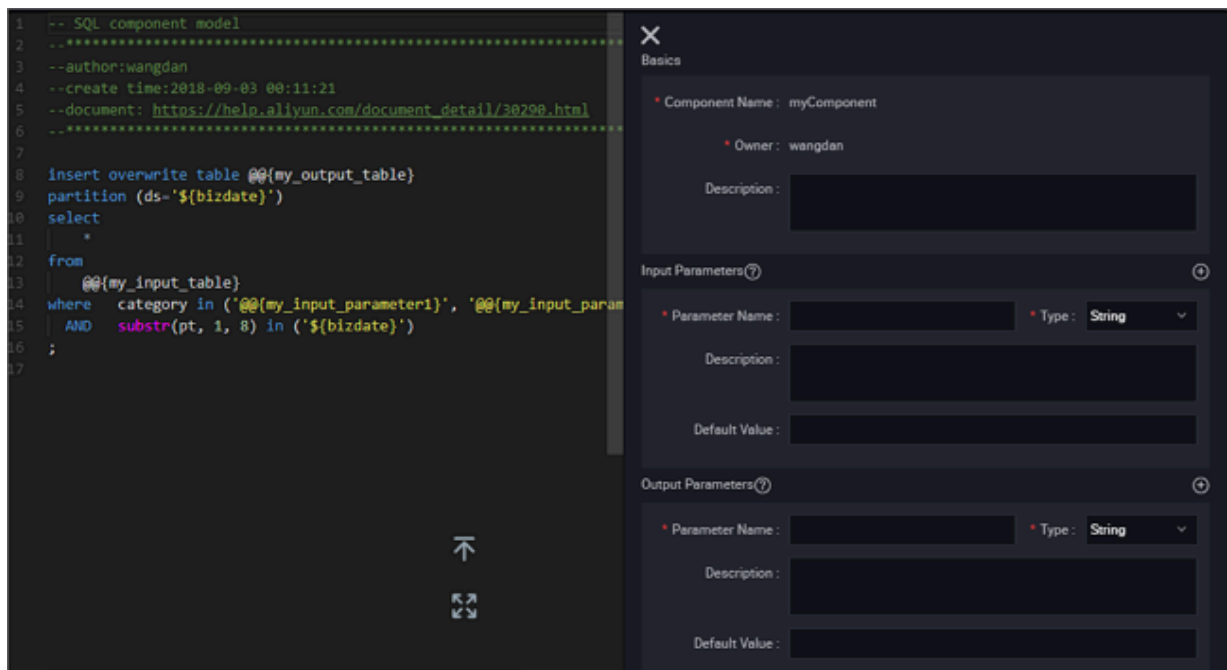
Definition of component process bodies

```
INSERT OVERWRITE TABLE @@{myoutput} PARTITION (pt='${bizdate}')
  SELECT r3.area_id,
  r3.city_id,
  r3.order_amt,
  r3.rank
from (
SELECT
  area_id,
  city_id,
  rank,
  order_amt_1505468133993_sum as order_amt ,
  order_number_1505468133991_sum,
  profit_amt_1505468134000_sum
FROM
  (SELECT
    area_id,
    city_id,
    ROW_NUMBER() OVER (PARTITION BY r1.area_id ORDER BY r1.order_amt_
1505468133993_sum DESC)
  AS rank,
    order_amt_1505468133993_sum,
    order_number_1505468133991_sum,
    profit_amt_1505468134000_sum
  FROM
    (SELECT area AS area_id,
      city AS city_id,
      SUM(order_amt) AS order_amt_1505468133993_sum,
      SUM(order_number) AS order_number_1505468133991_sum,
      SUM(profit_amt) AS profit_amt_1505468134000_sum
    FROM
      @@{myinputtable}
    WHERE
      SUBSTR(pt, 1, 8) IN ( '${bizdate}' )
    GROUP BY
      area,
      city )
    r1 ) r2
  WHERE
    r2.rank >= 1 AND r2.rank <= @@{topn}
  ORDER BY
    area_id,
    rank limit 10000) r3;
```

Sharing scope of components

There are two sharing scopes: project component and public component.

After a component is published, it is visible to users within the project by default. The component developer can click the Publish Component icon to publish a universal global component to the entire tenant, allowing all users in the tenant to view and use the public component. Whether a component is public depends on whether the icon in the following figure is visible:



Use of components

How can users use a developed component? For more information, see [Use components](#)

Reference records of components

The component developer can click the **Reference Records** tab to view the reference record of a component.

Project Name	Node ID	Node Name	Referenced Component Name	Owner	Create At	Development Version	Production Version	Parameters
No data								Version
<div><div><</div><div>1</div><div>></div></div>								Reference Records

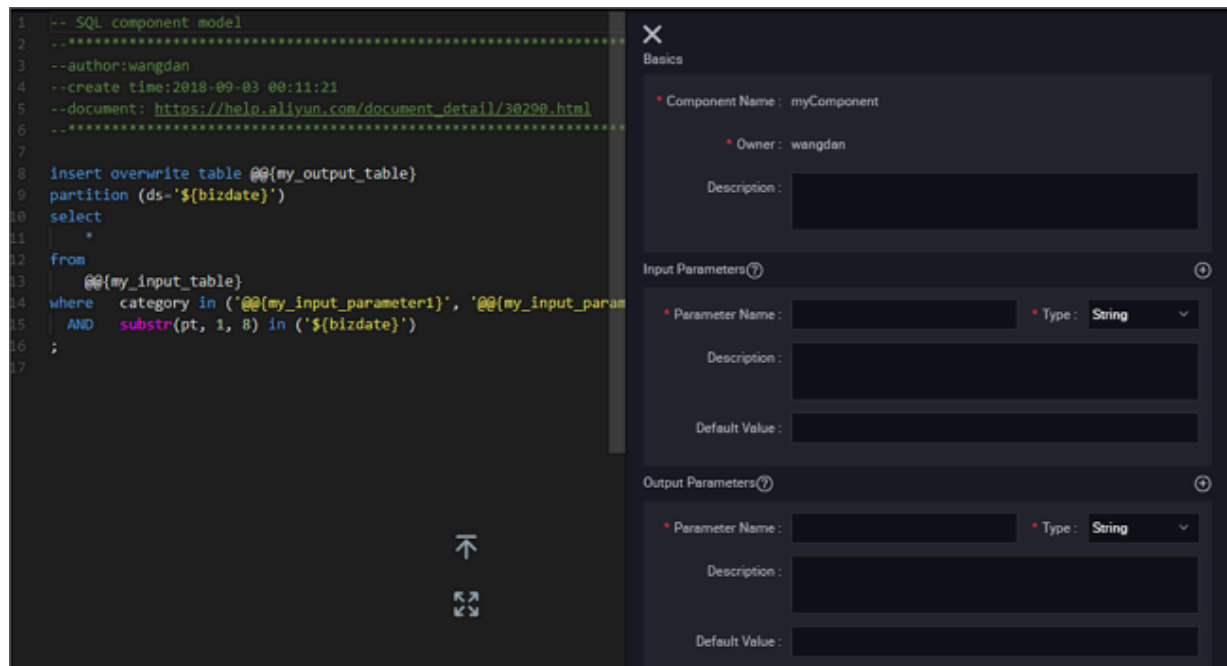
3.12.2 Use components

To improve the development efficiency, data task developers can use components contributed by project and tenant members to create data processing nodes.

- Components created by members of the local project are located under Project Components.
- Components created by tenant members are located under Public Components.

For more information about how to use the components, see [SQL Component node](#).

Interface functions



The interface functions are described below:

No.	Function	Description
1	Save	Click it to save settings of the current component.
2	Steallock Edit	Click it to steallock edit the node if you are not the owner of the current component.
3	Submit	Click it to submit the current component to the development environment.
4	Publish Component	Click it to publish a universal global component to the entire tenant, so that all users in the tenant can view and use the public component.
5	Resolve Input and Output Parameters	Click it to resolve the input and output parameters of the current code.

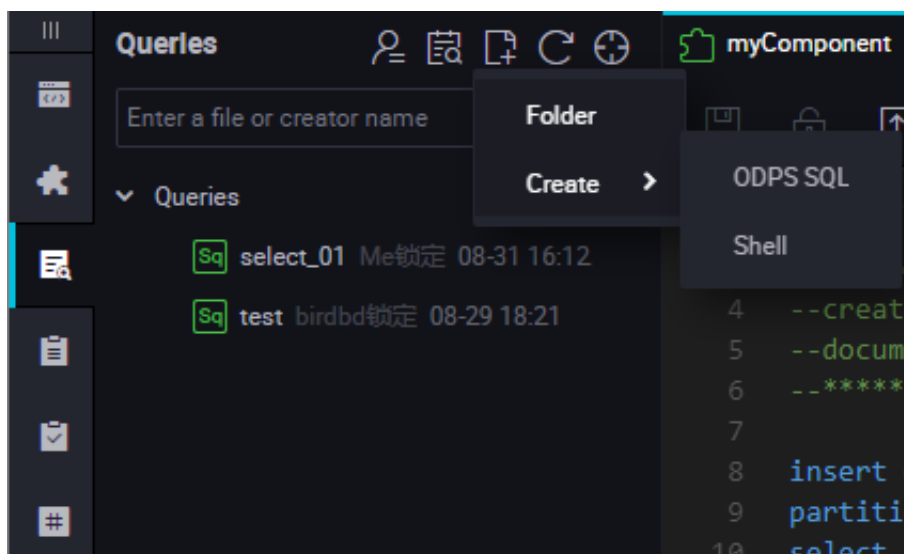
No.	Function	Description
6	Pre-compile	Click it to edit custom and component parameters of the current component.
7	Run	Click it to run the component locally in the development environment.
8	Stop Run	Click it to stop a running component.
9	Format	Click it to sort the current component code by keyword.
10	Parameter settings	Click it to view the component information, input parameter settings, and output parameter settings.
11	Version	Click it to view the submission and release records of the current component.
12	Reference Records	Click it to view the use record of the component.

3.13 Queries

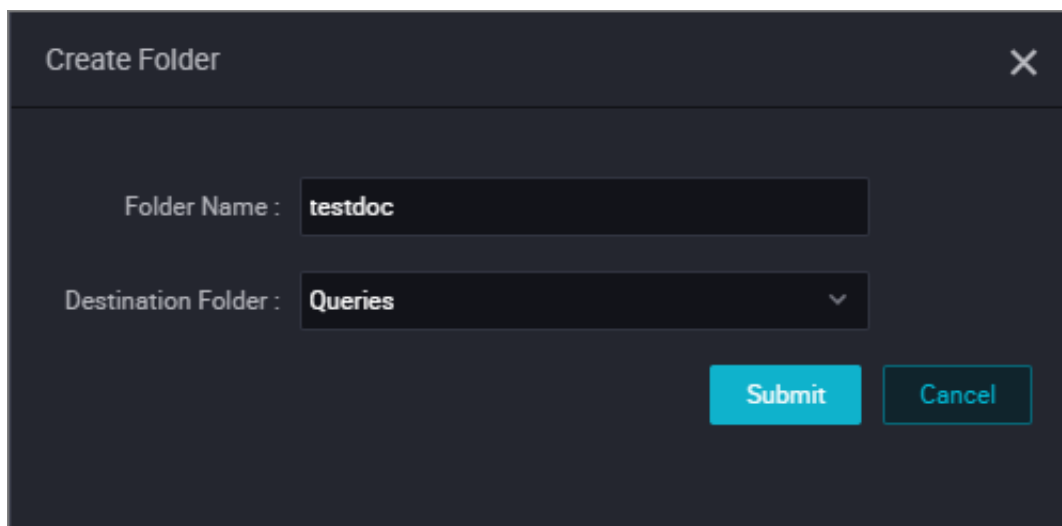
Temporary query facilitates you to use the editing code, test whether the actual conditions of the local code meets the expectations, and check the code status. Therefore, temporary query does not support submitting, releasing, and setting the scheduling parameters. To use the scheduling parameters, create a node in Data development or Manual business flow.

Create a folder

1. Click the **Queries** in the left-hand navigation bar, select **folder**.



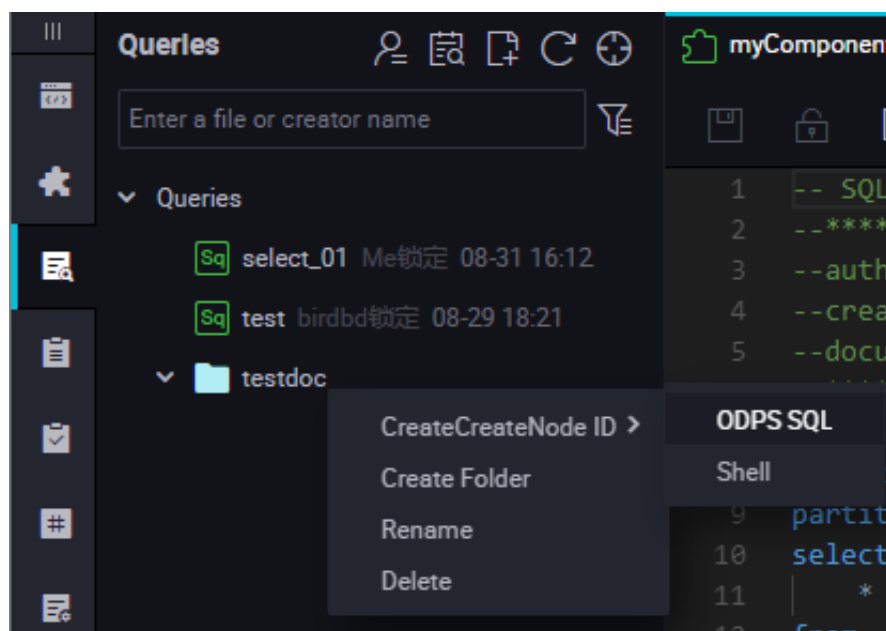
2. Enter the folder name, select the folder directory, and click **Submit**.

**Note:**

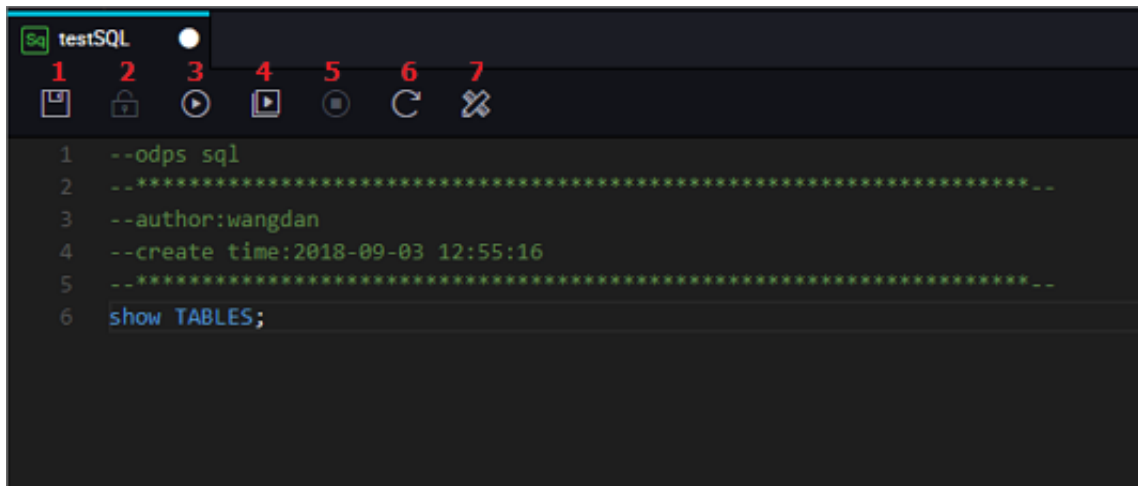
A multi-level folder directory is supported. Therefore, you can store the folder in another folder that has been created.



Create a node

Temporary query only supports the SHELL and SQL nodes.



Take the new ODPS SQL node as an example, right-click the folder name and select **Create Node > ODPS SQL**.



No.	Function	Description
1	Save	Click it to save the entered code.
2	Steallock Edit	A user other than the node owner can click it to edit the node.
3	Run	Click it to run the code locally (in the development environment).
4	Advanced Run (with Parameters)	Click it to run the code of the current node using the parameters configured for the code. <div>  Note: Advanced Run is unavailable to a Shell node. </div>
5	Stop Run	Click it to stop the code that is being run.
6	Reload	Click it to refresh the page, reload, and restore the last saved status. Unsaved content will be lost. <div>  Note: If the cache has been enabled in the configuration center, a message is displayed after page refreshing, indicating that the unsaved code has been cashed. Select a required version. </div>
7	Format	Click it to sort the current node code by keyword format. It is often used when a row of code is too long.

3.14 Running log

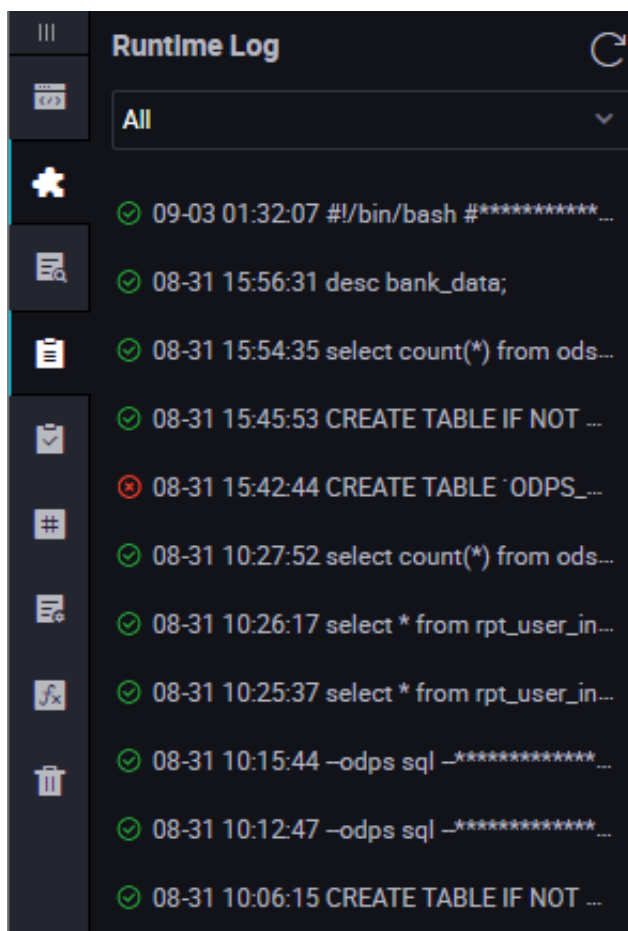
The Running Log page displays the record of all tasks that have locally run in the past three days. You can click it to view the task history and filter the running records by task status.

**Note:**

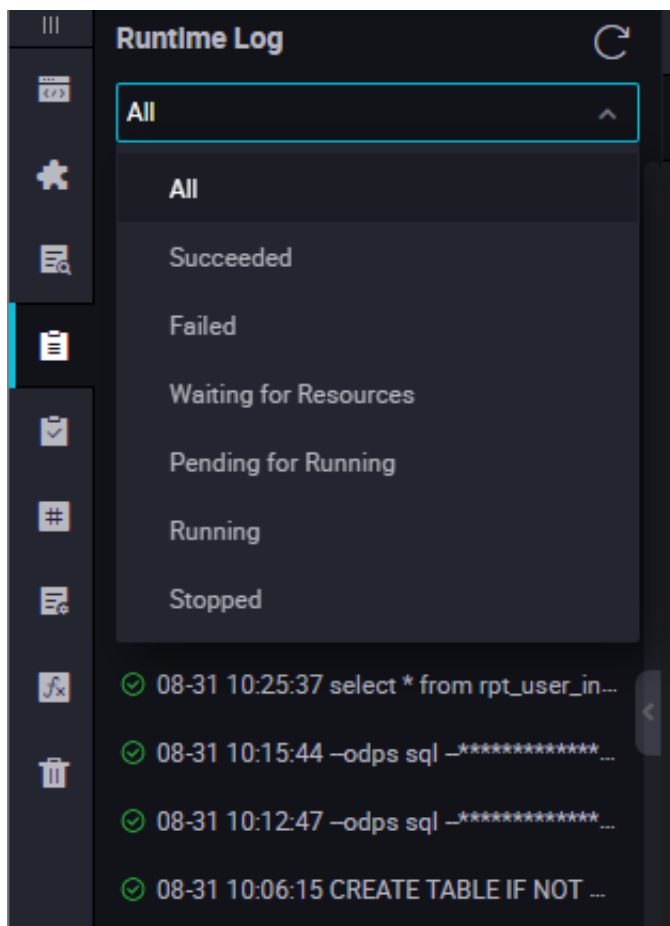
The Running Log is only retained for three days.

View the Running Log

1. Click to switch to the **Running Log** page (tasks in all status are displayed by default).



2. Click the drop-down list box and select the task filter criterion.



3. Click the target running record. The Running Log page displays the log of the running record.

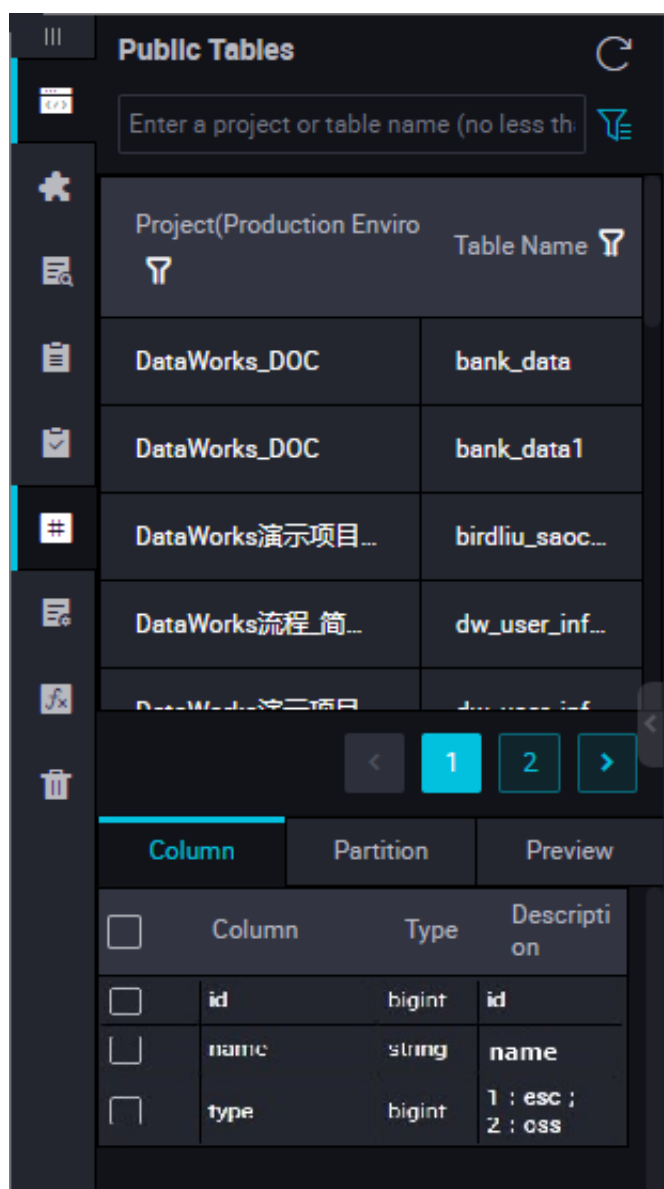
Save the log to a temporary file

To save the SQL statements in the running record, click the **Save** icon to save the SQL statements that have run to a temporary file.

Enter the file name and directory, and click **Submit**.

3.15 Public Tables

In the Public Table area, you can view tables created in all projects under the current tenant.



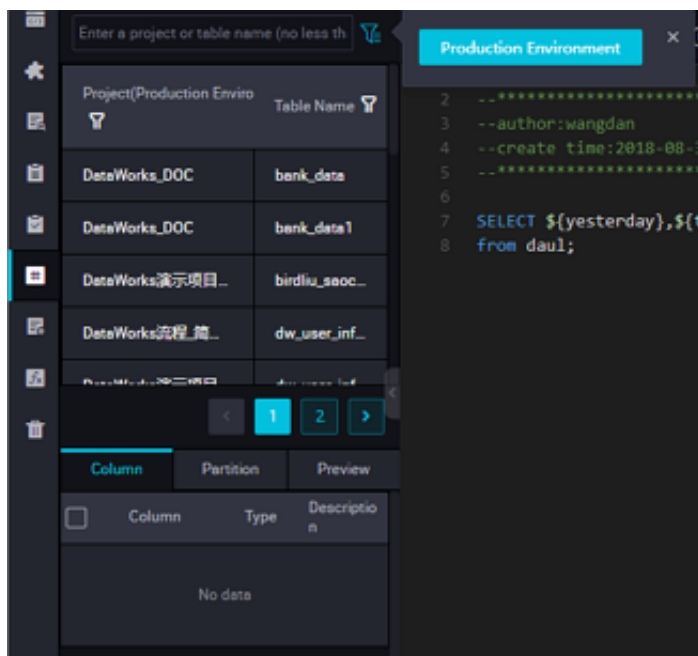
- **Project:** Project name. The prefix "odps." is added to each project name. For example, if a project name is "test", "odps.test" is displayed.
- **Table Name:** Name of the table in the project.

Click a table name to view the column and partition information of the table, and preview the table data.

- **Column Information:** Click it to view the field quantity, field type, and field description of the table.
- **Partition Information:** Click it to view the partition information and partition quantity of the table. A maximum of 60,000 partitions are allowed. If you have set the life cycle, the actual number of partitions depends on the life cycle.
- **Data Preview:** Click it to preview data in the current table.

Environment switchover

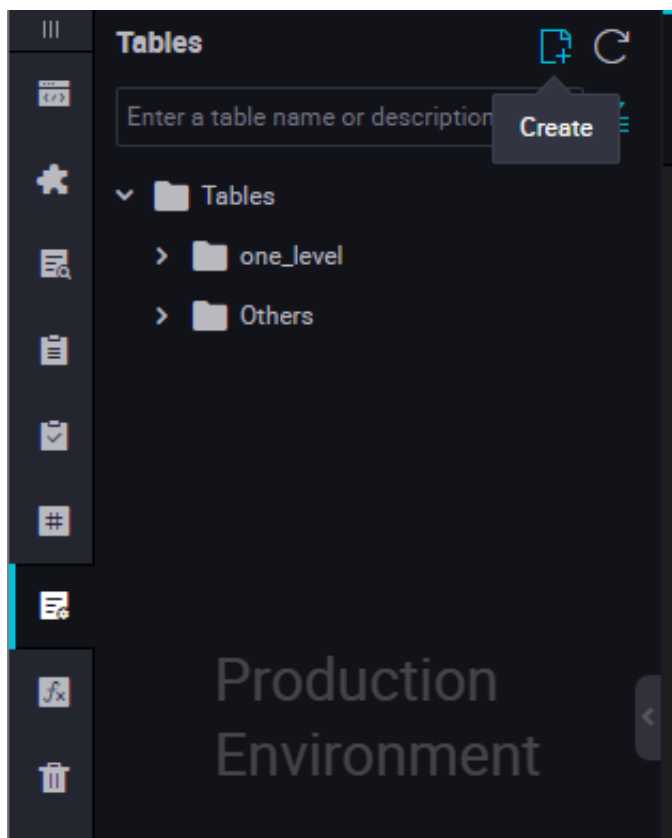
Similar to Table Management, Public Table supports the development and production environments. The current environment is displayed in blue. After you click an environment to be queried, the corresponding environment is displayed.



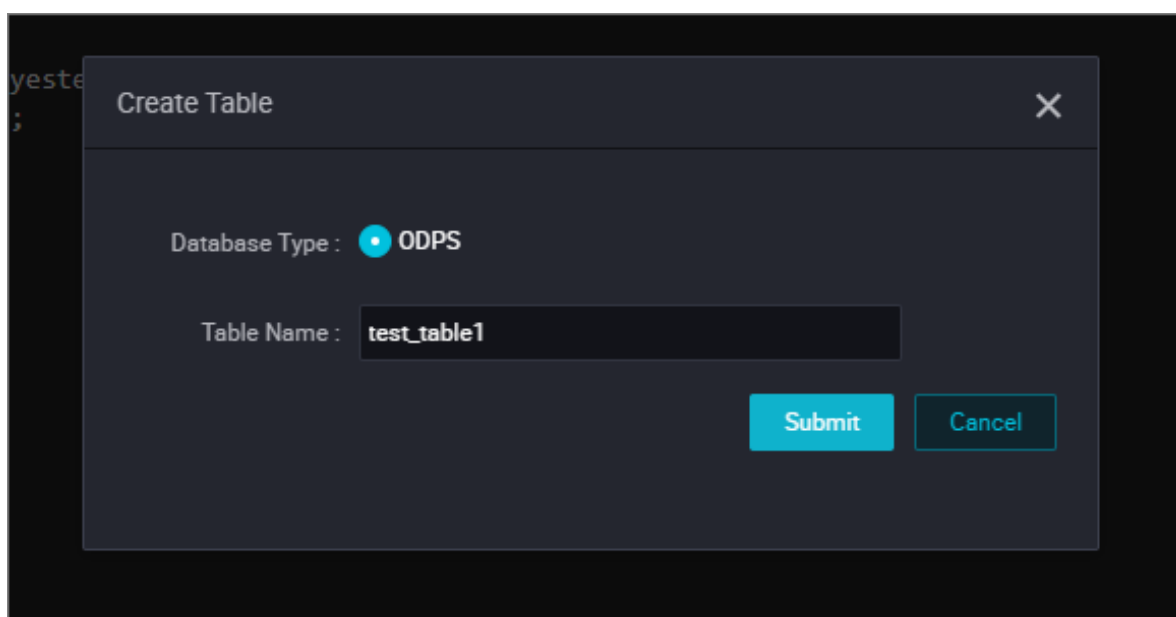
3.16 Table Management

Create a table

1. Click **Table Management** in the upper left corner of the page.
2. Select the **+** icon to create a table.



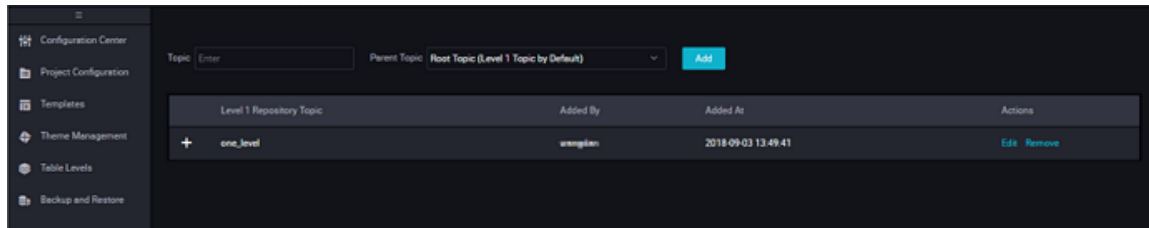
3. Enter the table name, only MaxCompute tables are supported currently, click **Submit**.



4. Set basic attributes.

- Chinese Name: Chinese name of the table to be created.
- Level-1 Topic: Name of the level-1 target folder of the table to be created.
- Level-2 Topic: Name of the level-2 target folder of the table to be created.
- Description: Description of the table to be created.

- Click **Create Topic**. On the displayed Topic Management page, create level-1 and level-2 topics.



5. Create a table in DDL mode.

Click **DDL Mode**. In the displayed dialog box, enter the standard table creation statements.

After editing the table creation SQL statements, click **Generate Table Structure**. Information in the Basic Attributes, Physical Model Design, and Table Structure Design areas is automatically entered.

6. Create a table on the GUI

If creating a table in DDL mode is not applicable, you can create the table on the GUI by performing the following settings.

- Physical model design
 - Table type: It can be set to Partitioned Table or Non-partitioned Table.
 - Life Cycle: Life cycle function of MaxCompute. Data in the table (or partition) that is not updated within a period specified by Life Cycle (unit: day) will be cleared.
 - Level: It can be set to DW, ODS, or RPT.
 - Physical Category: It can be set to Basic Business Layer, Advanced Business Layer, or Other. Click **Create Level**. On the displayed Level Management page, create a level.
- Table structure design
 - English Field Name: English name of a field, which may contain letters, digits, and underscores (_).
 - Chinese Name: Abbreviated Chinese name of a field.
 - Field Type: MaxCompute data type, which can only be String, Bigint, Double, Datetime, or Boolean.
 - Description: Detailed description of a field.
 - Primary Key: Select it to indicate the field is the primary key or a field in the joint primary key.
 - Click **Add Field** to add a column for a new field.
 - Click **Delete Field** to delete a created field.

**Note:**

If you delete a field from a created table and submit the table again, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.

- Click **Move Up** to adjust the field order of the table to be created. However, to adjust the field order of a created table, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.
- Click **Move Down**, the operation is the same as that of **Move Up**.
- Click **Add Partition** to create a partition for the current table. To add a partition to a created table, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.
- Click **Delete Partition** to delete a partition. To delete a partition from a created table, you must drop the current table and create one with the same name. This operation is not allowed in the production environment.
- Action: You can confirm to submit a new field, delete a field, and edit more attributes.

More properties mainly contain information related to data quality, which is provided for the system to generate validation logic. They are used in scenarios such as data profiling, SQL scan, and test rule generation.

- **0 Allowed:** If it is selected, the field value can be zero. This option is applicable only to bigint and double fields.
- **Negative Value Allowed:** If it is selected, the field value can be a negative number. This option is applicable only to bigint and double fields.
- **Security Level :** The security level is 0-4. The higher the number, the higher the security requirement. If your security level does not meet the digital requirements, you cannot access the corresponding fields in the form.
- **Unit:** Unit of the amount, which can be dollar or cent. This option is not required for fields unrelated to the amount.
- **Lookup Table Name/Kay Value:** It is applicable to enumerated value-type fields, such as the member type and status. You can enter the name of the dictionary table (or dimension table) corresponding to the field. For example, the name of the dictionary table corresponding to the member status is dim_user_status. If you use a globally unique dictionary table, enter the corresponding key_type of the field in the

dictionary table. For example, the corresponding key value of the member status is AOBABO_USER_STATUS.

- **Value Range:** The maximum and minimum values of the current field. It is applicable only to bigint and double fields..
 - **Regular Expression Verification:** Regular expression used by the current field. For example, if a field is a mobile phone number, its value can be limited to an 11-digit number by regular expression (or more strict limitation).
 - **Maximum Length:** Maximum number of characters of the field value. It is applicable only to string fields.
 - **Date Precision:** Precision of the date, which can be set to Hour, Day, Month, or others . For example, the precision of month_id in the monthly summary table is Month , although the field value is 2014-08-01 (it seems that the precision is Day). It is applicable to date values of the Datetime or String type.
 - **Date Format:** It is applicable only to date values of the string type. The format of the date value actually stored in the field is similar to yyyy-mm-dd hh:mm:ss.
 - **KV Primary Separator/Secondary Separator:** It is applicable to a large field (of the string type) combined by KV pairs. For example, if a product expansion attribute has a value similar to "key1:value1;key2:value2;key3:value3;...", the semicolon (;) is the primary separator of the field that separates the KV pairs, and the colon (:) is the secondary separator that separates the key and value in a KV pair.
- **Partition Field Design:** This option is displayed only when Partition Type in the Physical Model Design area is set to Partitioned Table.
 - **Field Type:** We recommend that you use the string type for all fields.
 - **Date Partition Format:** If a partition field is a date (although its data type may be string), select or enter a date format, such as yyyyymmdd.
 - **Date Partition Granularity:** For example, Day, Month, or Hour. Configure the partition granularity as per your needs. By default, if multiple partition granularities are required, the greater the granularity is, the higher the partition level is. For example, if three partitions (hour, day, and month) exist, the relationship among the multiple partitions is: level-1 partition (month), level-2 partition (day), and level-3 partition (hour).

Submit a table

After editing the table structure information, submit the new table to the development environment and production environment.

- Click **Load from Development Environment**. If the table has been submitted to the development environment, this button is highlighted. After you click the button, the information of the created table in the development environment overwrites the information on the current page.
- Click **Submit to Development Environment**. The system checks whether all required items on the current editing page are completely set. If any omission exists, an alarm is reported, forbidding you to submit the table.
- Click **Load from Production Environment**. The detailed information of the table submitted to the production environment overwrites the information on the current page.
- Click **Create in Production Environment**. The table is created in the project of the production environment.

Query tables by type

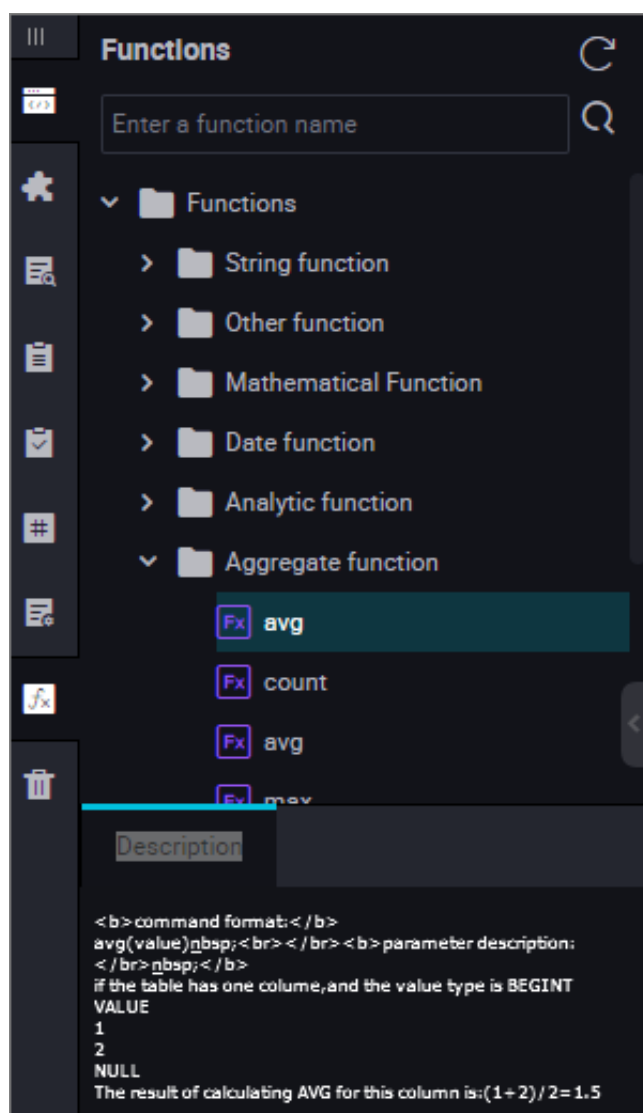
On the Table Management page, you can select Development Environment or Production Environment to query tables. The query results are sorted by folder of topics.

- If you select Development Environment, you can only query tables in the development environment.
- If you select Production Environment, you can query tables in the production environment. Be cautious when operating the tables in the production environment.

3.17 Functions

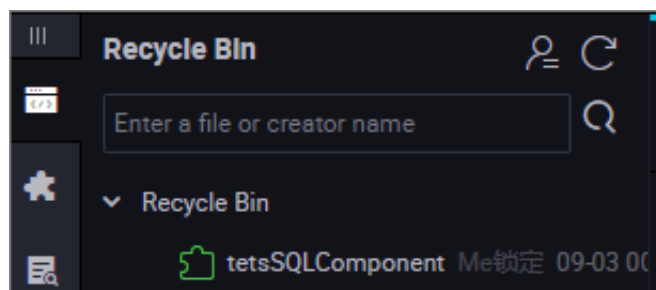
The function list provides the currently available functions, function classification, function usage description, and instances.

The function list contains six parts, including other functions, string processing functions, mathematical functions, date functions, window functions, and aggregate functions. These functions are provided by the system. You can view the description and example of a function by dragging the function.



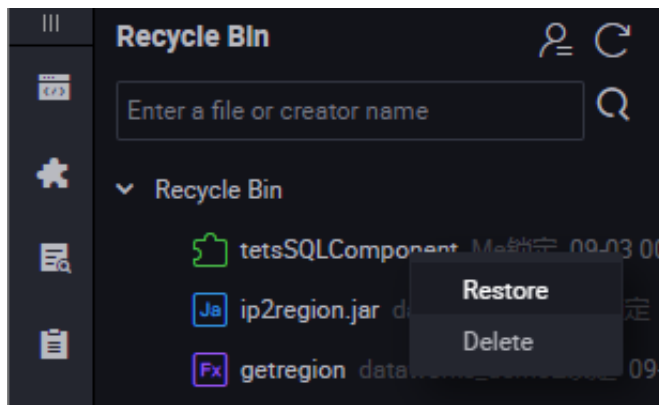
3.18 Recycle Bin

DataWorks has its own recycle bin, click **Recycle Bin** in the upper left corner of the page.



On the Recycle Bin page, you can check all deleted nodes in the current project. You can also right-click a node to restore or permanently delete it.

Click **Show My Files** on the right of the Recycle Bin page to view your deleted nodes.

**Note:**

If a node is permanently deleted from the recycle bin, it cannot be restored.

3.19 Editor shortcut list

Common shortcuts for code editing.

Windows chrome version

`Ctrl + S` Save

`Ctrl + Z` Undo

`Ctrl + Y` Redo

`Ctrl + D` Select the same word

`Ctrl + X` Cut a row

`Ctrl+Shift+K` Delete a row

`Ctrl + C` Copy the current row

`Ctrl+i` Select a row

`Shift+Alt+Dragging with the mouse` Column mode editing, modifying all the contents in this part

`Alt + mouse` Click multi-column mode edit, multi-line indents

`Ctrl + Shift + L` Add a cursor for all the identical string instances, batch changes

`Ctrl + F` Find

`Ctrl + H` Replace

`Ctrl + G` Locate to a specified row

`Alt + Enter` Select all the matching keywords in search

`Alt↓ / Alt↑` Move the current row down/up

`Shift + Alt + ↓ / Shift + Alt + ↑` Copy the current row down/up

`Shift + Ctrl + K` Delete the current row

`Ctrl + Enter / Shift + Ctrl + Enter` Move the cursor down/up

`Shift + Ctrl + \` Jump the cursor to the matching brackets

`Ctrl +] / Ctrl + [` Increase/decrease indent

`Home / End` Move to the beginning/end of the current row

`Ctrl + Home / Ctrl + End` Move to the beginning/end of the current file

`Ctrl + → / Ctrl + ←` Move the cursor right/left by words

`Shift + Ctrl + [/ Shift + Ctrl +]` Hide/Show block pointed by cursor

`Ctrl + K + Ctrl + [/ Ctrl + K + Ctrl +]` Hide/Show subblock pointed by cursor

`Ctrl + K + Ctrl + 0 / Ctrl + K + Ctrl + j` Fold/unfold all areas

`Ctrl + /` Write/Cancel comments for the row or code block where the cursor stays

MAC chrome version

`cmd + s` Save

`cmd + z` Undo

`cmd + y` Redo

`cmd + D` Select the same word

`cmd + x` Cut a row

`cmd + shift + K` Delete a row

`cmd + C` Copy the current row

`cmd + i` Select the current row

`cmd + F` Find

`cmd + alt + F` Replace

`alt↓ / alt↑` Move the current row down/up

`shift + alt + ↓ / shift + alt + ↑` Copy the current row down/up

`shift + cmd + ⌘`Delete the current row

`cmd + Enter / shift + cmd + Enter` Move the cursor down/up

`shift + cmd + \` Jump the cursor to the matching brackets

``cmd +] / cmd + [` Increase/decrease indent

`cmd + ← / cmd + →` Move to the beginning/end of the current row

`cmd + ↑ / cmd + ↓` Move to the beginning/end of the current file

`alt + → / alt + ←` Move the cursor right/left by words

`alt + cmd + [/ alt + cmd +]` Hide/Show block pointed by cursor

`cmd + K + cmd + [/ cmd + K + cmd +]` Hide/Show subblock pointed by cursor

`cmd + K + cmd + 0 / cmd + K + cmd + j` Fold/unfold all areas

`cmd + /` Write/Cancel comments for the row or code block where the cursor stays

Multiple cursors/select

`alt + Clicking with the mouse` Insert the cursor

`alt + cmd + ↑/↓` Insert the cursor up/down

`cmd + ⌘` Undo the last cursor operation

`shift + alt + I` Insert a cursor to the end of each row of the selected code block

`cmd + G / shift + cmd + G` Find the next/previous item

`cmd + F2` Select all the characters that the mouse has chosen

`shift + cmd + L` Select all the parts that the mouse has chosen

`alt+Enter` Select all the matching keywords in search

`shift + alt + Dragging with the mouse` Select multi-columns for editing

`shift + alt + cmd + ↑ / ↓` Move the cursor up/down to select multi-columns for editing

`shift + alt + cmd + ← / →` Move the cursor right/left to select multi-columns for editing

4 Operation center

4.1 Operation center overview

The Operation center offers four modules described as follows:

- O&M Overview

Overview makes a report presentation on the task running status.

- Task list

The Task List displays all the tasks submitted to the scheduling system, which are classified as **Cyclic Tasks** and **Manual Tasks**.

- Task Maintenance

This module displays the list of instances generated after a task is submitted to the scheduling system and then it is either triggered by the scheduling system or carried out manually. The instances are classified as **Cyclic Tasks**, **Test Instances** and **Data Completion Instances**.

- Alarm

[Alarm](#) monitors the running status of tasks. If a monitored task does not run as scheduled or fails, an alarm is generated and a notification is sent to the added contact.

Use cases

- The Operation Center is a place where tasks and instances are displayed and operated. You can view all your tasks in the Task List and perform operations on the displayed tasks, such as testing tasks and completing.
- In Task Maintenance, you can view the instances of all your tasks and terminate, re-run, or unfreeze the displayed instances.



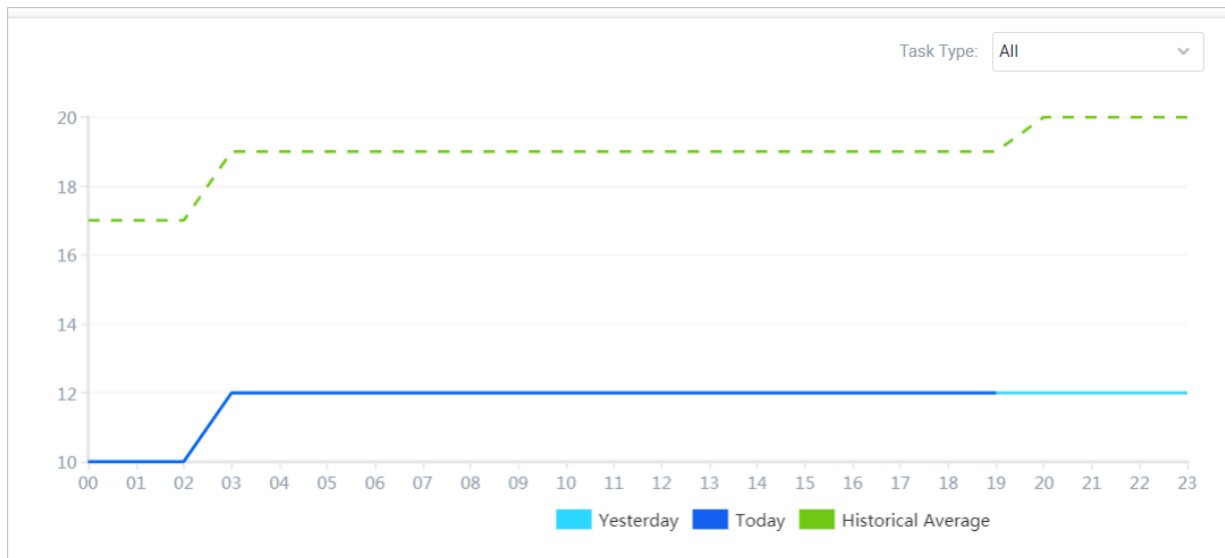
Note:

An instance is generated when a task in the scheduling system is triggered by the system or manually. An instance is a snapshot of a task at a certain time point, which includes the running time, status, and log of the task.

4.2 O&M overview

Task completion status

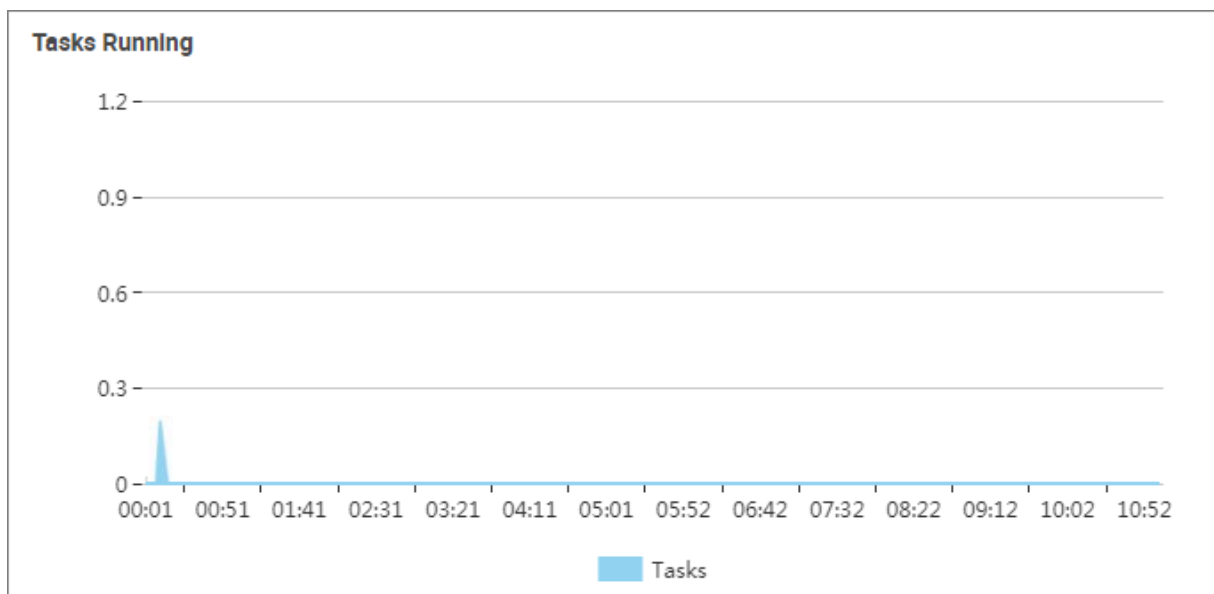
This module compares and generates statistical data for the completion of normal cyclic scheduling tasks for today, yesterday, and an average history level. If sharp misalignments occur between the three curves, it indicates exceptions within a certain period of time, and further checks and analyses are required as a result.



As shown in the above line statistics, three different color lines are displayed on the same day ~ Statistics on progress in the completion of all types of tasks in the current project space during the period of 24: 00, including today's completion of the task, yesterday's completion of the task and the history of the average level of completion.

Task running status

This section displays the number of currently running tasks by time. You can view the peak number of concurrent tasks at a certain point in time, and adjust the scheduled running time to avoid the concurrency peak.



Ranking of running durations of tasks

This section displays the ranking of running durations of tasks within the business period in the current project space. By default, the top ten tasks are displayed in descending order. The name, owner, and running duration of the task are displayed.

The tasks are displayed by business date. You can switch the business date to view the ranking of other dates.

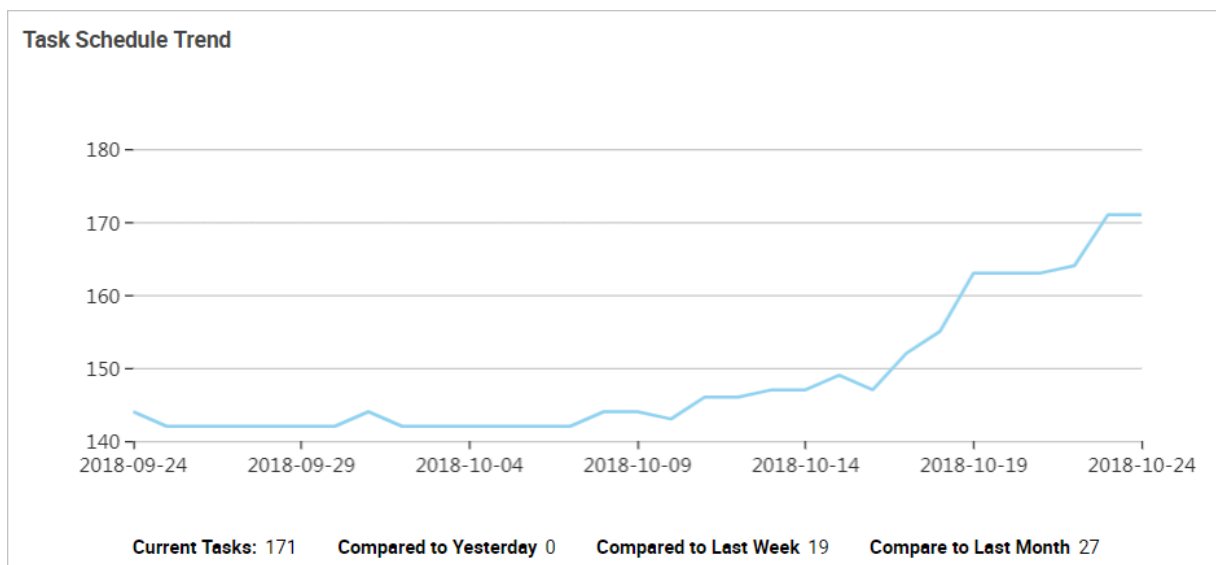
Ranking of failures in the last month

This section displays the top ten tasks with errors in the last month in descending order. You can view the task name, the owner and the occurrence of errors.

You can click a task name to jump to the details page of the task error history.

Trend in the number of scheduling tasks

This section displays the total number of current tasks and the task count changes compared with yesterday, last week, and last month. as shown in the following figure.



Task type distribution

Move the mouse over a sector to display the number and proportion of the task.

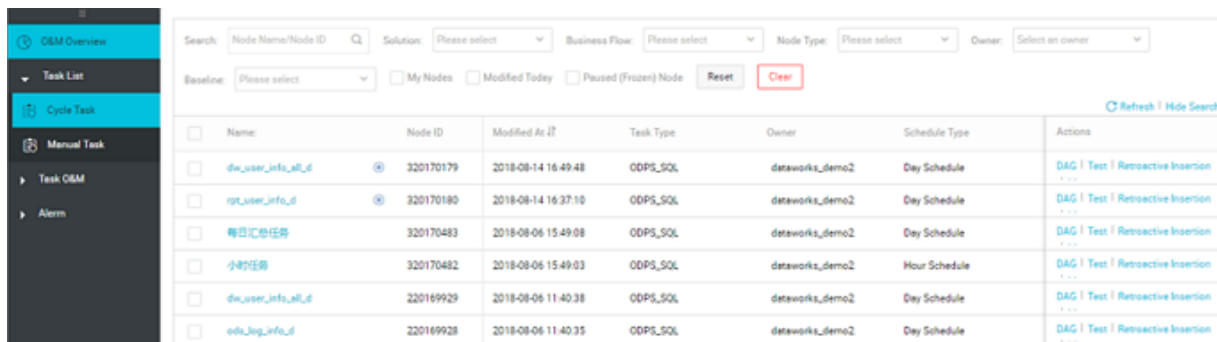


4.3 Task list

4.3.1 Cyclic task

Cyclic Task: Tasks automatically triggered by the scheduling system.

Click the **Cycle Task**, default display the current landing responsibility person node.

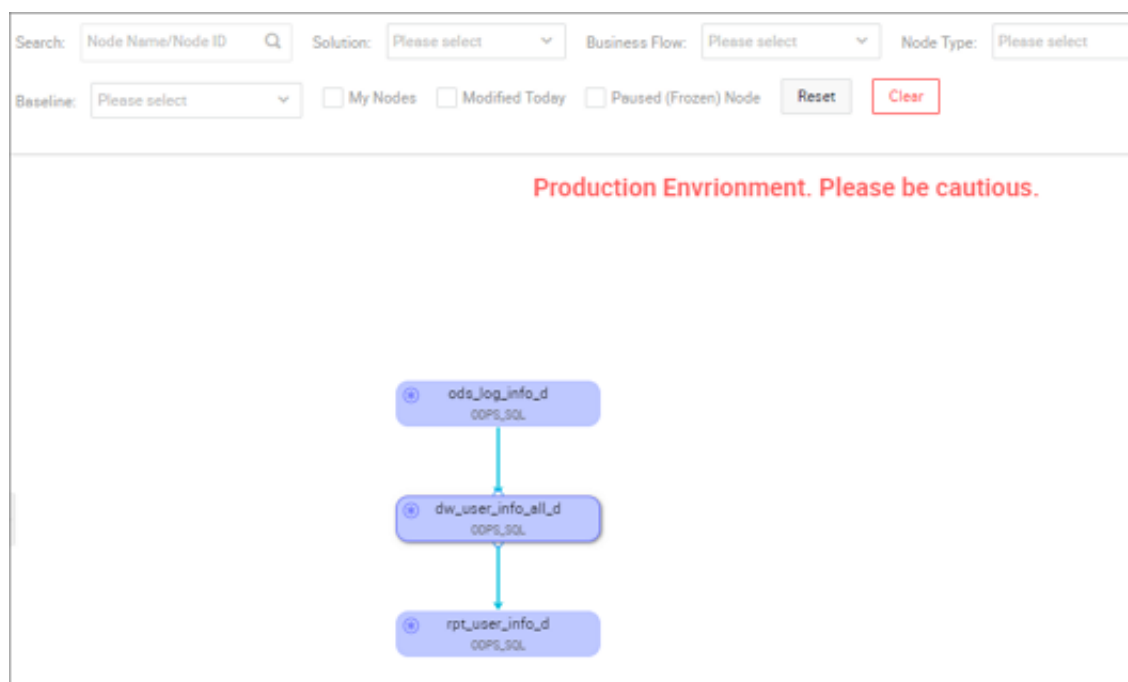


Name	Node ID	Modified At	Task Type	Owner	Schedule Type	Actions
dw_user_info_all_d	320170179	2018-08-14 16:49:48	ODPS_SQL	dataworks_demo2	Day Schedule	DAG Test Retroactive Insertion
rpt_user_info_d	320170180	2018-08-14 16:37:10	ODPS_SQL	dataworks_demo2	Day Schedule	DAG Test Retroactive Insertion
每日汇总任务	320170483	2018-08-06 15:49:08	ODPS_SQL	dataworks_demo2	Day Schedule	DAG Test Retroactive Insertion
小时任务	320170482	2018-08-06 15:49:03	ODPS_SQL	dataworks_demo2	Hour Schedule	DAG Test Retroactive Insertion
dw_user_info_all_d	220169929	2018-08-06 11:40:38	ODPS_SQL	dataworks_demo2	Day Schedule	DAG Test Retroactive Insertion
ods_log_info_d	220169928	2018-08-06 11:40:35	ODPS_SQL	dataworks_demo2	Day Schedule	DAG Test Retroactive Insertion

As shown in the figure above, task nodes can be filtered, providing name search, responsible person, baseline and other conditional search.


Default displays the name of the current task, modification date, task type, responsible person, scheduling type, resource group, alarm settings, operations. The operation button contains the following functions:

- DAG diagram: the DAG diagram of this node is displayed.

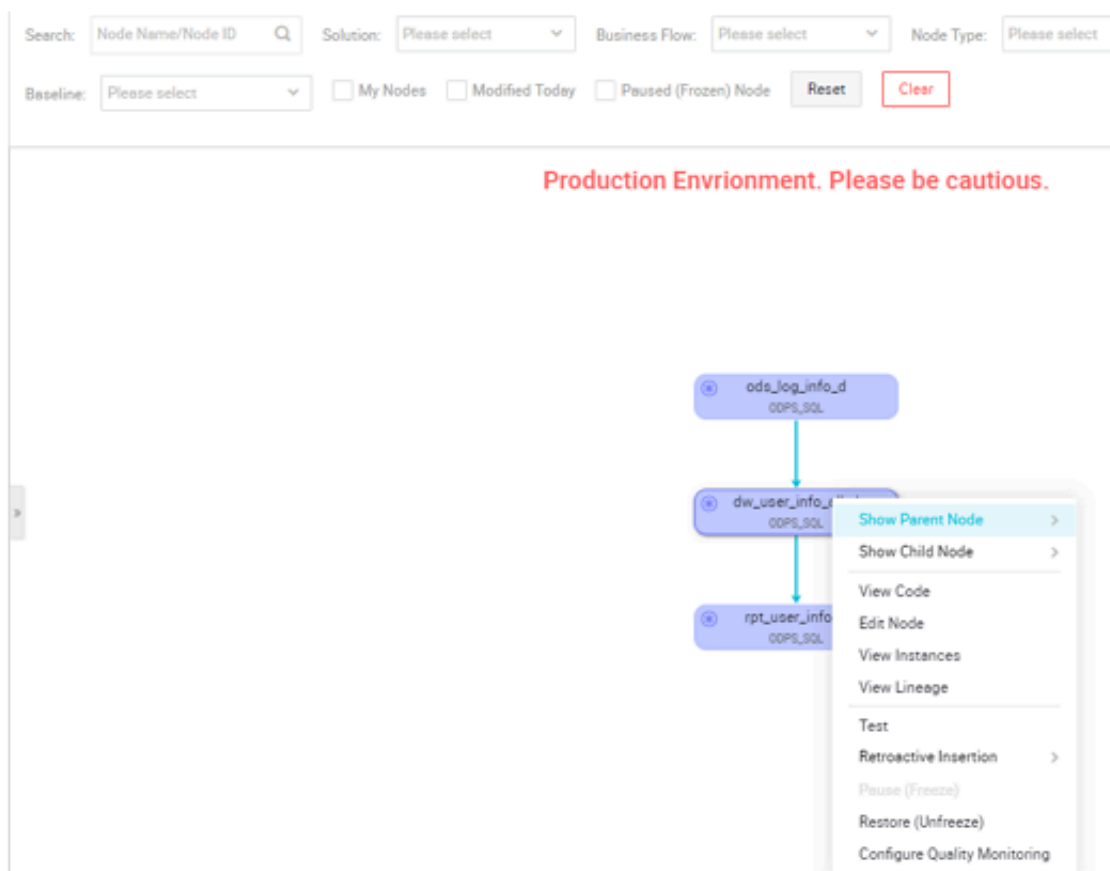


- Test: to test the current node.
- Data complement: data complement for the current node, see [Data completion instances](#).
- More: including node status modification and more functions.

More functions:

- Pause (freeze): Set the current node to a pause (freeze) state and stop scheduling. When the node state is paused, the  icon appears after the node name.
- Restore (thaw): restore the suspend (frozen) node to schedule.

- View instances: view the cycle instance of this node.
- Add alarm: configure alarm for node
- Modify the responsible person: modify the person responsible for the node
- Modify resource group: modify the resource group of nodes (if there are multiple resource groups in the project).
- Configuring quality monitoring: configuring DQC data quality and checking data.
- Look at blood ties: see the kinship map of the node.
- Upstream and downstream: this node in the DAG diagram, the right-click node will pop up the operable window. The detailed operation is as follows:



- Expanding parent / child nodes: When a workflow has three or more nodes, the operation and maintenance center will automatically hide the nodes when displaying tasks. Users can see more node dependencies by expanding the parent-child hierarchy. The larger the hierarchy, the more comprehensive the display.
- View node code: You can view the current code of the node.
- Edit nodes: You can jump to the page to edit the node.
- Testing: A prompt window pops up to edit the instance name and you can select the business date, which automatically jumps to the test instance page.

- Complement data: you can choose "include this node" and "include this node and downstream node".
- Pause (freeze): place the current node into a pause (freeze) state and stop scheduling.
- Restore (thaw): restore the suspend (frozen) node to schedule.
- View instances: view the cycle instance of this node.
- View kinship : see the kinship map of the node.

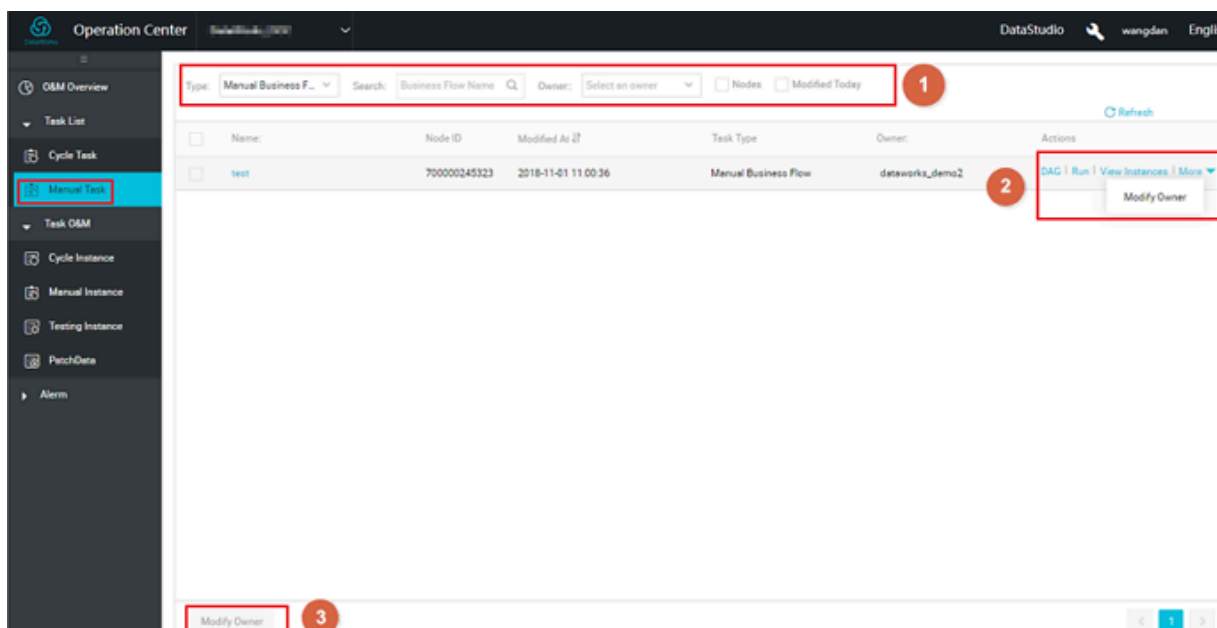
4.3.2 Manual task

Manual Task: Manual tasks do not run unless manually triggered.



Note:

- Manual tasks are submitted to the scheduling system and will not run automatically. Only manual triggers will run.
- The data under manual task is created in the old version of DataWorks. At present, the manual tasks created by users in the V2.0 version will be displayed under the **Manual Business Flow** options.



- DAG diagram: Click on the node name or DAG diagram, you can open the node's DAG diagram, DAG diagram click on the node can see the node's properties, operation log, code and other information.

Type: **Manual Business F...** Search: Owner: ☐ Nodes ☐ Modified Today

<input type="checkbox"/>	Name:	Node ID
<input type="checkbox"/>	test	700000245323

Production environment, please be cautious!

```

graph TD
    sh_1[sh_1 SHELL] --> testSQL[testSQL OOPS_SQL]
    testSQL --> test2SQL[test2SQL OOPS_SQL]
    testSQL --> rds[rds Data Integration]
    test2SQL --> ftyg[ftyg SHELL]
    rds --> ftyg
  
```

- Run: run this manual task to generate manual instances.
- View examples: jump to manual instance interface to see the result of manual task operation.
- More buttons contain two functions: modify the responsible person, modify the resource group.
 - Modify the responsible person: modify the node responsibility of this manual task.
 - Modify resource group: modify the resource group where this manual task is located.

In the DAG diagram, the right-click node will pop up the operable window. The detailed operation is as follows:

Type: **Manual Business F...** Search: Owner: ☐ Nodes ☐ Modified Today

<input type="checkbox"/>	Name:	Node ID
<input type="checkbox"/>	test	700000245323

Production environment, please be cautious!

```

graph TD
    sh_1[sh_1 SHELL] --> testSQL[testSQL OOPS_SQL]
    testSQL --> test2SQL[test2SQL OOPS_SQL]
    testSQL --> rds[rds Data Integration]
    test2SQL --> ftyg[ftyg SHELL]
    rds --> ftyg
  
```

- View node code: You can view the current code of the node.

- **Edit nodes:** You can jump to the page to edit the node.
- **View instances:** view the cycle instance of this node.
- **Look at blood ties:** see the kinship map of the node.
- **Run:** run this manual task to generate manual instances.

4.4 Task O&M

4.4.1 Cycle instance

Cycle instances are instance snapshots that are automatically scheduled when any cyclic task reaches the cyclic running time for scheduling.

One instance workflow is generated after each scheduling, which allows O&M management of scheduled instance tasks such as to view the running status and killing, re-running, and unfreezing tasks.

Instance list

The instance list provides operations and management for the tasks that have been scheduled in the form of a list. including checking running logs, re-running tasks, and killing running tasks.

OSM Overview

Task List

Cycle Task

Manual Task

Task OSM

Cycle Instance

Manual Instance

Testing Instance

PatchData

Alarm

Search: Node Name/Node ID

Business Date: yesterday The day before yesterday All 2018-10-10 2018-10-10

Node Type: Please select

☐ Nodes ☐ Error Nodes ☐ Unfinished Nodes

Refresh | Show Search

	Basic Information	Task Type	Owner	Priority	Timer	Business Date	Started	Actions
<input type="checkbox"/>	<div>movie_tr_score</div> <div>#700000420698 10-11 00:14:26 ~ 00:14:52 (dur 26s)</div>	ODPS_SQL		1	2018-10-11 00:08:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>start</div> <div>#700000420688 10-11 00:07:04 ~ 00:07:04 (dur 0s)</div>	Virtual Node		1	2018-10-11 00:07:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>create_dsl</div> <div>#700000420689 10-11 00:12:05 ~ 00:12:38 (dur 33s)</div>	ODPS_SQL		1	2018-10-11 00:12:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>user_rating_action</div> <div>#700000420690 10-11 00:13:23 ~ 00:13:46 (dur 23s)</div>	ODPS_SQL		1	2018-10-11 00:05:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>ftp数据同步</div> <div>#320170260 10-11 00:20:12 ~ 00:21:43 (dur 1m31s)</div>	Data Integration		1	2018-10-11 00:20:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>create_table_dsl</div> <div>#320170258 10-11 00:17:15 ~ 00:17:33 (dur 18s)</div>	ODPS_SQL		1	2018-10-11 00:17:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>test_sb12</div> <div>#700000420759 10-11 00:11:25 ~ 00:13:24 (dur 1m59s)</div>	Data Integration		1	2018-10-11 00:11:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>user_prefer_movie</div> <div>#700000420700 10-11 00:27:48 ~ 00:28:14 (dur 26s)</div>	ODPS_SQL		1	2018-10-11 00:26:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>ods_log_info_d</div> <div>#320170261 10-11 00:27:22 ~ 00:28:40 (dur 1m18s)</div>	ODPS_SQL		1	2018-10-11 00:27:00	2018-10-10	2018-10-10	DAG Terminate Rerun More
<input type="checkbox"/>	<div>movie_tag_score</div>							

1

2

3

Terminate




Rerun

Configured

Freeze

Unfreeze

Operation	Description
Filter	As the modules in the figure above, there are abundant Screening Conditions, the default filtering business date is a workflow task that is a day before the current time. You can add criteria such as Task Name, run time, owner, and so on for more precise filtering.
Terminate	It only applies to the instances in "Waiting" and "Running" statuses. If you perform this operation on an instance, the instance becomes "Failed".

Operation	Description
Rerun	<p>You can re-run a certain task. When the task is executed successfully, the scheduling of its downstream tasks that are not running can be triggered. This feature is often used for handling error nodes or missed nodes.</p> <div>  Note: Only tasks in the state of "Not Running", "Succeeded" and "Failed" can be re-run. </div>
Rerun Downstream	<p>It allows you to re-run the selected task and its downstream tasks. When the selected job re-runs successfully, scheduling can be triggered for its downstream jobs in the "Not Running" status. It is usually used for data restoration.</p> <div>  Note: Prerequisite: Only a task in the Not Running, Succeeded, or Failed state can be selected. Otherwise, a prompt An ineligible node is selected is displayed and re-running is prohibited. </div>
Set as Succeeded	<p>It allows you to change the status of the current node to "Succeeded" and run the downstream tasks in the "Not Running" status. This feature is often used for handling error nodes.</p> <div>  Note: Only tasks in a failed state can be successful, and workflow tasks cannot be successful. </div>
Freeze	<p>the freeze in the cycle instance is directed only at the current instance and is in the running instance, the freeze operation has no practical effect and does not kill the running instance.</p>
Unfreeze	<p>You can unfreeze an instance of the frozen state.</p> <ul style="list-style-type: none"> • If the instance is not already running, the upstream task runs automatically after it has finished running. • If the upstream task runs, the task is directly set to fail, the instance needs to be rerun manually before it can run properly.
Bulk operation	<p>As in the module above, bulk operation includes: stop running, run again, make successful, freeze, unfreeze 5 features.</p>

Instance DAG Graph

Click the task name to view the instance DAG.

The screenshot displays the DataWorks O&M Overview interface. On the left, a sidebar contains navigation options: O&M Overview, Task List, Cycle Task, Manual Task, Task O&M, Cycle Instance, Manual Instance, Testing Instance, Patch Data, and Alarm. The main area shows a list of tasks with columns for Name, ID, and Time. A task named 'movie_tag_score' is highlighted. To the right, a detailed view of this task instance is shown, including its dependencies (movie_tag_score, user_rating_action, movie_tr_score, user_prefer_movie) and a context menu with options like 'Show Parent Node', 'Show Child Node', 'View Running Log', 'View Code', 'Edit Node', 'View Nodes Affected', 'View Lineage', 'More', 'Terminate', 'Rerun', 'Rerun Downstream', 'Configured', 'Run', 'Freeze', and 'Unfreeze'. A warning message 'Production environment, please be cautious!' is displayed at the top right.

- Right-click an instance, you can view the dependencies and details of this instance and perform specific actions such as stopping, rerunning, and so on.







Operation	Description
Show Parent Node/Child Node	<p>When a workflow has 3 nodes and above, nodes are automatically hidden when the operations center displays tasks, and you can expand the parent-child level, to see the contents of all nodes.</p>
View running log	It allows you to view the running logs of the task when the node is in the status of "Running", "Succeeded" or "Failed".
View Code	It allows you to view the code of the instance task.
Edit Node	You can jump to the data development page to edit the node.
View Lineage	see the kinship map of the node.
Terminate	Kill task, valid only for this instance
Rerun	Failed task or abnormal status task re-run instance.

Operation	Description
Rerun Downstream	It allows you to re-run the selected task and its downstream tasks. When the selected job re-runs successfully, scheduling can be triggered for its downstream jobs in the "Not Running" status. It is usually used for data restoration.
Configured	It allows you to change the status of the current node to "Succeeded" and run the downstream tasks in the "Not Running" status. This feature is often used for handling error nodes.
Freeze	the freeze in the cycle instance is directed only at the current instance and is in the running instance, the freeze operation has no practical effect and does not kill the running instance.
Unfreeze	You can unfreeze an instance of the frozen state.

- Double-click an instance to pop up task properties, run logs, operation logs, code, and so on.

View content	Description
Properties	the attributes of this node are described, including schedule type, status, time, and so on.
Running Log	this node is running or running log information.
Operations Log	The operation log for the node, including the records of node changes , replenishment data, and so on.
Code	Code edited by the node.

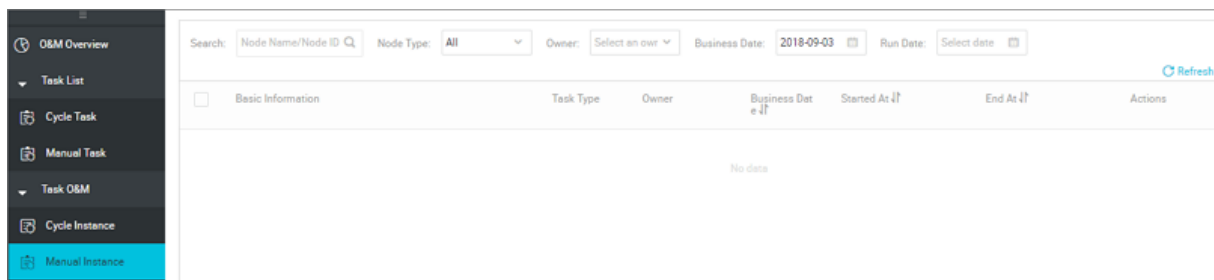
Description of instance status

SN	Status	State Mark
1	Running succeeded	
2	Not running	
3	Running failed	
4	Under running	
5	Waiting status	
6	Frozen status	

4.4.2 Manual instance

Manual instances are generated after a manual task is triggered, which allows O&M management of scheduled instance tasks such as viewing running status and killing and re-running tasks.

A manual instance, as the name implies, is an instance of a manual task, and a manual task is characterized by No scheduling dependency, you only need to trigger manually.



- Instance name/DAG graph: You can open the DAG graph for this node to view the results of the Instance run.
- Stop running: If the instance is running, click STOP to run the kill task.
- Re-run: re-schedule this instance.

Manual tasks have no dependencies, so the DAG graph only displays this instance, click the instance to see the properties, run log, operation log, code four columns. Right-click instance to see run log, code, edit node, view blood, terminate run, run again.

- Attributes: the attributes of this node are described, including schedule type, status, time, and so on.
- Run log: this node is running or running log information.
- Operation Log: The operation log for the node, including the records of node changes, replenishment data, and so on.
- Code: Code edited by the node.

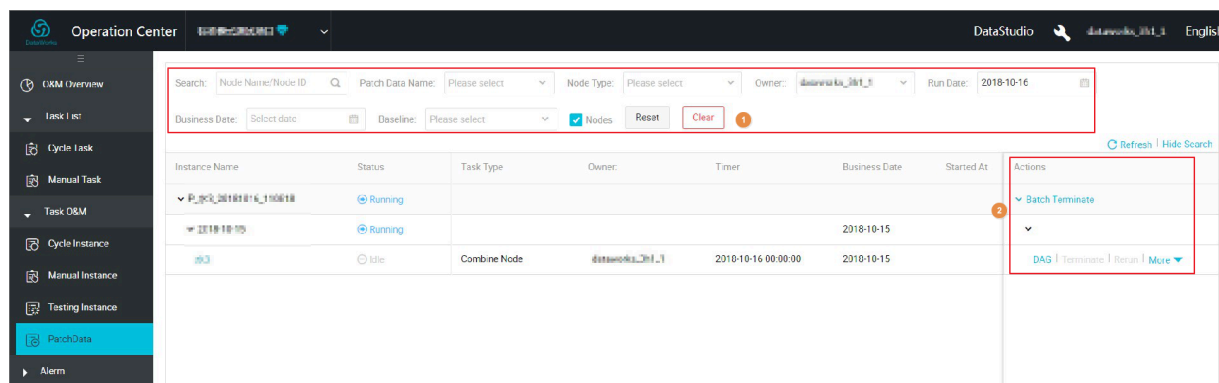
Introduction to the right-click node instance function:

- View running logs: Enter the Operations Log interface, where you can see information such as logview in the Operations Log.
- View node code: You can view the current code of the node.
- Edit nodes: You can jump to the data development page to edit the node.
- Look at blood ties: see the kinship map of the node.
- Stop operation: Kill task, valid only for this instance
- Re-run: Failed task or abnormal status task re-run instance.

4.4.3 PatchData


PatchData instances are generated during the completion of data for cyclic tasks, which allows O&M management of scheduled instance tasks such as viewing running status and terminating, re-running, and unfreezing tasks.

Instance list



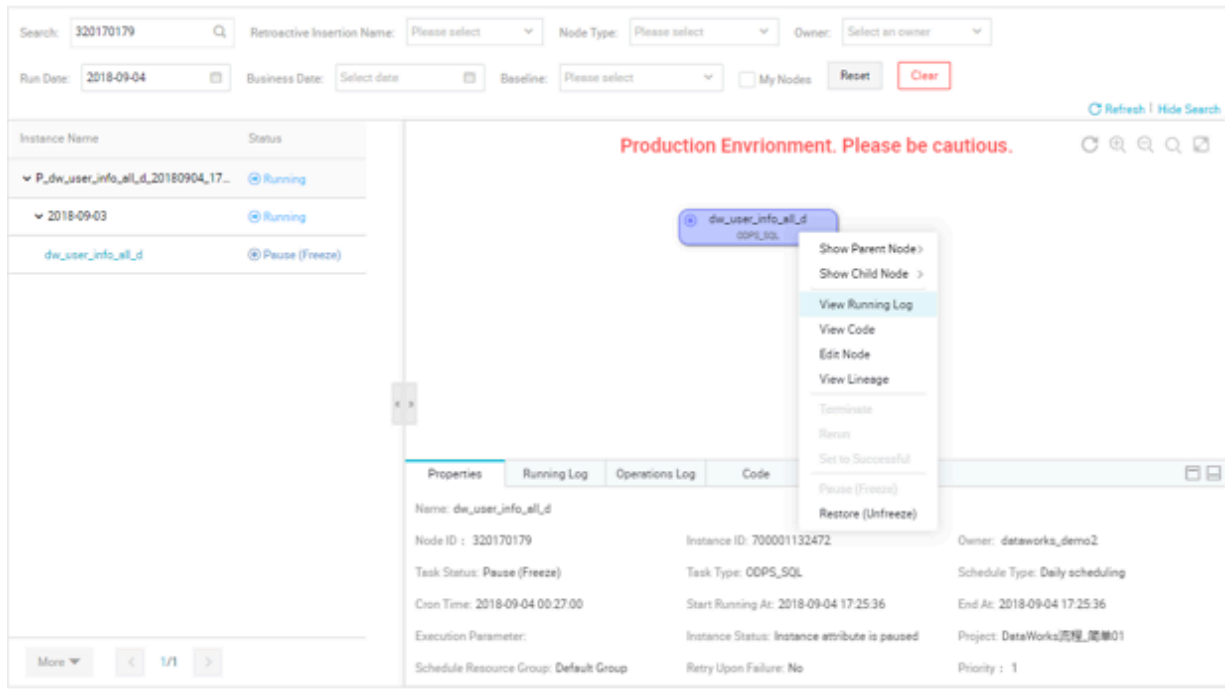
- Instance name/DAG graph: You can open the Dag graph for this node to view the results of the Instance run.
- Stop running: If the instance is running, click STOP to run the kill task.
- Re-run: re-schedule this instance.
- More: including node status modification and more functions.

Introduction to more features:

- Re-run downstream: re-run the downstream task for this node.
- Success: If the node fails to run, the node is successfully activated downstream.
- Pause (freeze): sets the current node to a pause (freeze) State and stops scheduling, when the node state is suspended, an icon  appears after the node name.
- Restore (thaw): restore the suspend (frozen) node to schedule.
- Look at blood ties: see the ki-nship map of the node.

Dag graph Introduction

Click the node name or dag map to open the Dag graph interface for this instance, right-click the node to see the operational features of this node.









- Attributes: the attributes of this node are described, including schedule type, status, time, and so on.
- Run log: this node is running or running log information.
- Operation Log: The operation log for the node, including the records of node changes, replenishment data, and so on.
- Code: Code edited by the node.

The right-click node function describes:

- View running logs: Enter the Operations Log interface, where you can see information such as logview in the Operations Log.
- View node code: You can view the current code of the node.
- Edit nodes: You can jump to the data development page to edit the node.
- View node impact: Enter the node information interface to view information such as baseline impact.
- Look at blood ties: see the kinship map of the node.
- Stop operation: Kill task, valid only for this instance
- Re-run: Failed task or abnormal status task re-run instance.
- Rerunning downstream: downstream rerunning instances of the current node, if there are multiple downstream instances, all of these instances will run again.
- Success: the node status is set to success.

- Emergency Operation: Emergency Operation refers to the operation of the current instance in a very urgent situation, emergency operations are only valid for the current node, including removing dependencies, modifying priorities, and forcing rerunning.
 - Remove dependencies: undependency this node, this node is often started when upstream fails and there is no data relationship to this instance.
 - Modify priority: Modify the priority of the current instance when the node is very important, used when running slowly (not recommended).
 - Force run again: ignores the status of the current instance and forces a restart (not recommended).
- Pause (freeze): place the current node into a pause (freeze) state and stop scheduling.
- Restore (thaw): restore the suspend (frozen) node to schedule.

Description of instance status

States
Mark
 ning
succeeded
 running
running
 ning
failed
 er
running
 ing
status
 zen
status

4.4.4 Testing instances

When the periodic task reaches the periodic run time configured to enable the modulation,, an instance snapshot that is automatically scheduled is a periodic instance. An instance workflow is generated at each scheduling. Daily O&M is performed for jobs on the started instance as scheduled, such as operations including viewing run statuses, or stopping, rerunning, or repairing a job,

Instance list

The instance list provides operations and management for the tasks that have been scheduled in the form of a list. including checking running logs, re-running tasks, and killing running tasks. The specific functions are described as follows:

Basic Information	Task Type	Owner	Timer ID	Business Date ID	Actions
<input type="checkbox"/> workshop_start #700000461343 09-12 00:05:13 ~ 00:05:13 (dur 0s)	Virtual Node	王丹	2018-09-12 00:05:00	2018-09-11	DAG Terminate Rerun More
<input type="checkbox"/> ftp_sync #700000461345 09-12 00:13:34 ~ 00:15:32 (dur 1m58s)	Data Integration	王丹	2018-09-12 00:12:00	2018-09-11	DAG Terminate Rerun More
<input type="checkbox"/> du_user_info_ddl #700000461554 ~ (dur 0s)	CCPS_SQL	王丹	2018-09-12 00:03:00	2018-09-11	DAG Terminate Rerun More
<input type="checkbox"/> ods_log_info_d #700000461553 09-12 00:15:41 ~ 00:19:12 (dur 3m31s)	CCPS_SQL	王丹	2018-09-12 00:11:00	2018-09-11	DAG Terminate Rerun More
<input type="checkbox"/> rpt_user_info_d #700000461555 ~ (dur 0s)	CCPS_SQL	王丹	2018-09-12 00:21:00	2018-09-11	DAG Terminate Rerun More
<input type="checkbox"/> rds_sync #700000461346 09-12 00:13:18 ~ 00:14:14 (dur 56s)	Data Integration	王丹	2018-09-12 00:11:00	2018-09-11	DAG Terminate Rerun More
<input type="checkbox"/> create_table_ddl #700000461344 09-12 00:11:44 ~ 00:12:40 (dur 56s)	CCPS_SQL	王丹	2018-09-12 00:11:00	2018-09-11	DAG Terminate Rerun More

Terminate Rerun Set to Successful Pause (Freeze) Restore (Unfreeze)

- **Filter Function:** As the modules in the figure above, there are abundant Screening Conditions, the default filtering business date is a workflow task that is a day before the current time. You can add criteria such as Task Name, run time, owner, and so on for more precise filtering.
- **Kill:** It only applies to the instances in "Waiting" and "Running" statuses. If you perform this operation on an instance, the instance becomes "Failed".
- **You can re-run a certain task.** When the task is executed successfully, the scheduling of its downstream tasks that are not running can be triggered. This feature is often used for handling error nodes or missed nodes.



Note:

Only tasks in the state of "Not Running", "Succeeded" and "Failed" can be re-run.

- **Re-run Downstream Tasks:** It allows you to re-run the selected task and its downstream tasks. When the selected job re-runs successfully, scheduling can be triggered for its downstream jobs in the "Not Running" status. It is usually used for data restoration.

**Note:**

You can only check tasks that are not running, completed, or failed. If you check tasks in other states, the page prompts the **selected node to contain nodes that do not meet the running conditions** and prohibits committing to run.

- **Set as Succeeded:** It allows you to change the status of the current node to "Succeeded" and run the downstream tasks in the "Not Running" status. This feature is often used for handling error nodes.

**Note:**

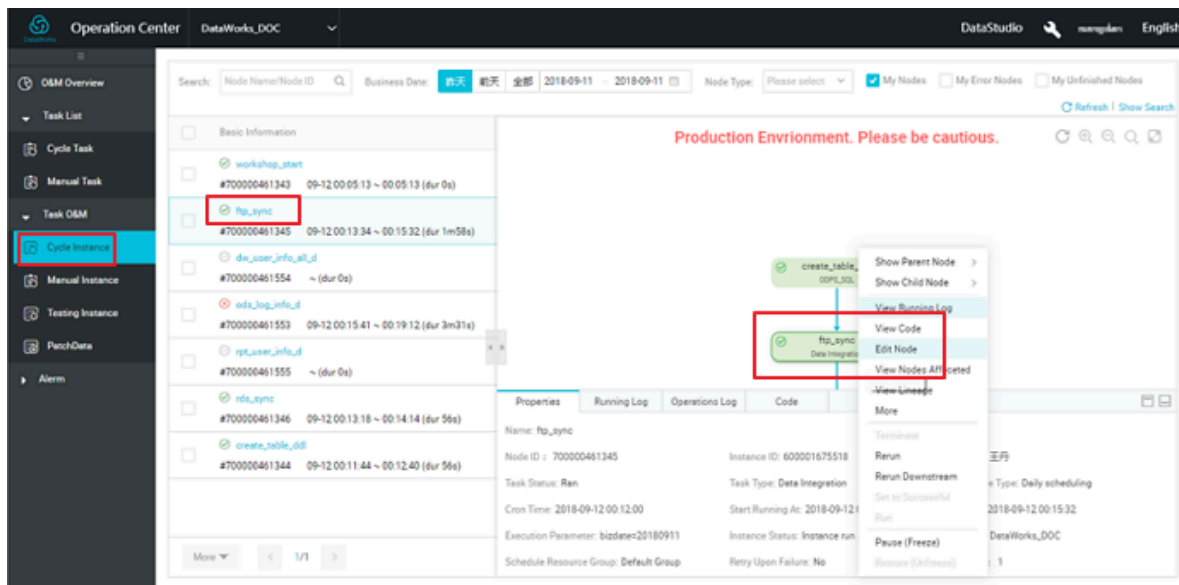
Only tasks in a failed state can be successful, and workflow tasks cannot be successful.

- **Freeze:** the freeze in the cycle instance is directed only at the current instance and is in the running instance, the freeze operation has no practical effect and does not kill the running instance.
- **Unfreezing:** You can unfreeze an instance of the frozen state.
 - If the instance is not already running, the upstream task runs automatically after it has finished running.
 - If the upstream task runs, the task is directly set to fail, the instance needs to be rerun manually before it can run properly.
- **Bulk operation:** As in the module above, bulk operation includes: stop running, run again, make successful, freeze, unfreeze features.

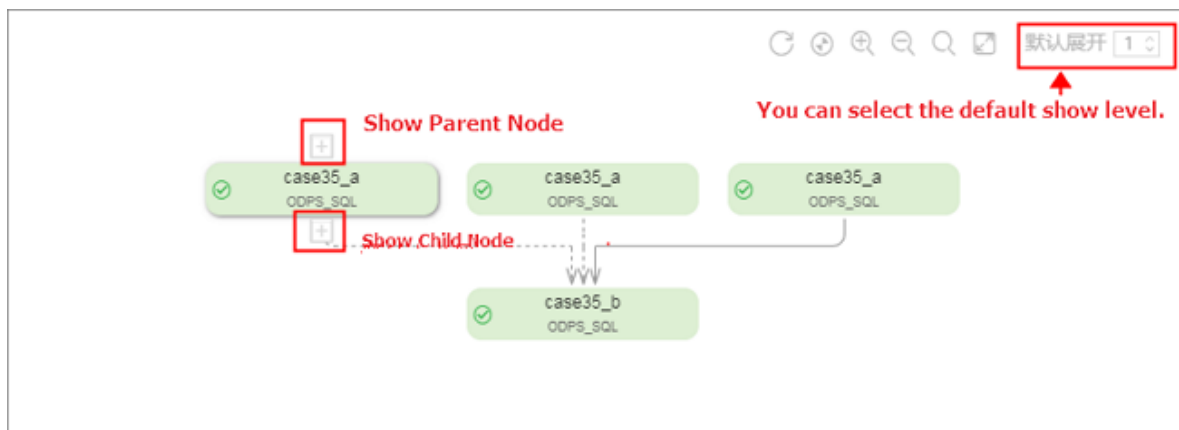
Instance DAG Graph

Click the task name to view the instance DAG. In the instance DAG View:

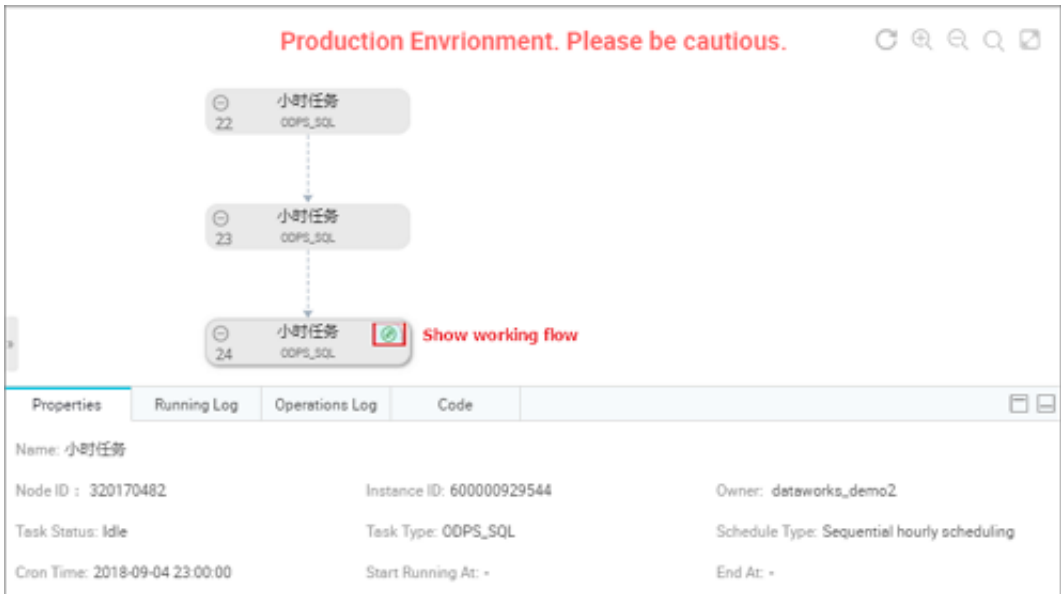
- Right-click an instance, you can view the dependencies and details of this instance and perform specific actions such as stopping, rerunning, and so on.
- Double-click an instance to pop up task properties, run logs, operation logs, code, and so on, as shown in the following figure:



- **Refresh node instance:** If you have modified the code or schedule parameters after the instance has been generated, you can click this button to use the latest code and parameters (bulk operations are not supported). **Use this function with caution because refreshing node instances is not refreshing the node status.**
- **Properties:** View instance properties, including various time information about the instance Run, Run Status, and so on.
- **View running log:** It allows you to view the running logs of the task when the node is in the status of "Running", "Succeeded" or "Failed".
- **Operational Log:** It records the operations performed on the instance, such as killing and re-running.
- **Code:** It allows you to view the code of the instance task.
- **Expand parent node/child node:** When a workflow has 3 nodes and above, nodes are automatically hidden when the operations center displays tasks, and you can expand the parent-child level, to see the contents of all nodes. As shown in the following illustration:



- **Expand/Close workflow:** When you have a workflow task, you can expand a workflow task, view the Run Status of the internal node task. As shown in the following illustration:



Description of instance status

States
Mark
ning
succeeded
running
ning
failed
er
running
ing
status
zen
status

4.5 Alarm

4.5.1 Alarm overview

Alarm is a monitoring and analysis system for the running of DataWorks tasks. Alarm, according to the monitoring rules and task running situation, determines whether, when, and how to report an alert as well as the object to which the alert is reported. Alarm automatically selects the most appropriate alert time, alert method, and alert object. Alarm aims to:

- Reduce the configuration costs for users.
- Prevent invalid alerts.
- Automatically cover all important tasks (the task quantity is beyond the handling capacity of users).

Conventional monitoring systems need users to configure relevant monitoring rules, which cannot meet the requirements of DataWorks because of the following reasons:

- DataWorks has considerable tasks, and users cannot accurately sort out the tasks that need to be monitored. Some DataWorks services involve thousands of tasks and the dependency between tasks is very complex. Even if you know what are the most important tasks, they have difficulties in figuring out all the upstream nodes of these tasks and putting them under monitoring. In this case, if you need to monitor all tasks, many invalid alerts may be triggered and valid alerts may be overlooked, which is equivalent to the absence of monitoring.
- The alert methods of monitored tasks are different: An alert is reported for some monitored tasks after they run for more than one hour, but is reported for other monitored tasks after they run for more than two hours. Therefore, it is very tedious to set the monitoring for each task separately, and users have difficulties to estimate the alert threshold of each task.
- The alert time of each monitored task is different: For example, an alert is reported after the work start time in the morning for unimportant tasks but is reported for important tasks immediately after they experience an exception. The importance of tasks cannot be differentiated.
- How to close alerts: If alerts are always present, an entry for closing such alerts must be available when users respond to the alerts.

Alarm has a set of alert monitoring logics. You need to only provide the names of important tasks about concerned services. Then, Alarm is capable of monitoring the output of all tasks comprehensively and defining a standard and unified alert mechanism. In addition, Alarm provides

the lightweight self-help configuration monitoring function, which allows you to define alert policies based on their requirements.

Currently, Alarm has undertaken the task monitoring of all important services of Alibaba Group. The full path monitoring function of Alarm secures the overall task output links of all important services of Alibaba Group. The upstream and downstream path analysis function enables Alarm to identify risks in a timely manner and provide O&M information for the Business Unit. With the analysis system of Alarm, Alibaba Group maintains high stability of services in the long term.

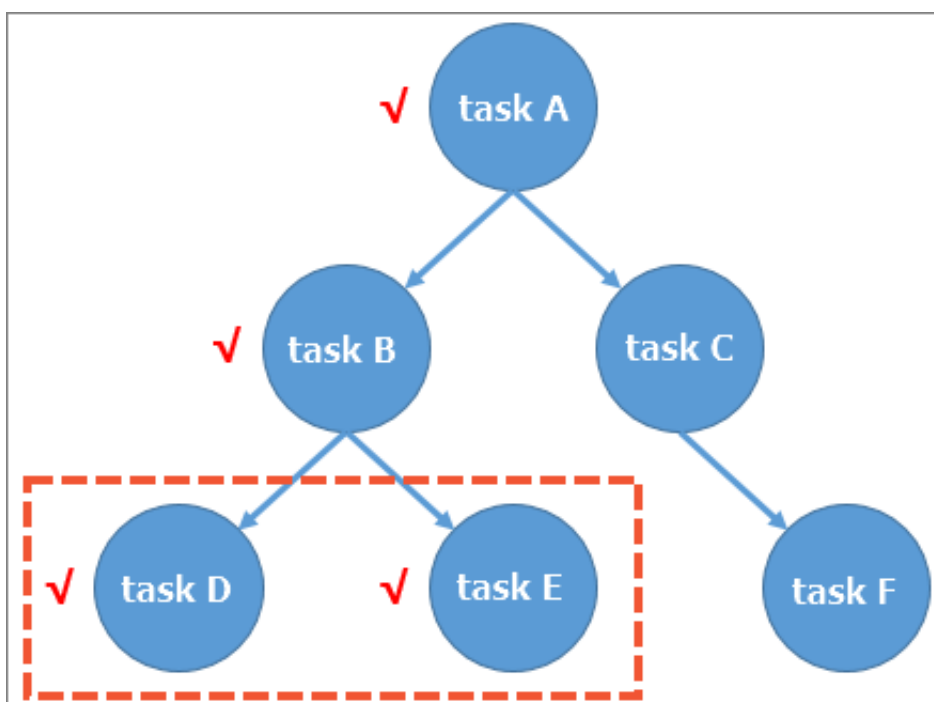
4.5.2 Function introduction

4.5.2.1 Baseline alarm and Event warning

This topic intuitively describes the logics of the baseline warning and event alarm functions in terms of the monitoring scope, task capture, alarm object judgment, alarm time judgment, alarm method judgment, and alarm escalation.

Monitoring scope

Tasks are put under monitoring through baselines (a baseline is the management unit of a group of nodes, which can be understood as a node group for the ease of management). After one baseline is put under monitoring, this baseline and all upstream tasks of the baseline are monitored. Alarm does not monitor all tasks by default but the downstream node of a monitored task must have tasks incorporated into a monitoring baseline. If a downstream node of the monitored task does not have tasks incorporated into a monitoring baseline, Alarm does not report an alarm even if the task has an error.



As shown in the above figure, assume that DataWorks has only six task nodes and Task D and Task E are incorporated into a baseline. Task D, Task E, and all their upstream nodes are included in the monitoring scope. That is, exceptions (error or slowdown), if any, occurring on Task A, Task B, Task D, and Task E can be spot by Alarm, but Task C and Task F are not monitored by Alarm.

Task capture

After the monitoring scope is determined, Alarm generates an event if any task within this monitoring scope has an exception. All alarm decisions are based on the analysis of this event. There are two types of task exceptions, you can select **Event Management > Event Type** to view the task exceptions.

- **Error:** a task running failure.
- **Slowdown:** The running duration of a task is much longer in comparison with the average running duration of tasks in a previous time range.



Note:

If a task times out and then encounters an error, two events are generated.

Alarm object judgment

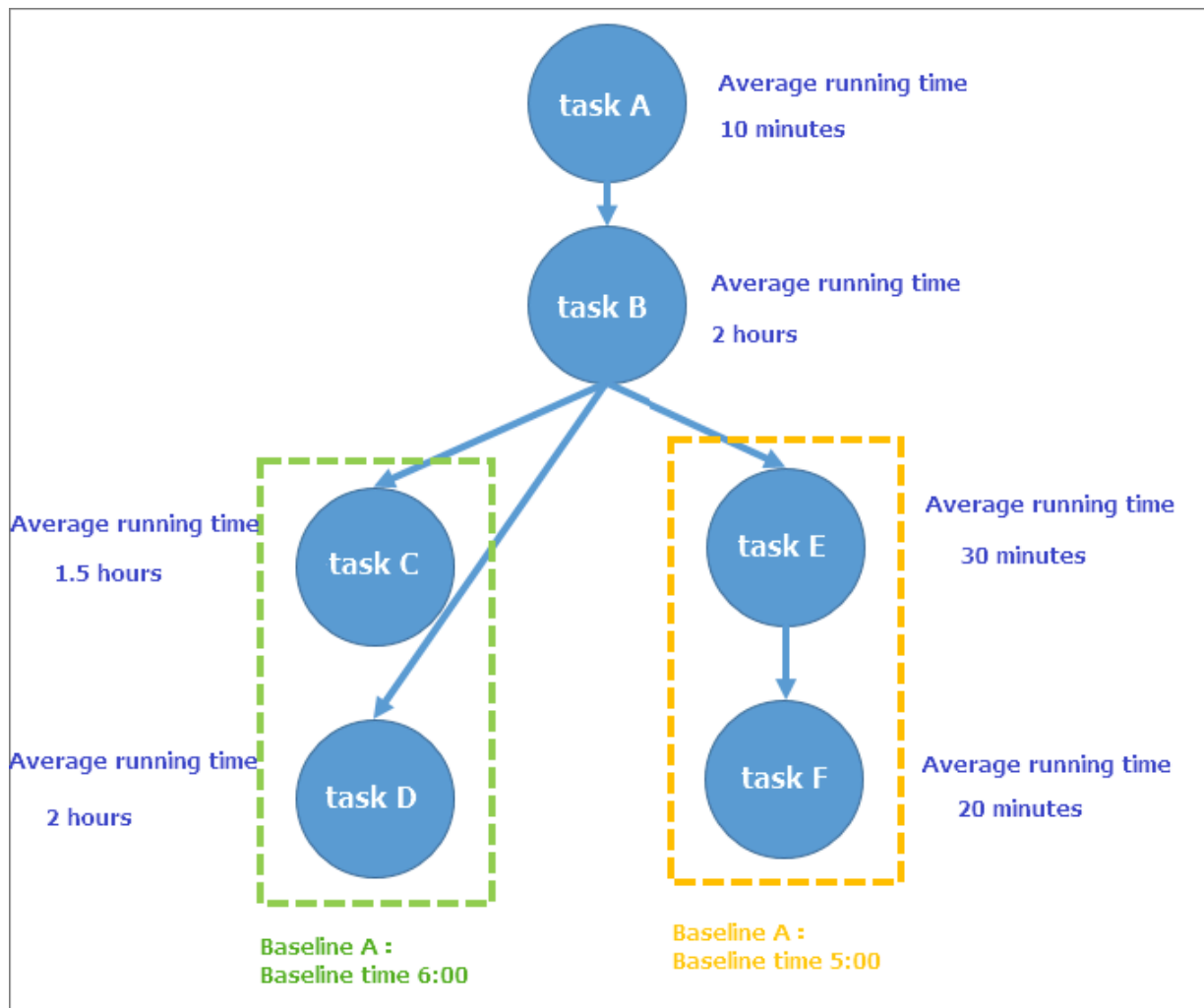
After capturing an abnormal task and generating an event, Alarm determines the alarm object first as follows.

1. Alarm checks whether the rule of the task has a duty schedule. If yes, Alarm considers the on-duty operator in the duty schedule as the alarm recipient.
2. If no duty schedule exists, Alarm sets the task owner as the alarm recipient.

In the task rule, on-duty operators in the duty schedule serve as recipients of alarms using this task rule. Owners of some applications implement the on-duty system and specify an operator for receiving alarms in a period of time. If the duty schedule is absent, Alarm determines that the task owner is responsible for the exception.

Alarm time judgment

Alarm time involves a key concept **margin** in Alarming. Margin indicates the maximum allowable delay before a task is started.



Latest start time of a task = Baseline time – Average running time. As shown in the above figure, in order to meet the baseline time (5:00) of Baseline A, it is required to calculate the latest start time of Task E backwards. The latest start time of Task E is 5:00 minus the sum of the running time of Task F (20 min) plus the running time of Task E (30 min), that is, 4:10, which is also the latest completion time of Task B that meets Baseline A.

To meet the baseline time (6:00) of Baseline B, it is required to calculate the latest completion time of Task B backwards. The result is 6:00 minus the running time of Task D (2 hours), that is, 4:00, which is earlier than 4:10. If both Baseline A and Baseline B need to be met, the latest completion time of Task B is 4:00. The latest completion time of Task A is 4:00 minus the running time of Task B (2 hours), that is, 2:00. The latest start time of Task A is 2:00 minus the running time of Task A (10 min), that is, 1:50. If Task A cannot start at 1:50, it is difficult to meet Baseline A.

Assume that Task A has an error during running at 1:00. The margin time of Task A is the difference between 1:50 and 1:00, that is, 50 minutes. This example shows that the margin reflects the alarm level of a task exception.

Baseline alarm

Baseline alarm is an additional function targeted for baselines with the baseline function enabled. Each baseline must provide the warning margin and commitment time. When Alarm predicts that the baseline completion time is beyond the warning margin at a specific time, it directly notifies the alarm object of the case three times at an interval of 30 minutes. This is called baseline alarm.

Alarm method

You can set the alarm trigger mode and alarm behavior on the **Rule Management** page.

Alarm escalation

If you fail to close an event alarm on Alarm within 40 minutes, the alarm is escalated. The alarm escalation process is as follows:

1. Alarm checks whether the rule of an abnormal task has a duty schedule. If yes, Alarm sends the alarm to the on-duty operator specified in the duty schedule.
2. If no duty schedule exists, Alarm sends the alarm to the supervisor of the task owner.

You can close an alarm by closing the event on the homepage of Alarm.

Gantt chart function

The Gantt chart function is embedded in the **baseline instance** module of Alarm. It reflects the key path of a task.



Note:

A key path is the slowest upstream link that causes the task completion at a time point.

4.5.2.2 Custom notifications

Custom notification is a lightweight monitoring function of Alarm. Its design idea complies with the general monitoring system concept. All alert policies are set by you and the configuration covers the following.

- Monitored object (node, baseline, or project)
- Monitoring trigger condition (error, complete, incomplete, or time-out)
- Alert method (email, SMS)
- Alert object (owner, duty schedule, or others)
- Maximum alert count (maximum number of alerts triggered by an exception, after which the alert is no longer reported. The default value is 3)
- Minimum alert interval (alert interval, which is 30 minutes by default)

- Alert do-not-disturb time

Monitoring trigger conditions are described as follows.

Error

You can set alerts for errors occurring on tasks, baselines, or projects. Once a task has an error, an alert is sent to the preset alert object. Then, detailed task error information is pushed to a relevant user.

Complete

You can set alerts for the completion of tasks, baselines, or projects. Once all tasks of an object are completed, an alert is sent. If alerts are set for the completion of baselines, an alert is sent when all tasks of a baseline are completed.

Incomplete

You can set alerts for tasks, baselines, or projects that are not completed at a time point. For example, when the completion time of a baseline is set to 10:00, if any task of the baseline is not completed at 10:00, an alert is sent and the list of incomplete tasks is pushed to a relevant user.

Time-out

You can set alerts for the time-out of tasks, baselines, or projects. If a monitored task on a preset object is not completed within specified time, an alert is sent.

4.5.2.3 Other functions

Duty schedule function

Alarm provides the duty schedule function. Like the calendar function, the duty schedule function allows you to set a duty schedule and specify a person to receive alerts within a period of time. The duty schedule takes effect only after it is configured as an object for receiving alerts in the alert policy. The duty schedule function supports the cycle rule configuration and the active/standby mode.

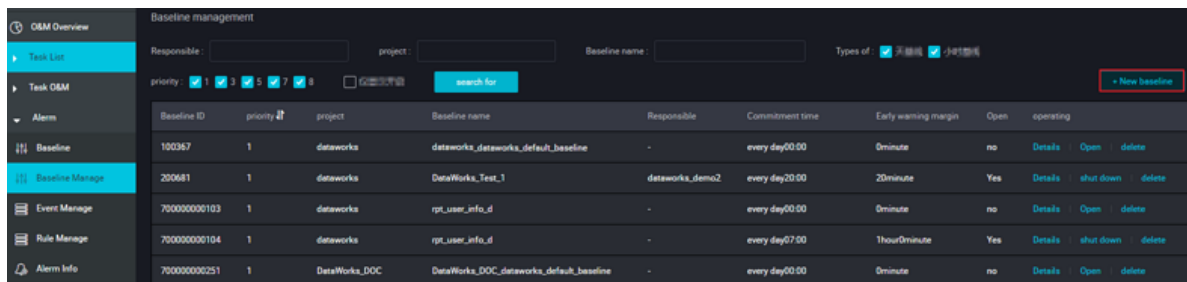
4.5.3 User guide

4.5.3.1 Baseline management and baseline instance

The baseline function involves the **Baseline Management** and **Baseline Instance** pages. On the **Baseline Management** page, you can create and define a baseline while on the **Baseline Instance** page, you can view baseline-relevant information.

Baseline management

1. On the **Baseline Management** page, click **New Baseline** in the upper right corner to create a baseline.



2. On the displayed page, set the baseline and click **determine** in the lower right corner to complete the creation.

New baseline

Baseline name : test-baseline

It's not played : DataWorks_DOC

Responsible : Please enter the responsible person's name/ID

Baseline type : ☒ 天级别 ☐ 小时级别

Safeguard task :

No.	Node name	Responsible
No data		

Please enter task node name/ID

priority : 1

estimated finish time (insufficient historical data, temporarily unable to estimate)

Commitment time : every day 16:00

Early warning margin 15 minute

determine **cancel**

The configuration items are as follows:

- Project: the project to which a task associated with the baseline belongs.
- Baseline Type: determines whether the baseline is detected by day or hour. The option includes day baseline and hour baseline.
- Support Task: a task node associated with the baseline. Enter the task node name or ID and then click the icon behind to add the task node. You can add multiple task nodes.
- Priority: A baseline with a large number is scheduled at a higher priority.
- Estimated finish time: The expected completion time is estimated based on the average completion time of task nodes in the previous periodical scheduling.
- Commitment Time: An alert is triggered if the actual completion time is later than the difference of the commitment time minus the warning margin time.

3. After a baseline is created, click **Enable** in the Operations column to enable the baseline function.

Baseline ID	priority	project	Baseline name	Responsible	Commitment time	Early warning margin	Open	operating
100367	1	dataworks	dataworks_dataworks_default_baseline	-	every day00:00	0minute	no	Details Open delete
200681	1	dataworks	DataWorks_Test_1	dataworks_demo2	every day20:00	20minute	Yes	Details shut down delete

Baseline instance

After a baseline is created, you need to enable the baseline function so that baseline instances can be generated. On the Baseline Instance page, you can search for instances by owner, baseline name, project name, or baseline status, and click **Details**, **deal with**, or **Gantt Chart** in the Operations column to perform operations.

project	Responsible	Baseline name	priority	Baseline status	carry out	Baseline time	margin	Expected latest instance	Current key instance	operating
dataworks	dataworks_demo2	DataWorks_Test_1	1	undone	09-03 14:39 (Expected)	Early warning: 09-03 19:40 committed to: 09-03 20:00	300minute			Details deal with Gantt chart

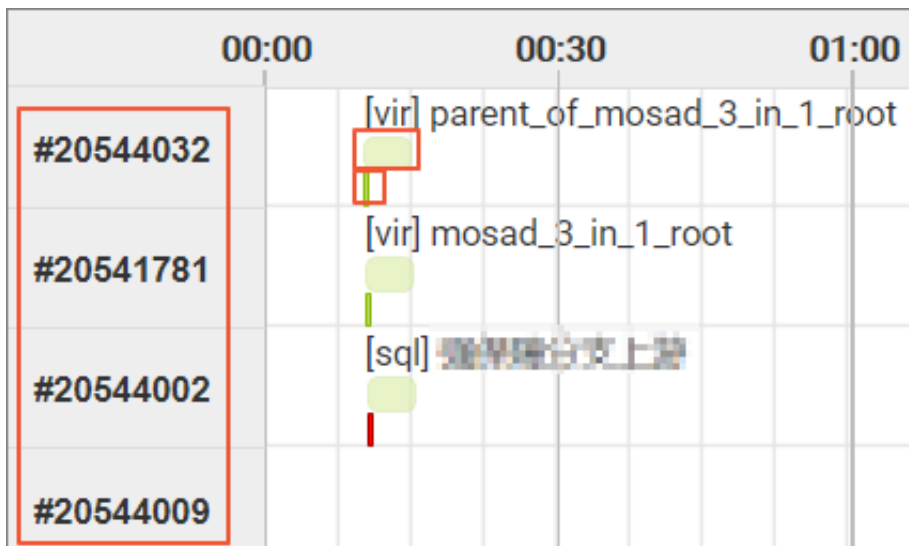
The baseline status is described as follows.

- **Secure:** A task is completed prior to the warning time.
- **Warning:** A task is not completed after the warning time expires but the commitment time is not reached.
- **Breakage:** A task is not completed yet after the commitment time expires.
- **Other:** All tasks of a baseline are paused or the baseline has no task associated.

Operation buttons are described as follows.

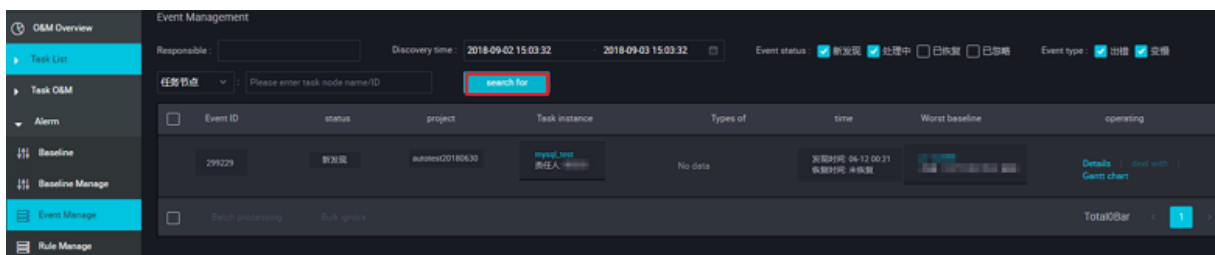
- **Details:** Click this button to go to the Baseline Management page.
- **deal with:** The baseline that generates an alert stops reporting the alert within the handling time.
- **Gantt Chart:** Click this button to view the key path of a task in a Gantt chart.

Gantt chart reflects the key path of a task. The chart displays the average running time of a task, task running status, task running history, and generated exception events. As shown in the following figure, the Gantt chart shows the key path of a task on the left side, the frame in light green shows the average running time of the task, and the frame in dark green shows the actual running time of the task.



4.5.3.2 Event Management

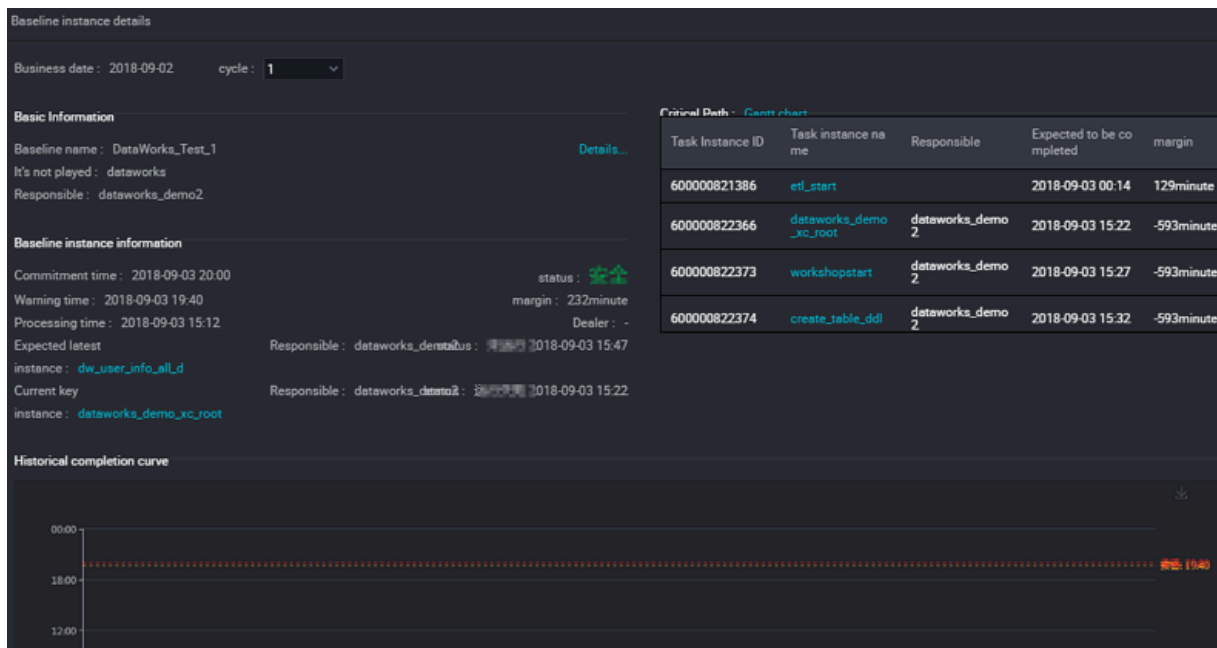
The **Event Management** page lists all **slowdown** and **error** events. You can search for events by owner, name/ID of task node or instance, or event discovery time, as shown in the following figure.



In the search results, each row indicates one event (associated with an abnormal task). The worst baseline indicates a baseline with the minimum margin among the baselines affected by this event.

- Click **Details** in the Actions column of an event to view the event details.
- Click **deal with** to record the event handling operation and pause the alarm in the operation period.
- Click **Ignore** to record the event ignorance record and stop the alarm permanently.

As shown in the following figure, after **Details** is clicked, the event generation time, alarm time, clearing event, previous running record of the task, and detailed task logs are displayed.

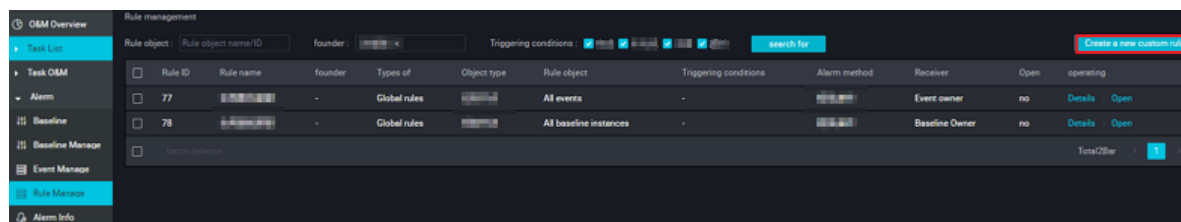


The actual alarm recipient is the person whom an alarm is assigned to. You can click **Alarm Info** to redirect to the alarm details page of an event. Baseline influence displays all downstream baselines affected by tasks related to the event. You can check downstream baselines and baseline breaking severity, in combination with task logs, to investigate causes for the event.

4.5.3.3 Rule Management

This article show you how to customize alarm rules on the **Rule Management** page.

1. On the **Rule Management** page, click **Create a new custom rule** on the right side to define alarm policies.



2. In the displayed **Basic information** dialog box, enter the policy name, policy object, trigger method, and alarm behavior, and click **determine** to generate a policy.

Create a new custom rule


Basic Information

Rule name :

Object type :

Rule object :

No.	mission name	Responsible
No data		



Trigger method



Triggering cond


Alarm behavior

Maximum num: Times

Minimum alarm: minute

Do not disturb time: to

Alarm method: ☒  ☐ 

Receiver: ☐ Task owner ☐ other 

The configuration items are described as follows.

- Object Type: controls the monitoring granularity. A baseline, project, or task node can be selected as a monitored object.
- Trigger Condition: It can be set to complete, incomplete, error, or time-out.
- Minimum alarm Interval: a time interval between two alarms.
- Maximum alarm Count: maximum number of alarms, after which the alarm is not reported regardless of the status of the monitored object.
- Recipient: alarm object, which can be set to owner, duty schedule, or others.
- Do-Not-Disturb Time: No alarm is sent within this period of time.

3. After completing the preceding settings, you can click **Details** in the Operations column of a policy on the **Rule Management** page to view rule details.

4.5.3.4 Alarm info

On Alarm, all alarms can be queried. You can search for specific alarms by rule ID/name, alarm time, or recipient.

Baseline instance

Business date: 2018-09-02 Responsible: Please enter the responsible person's name/ID Related event ID: Please enter the event ID project: Please enter the item

Baseline name: Please enter the baseline name Types of: ☒ ☒ priority: ☒ 1 ☒ 3 ☒ 5 ☒ 7 ☒ 8 Baseline status: ☒ ☒ ☒ ☒ carry out: ☒ ☒ [search for](#)

project	Responsible	Baseline name	priority	Baseline status	carry out	Baseline time	margin	Expected latest instance	Current key instance	operating
dataworks	dataworks_demo2	DataWorks_Test_1	1		undone 09-03 14:39 (Expected)	Early warning: 09-03 19:40 committed to: 09-03 20:00	300minute		<ul style="list-style-type: none"> dw_user_info_all_d dataworks_demo_ac_root 	Details Gantt chart

Each row indicates an alarm, in which the alarm method and alarm transmission status are displayed. You can click **Details** in the Operations column on the right side to view alarm details.

Baseline instance details

Business date: 2018-09-02 cycle: 1

Basic Information

Baseline name: DataWorks_Test_1 [Details...](#)

It's not played: dataworks

Responsible: dataworks_demo2

Baseline Instance Information

Commitment time: 2018-09-03 20:00 status:

Warning time: 2018-09-03 19:40 margin: 276minute

Processing time: 2018-09-03 14:29 Dealer: -

Expected latest instance: dw_user_info_all_d Responsible: dataworks_demo2 status: 2018-09-03 15:04

Current key instance: dataworks_demo_ac_root Responsible: dataworks_demo2 status: 2018-09-03 14:39

Critical Path: Gantt chart

Task instance ID	Task instance name	Responsible	Expected to be completed	margin
600000821386	ed_start		2018-09-03 00:14	129minute
600000822366	dataworks_demo_ac_root	dataworks_demo2	2018-09-03 14:39	-550minute
600000822373	workshopstart	dataworks_demo2	2018-09-03 14:44	-550minute
600000822374	create_table_ddl	dataworks_demo2	2018-09-03 14:49	-550minute
600000822374	create_table_ddl	dataworks_demo2	2018-09-03 14:49	-550minute

Historical completion curve

4.5.4 Intelligent monitor FAQ

4.5.4.1 Why did my alarm report to someone else?

- Check with custom notification creators about rules of custom alarms.
- For alarms generated by baselines with the baseline function enabled, check the specific event page, on which the alarm transmission cause is provided in the lower part.
- If the project of a task is associated with a duty schedule, an alarm is sent to the recipient specified in the duty schedule first. If no duty schedule is available, Alarm checks whether a person has an associated duty schedule. If no, Alarm sends the alarm to the task owner.

4.5.4.2 Task is not important and I do not want to receive alarm. What should I do?

Click **Details** on the **Event Management** page to view downstream baselines affected by the task. If an error occurs within the range of these baselines, a task alarm may be triggered. Contact the baseline owners.

4.5.4.3 Baseline is broken. Why not call the alarm?

The monitoring of a baseline with the baseline function enabled is targeted for tasks. If all tasks of the baseline are normal, no alert is reported even if the baseline is broken because Intelligent Monitor cannot judge which task has an error.

The possible causes for baseline breakage while tasks are normal are as follows.

- The baseline time is set improperly.
- The task dependency is incorrect, and no alert is reported even if the baseline is broken.

4.5.4.4 My task is slowing down but I don't want to receive an alarm.

The following conditions must be met before an alarm is reported for task slowdown:

- The task is on the upstream node of an important baseline.
- The task becomes slow in comparison with its previous running behavior.

If the task slowdown is insignificant, you can ignore it and check with the downstream baseline that has monitored tasks (downstream baseline information is displayed on the **Event Management** page). If the downstream baseline is affected, maintain the task properly.

4.5.4.5 Why is the task wrong but I didn't receive an alarm?

An alarm is reported only when a task meets either of the following conditions.

- The task is on the upstream node of a baseline with the baseline function enabled.
- Associated custom notification rules are set for the task.

4.5.4.6 What should I do when receiving an alarm at night?

When you receive an alarm call at night, you can log on to the event page to close the event alarm for a period of time.

The preceding operations can only close the alarm for a period of time. You should handle received alarms timely.

5 Project management

5.1 Project configuration

You can use the **Project Management** page in the administration console, manages and configures the properties of the current project space.

Procedure

1. Log in to the dataworks management console and navigate to the **Project List** page.
2. Click **Config** after the corresponding project to enter the dataworks project configuration page.
3. Configure your project as needed,
 - Basic Attributes
 - Project name: the name of the current project in dataworks, only letters or numbers (must begin with letters) are supported, not case-sensitive. It is the unique identifier of the project and cannot be changed once created.
 - Project display name: The project display name of the current project in dataworks, used to identify the project, letters, numbers, or Chinese are supported and can be modified.
 - Project Owner: the owner of the current project, who has permission to delete and disable the project, and the identity cannot be changed.
 - Creation date: The date on which the current project is created. Alibaba Cloud's Chinese sites observes the time zone UTC+08:00 and cannot be changed.
 - Status: the item is divided into four states: initialization, initialization failure, normal, and disable.
 - The status of a new project is initializing.
 - The status becomes initialization failed if the creation fails, in which case you can try it again.
 - The normal item can be disabled by the Administrator, and all features of the item are unavailable and data is retained, tasks that have been submitted perform normally.
 - The disabled project can be reset to be normal by using the restoration function.
 - Description: The description of the current project, which is used to comment on the project-related information, you can edit the changes, supports 128 Chinese, letters, symbols, or numbers.
 - Project mode: simple mode and standard mode.

- Enable scheduling cycle: This option determines whether to enable the scheduling system for the current project. If it is off, you cannot schedule tasks cyclically.
- SandBox whitelist (IP address or domain name that can be accessed by configuring Shell)

With SandBox whitelist configured here, even if the Shell task run on the default Resource Group, you can also access the IP directly (where the whitelist can be configured with IP and domain names).

- Calculation engine information
 - Development Environment Project name: Current dataworks project, project name of the maxcompute Project Development Environment used by the underlying layer (this maxcompute project acts as a resource for calculation and storage).
 - Production Environment Project name: the name of the project for the current dataworks project, the maxcompute project production environment that is used at the bottom.
 - Development Environment access identity: default is a personal account, not modifiable.
 - Production Environment System Account: Default select SYSTEM account. Project leader's account execution SQL uses the main account's AK, personal Accounts Execute SQL using sub-account AK, the system account has the highest authority to operate a table of all the items under this account, A personal account can only operate on a table with permission.

**Note:**

When the production environment system account is using a personal account, tasks that run in a production environment may fail in large quantities due to insufficient permissions, please be careful.

5.2 User management

On the User management page under the **Project Management** module of the Alibaba Cloud DTplus platform, you can manage and configure members of the current project.

Page description

Click Project Member Manage in the left-side navigation pane on the Project Management page to enter the Project **User Management** page.

Concepts of listed items:

- Member name: The alias/nick name of the member. The member name is the Alibaba Cloud account currently logged on by default.

- Login name: The Alibaba Cloud account currently logged on.
- Member role: The role of a member in the current DataWorks project (owner, administrator, development, O&M, deploy, Safety Manager or visitor). For specific permissions for different member roles, click the **Permissions List** to view.
- Add members: The system can synchronize all the sub-user accounts under the main account and provide the searching and filtering functions. You can select one or more matched items in the search result and set roles for them in batch.
- Then, you can add selected members to the project, and these members can perform other data and project operations in the current project. You can select one or more matched items in the search result and set roles for them in batch. Then, you can add selected members to the project, and these members can perform other data and project operations in the current project.

**Note:**

If the member account to be added is not found in the Add member list, click **Refresh**, refresh the sub-account to the Count Plus. After the refresh is successful, the check box for the optional neutron account, transfer the sub-account to the account column that you added on the right, and select the role that you want to grant at the bottom, click **confirm** to complete the add operation.

View permissions

In a MaxCompute_SQL task, you can run the following statements to view your permissions:

```
show grants -- View the permissions of the current user
show grants for <username> -- View the permissions of a specified user
, which is only available to the project administrator.
```

For more permission viewing commands, see [Permission check](#).

5.3 Permission list

DataWorks provides seven roles for project owners (non-authorizable), project administrators, development, operations, deployment, guest, and Security Administrators. This article will introduce you to the permissions descriptions for specific roles.

Data Management

Permission Point	Owner	Administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Delete tables created by one-self	√	√	√	N/A	N/A	N/A	N/A
Settings of table categories by one-self	√	√	√	N/A	N/A	N/A	N/A
View your own collection of tables	√	√	√	N/A	N/A	N/A	N/A
New table	√	√	√	N/A	N/A	N/A	N/A
Unhide the table you created	√	√	√	N/A	N/A	N/A	N/A
Self-created table structure changes	√	√	√	N/A	N/A	N/A	N/A
Self-created table view	√	√	√	N/A	N/A	N/A	N/A
Viewing the content of the Right applied by one-self	√	√	√	N/A	N/A	N/A	N/A
Self-created table Hiding	√	√	√	N/A	N/A	N/A	N/A
Self-created table lifecycle settings	√	√	√	N/A	N/A	N/A	N/A
Non-self-created table data permission application	√	√	√	N/A	N/A	N/A	N/A
Update table	√	√	√	√	√	N/A	N/A
Delete a table	√	√	√	N/A	N/A	N/A	N/A

release management

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Create a publishing package	√	√	√	√	N/A	N/A	N/A
View the Publishing Package List	√	√	√	√	√	√	N/A
Delete package	√	√	√	√	N/A	N/A	N/A
Perform publish	√	√	N/A	√	√	N/A	N/A
see release package content	√	√	√	√	√	√	N/A

button control

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
button-stop	√	√	√	N/A	N/A	N/A	N/A
button-format	√	√	√	N/A	N/A	N/A	N/A
button-Edit	√	√	√	N/A	N/A	N/A	N/A
button-run	√	√	√	N/A	N/A	N/A	N/A
button-Amplification	√	√	√	N/A	N/A	N/A	N/A
button-save	√	√	√	N/A	N/A	N/A	N/A
button-expand/collapse	√	√	√	N/A	N/A	N/A	N/A
button-delete	√	√	√	N/A	N/A	N/A	N/A

Code development

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Save submitted code	√	√	√	N/A	N/A	N/A	N/A
view code content	√	√	√	√	√	√	N/A
Create Code	√	√	√	N/A	N/A	N/A	N/A

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Delete Code	√	√	√	N/A	N/A	N/A	N/A
view code list	√	√	√	√	√	√	N/A
run code	√	√	√	N/A	N/A	N/A	N/A
modify code	√	√	√	N/A	N/A	N/A	N/A
Your files download	√	√	√	N/A	N/A	N/A	N/A
Your files upload	√	√	√	N/A	N/A	N/A	N/A

Function development

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
View function details	√	√	√	√	√	√	N/A
Create Function	√	√	√	N/A	N/A	N/A	N/A
query function	√	√	√	√	√	√	N/A
delete function	√	√	√	N/A	N/A	N/A	N/A

Node Type Control

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
node-PAI	√	√	√	N/A	N/A	N/A	N/A
Node -- MR	√	√	√	N/A	N/A	N/A	N/A
node-CDP	√	√	√	N/A	N/A	N/A	N/A
Node -- sql	√	√	√	N/A	N/A	N/A	N/A
Node -- xlib	√	√	√	√	√	√	N/A
node-Shell	√	√	√	N/A	N/A	N/A	N/A
node-virtual node	√	√	√	√	√	√	N/A
node-script_seahawks	√	√	√	N/A	N/A	N/A	N/A
node-dtboost_analytic	√	√	√	N/A	N/A	N/A	N/A

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Node -- dtboost_re command	√	√	√	N/A	N/A	N/A	N/A
Node -- pyodps	√	√	√	N/A	N/A	N/A	N/A

Resources Management

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
view resources list	√	√	√	√	√	√	N/A
delete Resources	√	√	√	N/A	N/A	N/A	N/A
create resources	√	√	√	N/A	N/A	N/A	N/A
Upload jar your files	√	√	√	N/A	N/A	N/A	N/A
Upload text your files	√	√	√	N/A	N/A	N/A	N/A
Upload archive your files	√	√	√	N/A	N/A	N/A	N/A

workflow development

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Run/Stop Workflow	√	√	√	N/A	N/A	N/A	N/A
save workflow	√	√	√	N/A	N/A	N/A	N/A
View workflow content	√	√	√	√	√	√	N/A
Submitted Node Code	√	√	√	N/A	N/A	N/A	N/A
Modify Workflow	√	√	√	N/A	N/A	N/A	N/A
View workflow list	√	√	√	√	√	√	N/A
Modify the Owner property	√	√	N/A	N/A	N/A	N/A	N/A
Open Node Code	√	√	√	N/A	N/A	N/A	N/A
delete Workflow	√	√	√	N/A	N/A	N/A	N/A
create workflow	√	√	√	N/A	N/A	N/A	N/A

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Create folder	√	√	√	N/A	N/A	N/A	N/A
delete folder	√	√	√	N/A	N/A	N/A	N/A
Modify folder	√	√	√	N/A	N/A	N/A	N/A

Data Integration

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Data Integration-node Edit	√	√	√	N/A	N/A	N/A	N/A
Data Integration-node View	√	√	√	N/A	N/A	N/A	N/A
Data Integration-node Delete	√	√	√	N/A	N/A	N/A	N/A
project resources consumption monitoring menu	√	√	N/A	N/A	N/A	N/A	N/A
Project synchronous Resources Management menu	√	√	√	√	√	N/A	N/A
Project synchronous Resources Group list	√	√	√	√	√	√	N/A
Project synchronous Resources Group create	√	√	√	√	√	N/A	N/A
Project synchronous Resources Group Management machine list	√	√	√	√	√	N/A	N/A
Project synchronous Resources Group Add Machine	√	√	√	√	√	N/A	N/A

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
Project synchronous Resources Group Delete Machine	√	√	√	√	√	N/A	N/A
Project synchronous Resources Group modify Machine	√	√	√	√	√	N/A	N/A
Project synchronous Resources Group get Resources Group AK	√	√	√	√	√	N/A	N/A
Project synchronous Resources Group Delete	√	√	√	√	√	N/A	N/A
project resources consumption monitoring	√	√	N/A	N/A	N/A	N/A	N/A
Operation and Maintenance Center task modify Resources Group	√	√	√	√	√	N/A	N/A
Synchronous task list menu	√	√	√	√	√	N/A	N/A
The task is moved script	√	√	√	√	√	N/A	N/A
Get project members list	√	√	√	√	√	N/A	N/A
New Code Interface	√	√	√	√	√	N/A	N/A
Save/update code Interface	√	√	√	√	√	N/A	N/A
According to fileId get code Interface	√	√	√	√	√	√	N/A
Get Data Integrated node list	√	√	√	√	√	N/A	N/A
Search table Interfaces	√	√	√	√	√	N/A	N/A
search field interface	√	√	√	√	√	N/A	N/A
query data source list Interface	√	√	√	√	√	√	N/A

Permission Point	Owner	Project administrator	Developer	O&M	Deploy	Visitor	Security Administrator
new data source interface	√	√	N/A	N/A	N/A	N/A	N/A
query data source details Interface	√	√	√	√	√	N/A	N/A
update data source interface	√	√	N/A	N/A	N/A	N/A	N/A
delete data source interface	√	√	N/A	N/A	N/A	N/A	N/A
Test connectivity	√	√	√	√	√	N/A	N/A
Data Preview	√	√	√	√	√	N/A	N/A
Check whether open OTSStream	√	√	√	√	√	N/A	N/A
Open Table Store	√	√	√	√	√	N/A	N/A
Query ODPS table building statement	√	√	√	√	√	N/A	N/A
New ODPS table	√	√	√	√	√	N/A	N/A
Query ODPS table status	√	√	√	√	√	N/A	N/A
Migration Database Table	√	√	N/A	N/A	N/A	N/A	N/A

5.4 Project mode upgrade

In DataWorks V2.0, a standard project model, a standard project model, was introduced, A DataWorks project corresponds to two MaxCompute projects that isolate the development and production environments, increase the release process of the task to ensure the correctness of the task code.

Benefits of a standard pattern

In previous versions, the projects that you created were a DataWorks project that corresponds to a MaxCompute project, this is a simple pattern in DataWorksV2.0. Simple mode directly causes table permissions to be uncontrollable, such: just want to query some of the table for some of the students in the project, this scenario cannot be implemented directly in simple mode , because a DataWorks project corresponds to a MaxCompute, the development role permission

s of DataWorks have the operation privileges of all tables under the MaxCompute project, so it is not possible to control table permissions precisely, and it is necessary to create a separate DataWorks project, to complete the isolation of data using the method of project isolation.

DataWorks V1. for the scenario of table permission control, a scenario is derived: manually bind two DataWorks projects, set the project to be a published Project for the B project, project A receives tasks published in Project B without having to develop code directly. So that project A became a project similar to the production environment, B is a project similar to the development environment.

There are also vulnerabilities in the mode of two DataWorks project bindings, project A is also a normal DataWorks project, can be directly in the data development module for task development, resulting in (production) the Code Update portal for the environment is not unique, and there is a logical vulnerability throughout the development process.

In response to the above-mentioned problems, we launched a standard project model.

In a standard project model, there are several benefits for data developers:

1. A DataWorks project corresponds to two MaxCompute projects that perfectly separate the development and production computing engines, project members have only the permissions of the development environment, and default no permission actions on the Production Project's tables, improves the security of production data.
2. In standard mode, the data development interface defaults to the task of operating the development environment, the tasks of the production environment are published to production through the publishing function, ensure the uniqueness of production environment code editing entry, improve the safety of production environment code.
3. In standard mode, the development environment does not do periodic scheduling by default , it can reduce the consumption of computing resources under account, and guarantee the resources running in production environment task.

Project mode upgrade

In DataWorks V1.0, we create simple schema projects, and that is, under simple schema projects , how can we upgrade to a standard model?

1. In project management, you can see the buttons that are upgraded to the standard mode.

Basic Properties	
Project ID : 77385	Created At : 2018-07-26 16:17:55
Project Name : dataworks_demo_xc	Project Mode : Simple Mode Upgrade to standard mode
Project Display Name : DataWorks流程_简单01	You can down SELECT results in this project. : <input checked="" type="checkbox"/>
Project Owner : dataworks_demo2	Enable Schedule Period : <input checked="" type="checkbox"/>
Project Status : Normal	

As you can see from the figure above, the original project will become a production project in the dual project, the user needs to create a new development environment for MaxCompute, and the project name can be selected by itself. When you click confirm, DataWorks joins the members of the original project in the newly created MaxCompute development project, the members and roles of the original project are retained, however, the project member's permissions on the Production Project are abolished, and only the project owner has permissions on the production item.

For example: A company has an project on DataWorks, and after you click on the project upgrade, create a Development Environment Project, the members, roles, tables, and resources in the original a project are all created under the'dev Project (only tables are created, do not clone the table data as well). Member A1 (development role), B1 (operation and maintenance role), former project), it also joins under the'dev project and retains the role permissions. Project A becomes a production project, the A1 and B1 users' data permissions in Project A are abolished, by default, there is no select and drop permission for the table, and the data for the production item is directly protected. In the data studio interface, the default operation of the MaxCompute project is a'dev, to query the data of the production environment in the data development interface, you need to use the project name. The way the table is called. The data development interface can only edit code for the AHA Dev environment, to update the code in Project A, you can submit a task to the scheduling system only by the'dev, how to publish to the production environment for updates. A process of task release (Audit) was added to ensure the correctness of the production environment code.

2. When you click on the project mode, the following prompts appear, and you need to enter the project name for the development environment.

Project mode upgraded to standard mode

Project mode upgrades are expected to take develop environment minutes.

MaxCompute

MaxCompute项目名称: nodi_dev

MaxCompute访问身份: 个人账号

发布

MaxCompute

MaxCompute项目名称: nodi

MaxCompute访问身份: 项目负责人账号

我确认要升级此项目: ☐

确定 cancel

**Note:**

After the project has been upgraded, you cannot directly access the data of the original project, and you need to apply for role permissions. The tables that you query in the data development interface, by default, are the tables of your development environment, to access the production table, you need to apply for the role permission after using the project name. The way the table name is accessed.

6 Data quality

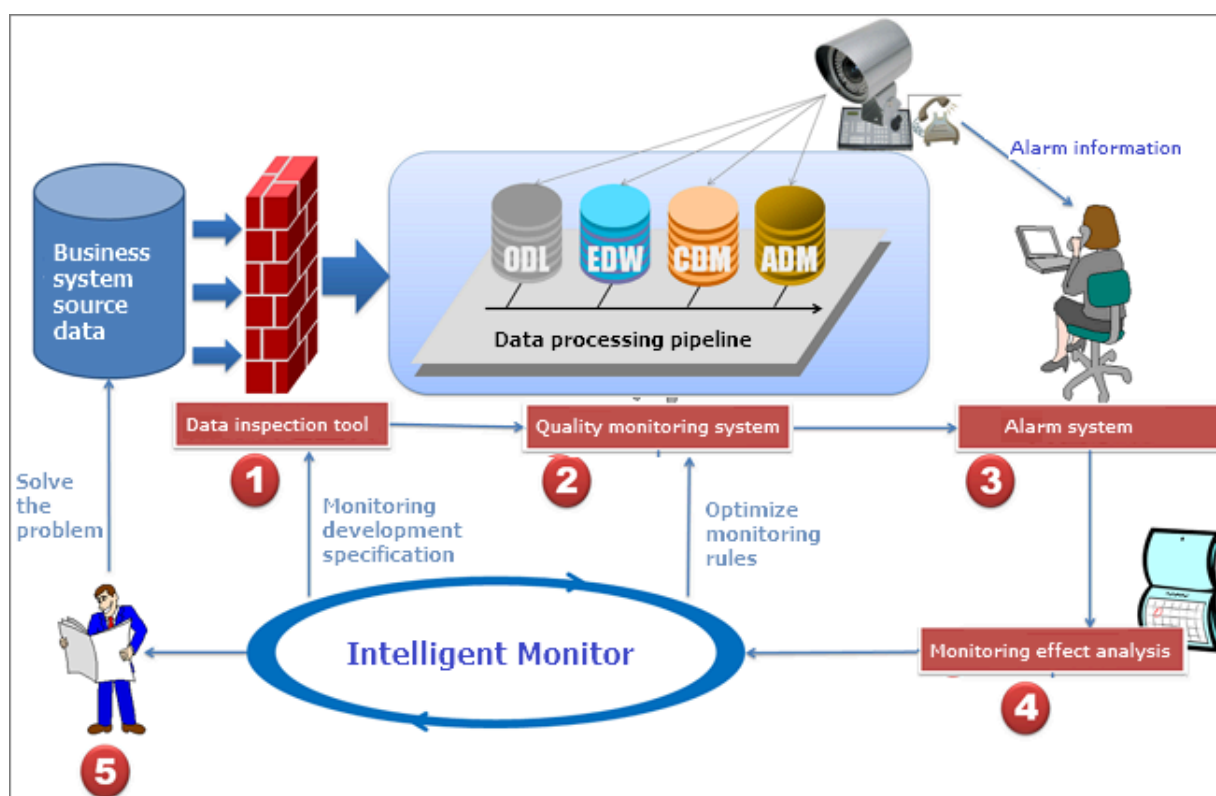
6.1 Data quality overview



Note:

Currently, Data Quality Center service is in the internal beta stage. It can be activated only in Shanghai, Hangzhou, Shenzhen, Beijing, UK, Malaysia region. Therefore, if you have related requirements, join DataWorks communication group 0 (group number is 11718465) to apply for service activation.

DataWorks Data Quality Center (DQC) is a one-stop platform supporting multiple heterogeneous data sources quality check, notifications, and management services.



Data Quality monitors DataSet. Currently, Data Quality supports monitoring of MaxCompute data tables and DataHub real-time data streams. When the offline MaxCompute data changes, the Data Quality verifies the data, and blocks the production links to avoid spread of data pollution. Furthermore, Data Quality provides verification of historical results. Thus, you can analyze and quantify data.

In the streaming data scenario, Data Quality can monitor the disconnections based on the Datahub data tunnel. For the first time, warning is sent to the subscriber. Data Quality also provides orange and red alarm levels, and supports alarm frequency settings to minimize redundant alarms.

This article briefly introduces the main interface components of Data Quality. The interface consists of four function modules, as follows:

- **Overview:** By default, home page is the overview page that shows MaxCompute data tables alarms and blockings, DataHub Topic alarms, and current and historical tasks. Current tasks include personal subscriptions, alarms, and blockings for all tasks under the project. You can also browse historical tasks for last seven and last thirty days (date range of up to three months). Additionally, a quick way to go to the task query page is provided.
- **My subscriptions:** The page shows the running status of all subscribed tasks. You can switch between MaxCompute and DataHub data sources to find subscribed tables or Topics. You can also change the notification method (currently, email notification, and email and SMS notifications are supported).

Select MaxCompute data source, click partition expression on the right (or select the DataHub data source, and click Topic name) to enter the currently selected rule configuration interface.

- **Rule configuration:** Rule configuration is the core function module of Data Quality. Using this module, you can manage the features related to partition expressions and rule configurations (template rules and customized rules).
- **Mission inquiries:** The task query module mainly queries the rule validation situation.

6.2 Prerequisites

6.2.1 Prepare your data

DQC is mainly to monitor the quality of MaxCompute dataset and DataHub dataset. You need to create a table first, and then insert some sample data into the table.



Note:

You can create a MaxCompute table and insert into sample data by using MaxCompute console or DataWorks.

6.2.2 Establish DQC

Procedure

1. Create an account.

Organization Administrator create accounts by using RAM.

2. Log on to the Console with your Alibaba cloud account, go to DQC.

You will see You haven't join any organization, please contact with your administrator.

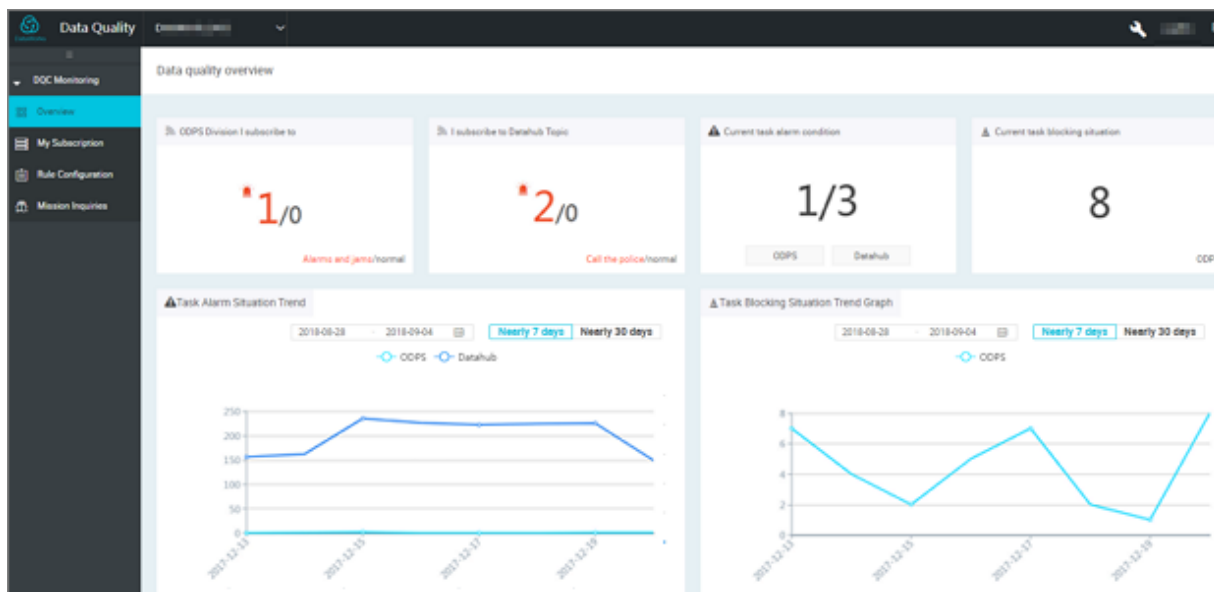
3. Add your account as a member of the project.

Contact with Organization Administrator to add your account to the organization as a member.

After these steps, when you log on to Alibaba cloud and visit DQC, you can work now.

6.3 Overview

Data quality home page mainly includes ODPS Division I subscribe to, DataHub Topic I subscribe to, Current task alarm condition, Current task blocking situation, Task Alarm Situation Trend and Task Blocking Situation Trend Graph.



The module is described below:

- **ODPS Division I subscribe to:** Displays the subscribed MaxCompute partition alarms, and blocked and normal tasks for the current day. Click this module to quickly jump to the task query page of The MaxCompute data source for details.

- **DataHub Topic I subscribe to:** Displays the DataHub data source alarm that I subscribe to the same day, the normal two situations, click this module to quickly jump to the task query page of The DataHub data source for details.
- **Current task alarm condition:** Displays the task alarm status for both the day and the currently applied MaxCompute and DataHub data sources.
- **Current task blocking situation:** Displays the day that the task blocking is currently applied to the MaxCompute data source.
- **Task Alarm Situation Trend:** Optional 7 days, 30 days, and custom time periods, supports task alarm trend diagrams for MaxCompute and DataHub data sources for a date range of nearly three months.
- **Task Blocking Situation Trend Graph:** Optional 7 days, 30 days, and custom time periods, supports task blocking for MaxCompute for a date range of nearly three months.

6.4 My subscription

My subscription page shows the current status of all subscribed tasks. You can select the corresponding data source to find your subscription task. You can also change the notification method (currently email notification, and email and SMS notifications are supported).

You can select the following two data sources to perform the related operations.

- Select MaxCompute data source

Click the corresponding partition expression on the right to enter the rule configuration interface

.

1. Click **Subscribed** in the corresponding partition expression action bar to cancel the subscription.
2. Click **Last check results** to go to the task query interface. For more information, see [Configure MaxCompute data source rules](#). See for details [Rules Configuration for ODPS data source](#).

- Select DataHub data source

1. Select DataHub data source Click **Unsubscribe** in the corresponding Topic action bar to cancel the subscription.
2. Click Topic name to enter the rule configuration interface. For more information, see [Configure DataHub data source rules](#).

6.5 Rule Configuration

6.5.1 Rules configuration for DataHub data source

This article describes how to configure the DataHub data source.

Go to Operation center, you can create new data sources. You can configure the Endpoint of DataHub, data source name, AccessKeyID, and AccessKeySecret to create a connection string. After this, you can query on DQC.

Choose data source

1. Click **Rules configuration** at the left navigation;
2. Select **DataHub data source**, you can see all topics in this data source.

Select DataHub data source, you can see all topics in this data source.

Table Name	data source	Application name	Responsible	operating
ods_user_log_id	Datahub	dataworks_demo2	dataworks_demo2	Configure monitoring rules
ods_user_info_id	Datahub	dataworks_demo2	dataworks_demo2	Configure monitoring rules

Monitor rules configuration

1. Select a specific topic, click **Configure monitoring rules** to enter monitor rules page.

You can also navigate to **My Subscription > DataHub data source > Topic name** to enter the subscribed topic quickly.

2. Click **create rule**, and the datahub data source can now only create a template rule with a data type of cut-off monitoring.

Configurations:

- **Alarm frequency:** You can set how often to alarm, there are 10 minutes, 30 minutes, 1 hour, 2 hours four options.
- **Orange threshold:** in minutes, you can only enter an integer, and you must be less than the red threshold.
- **Red threshold:** in minutes, you can only enter an integer, which must be greater than the orange threshold.

3. When the settings are complete, click **Save** to add the rules that you created to the topic.

6.5.2 Rules Configuration for ODPS data source

This article introduces how to configure ODPS data source.

Rules configuration is the core function module of Data Quality. Data sources are divided into ODPS data source and DataHub data source.

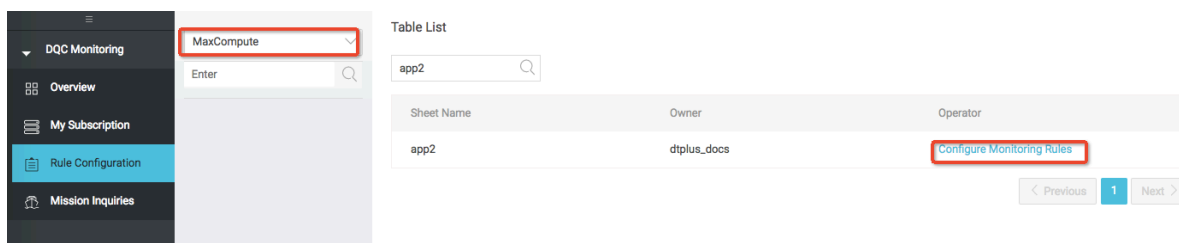
Select a data source

1. Click **Rules Configuration** in the left-side navigation pane to enter the Rules configuration page.
2. Select **MaxCompute** to display all the tables in the project you have joined.



Note:

You can use the search box to find topics in other data sources quickly.



3. Click **Configure Monitoring Rules** on the right side.



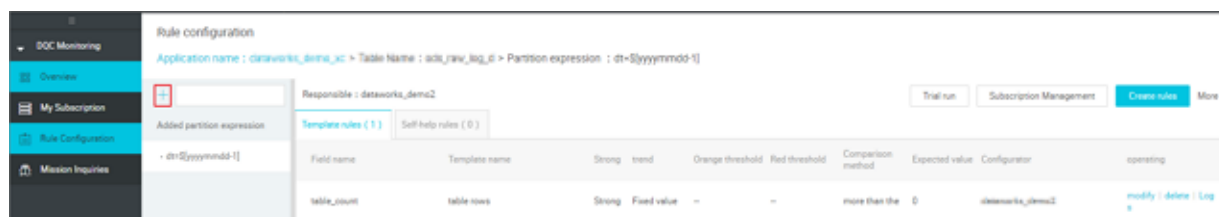
Note:

Additionally, you can select **ODPS data source** in My subscriptions by [My subscription](#), and click **Partition expressions** on the right to enter the Rules configuration page.

Configure the partition expression

A partition expression is a filtering condition used to match a validation rule.

In the Rule Configuration page, click the plus sign⁺ in the upper left corner to add a partition expression.



- Expression for new partition: Click **+** in the upper left corner to pop up **Add a partition**, you can edit a syntax-compliant partition expression to suit your needs. Non-partition table can be directly selected **NOTAPARTITIONTABLE** in the recommended partition expressions list.
- Format of the first-level partition expressions: Partition name = partition value. Partition value can be a fixed value or a built-in parameter expression.
- Format of the multi-level partition expressions: First-level partition name = partition value / second-level partition name = partition value / N-level partition name = partition value. Partition value can be a fixed value or a built-in parameter expression.

Built-in parameter expression

- `${yyyymmddmiss-1}`

The format is `yyyymmddmiss-1`. The previous day's (year-month-day) scheduled time of the daily scheduled instance; and is equal to the time (year-month-day) for the instance of the automatically scheduled daily node, minus 1 day.

- `${yyyymmddhh24miss}`

The format is `yyyymmddhh24miss`. It specifies the scheduled time (year-month-date-hour-minute-second) for the routinely scheduled instance.

- Yyyy indicates 4-digit year
- Mm for 2-digit month
- DD for 2-digit days
- Hh24 is a 24-hour system.
- MI 2-digit minutes
- SS for 2-digit seconds

Get +/- period method

The partition expression cycle is determined by the configured run time, for example, the configuration run time is the first 5 days, the cycle is scheduled every 5 days.

- N days before: `${yyyymmdd-N}`
- The 1st day of each month: `${yyyymm01-1}`
- The 1st day of N months before: `${yyyymm01-Nm}`
- The last day of each month: `dt=${yyyymmld-1}`
- The last day of N months before: `dt=${yyyymmld-Nm}`
- One hour ago: `$(hh24miss-1/24)`
- Half an hour ago: `$(hh24miss-30/24/60)`

- Added partition expressions: Indicates the partition expressions already added to the table.
- Recommended partition expressions: Indicates the partition expressions recommended by Data Quality. In the list of recommended partition expressions, you can find the partition expression that meets your requirements, and select to add it. When a recommended partition is successfully added to the table, it is displayed in the Added Partitions section.

If you don't know if the recommended and custom expressions match your expectations, you can use the partition calculation function for calculations.

- Delete partition expression: Partition expressions that are no longer used can be deleted. If the partition expression has been configured with rules, all rules under the expression are also deleted.



Note:

In the following example, the partition name dt is taken as an example. If the table is a dynamic partition table, the use of a regular partition expressions is not recommended.

Partition expression	Description
ALL_PARTITIONS	This partition expression can be selected for non-partition tables.
dt = [[a-zA-Z0-9 _-] *>	The expression is generally used for hours tasks. If the table partition is an hour partition, it automatically replaces the regular expression with the partition expression.
dt=\${yyyymmdd-N}	Indicates N days before.
dt=\${yyyymm01-1}	Indicates the 1st day of each month.

Partition expression	Description
dt=[yyyyymm01-Nm]	Indicates the 1st day of N months before.
dt=[yyyyymmld-1]	Indicates the last day of N months before.
dt=[yyyyymmld-1m]	Represents the last day of N months ago.
dt=[hh24miss-1/24]	Represents an hour ago.
dt=[hh24miss-30/24/60]	Representing half an hour ago.

Click the input expression window, and the recommended partition expressions are displayed in the drop-down list.

- If an appropriate expression is in the list, click the line to automatically synchronize it to the output window.
- If none of partition expressions meet your requirements, you can input partition expressions as needed.

After the operation is complete, click **Calculate**. Data Quality calculates the value of partition expressions according to the current time (scheduled time) to verify the correctness of the partition expressions.

Click **Ok** to complete the operation.

Associated scheduling

To monitor offline data on the production links, you can use Data Quality associated scheduling function. Please ensure at least one of those three roles , which are **Project Manager, Development, O&M** , has been granted in both projects.



Note:

Please refer to [User management](#) for how to check project role.

You can add associated scheduling to existing Task node. After associating with the schedule, the data quality monitoring task would run automatically. (You can skip below steps if you do not want to monitor the data quality.)

You can enter **Operation Center** to set the associated scheduling quality monitoring configuration.

1. Click **More > Configure Quality Monitoring** in corresponding task tab.
2. Select specific Project Name and Table Name , and click **Configurations** in the corresponding partition expression tab (you can also add a partition expression by yourself) to configure this partition expression.

Create rules

Creating rules according to the actual needs of the table is the core function module of Data Quality.

Currently, rules can be created in two ways: Template rules and Customized rules, specific usage depends on the actual needs. These two kinds of rules are divided into **Add monitoring rules** and **Quick add**.

After creating the rules, click **Save batches**, you can save all the rules to the already created partition expressions.

Template rules

- **Add monitoring rules**

- Field type: Consists of table-level rules and field-level rules. The field-level rules configure monitoring rules for specific fields in the table. The table-level rules are selected here, and other setting items in the interface correspond to the table-level rules configuration.
- Intensity: You can configure the intensity of the rule. For example, when strong is selected, if the red threshold is triggered while the task is running, the task is set to fail.
- Template type: The system has a built-in table-level monitoring rules module.
- Tendency: Depending on the type of template selected, tendency can include the following types: absolute value, increasing, and decreasing.
- Comparison of fluctuation values: Set the orange and red thresholds of the fluctuation value. You can manually drag the progress bar, or directly input the threshold value.

- **Quick add**

- Field name: Can be used only for field-level rules. Field-level rules configure monitoring rules for specific fields in the table. Select specific fields to set the field-level rules.
- Rule type: Select the field null value or field repetition value.

If the template rules do not meet your requirements for partition expressions quality monitoring, you can use customized rules to create the custom monitoring rules.

Customized rules

On the Customized rules page, you can select to create table-level rules or custom SQL.

• Add monitoring rules

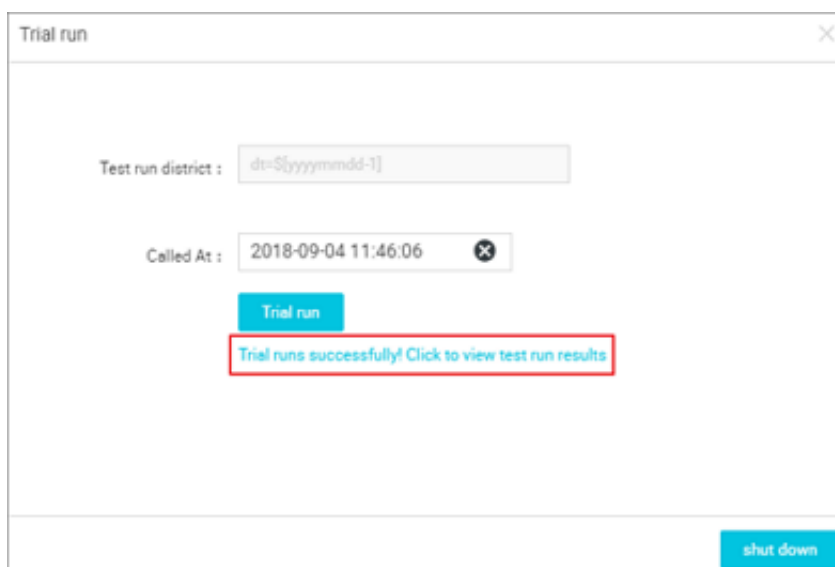
- Field Type: Consists of table-level rules, field-level rules, and custom SQL. The table-level rules are selected here, and other settings items in the interface correspond to the table-level customized rules configuration.
- Intensity: When strong is selected, if the red threshold is triggered while the task is running, the task is set to fail.

- Statistical functions: Include two types: count and count/table_count.
- Filter conditions: Custom SQL.
- Verification method: The built-in verification method can be selected. The verification method defaults to a fixed value.
- Tendency: Includes three types: absolute value, increasing, and decreasing. If the statistical function is set to count/table_count, the tendency defaults to a fixed value.
- Comparison method: According to the actual needs, there are many options: greater than, greater than or equal to, equal to, not equal to, less than, less than or equal to.
- Expected value: The expected target value.
- Description: The detailed description of the customized rule.
- **Quick add**
 - Rule type: Includes two types: Number of rows in the table is greater than null and Multiple fields repetition value.
 - Field name: When the rule type is Multiple fields repetition value, the field names that must be added are displayed, and the multiple field names can be added.

Test run

After the rules are configured, you can perform a test run for all the rules under a partition expression, and view the test run results.

1. Select the required scheduling date, and click **Test run**.



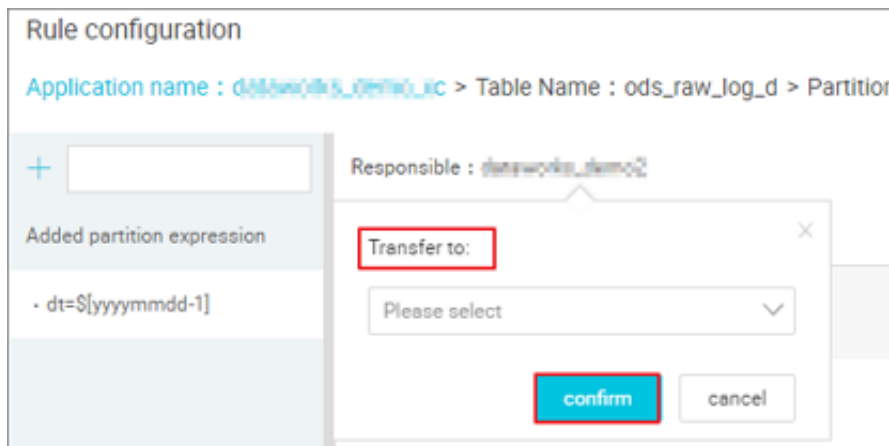
- **Test run partition:** the actual partition changes with the change of business date. If NOPARTITIONTABLE, the actual partition is automatically added.
- **Scheduling time:** The default is the current time.

2. Click **test run success! Click Trial Run Success** , Click to view the test run results, and go to the [task query](#) page to check the results.

Change the responsible person

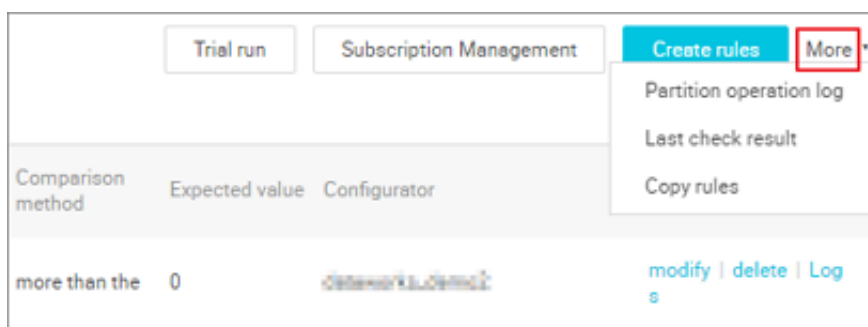
When the responsible person leaves or changes job, person in charge of the partition expressions can be changed with another project member. Place the mouse over the responsible person, and the hidden button is displayed.

Place the mouse over the responsible person, and the hidden button is displayed. Click to modify the responsible person, input the name of the new person in charge, and click **Confirm** to submit.

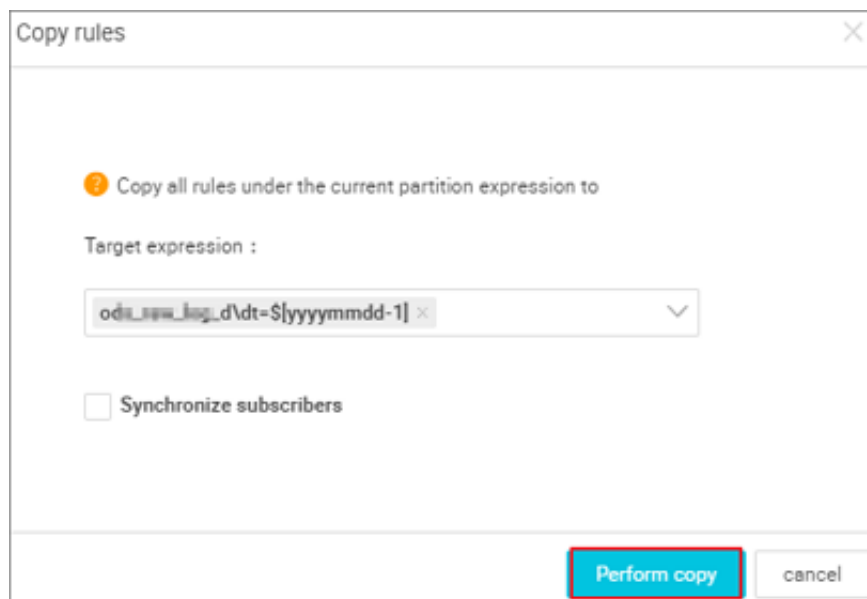


More

Option **More** includes the following options: Partition operations logs, Last verification results, and Copy rules.



- **Partition operations logs:** Displays a record of all the rule settings for the current partition expression.
- **Last verification results:** Redirects to the task query interface where you can view the running results under the current partition expression. You can also check the historical results.
- **Copy rules:** You can copy the currently set rules into the target expression, and the transmissions can be synchronized.



For more information about template rules supported by ODPS data source, see [Template rule](#).

6.6 Mission Inquiries

6.6.1 Viewing DataHub data source tasks

1. Visit the Data Quality Center, click **Mission Inquiries**, and enter the query page.
2. Step 2: Choose **DataHub Data Source**, and enter key words as prompted in the search box to find the specific topic.

DOC Monitoring

Overview

My Subscription

Rule Configuration

Mission Inquiries

Task query

Datahub data source

Please select a data source

☐ my subscription

Empty

Topic name	Data source type	Data source name	project name
<input type="radio"/> test_many_shard	datahub	wengji_test_datahub	wengji_test
<input checked="" type="radio"/> test_sale	datahub	wengji_test_datahub	wengji_test

Alarm time	status	Number of rules	Abnormal number	Monitoring rules	operating
2017-12-20 14:34:38	<div></div>	1	0	<div></div>	<div></div>
2017-12-20 14:24:34	<div></div>	1	0	<div></div>	<div></div>
2017-12-20 14:14:29	<div></div>	1	0	<div></div>	<div></div>

- View task run details

Click **Details** on the right of the topic to check the topic details.

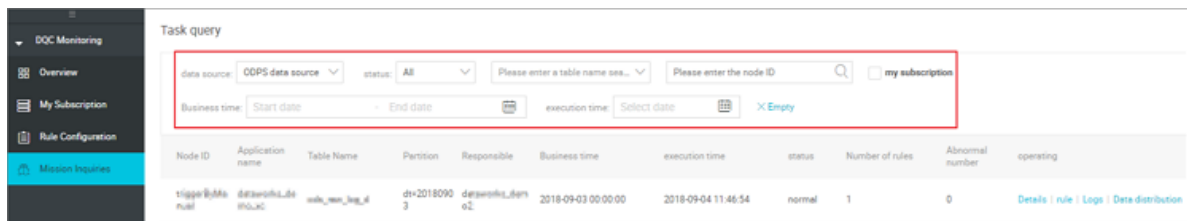
- Viewing rule configurations

Click the **Rule** to the right of the topic to enter the rule configuration page of The datahub data source, view or modify the rules created by the current topic. See for details [Rules configuration for DataHub data source](#).

6.6.2 View ODPS data source tasks

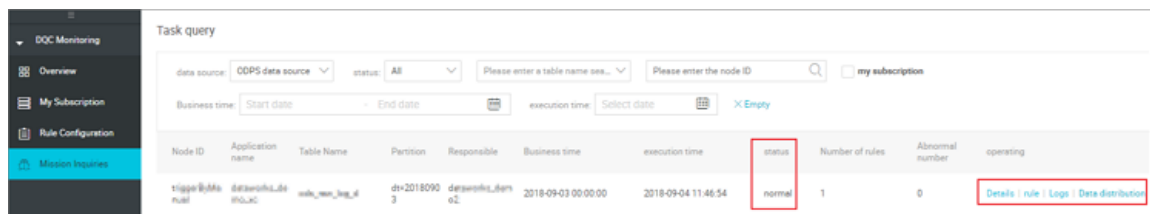
The task query module allows you to query and view rule verification results. Rule run is the task run, where the rule run record can be viewed in the **Mission Inquiries** module.

1. Visit the Data Quality Center, click **Mission Inquiries**, and enter the query page.
2. Select the **ODPS data source** and, according to the search box, enter content to locate exactly the table you want to find.



- Display the task running state

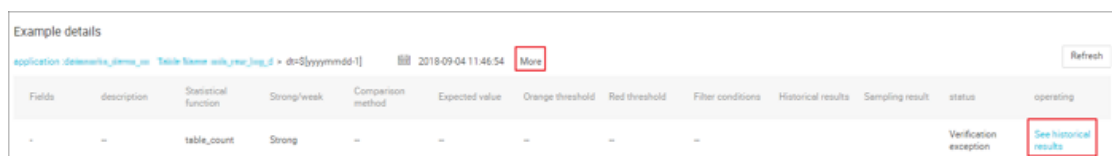
You can view the task execution status, number of rules, and number of exceptional rules in the task list. By clicking the hyperlinks on the right side, you can go to the relevant pages, and view details and make modifications.



- View Details of the partition expression

Click the **details** of the corresponding task to enter the instance details page. This page shows the running status of all rules created for the current partition expression.

- Click **More** to view information about the data source, app name, node ID, and owner.
- Click **view history** after the corresponding field to view the running records after each schedule.



- Viewing rule configurations

Click **Rules** after the corresponding task to jump to the rule configuration page. On this page, you can view and modify existing partition expressions and rules. See for details [Rules Configuration for ODPS data source](#).

- View log

Click the **log** after the corresponding task to view the running log for the current task.

- View data distribution

Click **data distribution** after the corresponding task to view the task from creation to date, the situation of each run.

6.7 Template rule

Currently, Data Quality Center (DQC) has 36 template rules every of which is described in this article.

Fluctuation calculation

$$\text{Fluctuation} = (\text{Sample} - \text{Reference value}) / \text{Reference value}$$

Fluctuation variance calculation

$$(\text{Current sample} - \text{historical N-day average values}) / \text{standard deviation}$$

Glossary

- Sample: The value of the specific samples collected per day, such as the number of rows in the SQL task table, one-day fluctuation detection. Sample is the number of partitions of the table in the current day.
- Reference value: Comparison of historical samples.
 - For example, rule is the number of rows in the SQL task table and one-day fluctuation detection, then the reference value is the number of partitions of the table generated in the previous day.
 - For example, rule is the number of rows in the SQL task table and seven-day fluctuation detection, then the reference value is the average data value in rows of the table for the previous seven days.

Verification logic

Currently, Data Quality only supports **Fluctuation detection value** and **Comparison of fixed value** verification methods.

Verification method	Verification logic
---------------------	--------------------

Fluctuation detection value	<ul style="list-style-type: none"> • If the absolute value of the check value is less than or equal to the orange threshold, it returns normal. • If the absolute value of the check value does not meet the first condition and is less than or equal to the red threshold, orange alarm is triggered. • If the check value does not meet the second condition, red alarm is triggered. • If there is no orange threshold, only two cases are possible: red alarm and normal. • If there is no red threshold, only two cases are possible: orange alarm and normal. • If none of them is filled, red alert is triggered, as the front end is not allowed to leave two thresholds blank.
Comparison of fixed value	<ul style="list-style-type: none"> • According to the check expression, calculate s opt expect, returns Boolean value, opt supports greater than, less than, equal to, greater than or equal to, less than or equal to, not equal to. • According to the preceding formula, if true, it returns normal, otherwise, red alarm is triggered.

Template rule

Template level	Template name	Description
1	The average value of the field, fluctuation compared to the one day, one week, one month before.	Take the average value of this field, compare with the one-day, seven-day, one-month period , calculate the fluctuation. Then compare it with the threshold, if there is an alarm, it is triggered.
2	The summary value of the field, fluctuation compared to the one day, one week, one month before.	Take the sum value of this field, compare with the one-day, seven-day, one-month period, calculate the fluctuation. Then compare it with the threshold, if there is an alarm, it is triggered .
3	The minimum value of the field, fluctuation compared to the one day, one week, one month before.	Take the minimum value of this field, compare with the one-day, seven-day, one-month period , calculate the fluctuation. Then compare it with the threshold, if there is an alarm, it is triggered.

4	The maximum value of the field, fluctuation compared to the one day, one week, one month before.	Take the maximum value of this field, compare with the one-day, seven-day, one-month period, calculate the fluctuation. Then compare it with the threshold, if there is an alarm, it is triggered.
5	The number of unique values in the field.	Count the number after removing duplicates, then compare with an expected number, that is, fixed value verification.
6	The number of unique values in the field, volatility compared to the one day, one week, one month before.	Count the number after removing duplicates, compare with one day, one week, one month, that is, fixed value verification.
7	The number of rows in the table, fluctuation compared to the one day, one week, one month before.	Compare the number of rows in the table collected one day, one week, and one month before, and compare the fluctuation.
8	The number of null values in the field.	The number of null values in this field compare to the fixed value.
9	The number of null values in the field / Total number of rows.	Calculate the number of null values and the total number of rows to get a rate, then compare with a fixed value. Note: The fixed value is a decimal.
10	The number of duplications in the field / Total number of rows.	The rate of the number of repeated values to the total number of rows, then compare with a fixed value.
11	The number of duplicated values in the field.	The total number of rows minus the number after removing duplicates (that is the number of duplicated values in the field), and the number of duplicated values compared to the fixed value.
12	The number of unique values in the field / Total number of rows.	The rate of the number of unique values to the total number of rows, then compare with a fixed value.
13	The average value of the field, fluctuation compared to the one day before.	Take the average value of the field, compare with the previous period. Calculate the fluctuation, then compare with a threshold value.
14	The summary value of the field, fluctuation compared to the one day before.	Take the sum value of this field, compare with the previous period. Calculate the fluctuation, then compare with a threshold value.

15	The minimum value of the field, fluctuation compared to the one day before.	Take the maximum value of this field, compare it to the one day before. Calculate the volatility, then compare with a threshold value.
16	The maximum value of the field, fluctuation compared to the one day before.	Take the maximum value of this field, compare it to the one day before, calculate the fluctuation, then compare with a threshold value.
17	The summary value of the field, fluctuation compared to the previous period.	Take the sum value of this field, compare it with the previous period, calculate the fluctuation. Then compare it with the threshold, if there is an alarm, it is triggered.
18	The minimum value of the field, fluctuation compared to the previous period.	Take the minimum value of this field, compare it with the previous period, calculate the volatility. Then compare it with the threshold, if there is an alarm, it is triggered.
19	The maximum value of the field, fluctuation compared to the previous period.	Take the maximum value of this field, compare it with the previous period, calculate the fluctuation. Then compare it with the threshold, if there is an alarm, it is triggered.
20	Table size (bytes) is unchanged, compared to the previous period.	Table size (bytes) is unchanged, compared to the previous period.
21	Table size (bytes) has changed, compared to the previous period.	Table size (bytes) has changed, compared to the previous period.
22	The number of rows in the table has changed, compared to the previous period.	The number of rows in the table has changed, compared to the previous period.
23	The number of rows in the table is unchanged, compared to the previous period.	The number of rows in the table is unchanged, compared to the previous period.
24	Table size, difference value compared to the previous period (bytes).	Table size, difference value compared to the previous period (bytes).
25	The number of rows in the table, difference value compared to the previous period.	The reference value is the number of partitions of the table generated in the previous period. Compare to the number of table rows collected on the current day, then compare the difference value.
26	The number of rows in the table.	The number of rows in the table.
27	Table space size (bytes).	Table space size (bytes).

28	The number of rows in the table, difference value compared to one day before.	The reference value is the number of partitions of the table generated one day before. Compare to the number of table rows collected on the current day, then compare the difference value.
29	Table space size, difference value compared to one day before (bytes).	Table space size, difference value compared to one day before (bytes).
30	Table space size, fluctuation compared to the one day before.	The template is the fluctuation of the table size monitoring. The sample is compared with the quota sample of the previous day. If the orange threshold is 5% and the red threshold is 10%, the orange alarm is triggered when the fluctuation is greater than 5% and less than or equal to 10%. The red alarm is triggered when the orange threshold is greater than 10%.
31	Table space size, fluctuation compared to the one week before.	The template is the fluctuation of the table size monitoring. The sample is compared with the quota sample of the previous week. If the orange threshold is 5% and the red threshold is 10%, the orange alarm is triggered when the fluctuation is greater than 5% and less than or equal to 10%. The red alarm is triggered when the orange threshold is greater than 10%.
32	Table space size, fluctuation compared to the one month before.	The template is the fluctuation of the table size monitoring. The sample is compared with the quota sample of the previous month. If the orange threshold is 5% and the red threshold is 10%, the orange alarm is triggered when the fluctuation is greater than 5% and less than or equal to 10%. The red alarm is triggered when the orange threshold is greater than 10%.
33	The number of rows in the table , average fluctuation value compared to the last seven days.	The reference value is the average value of the number of table rows in the last seven days.
34	The number of rows in the table , average fluctuation value compared to the last thirty days.	The reference value is the average value of the number of table rows in the last thirty days.

35	The number of rows in the table, fluctuation compared to the one day before.	The reference value is the number of partitions of the table generated one day before. Compare to the number of table rows collected on the day, then compare the fluctuation.
36	The number of rows in the table, fluctuation compared to the one week before.	The reference value is the number of partitions of the table generated one week before. Compare to the number of table rows collected on the current day, then compare the fluctuation.
37	The number of rows in the table, fluctuation compared to the one month before.	The reference value is the number of partitions of the table generated one month before. Compare to the number of table rows collected on the current day, then compare the fluctuation.
38	The number of rows in the table, the first day of the current month fluctuation compared to the one day, one week, one month before.	Compare the number of table rows collected on the first day of the current month to one day, one week, one month before, and compare the fluctuation.
39	The number of rows in the table, fluctuation compared to the previous period.	The reference value is the number of partitions of the table generated in the previous period. Compare to the number of table rows collected on the current day, and compare the fluctuation.
40	Discrete value monitoring (number of packets)	The number of packets is compared with a fixed value.
41	Discrete value monitoring (group number fluctuation)	The number of divisions for fluctuation detection, one day, seven days, one month ago that day the number of groups is the benchmark.
42	Discrete value monitoring (state value)	As in select count (*) from table group by table .id, the value of each group after grouping is compared to a certain number.
43	Discrete value monitoring (state value and fluctuation of state value)	Like select count (*) from table group by table .id, it compares the value of each group after grouping with a certain number; and if the number of groupings increases, it will alarm, without alarming.

7 Data management

7.1 Introduction

The Data Management module of the Alibaba Cloud DTplus platform displays the global data view and metadata details of an organization, and enables operations such as permission management, data lifecycle management, and approval and management of data table/resource/function permissions.

such as:

[*Search for data*](#)

[*Apply for data permissions*](#)

[*Create a table*](#)

[*Collection table modifying Life Cycle*](#)

[*Modify a table structure*](#)

[*Hide a table*](#)

[*Change a table owner*](#)

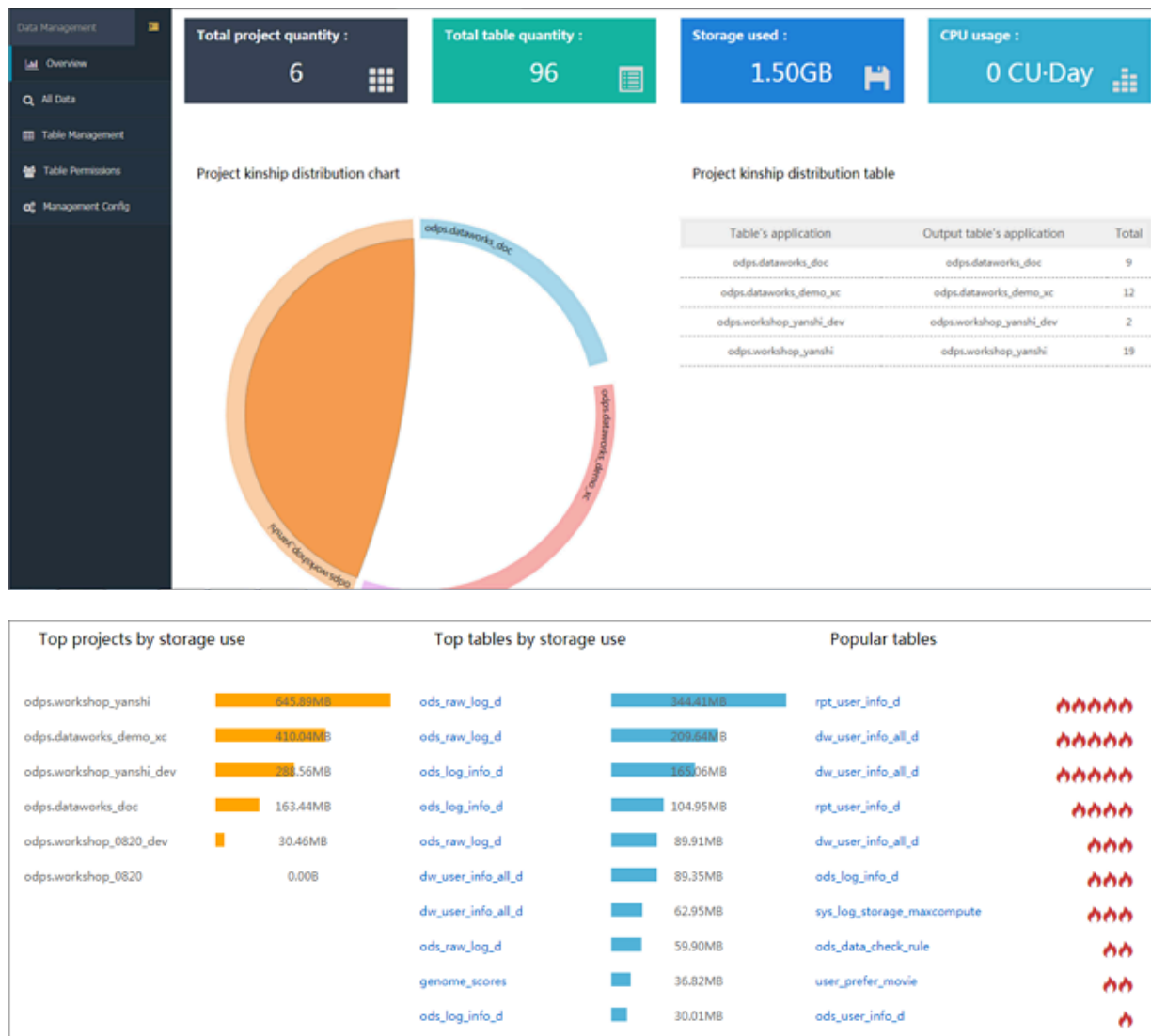
[*Delete a table*](#)

[*View the table details*](#)

[*Category navigation configuration*](#)

7.2 Overview

You can go to the global overview page through **Data Management** > **Overview**, the statistics on this page are measured on the premises of the entire organization, at the same time, the data information for the entire page is produced offline, that is, the data information for the page is yesterday's statistics.



List items description:

- Total project quantity, Total table quantity, Storage used, CPU usage: From an organizational perspective, the number of project spaces, data tables, data tables used by the data table, and the storage used by the task runtime. calculation (CPU/minute or second, etc).
- Project kinship distribution chart: From an organizational perspective, the network is used to describe the relationship between project spaces, the arc represents the project space, and the relationship between the two project spaces is connected if there is a blood relationship.
- Project kinship distribution table: From an organizational perspective, the left side is the project space in which the upstream table is located, to the right is the project space to which the downstream table belongs, with the total amount representing the number of blood relationships that exist for the two project spaces.
- Top projects by storage use: The top ten projects, in terms of storage spaces used in the organizational perspective.

- Top tables by storage use: From an organizational point of view, the display data table occupies the top 10 of the storage volume, you can click the specific table name to jump to the table details page.
- Popular tables: From an organizational perspective, the list of data tables with the most cited numbers displays the top 10, you can click the specific table name to jump to the table details page.

7.3 All data

In the organization, to search for the data tables (of multiple projects) you must log on to the **Data Management > All Data** page. Search for the tables by selecting the filter conditions and entering the table name in the search box on the All Data page.

Category: All

Application: All

Enter Search

bank_data [Apply permissions](#)

Application: odps.dataworks_doc Owner: dataworks_demo2 Last updated: 2018-08-27 17:24:04

Description:

Category attributes: Unclassified tables

bank_data1 [Apply permissions](#)

Application: odps.dataworks_doc Owner: dataworks_demo2 Last updated: 2018-08-27 17:16:14

Description:

Category attributes: Unclassified tables

You can follow any of the following three ways:

- Select a category to view all the tables under the selected category.
- Select a project name: View all the tables under the selected project. This can be used with the category filtering condition.
- Search condition: Enter the table name in the search box to for a search (supports fuzzy search by table name), and search by note is also supported.

7.4 Table detail page

On the table detail page you can view the basic information, storage information, field information, partition information, output information, change history, kinship information, and data preview of the table. To view the table details, click the name of a data table from the **Table Management** module lists.

odps_result [★Add to favorites](#) [Apply permissions](#) [Return all lists](#)

Basic table information

Table name: odps.dataworks_doc.odps_result

Chinese name: -

Project name: DataWorks_DOC

Owner: dataworks_demo2

Description: -

Permission status: Read permission

Other table information

Physical storage capacity: -

Lifecycle: Permanent

Is partition table: Yes

Table creation time: 2018-08-31 15:45:56

Last DDL modification time: 2018-08-31 15:45:56

Field information

Generate table creation statement

Non-partition field:

SN	Field name	Type	Description
1	education	STRING	Education
2	num	BIGINT	Num

Partition field:

SN	Field name	Type	Description
3	dt	STRING	-

Note: Regular daily update, not real-time data.

Add tables to favorites

Click **Add to favorites** in the upper corner of the page to add the table to your favorite list. You can view such tables in **Table Management > My Favorite Tables**.

odps_result [★Add to favorites](#) [Apply permissions](#) [Return all lists](#)

Basic table information

Table name: odps.dataworks_doc.odps_result

Chinese name: -

Project name: DataWorks_DOC

Owner: dataworks_demo2

Description: -

Field information

Generate table creation statement

Non-partition field:

SN	Field name	Type	Description
1	education	STRING	Education
2	num	BIGINT	Num

Application Permissions

You can apply for permissions for the current table on the table details page. The permissions can be applied for by the user himself/herself, or by someone else on behalf of the user.

odps_result [★Add to favorites](#) [Apply permissions](#) [Return all lists](#)

Basic table information

Table name: odps.dataworks_doc.odps_result

Chinese name: -

Project name: DataWorks_DOC

Owner: dataworks_demo2

Description: -

[Generate table creation statement](#)

Non-partition field:

SN	Field name	Type	Description
1	education	STRING	Education
2	num	BIGINT	Num

Basic table information

The basic information of a table includes the table name, the Chinese name of the table, the Alibaba Cloud DTplus platform project name, the owner name, description, and permission status (offline processed data, lagging by one day).

odps_result [★Add to favorites](#) [Apply permissions](#)

Basic table information

Table name: odps.dataworks_doc.odps_result

Chinese name: -

Project name: DataWorks_DOC

Owner: dataworks_demo2

Description: -

Permission status: Read permission

Other table information

Physical storage capacity: -

Lifecycle: Permanent

Is partition table: Yes

Table creation time: 2018-08-31 15:45:56

Last DDL modification time: 2018-08-31 15:45:56

Physical storage capacity

The storage information of a table includes the physical storage capacity (data lagging by one day), lifecycle, whether the table is a partition table, the table creation time, the last DDL modification time, and the last data modification time.

Physical storage capacity:	-
Lifecycle:	Permanent
Is partition table:	Yes
Table creation time:	2018-08-31 15:45:56
Last DDL modification time:	2018-08-31 15:45:56

Field information

The field information of a table includes a field name, type, whether the field is a partition field, and description. You can also click Generate table creation statement to generate the DDL statement of the table.

Field information	Partition information	Output information	Change history	Kinship information
Generate table creation statement				
Non-partition field:				
SN	Field name	Type	Description	
1	uid	STRING	UserID	
2	region	STRING	Region , get based on ip	
3	device	STRING	Client type	
4	pv	BIGINT	pv	
5	gender	STRING	Gender	
6	age_range	STRING	Agenrange	
7	zodiac	STRING	Zodiac	

Partition information

The Partition information module displays the current partition of the table, including the partition name, creation time, storage capacity, and record quantity.

Field information	Partition information	Output information	Change history	Kinship information	preview data	
Partition name		Creation time		Storage capacity		No. of records
dt=20180830		2018-08-31 09:49:05		0.00B		0
Note: Regular daily update, not real-time data.						

Output information

The Output information module shows which task outputs the table/partition, including the running time (in seconds) and the end time of data output in the table partition. You can select the start time and end time to filter tasks within the period.



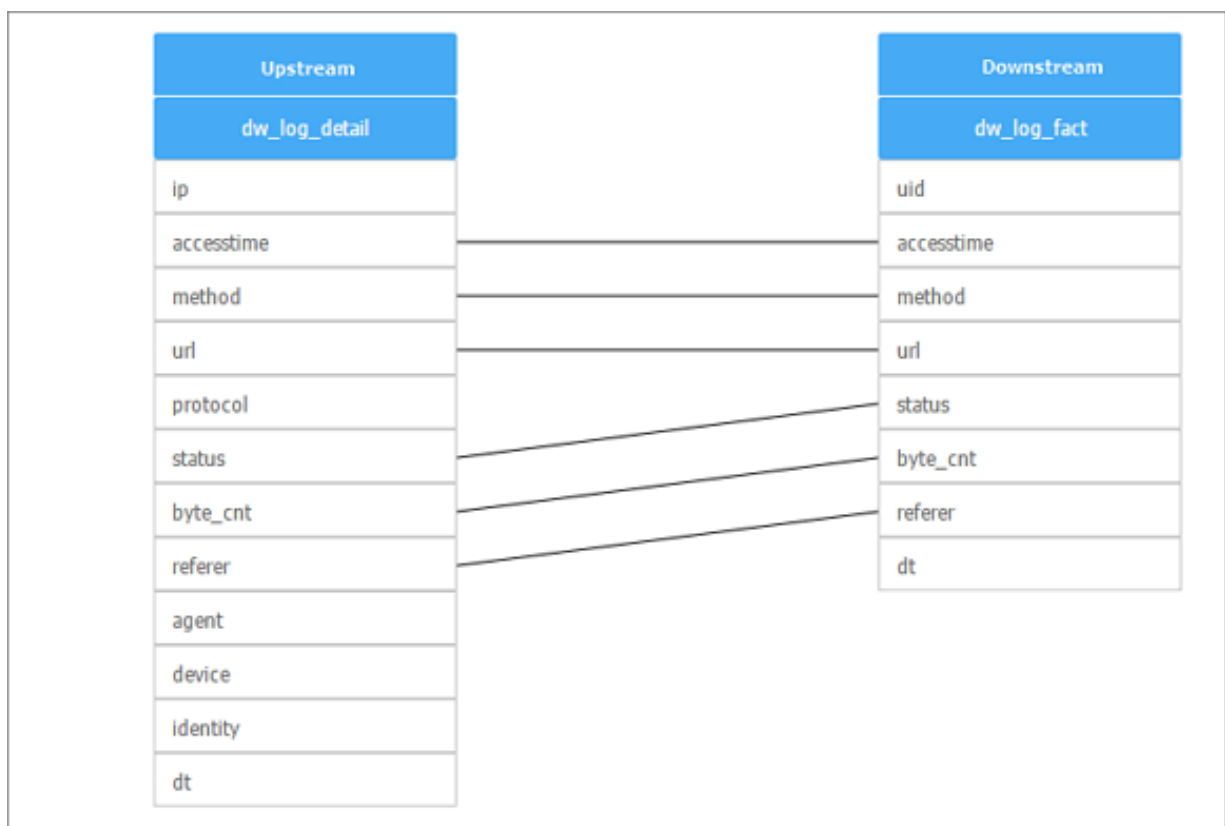
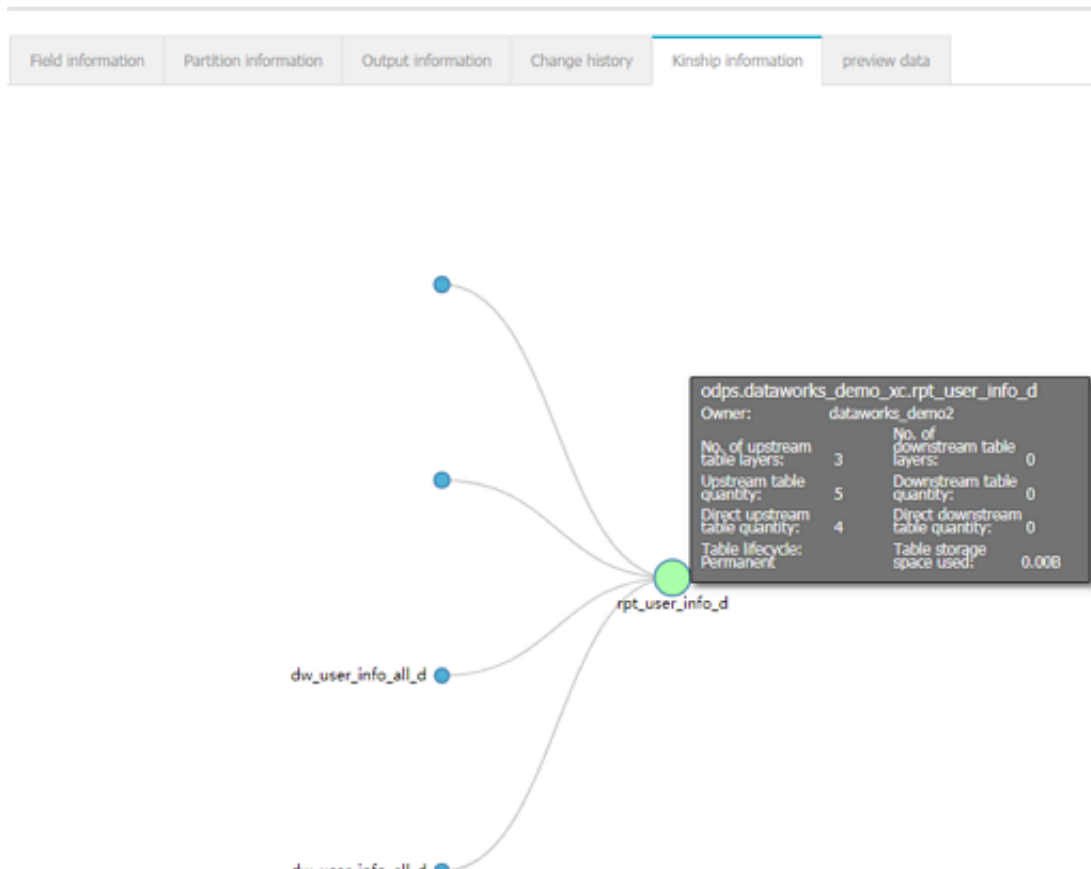
Change history

The Change history module displays the table change information, including the change history of the table and partition granularity.

Field information	Partition information	Output information	Change history	Kinship information	preview data
Granularity: All Start time: <input type="text"/> End time: <input type="text"/> Find					
Content			Granularity	Time	
Add partition:dt=20180830			PARTITION	2018-08-31 09:49:06	
New column [uid] added, with type [string], commented by [UserID]New column [region] added, with type [string], commented by [Region , get based on ip]New column [device] added, with type [string], commented by [Client type]New column [pv] added, with type [bigint], commented by [pv]New column [gender] added, with type [string], commented by [Gender]New column [age_range] added, with type [string], commented by [Agerange]New column [zodiac] added, with type [string], commented by [Zodiac]New column [dt] added, with type [string]			TABLE	2018-08-31 09:48:48	
Column [] with type [] deletedColumn [] with type [] deletedColumn [] with type [] deletedColumn [] with type [] deletedColumn [] with type [] deletedColumn [] with type [] deleted			TABLE	2018-08-26 11:26:46	

Kinship information

The Kinship information module shows the kinship information of the table data that flows through MaxCompute. The field kinship analysis is supported.



Data preview of a table

Click preview data to preview the data information of the current table.

Field information Partition information Output information Change history Kinship information preview data											
ip	uid	time	status	bytes	region	method	url		protocol	referer	device
14.136.107.248	022cee3696778	2014-02-12 03:08:03	200	92446	■■■■■	GET	/feed		HTTP/1.1		andro
106.120.203.227	d4df3947d448	2014-02-12 03:08:05	200	281306	■■■■■	GET	/feed		HTTP/1.1		unknc
69.10.179.41	d526a1e316471	2014-02-12 03:08:06	200	92446	■■■■■	GET	/feed		HTTP/1.1		unknc
81.144.138.34	ced52e0d16753	2014-02-12 03:08:09	200	21038	■■■■■	GET	/articles/1592.html		HTTP/1.1		unknc
112.64.235.91	28d2757601499	2014-02-12 03:08:11	200	15	■■■■■	GET	/wp-admin/admin-ajax.php?postviews_id=8638&action=...		HTTP/1.1		unknc
180.169.37.125	510241ebf8432	2014-02-12 03:08:11	200	92439	■■■■■	GET	/feed		HTTP/1.1		windc
61.55.185.134	5471e33b16235	2014-02-12 03:08:11	200	22667	■■■■■	GET	/articles/1379.html		HTTP/1.1	coolshell.cn	windc
204.236.179.67	73417d0610317	2014-02-12 03:08:15	304	0	■■■■■	GET	?feed=rss2		HTTP/1.1		macir
61.55.181.19	760373ae16204	2014-02-12 03:08:16	200	55144	■■■■■	GET	/feed		HTTP/1.1		windc
123.58.180.229	1ad89d77e5702	2014-02-12 03:08:16	200	121850	■■■■■	GET	/		HTTP/1.0		unknc
124.93.197.10	9f09e476e6210	2014-02-12 03:08:17	200	92446	■■■■■	GET	/feed		HTTP/1.1		andro

7.5 Apply for data permissions

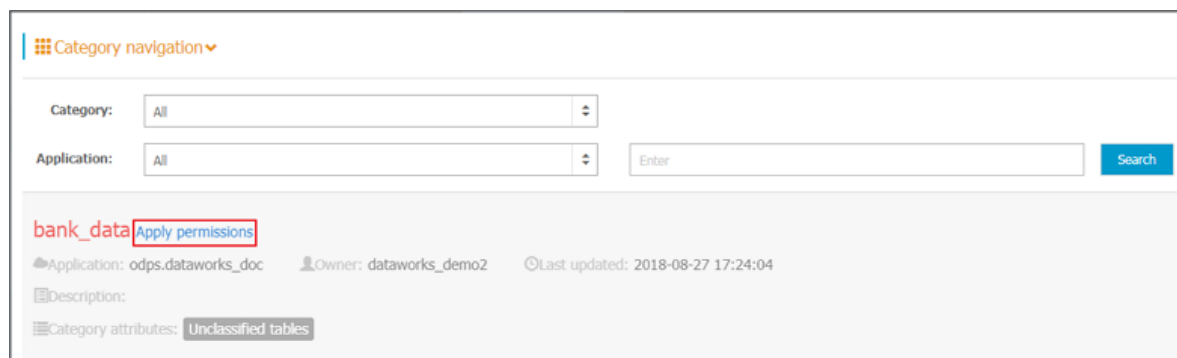
Alibaba Cloud DTplus DataWorks provides the following three data types.

- Table: Namely the data tables.
- Function: Namely the UDF, functions that can be used in SQL.
- Resource: For example, the text files and MapReduce JAR files.

These three data types have a strict permission control feature. You can use them after applying for the required permissions.

Apply for table permissions

1. Find the data table that needs to apply for permission by **Data Management > All Data** page.
2. Click **Application permissions** in the Actions column of the data table.



3. Complete the configurations in the **Apply for authorization** dialog box.

Apply for authorization

Applying for table:

`adps_dataworks_data`

* Permission owner:

☒ Self Apply

☐ Apply as agent

Permission expiration date:

1

?

* Application reason:

view data permission

Cancel

OK

Parameters:

- Permission owner: Select Self Apply or Apply as agent.
 - Self Apply: With this option selected, the permission is granted to the you, because you being the current logon user, after the application is approved.
 - Apply as agent: With this option selected, enter the account (the logon name in the upper-right corner of the system) to whom you want to apply the permission for. Once the application is approved, the permission is granted to the specified account.

- Permission expiration date: The duration of the applied table permission. The unit is in days . If not specified, the permission does not expire permanently by default. When the validity period expires, the permission is automatically revoked by the system.
 - Application reason: Enter a brief application reason for faster approval.
4. Click **OK** to submit the application and wait for approval. You can check the application status in **Permission Management > Application History**.

Apply for function and resource permissions

1. Enter the **Data Management > Query Data** page.
2. Click Apply for data permission in the upper-right corner of the list.
3. Complete the configurations in the Apply for authorization dialog box.

Parameters:

- Application type: Select Function or Resource.
- Permission owner: Select Self Apply or Apply as agent.
 - Self Apply: With this option selected, the permission is granted to the you because you being the current logon user, after the application is approved.
 - Apply as agent: With this option selected, enter the account (the logon name in the upper-right corner of the system) to whom you want to apply the permission for. Once the application is approved, the permission is granted to the specified account.

- **Project name:** Select the project name (MaxCompute project name) where the function or resource that you want to apply for permissions resides. Fuzzy searches within the organization is supported.
 - **Function name/Resource name:** Enter the name of the function or the resource in the project. Enter the full name of the resource, including the file suffix, such as my_mr.jar.
 - **Permission expiration date:** The duration of the applied permission. The unit is in days. If not specified, the permission does not expire permanently by default. When the expiration date arrives, the permission is automatically revoked by the system.
 - **Application reason:** Enter a brief application reason for faster approval.
4. Click **OK** to submit the application and wait for approval. You can check the application status in Permission Management > Application History.

7.6 Table management

The Table management module categorizes data tables and helps to manage information and operations for different tables in various categories. This enables the developers to manage their own data tables. On the Manage Data Tables page, you can follow these steps on your tables: setting the lifecycle, managing tables (including modifying the category, description, field, and partition of a table), hiding and unhiding tables, and deleting tables.

Table category

- **My favorite tables**

This section lists your favorite data tables. You can also remove the table from your favorite list.

- **My recently used tables**

This section displays the tables that you recently used. You can set the table lifecycle, manage tables (including modifying the category, description, field, and partition of a table), hiding and unhiding tables, and deleting tables. For more information, see the Manage tables section in this article.

- **Individual account table**

This section lists the data tables you have created within the organization. In other words, you are the owner of the tables as you are the current logon user.

Data table management

Refresh Create table

My favorite tables My Recently Used Tables **Individual account table** Production account table My managed tables

Enter table name/project name Search

Table name	Project	Project name	Creation time	Physical storage	Lifecycle	Favorites	Operation
odps_result	odps.dataworks_doc	DataWorks_DOC	2018-08-31 15:45:56	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_demo_mc	DataWorks流程_账单01	2018-08-31 09:48:48	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:40:01	504.05KB	Permanent	0	Lifecycle More
dw_user_info_all_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:35:11	25.59MB	Permanent	0	Lifecycle More
odi_log_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:13:19	30.01MB	Permanent	0	Lifecycle More
odi_user_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-29 16:41:13	696.28KB	Permanent	0	Lifecycle More
odi_raw_log_d	odps.dataworks_doc	DataWorks_DOC	2018-08-29 16:41:07	59.90MB	Permanent	0	Lifecycle More
result_table	odps.dataworks_doc	DataWorks_DOC	2018-08-27 17:37:43	680.00B	Permanent	0	Lifecycle More
bank_data1	odps.dataworks_doc	DataWorks_DOC	2018-08-27 17:36:14	0.00B	Permanent	0	Lifecycle More
bank_data	odps.dataworks_doc	DataWorks_DOC	2018-08-27 16:46:21	736.41KB	Permanent	0	Lifecycle More

Select all Batch hide Batch cancel hide Batch delete

1 2

You can search for the tables by table names and filter the tables according to the projects where the tables belong. The operations available here are the same as those for My recently used tables.

- Production account table

This section lists the tables with owners configured as Computing Engine Accounts (namely, the production account) with a MaxCompute access identity. The operations available here are the same as those for My recently used tables.

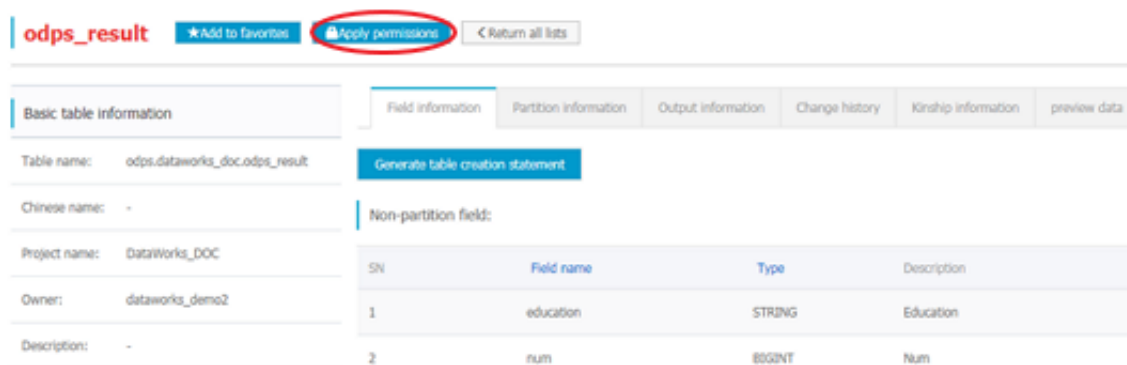
- My managed tables

If you are the project administrator, all the data tables in the project spaces you managed are displayed on this page. As an administrator, you can perform various operations on the tables such as modifying the table owner.

Manage tables

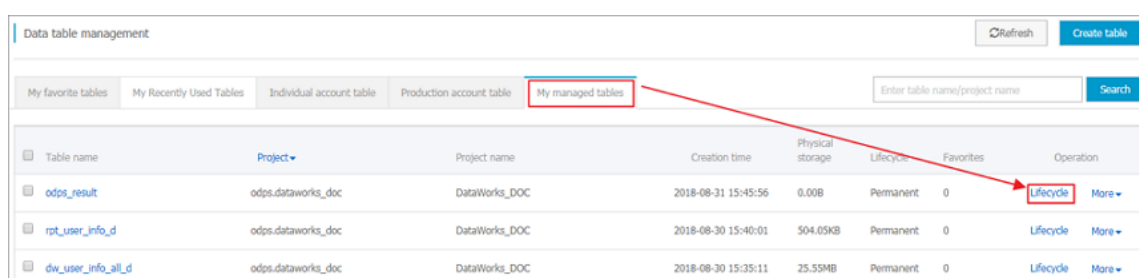
- Add tables to favorites

The Data Management module allows you to add tables to your favorites list. You can click **Add to favorites** on the table details page to add the table to your favorite list. Similarly, to remove a table from favorites list, click **remove**, on the My Favorite Tables page.

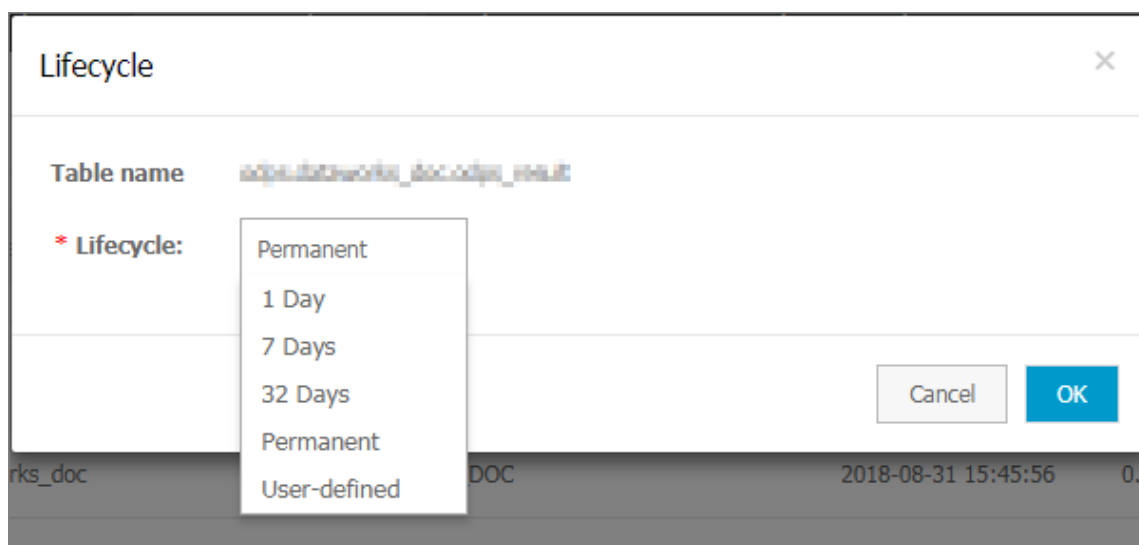


- Modify the lifecycle

1. Click the **LifeCycle** in the actions column of the list.



2. Modify the table lifecycle in the **Lifecycle** dialog box.



- Modify table structure

1. Click **More** in the Actions column of the list and select **Table Management** to modify the table structure.

Data table management Refresh Create table

My favorite tables My Recently Used Tables **Individual account table** Production account table My managed tables Search

Table name	Project	Project name	Creation time	Physical storage	Lifecycle	Favorites	Operation
odps_result	odps.dataworks_doc	DataWorks_DOC	2018-08-31 15:45:56	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_demo_xc	DataWorks流程_简单01	2018-08-31 09:48:48	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:40:01	504.05KB	Permanent	0	Lifecycle More

2. Modify the related information on the Table Management page.

Table management

Table name:

Chinese name:

Project:

Category:

Lifecycle:

Description:

Field information

Field's English name	Field type	Description	Operation
education	STRING	Education	Edit
num	BIGINT	Num	Edit

+Add field

Partition information

Field's English name	Field type	Description	Operation
dt	STRING	-	Edit

Submit

3. Click **Submit** to confirm the changes.

- Hide a table

The table owner or project administrator can hide a table to make table invisible to other members.

Click More in the Actions column of the list and select Hide to hide a table. To unhide the table, select **Unhide**.

Data table management Refresh Create table

My favorite tables My Recently Used Tables Individual account table **Production account table** My managed tables Search

Table name	Project	Project name	Creation time	Physical storage	Lifecycle	Favorites	Operation
odps_result	odps.dataworks_doc	DataWorks_DOC	2018-08-31 15:45:56	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_demo_xc	DataWorks流程_简单01	2018-08-31 09:48:48	0.00B	Permanent	0	Lifecycle Hide More
rpt_user_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:40:01	504.05KB	Permanent	0	Lifecycle More
dw_user_info_all_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:35:11	25.55MB	Permanent	0	Lifecycle More
ods_log_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:12:19	30.01MB	Permanent	0	Lifecycle More

A hidden table is marked with Hidden behind its name.

Data table management Refresh Create table

My favorite tables | My Recently Used Tables | Individual account table | **Production account table** | My managed tables

Enter table name/project name Search

Table name	Project	Project name	Creation time	Physical storage	Lifecycle	Favorites	Operation
odps_resu Hide	odps.dataworks_doc	DataWorks_DOC	2018-08-31 15:45:56	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_demo_xc	DataWorks流程_简单01	2018-08-31 09:48:48	0.00B	Permanent	0	Lifecycle More



Note:

The master account hidden table sub-accounts cannot view the hidden table content, click the appropriate prompt: table is hidden, contact the administrator or owner, sub-account hidden table master account can query the table contents.

- Modify table owner

The project administrator can modify the table owner by completing the following steps:

1. In the My managed tables section, click **More** in the Actions column of the list and select **Modify**

Owner.

Data table management

My favorite tables | My Recently Used Tables | Individual account table | Production account table | **My managed tables**

Table name	Project	Project name
odps_result Hide	odps.dataworks_doc	DataWorks_DOC
rpt_user_info_d	odps.dataworks_doc	DataWorks_DOC
dw_user_info_all_d	odps.dataworks_doc	DataWorks_DOC
ods_log_info_d	odps.dataworks_doc	DataWorks_DOC
ods_user_info_d	odps.dataworks_doc	DataWorks_DOC

2. Enter the cloud account name of the new owner in the Modify table owner dialog box. Note that the new owner must be a member of the project.

3. After the modification is complete, click Submit.

- Delete a table

1. Click **More** in the Actions column of the list and select **Delete**.

Data table management Refresh Create table

My favorite tables My Recently Used Tables Individual account table Production account table **My managed tables** Search

Table name	Project	Project name	Creation time	Physical storage	Lifecycle	Favorites	Operation
odps_result Hide	odps.dataworks_doc	DataWorks_DOC	2018-08-31 15:45:56	0.00B	Permanent	0	Lifecycle More
rpt_user_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:40:01	504.05KB	Permanent	0	Lifecycle More
dw_user_info_all_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:35:11	25.55MB	Permanent	0	Lifecycle More
ods_log_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-30 15:13:19	30.01MB	Permanent	0	Lifecycle More
ods_user_info_d	odps.dataworks_doc	DataWorks_DOC	2018-08-29 16:41:13	696.28KB	Permanent	0	Lifecycle More
ods_raw_log_d	odps.dataworks_doc	DataWorks_DOC	2018-08-29 16:41:07	59.90MB	Permanent	0	Lifecycle More

Table manager
 Modify owner
 Hide
Delete
 More

- Click **OK** to confirm the action. Once a data table is deleted, the table structure.

Confirm operation

Caution! This operation may delete the table structure and all table data and cannot be undone.

deleting table:odps.dataworks_doc.dw_user_info_all_d

OK

Cancel

Note that once you delete a table, all table data gets deleted and cannot be recovered. So, proceed with caution.

7.7 Create a table

Generally, you must create tables during data development to store the results of data synchronization and data processing. The Data Management module of Alibaba Cloud DTplus platform provides two ways to create a table.



Note:

Statement-based table creation The classification can facilitate metadata management for numerous businesses in the organization. For more information on creating tables with the maxcompute client, see [Create tables](#).

Prerequisites

- Real-name registration for cloud accounts to generate the access ID and AccessKey.

The cloud account used to build the table is the current logon account, you must have access Sid and accesskey to request a table to be built by maxcompute, so the cloud account must

have real name authentication to generate access Sid and accesskey. For more information, see [Prepare Alibaba Cloud account](#).

- Log on to Alibaba Cloud official website using the cloud account.

You must authorize the Alibaba Cloud account before creating tables. MaxCompute project owners can directly run the authorization statement to authorize the permissions. Examples are as follows:

```
use projectname; --Open a project
add user aliyun$Alibaba Cloud account; --Add an user
Grant CreateInstance,CreateTable,List ON PROJECT projectname TO
aliyun$Alibaba Cloud account; --Authorize the user
```

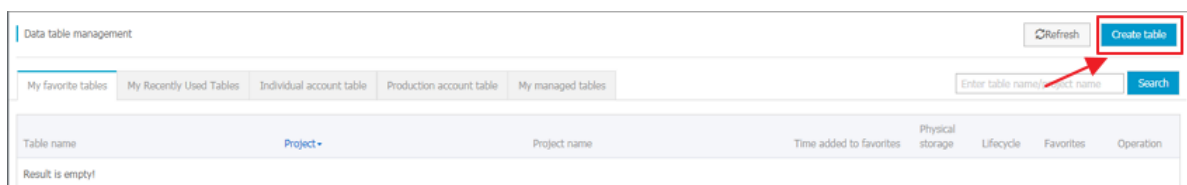


Note:

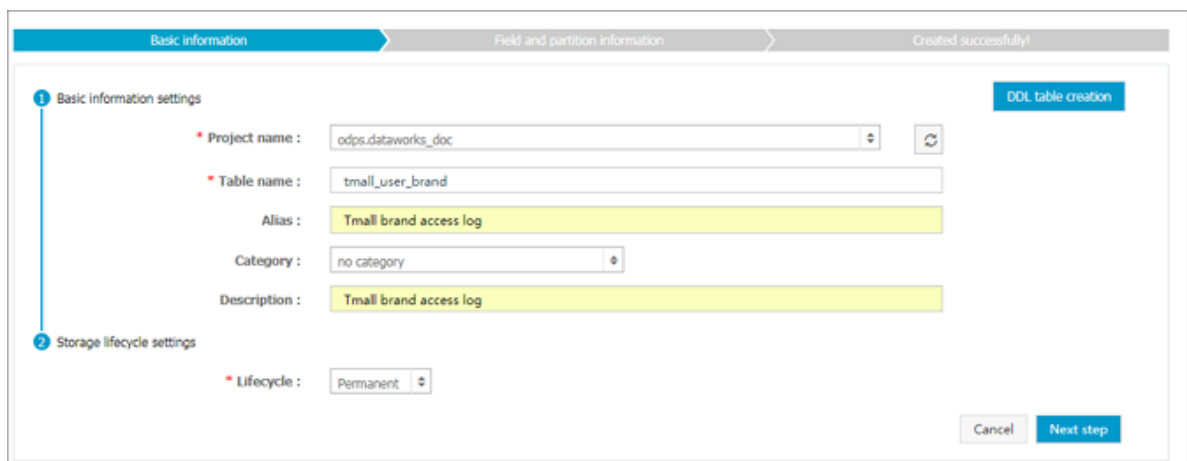
>The tables are created using the Alibaba Cloud account currently logged on, so the owner of the tables is the account currently logged on.

Visualization of creating a table

- Enter the [DataWorks management console](#) as a developer, and click Enter workspace after the corresponding project under the project list.
- Click Data Management in the upper navigation pane and navigate to Manage Data Tables page.
- Click **Create table**.



- Complete the configurations of the Basic information steps in the Create table dialog box.



Parameters:

- **Project Name:** The list shows the MaxCompute projects that the user is currently in.
- **Table Name:** It may contain letters, digits, and underscores.
- **Alias:** Chinese name of the table to be created.
- **Category:** the current table is in a category that supports a maximum of four levels. Class navigation, configuration see [Manage config](#).
- **Description:** brief description of the table to be created.
- **Lifecycle:** The lifecycle function of MaxCompute. Data in the table (or partition) that has not been updated within the period of time specified by "Lifecycle" (in days) will be cleared. Five options are available, including 1 day, 7 days, 32 days, Permanent, and User-defined.

5. Click **Next**.

6. Fill in configuration items on the Create a Table > Field and Partition Info. tab page.

- Add the field settings.
- Set the partitions.

Basic information > Field and partition information > Created successfully!

3 Field information settings

Field's English name	Field type	Description	Operation
table_name	STRING	table_level	Move up Move down Delete
age	DOUBLE	title	Move up Move down Delete
zodisc	STRING	hobby	Move up Move down Delete

+Add field

4 Set a partition: ☒ No ☐ Yes

Cancel Last step Submit

Parameters:

- **Field English Name:** English name of a field, which may contain letters, digits, and underscores.
- **Field type:** MaxCompute data type (string, bigint, double, datetime, or boolean).
- **Description:** detailed description of a field.
- **Operation:** The options include Move Up, Move Down, and Delete.
- **Whether to Set Partitions:** If you select "Yes", you need to configure the partition key information. The string and bigint data types are supported.

7. Click **Submit**.

Upon successful commit of the new table, the system will automatically jump back to the data table management interface, click the tables that I manage to view the new table.

Statement to create a table

1. Enter the [DataWorks management console](#) as a developer, and click Enter workspace after the corresponding project under the project list.
2. On the top menu bar, choose **Data Management**. Navigate to Table Management on the left.
3. Click **new table**, and then select **DDL build table**.
4. Write DDL statements to create a table. Examples are as follows:

```
create table if not exists table2
(
  id string comment 'user ID',
  name string comment 'user name'
) partitioned by(dt string)
LIFECYCLE 7;
```

5. Click Submit and the following page appears:

The screenshot shows the 'DDL table creation' interface in DataWorks. The interface is divided into three tabs: 'Basic information', 'Field and partition information', and 'Created successfully!'. The 'Basic information' tab is selected. It contains two sections: '1 Basic Information settings' and '2 Storage lifecycle settings'. Under '1 Basic Information settings', there are fields for 'Project name' (filled with 'odps.dataworks_doc'), 'Table name' (filled with 'tmail2'), 'Alias' (placeholder 'Enter Alias name'), 'Category' (filled with 'no category'), and 'Description' (placeholder 'Enter description'). Under '2 Storage lifecycle settings', there is a 'Lifecycle' field (filled with '7Day'). A 'DDL table creation' button is in the top right. 'Cancel' and 'Next step' buttons are in the bottom right.

Except Alias, Category, and Lifecycle, all the other configuration items on the Basic Information page are automatically filled in. You need to edit and provide the names and the security levels of fields on the Field and Partition Information page.

Basic information > Field and partition information > Created successfully!

3 Field information settings

Field's English name	Field type	Description	Operation
id	STRING	Userid	Move up Move down Delete
name	STRING	Username	Move up Move down Delete

+Add field

4 Set a partition: ☐ No ☒ Yes

Partition information settings

Field's English name	Field type	Description	Operation
dt	STRING		Delete

+Add partition

Cancel Last step Submit

6. Fill in the remaining configuration items on the Basic Info. tab page.

Basic information > Field and partition information > Created successfully!

1 Basic information settings

* Project name : odps.dataworks_doc

* Table name : table2

Alias : testtable

Category : no category

Description : newtesttable

DDL, table creation

2 Storage lifecycle settings

* Lifecycle : Permanent

Cancel Next step

7. Click **Next step**.

8. Click **Submit**.

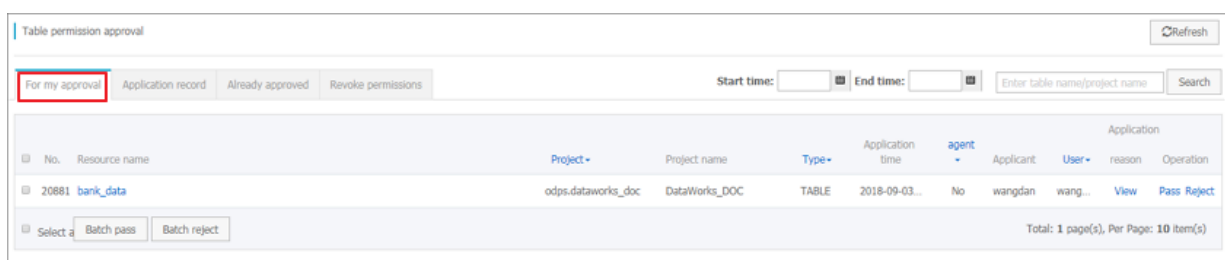
After the created table is submitted, the system automatically jumps back to the Data Table Management page. Click **My Tables** to view the created table.

7.8 Permission management

The Permission Management module is mainly used to manage the applications for permissions of tables, resources, and functions. It includes the following submodules: **For my approval**, **Application record**, **Already approved**, and **Revoke permissions**.

For my approval

In the **For my approval** module, you can view and approve the pending applications for permissions of tables, resources, and functions in all the projects where the current access account is as the **administrator**.



The screenshot shows the 'Table permission approval' interface. The 'For my approval' tab is selected and highlighted with a red box. The interface includes a search bar, filters for start and end time, and a table of pending applications. The table has columns for No., Resource name, Project, Project name, Type, Application time, agent, Applicant, User, reason, and Operation. A single application is listed with ID 20881, resource 'bank_data', project 'odps.dataworks_doc', and type 'TABLE'. The application time is '2018-09-03...'. The applicant is 'wangdan' and the user is 'wang...'. The operation options are 'View' and 'Pass Reject'. At the bottom, it says 'Total: 1 page(s), Per Page: 10 item(s)'.

No.	Resource name	Project	Project name	Type	Application time	agent	Applicant	User	reason	Operation
20881	bank_data	odps.dataworks_doc	DataWorks_DOC	TABLE	2018-09-03...	No	wangdan	wang...		View Pass Reject

Application record

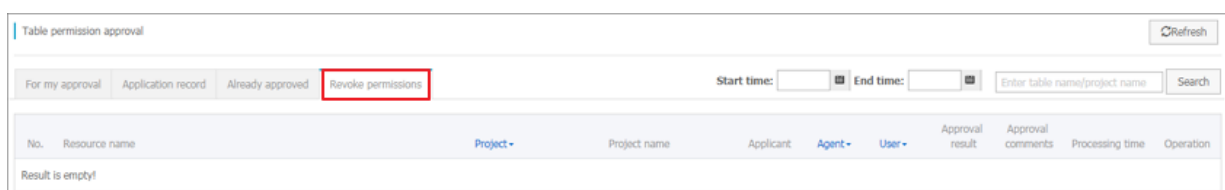
In the **Application record** module, you can view the permission application history of the current access account.

Already approved

In the **Already approved** module, you can view the processed applications for permissions of tables, resources, and functions in all the projects where the current access account is as the **administrator**.

Revoke permissions

In the **Revoke permissions** module, you can view and revoke the approved applications for permissions of tables, resources, and functions in all the projects where the current access account is as the **administrator**.




The screenshot shows the 'Table permission approval' interface with the 'Revoke permissions' tab selected and highlighted with a red box. The interface includes a search bar, filters for start and end time, and a table that is currently empty, displaying 'Result is empty!'. The table headers are No., Resource name, Project, Project name, Applicant, Agent, User, Approval result, Approval comments, Processing time, and Operation.

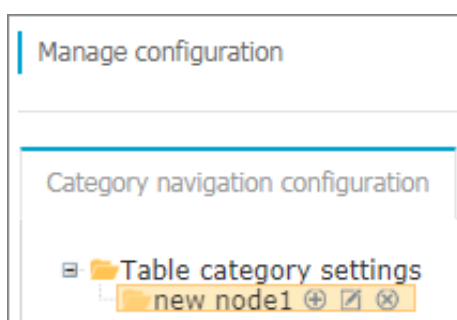
No.	Resource name	Project	Project name	Applicant	Agent	User	Approval result	Approval comments	Processing time	Operation
Result is empty!										

7.9 Manage config

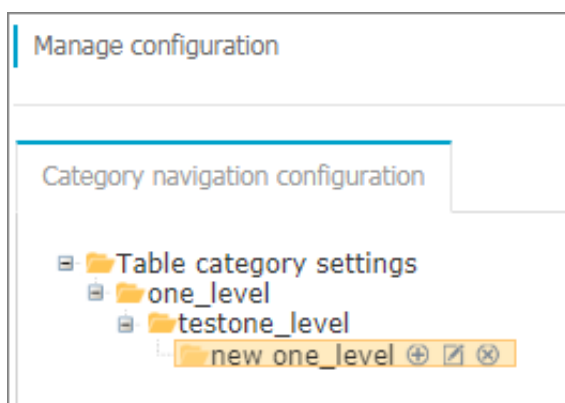
You can configure the categories of a newly created table on the Category Navigation Configuration page (organization administrator permission is required for this operation).



Procedure

1. Enter the [DataWorks console](#) as a developer, and click Enter Project to enter the project management page.
2. Click **Data Management** from the upper menu and go to the **Manage Config** page.
3. Click  after the Table category settings to add level 1 category.

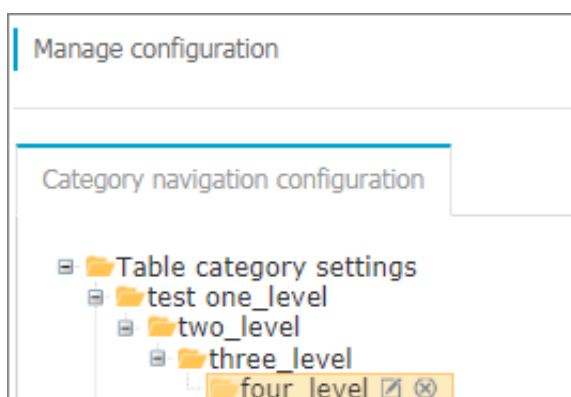


4. Click  after the level 1 category to add level 2 category.

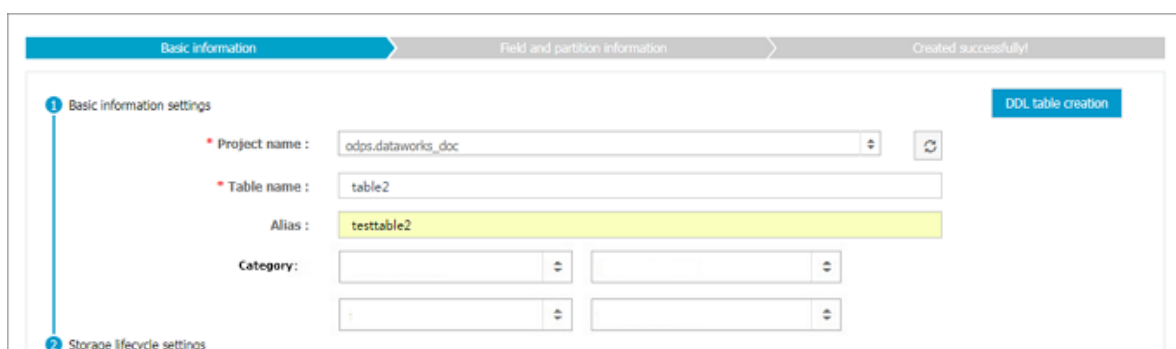


You can add up to four levels of categories.  indicates editing the category name, and  indicates deleting the category.

After the configurations, you can select the configured categories on the New Table page, as shown in the following figure:



The categories of a newly created table are as follows:



8 DataService studio

8.1 DataService studio overview

DataService Studio aims to build a data service bus to help enterprises centrally manage private and public APIs. DataService Studio allows you to quickly create APIs based on data tables and register existing APIs with the DataService Studio platform for centralized management and release. In addition, DataService Studio is connected to API Gateway. You can deploy APIs to API Gateway with one-click. DataService Studio works together with API Gateway to provide a secure , stable, low-cost, and easy-to-use data sharing service.

DataService Studio adopts the serverless architecture. All you need to care is the query logic of APIs, instead of the infrastructure such as the running environment. DataService Studio prepares the computing resources for you, supports elastic scaling, and requires zero O&M cost.

Creation of data APIs

DataService Studio currently supports the use of the visualized wizard to quickly create data APIs based on tables of the relational database and NoSQL database. You can configure a data API in several minutes without writing codes. To meet the personalized query requirements of advanced users, DataService Studio provides the custom SQL script mode to allow you compile the API query SQL statements by yourself. It also supports multi-table association, complex query conditions, and aggregate functions.

API registration

DataService Studio also supports centralized management of the existing API services that you register with DataService Studio and the APIs created based on data tables. Currently only RESTful APIs can be registered. Supported request methods include GET, POST, PUT, and DELETE. Supported data types include forms, JSON data, and XML data.

API gateway

API Gateway provides API management services, including API publish, management, and maintenance, and API subscription duration management. It provides you with a simple, fast, low-cost, and low-risk method to implement microservice aggregation, frontend-backend isolation, and system integration, and opens functions and data to partners and developers.

DataService Studio has been connected to API Gateway. You can deploy any APIs created and registered in DataService Studio to API Gateway for management, such as API authorization and authentication, traffic control, and metering.

API Market

The Ali cloud API market is the most comprehensive API trading market in China, covering finance, artificial intelligence, e-commerce, transportation geography, Living Services, corporate management and the eight main categories of public affairs, thousands of API products have been sold online.

After your APIs from DataService Studio have been published to API Gateway, you can then publish them to Alibaba Cloud API Marketplace. This is an easy way to achieve financial gains for your company.

8.2 Glossary

The data services related words are explained below.

- **Data sources:** database links. Data Service accesses data through data sources. Data sources can only be configured in Data Integration.
- **Create APIs:** create APIs based on data tables.
- **Register APIs:** register existing APIs to Data Service for central management.
- **Wizard:** guides you through the procedure of API creation. This method is suitable for beginners who want to create simple APIs. You do not need to write any code.
- **Script:** allows you to create APIs by writing SQL scripts. This method supports table join queries, complex queries, and aggregate functions. This method is suitable for experienced developers who want to create complex APIs.
- **API groups:** an API group is a set of APIs for a certain scenario or for consuming a specific service. API groups are the smallest group units in Data Service, as well as the smallest units managed by API Gateway. API groups are published in Alibaba Cloud API Market as API products.
- **API Gateway:** a service provided by Alibaba Cloud to manage APIs. API Gateway supports API subscription duration management, permission management, access management, and traffic control.
- **API Market:** Alibaba Cloud API Market is the most complete and integrated domestic API trading platform established on Alibaba Cloud Market.

8.3 Generate API

8.3.1 Configure the Data Source

Before you can use the data API to generate a service, you must configure the data source in advance. Data Service allows you to obtain schema information of data tables from data sources and handle API requests.

You can configure a data source on the **data integration** > **data source** page in the dataworks console, support for different data source types and how to configure them is shown in the following table.

Data source name	Wizard mode to generate data API	Script Mode generation data API	Configuration method
RDS (ApsaraDB for RDS)	Supported	Supported	The RDS includes MySQL, PostgreSQL, and SQL Server.
DRDS	Supported	Supported	Configure DRDS data sources
MySQL	Supported	Supported	Configure MySQL data source
PostgreSQL	Supported	Supported	Configure PostgreSQL data source
SQL Server	Supported	Supported	Configuring SQL server data source
Oracle	Supported	Supported	Configure Oracle data source
AnalyticDB(ADS)	Supported	Supported	Configure the AnalyticDB data source
Table Store(OTS)	Yes	No	Configure Table Store(OTS) data source
MongoDB	Supported	No	Configure MongoDB data source

8.3.2 Overview of generating API

The Data Service currently supports faster generation of tables from relational and neosql databases through a visually configured wizard mode. data API, you don't need to have the ability to code to configure a data API in a matter of minutes. To meet the personalized query requirements of advanced users, Data Service provides the custom SQL script mode to allow you compile the API query SQL statements by yourself. It also supports multi-table association, complex query conditions, and aggregate functions.

The functions of the wizard mode and the script mode are listed as follows:

Features	Features	Wizard mode	Script Mode
Query object	Query a single data table from one data source	Supported	Supported
	Query multiple joined tables from one data source	No	Supported
Filter bar	Query for an exact number	Supported	Supported
	Query for a range of numbers	No	Supported
	Match an exact string	Supported	Supported
	Fuzzy search for strings	Supported	Supported
	Set required and optional parameters	Supported	Supported
Query results	Return the field value	Supported	Supported
	Return a mathematical calculation of field values	No	Supported
	Return an aggregate calculation of field values	No	Supported
	Display results with pagination	Supported	Supported

8.3.3 Generate API in Wizard Mode

This article will introduce you to the steps and considerations of the wizard mode generation API.

Using the wizard mode to generate data, the API is simple and easy to get started without writing any code, the API can be quickly generated by checking the configuration from the product interface. We recommend that users who do not have high requirements for the functions of the API or have little code development experience use the wizard.



Note:

Before you configure the API, configure the data source in the **Data integration > Data Source** page of the dataworks console.

Configure the API basic information

1. Navigate to the **API Service list > Generate API**.
2. Click **Wizard Mode** to fill in the API basics.

1 API Basic Information — 2 API Parameters — 3 API Testing **Next**

* API Name
Support Chinese characters, English, numbers, underline, and must start with English or Chinese characters, 4 to 50 characters

* API Group + Add API Group

* Protocol ☒ HTTP

* API Path
Support for English, number, underscore, hyphens (-), and must start with /, not more than 200 characters, etc: /user

Request

Response

* Description

Note the settings for the API grouping during configuration. An API group includes a collection of APIs that are used for a specific scenario. It is the minimum management unit in API Gateway. In the Alibaba Cloud API Market, each API group corresponds to a specific API product.

**Note:**

The set up example for API grouping is as follows:

For example, you would like to configure an API product for weather inquiry, weather search API by city name weather search API, scenic spot name search weather API and zip search weather API three kinds of APIS, then you can create an API group called a weather query , and put the above three APIs in this group. The API is shown as a weather query product when published to the market.

Of course, if your generated API is used in your own app, you can use grouping as a classification.

Currently, the build API only supports HTTP protocol, GET request mode, and JSON return type.

3. After providing the API basic information, click **Next** to go to the API parameter configuration page.

Configure API parameters

1. Navigate to the **Data source type** > **Data source name** > **Table** and select the tables that you want to configure.

**Note:**

You need to configure the data source in advance in the data set, and the data table drop-down box supports the table name search.

2. Second, specify request and response parameters.

When a data table has been selected, all fields of the table are displayed on the left. Select the fields to be used as request parameters and response parameters, then add them to the corresponding parameter list.

3. Finally, edit and complete parameter information.

Click **Edit** in the upper-right corner of the request and return parameter lists to enter the parameter information Edit page, sets the name of the parameter, sample value, default, mandatory, fuzzy match (only string type is supported) settings) and the description. The optional and description fields are required.

Field	Field Type	Index
biddate	DATE	
region	VARCHAR	
pk	BIGINT	
vk	BIGINT	
browse_size	BIGINT	

Parameter Name	Binding Field	Type	Example Value	Default Value	Required	Fuzzy Matching	Description
region	region	string			Yes	No	

Parameter Name	Binding Field	Type	Example Value	Description
browse_size	browse_size	long		

You need to pay attention to the settings that return result paging during the configuration process.

- If you do not enable the **response pagination**, the API outputs up to 500 records by default.
- If the return result may exceed 500, turn on the **response pagination** function.

The following public parameters are available only when the response pagination feature is enabled:

- Common request parameters
 - pageNum: the current page number.
 - Pagesize: The page size, that is, the number of records per page.
- Common response parameters
 - pageNum: the current page number.

- Pagesize: The page size, that is, the number of records per page.
- totalNum: the total number of records.



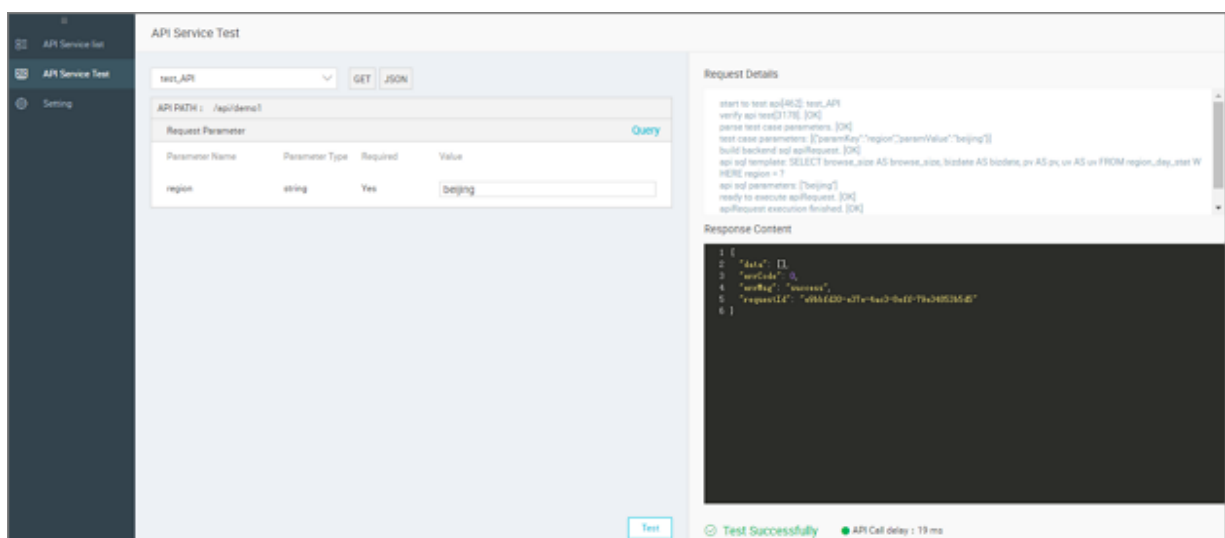
Note:

- The request parameter only supports the equivalent query, and the return parameter only supports the output of the field value as is.
- As far as possible, set an indexed field to a request parameter.
- You are allowed to specify no request parameters for an API. In that case, the pagination feature must be enabled.
- To make it easy for API callers to understand the details of an API, we recommend that you specify the sample value, default value, and description parameters of the API.
- Click on the configured API to view a list of the APIs that have been generated in the current table, avoid generating the same API.

When the configuration of the API parameters is complete, click **Next** to enter the API testing section.

API Testing

After completing configuration of API parameters, you can start the API test.

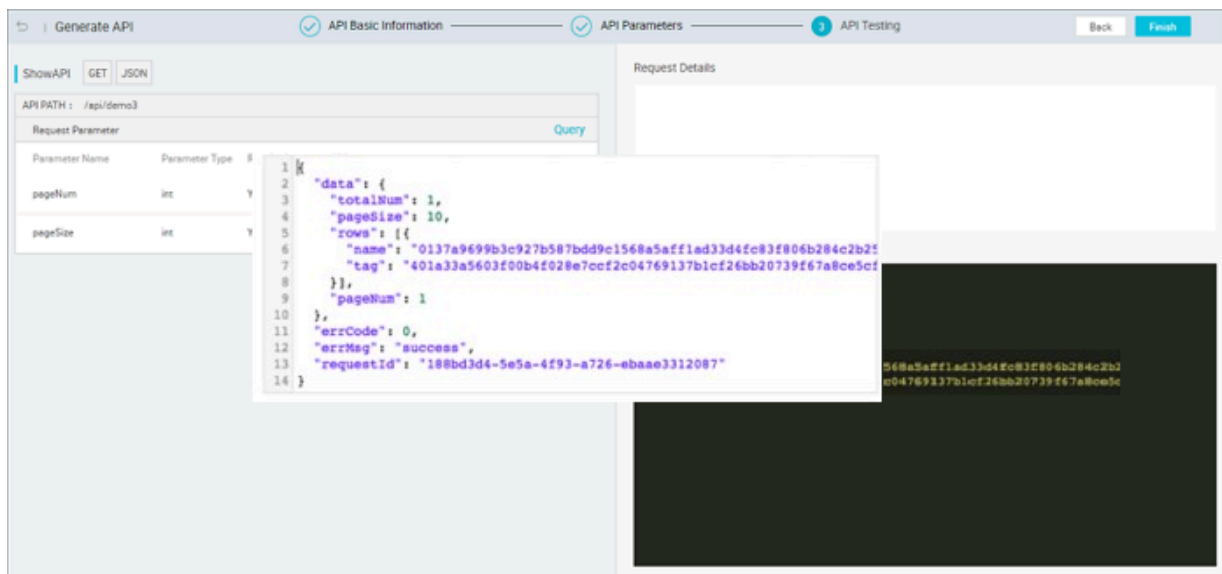


Set parameters and click **Start Test** to send the API request online. The API request details and response are displayed on the right. If the test fails, read the error message carefully and make the appropriate adjustments to test your API again.

You need to note the settings for the normal return example during the configuration process.

When testing an API, the system automatically generates exception examples and error codes.

However, normal response examples are not automatically generated. After the test succeeds, you need to click **Save as Normal Response Sample** to save the current test result as the normal response sample. If sensitive data is included in the response, you can manually edit it.



Note:

- Normal response examples provide an important reference value for the API callers. Specify an example if possible.
- The API calling delay is the delay of the current API request, which is used to evaluate the API performance. If the latency is too high, you may consider optimizing your database.

After completing the API test, click **Finish**. The data API is successfully created.

API details viewing

Back on the API service list page, click **details** in the Action column to view the details of the API. This page displays detailed information about an API from the view of a caller.

API Service Details Status: Draft API Service Test

test_API

API Basic Information

- API ID: 462
- API Group: Workshop
- Principal: suailin
- Create Time: 2018-09-04 15:57:13
- Description: API demo

HTTP API Info

- HTTP API address: http://db-server.cn-shanghai.data.aliyun-inc.com/project/79023/api/demo1
- Request: GET
- Response: JSON

Data Source Information

- Name: nds_workshop_log
- Type: mysql
- Connection: JDBC URL: jdbc:mysql://192.168.1.1:3306/nds_workshop
- Username: workshop
- Table Name: region_day_stat
- Description: nds log data sync

Request Parameters

Application-level request parameters

Parameter Name	Type	Example Value	Default Value	Required	Fuzzy Matching	Description
region	string			Yes	No	

Request Parameter

Application-level response parameters

Parameter Name	Type	Example Value	Description
browse_size	long		
bizdate	string		
pv	long		
uv	long		

Correct Response Example

8.3.4 Generate API in Script Mode

This article introduces you to the steps that script mode can take to generate the API.

To meet the needs of high-end users for personalized queries, the Data Service also provides a script pattern for customizing SQL, allows you to write your own SQL queries for the API, multi-Table Association, complex query conditions and Aggregate functions are supported.

Configure the API basic information

1. Navigate to the **API Service list > Generate API**.
2. Click **Script Mode** to fill in the API basics.

1 API Basic Information **2 API Parameters** **3 API Testing** Next

* **API Name** test_API
Support Chinese characters, English, numbers, underline, and must start with English or Chinese characters, 4 to 50 characters

* **API Group** Workshop + Add API Group

* **Protocol** ☒ HTTP

* **API Path** /api/demo
Support for English, number, underscore, hyphens (-), and must start with /, not more than 200 characters, etc: /user

Request GET

Response JSON

* **Description** API demo

Note the settings for the API grouping during configuration. An API group includes a collection of APIs that are used for a specific scenario. It is the minimum management unit in API

Gateway. In the Alibaba Cloud API Marketplace, each API group corresponds to a specific API product.

**Note:**

The set up example for API grouping is as follows:

For example, you would like to configure an API product for weather inquiry, weather search API by city name weather search API, scenic spot name search weather API and zip search weather API three kinds of APIs, then you can create an API group called a weather query , and put the above three APIs in this group. The API is shown as a weather query product when published to the marketplace.

Of course, if your generated API is used in your own app, you can use grouping as a classification.

Currently, the build API only supports HTTP protocol, GET request mode, and JSON return type.

3. After providing the API basic information, click **Next** to go to the API parameter configuration page.

Configure the API Parameters

1. Select the data source and table.

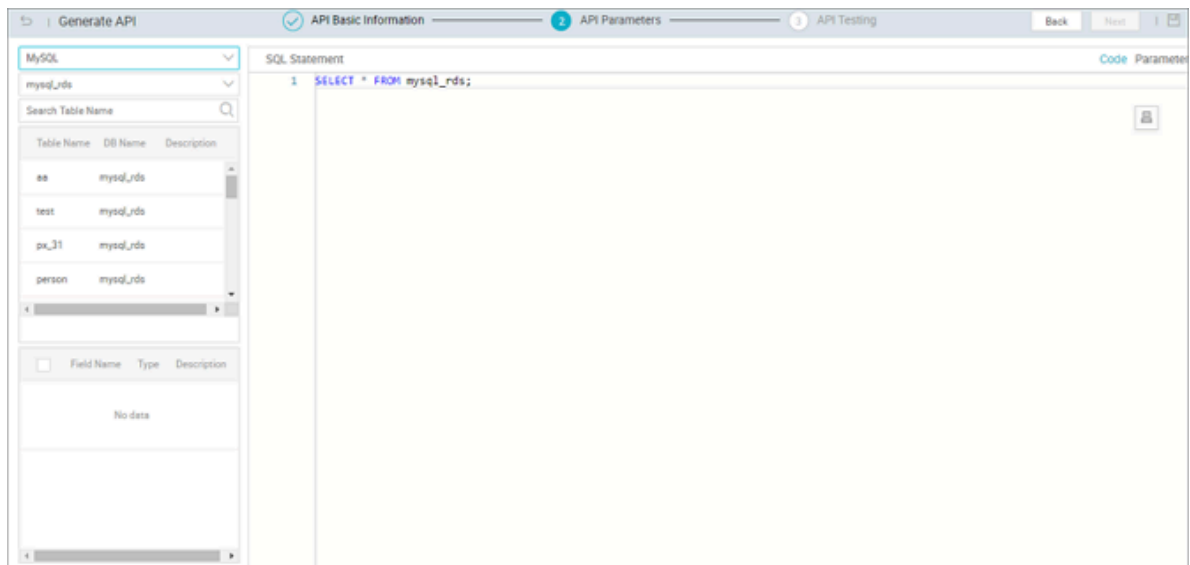
Navigate to the **data source type** > **data source name** > **data table**, click the appropriate table name in the data table list, you can view the field information for this table.

**Note:**

- You need to configure the data source in advance in the data set formation.
- You must select a data source. Table join queries across data sources are not supported.

2. Write SQL queries for the API.

You can enter the SQL code in the code box on the right side. The system supports one-click SQL function, checking fields in the list of fields, and clicking **Generate SQL**, the SQL statement for `SELECT xxx FROM xxx` is automatically generated and inserted at the right cursor.

**Note:**

- One-click SQL addition is especially useful when the number of fields is relatively large, which can greatly improve the efficiency of SQL writing.
- The field of the SELECT query is the return parameter of the API, the parameter at the where condition is the request parameter for the API, And the request parameter is identified with \$.

3. Finally, edit and complete parameter information.

After writing the API query SQL, click the **parameters** in the upper-right corner to switch to the parameter information Edit page, you can edit the type, sample values, default values, and descriptions of the parameters here, where Type and description are required.

**Note:**

To help the caller of the API get a more comprehensive understanding of the API, please complete the API parameter information as much as possible.

You need to pay attention to the settings that return result paging during the configuration process.

- If you do not enable the **response pagination**, the API outputs up to 500 records by default.
- If the return result may exceed 500, turn on the **response pagination** function.

The following public parameters are available only when the response pagination feature is enabled:

- Common request parameters
 - pageNum: the current page number.
 - Pagesize: The page size, that is, the number of records per page.
- Common response parameters
 - pageNum: the current page number.
 - pageSize: The page size, that is, the number of records per page.
 - totalNum: the total number of records.



Note:

SQL rule prompt.

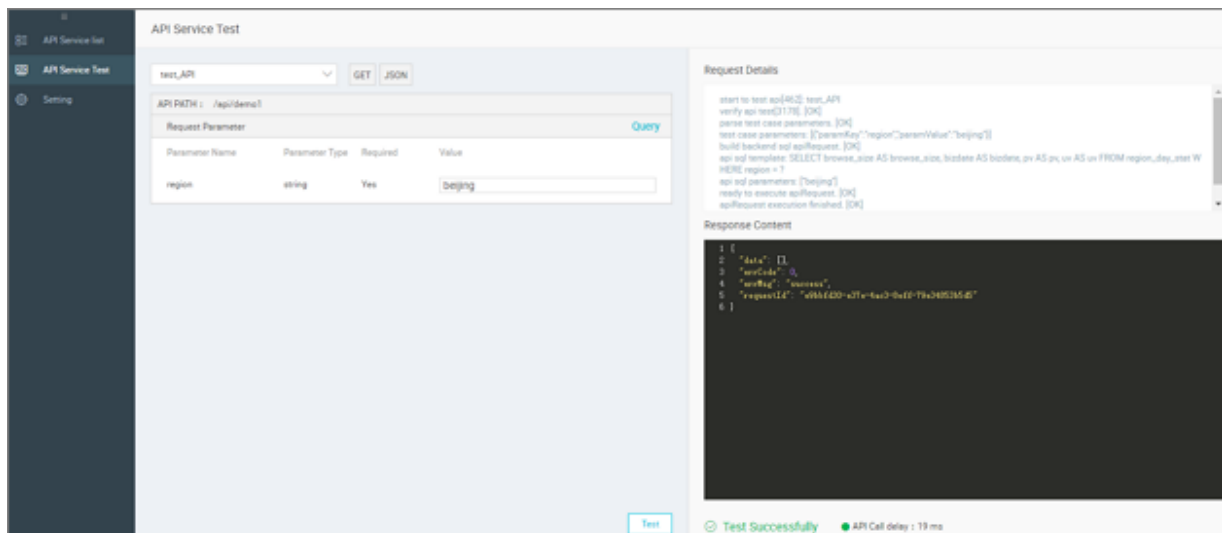
- Only one SQL statement is supported, and multiple SQL statements are not supported.
- Only the `SELECT` clause is supported. Other clauses such as `INSERT`, `UPDATE`, and `DELETE` are not supported.
- The query field for select is the return parameter for the API, the variable Param in the \${Param} in the where condition is a request parameter for the API.
- `SELECT *` is not supported, columns of the query must be specified explicitly.

- Single table queries, table join queries, and nested queries within one data source are supported.
- If the column name of the SELECT query column has a table name prefix (such as T. name), the alias must be taken as the return parameter name (such as T. name as name).
- If you use the aggregate function (min/max/sum/count, etc), the alias must be taken as the return parameter name (such as sum (Num) as total \ _ num).
- In SQL, \$ {Param} is uniform when the request parameter is replaced, contains \$ {Param} in the string }. When \$ {Param} has an escape character \, it does not do request parameter processing, processed as an ordinary string.
- Putting \$ {Param} in quotation marks is not supported, such as '\$ {ID}', 'ABC \$ {xyz} 123 ', concat ('abc ', \$ {xyz}, '123') can be passed if necessary ') implementation.

When the configuration of the API parameters is complete, click **Next** to enter the API testing section.

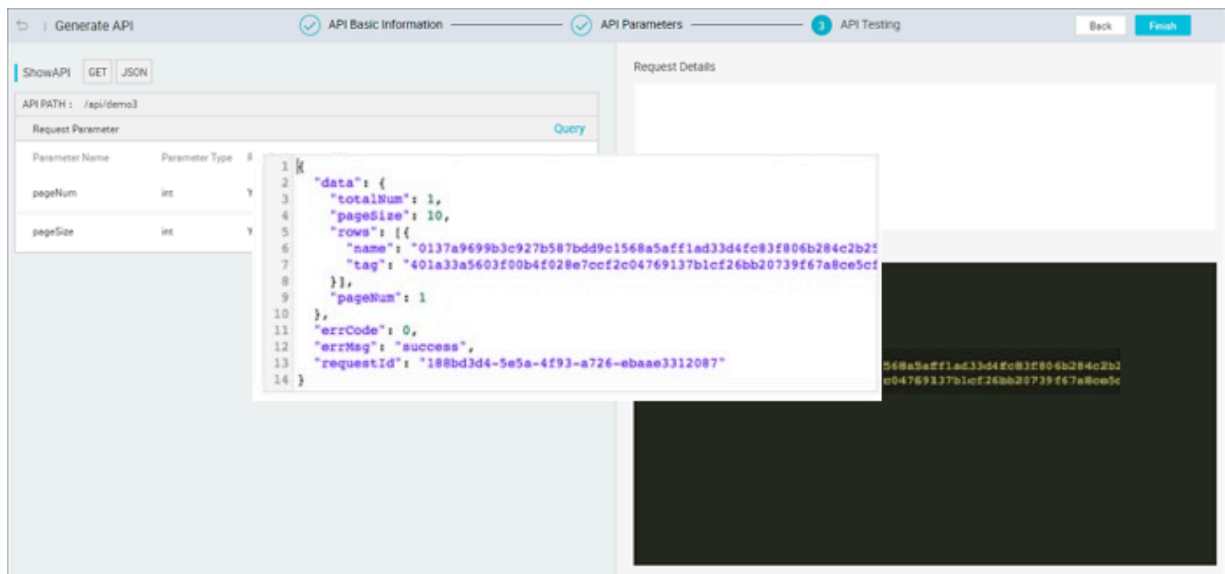
API Testing

After completing configuration of API parameters, you can start the API test.



Set parameters and click **Start Test** to send the API request online. The API request details and response are displayed on the right. If the test fails, read the error message carefully and make the appropriate adjustments to test your API again.

You need to note the settings for the normal return example during the configuration process. When testing an API, the system automatically generates exception examples and error codes. However, normal response examples are not automatically generated. After the test succeeds, you need to click **Save as Normal Response Sample** to save the current test result as the normal response sample. If sensitive data is included in the response, you can manually edit it.



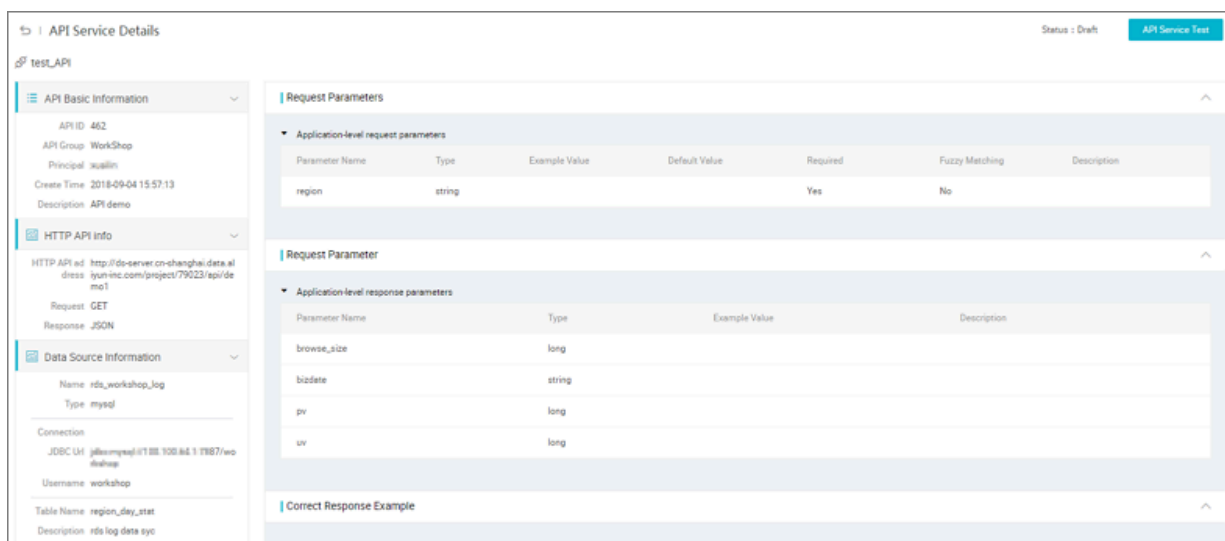
Note:

- Normal response examples provide an important reference value for the API callers. Specify an example if possible.
- The API calling delay is the delay of the current API request, which is used to evaluate the API performance. If the latency is too high, you may consider optimizing your database.

After completing the API test, click **Finish**. The data API is successfully created.

API details viewing

Back on the API service list page, click **details** in the Action column to view the details of the API. This page displays detailed information about an API from the view of a caller.



8.4 Register API

This section describes how to register an API.

You can register currently available APIs in Data Service. These APIs can be managed and published to API Gateway together with APIs created based on data tables. Currently, you can only register RESTful APIs supporting GET, POST, PUT, and DELETE requests and content types form,JSON,and XML.

Configure the API basic information

1. You can go to the registration API page by selecting the **Register API** in the API Service list.
2. Configure the API basic information.

The screenshot shows the 'Register API' wizard in the DataService studio. The wizard has three steps: 1. API Basic Information, 2. API Parameters, and 3. API Testing. The first step is active. The form contains the following fields:

- API Name:** registerAPI (Support Chinese characters, English, numbers, underline, and must start with English or Chinese characters, 4 to 50 characters)
- API Group:** WorkShop (Add API Group button)
- Protocol:** HTTP (checked)
- Background Services Host:** https://sqjson.com (Begin with http:// or https:// and do not include Path)
- Background Services Path:** /api/demo/work (Supports English character, number, underscore, hyphen(-), and must begin with /, no more than 200 characters. Back-end service Path. If there is request parameter in Path, place it in [], etc: /user/[userid])
- API Path:** /open/api/weather (API Path is an alias of backend service Path, supporting English character, number, underscore, hyphen(-), and must start with /, no more than 200 characters. If the API Path contains the Parameter in the request parameters, please place the parameter in [], and the parameter name should be the same as that in the background service Path)
- Request:** GET
- Response:** JSON
- Description:** dglfing

There are 'Add API Group' and 'Next' buttons.

Parameters:

- Protocol: Only HTTP is supported.
- Background Service Host: Enter the host of the API. The host must start with http:// or https://, and cannot contain the path.
- Background Service Path: Enter the path of the API. Put parameter names in brackets, for example, /user/[userid].

If a parameter is defined in the path, the system automatically adds the parameter in the path to the request parameter list in the second step of the API registration wizard.

- API path: The alias of the background service path. It allows an API for the background service host and path to register as multiple APIs.

Parameters defined in Background Service Path must also be defined in brackets in API Path.

- Request method: The options include GET, POST, PUT, and DELETE. Different request methods correspond to different subsequent configuration items.
- Return Type: Select JSON or XML.

3. After providing the API basic information, click **Next** to go to the API parameter configuration page.

Configure API parameters

After configuring the basic API information, you can configure the API parameters, including the request parameters, response example, and error code of the API.

- Request Parameters:
 - Parameter location: The options include Path, Header, Query, and Body. Different request methods support different optional parameter locations. You can select the options as required.
 - Constant parameters: The parameters that have the fixed values and are invisible to callers. The constant parameters do not need to be input during API calling. However, the background service always receives the defined constant parameters and their values. Constant parameters are applicable if you want to fix the value of a parameter or hide the parameters to the callers.
- Request Body is required only when the request mode is POST or PUT. You can enter the desc
The content types of the request body include JSON and XML.



Note:

If the request body is defined in the request body definition and the body location parameter is defined in the request parameter definition, the body location parameter is invalid. The request body is applied.

- You can enter a normal example or an exception example for API callers to refer to when writing the return parse code.
- Enter the common errors and solutions in API calling. This enables API callers to troubleshoot and solve these errors.

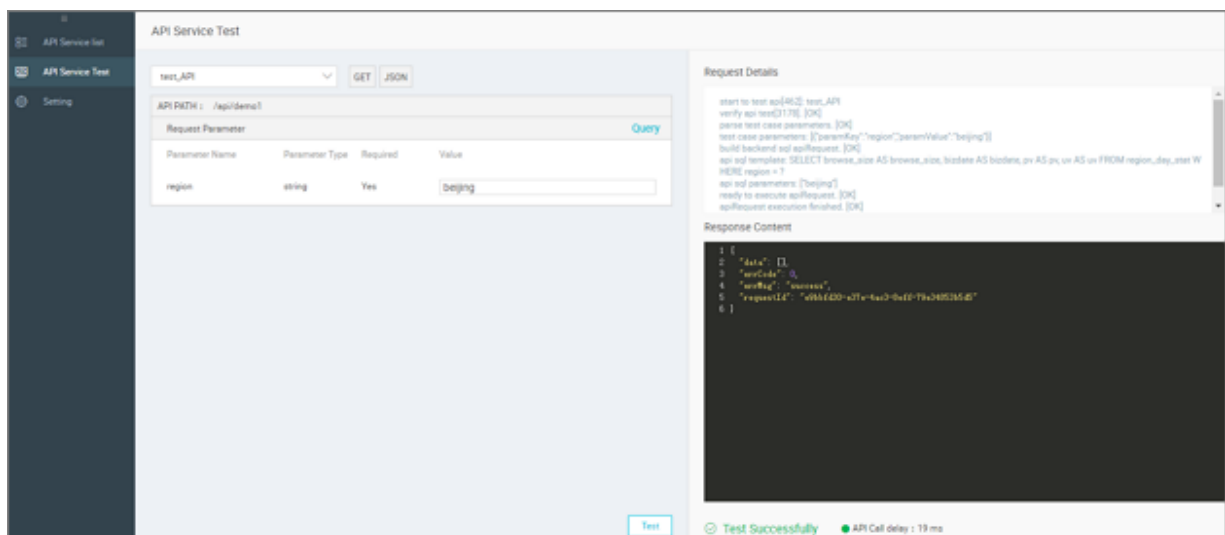


Note:

To ensure that the API is easily used by the callers, provide complete API parameter information if possible, especially the parameter sample values, default values, and response examples.

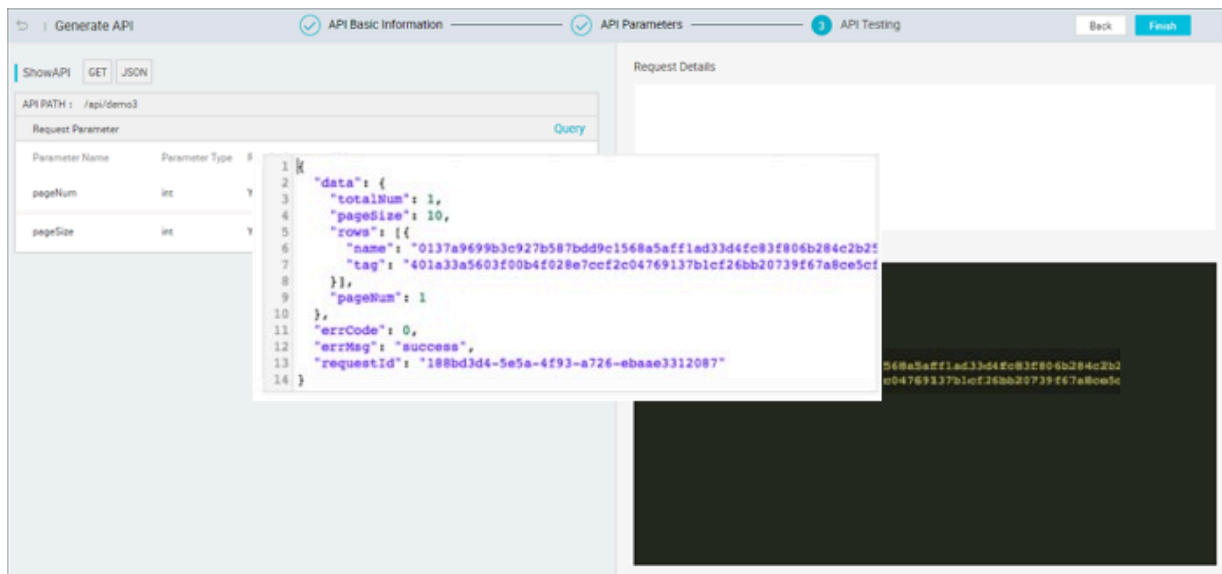
API Testing

After completing configuration of API parameters, you can start the API test.



Set parameters and click **Start Test** to send the API request online. The API request details and response are displayed on the right. If the test fails, read the error message carefully and make the appropriate adjustments to test your API again.

You need to note the settings for the normal return example during the configuration process. When testing an API, the system automatically generates exception examples and error codes. However, normal response examples are not automatically generated. After the test succeeds, you need to click **Save as Normal Response Sample** to save the current test result as the normal response sample. If sensitive data is included in the response, you can manually edit it.



Note:

- Normal response examples provide an important reference value for the API callers. Specify an example if possible.
- The API calling delay is the delay of the current API request, which is used to evaluate the API performance. If the latency is too high, you may consider optimizing your database.

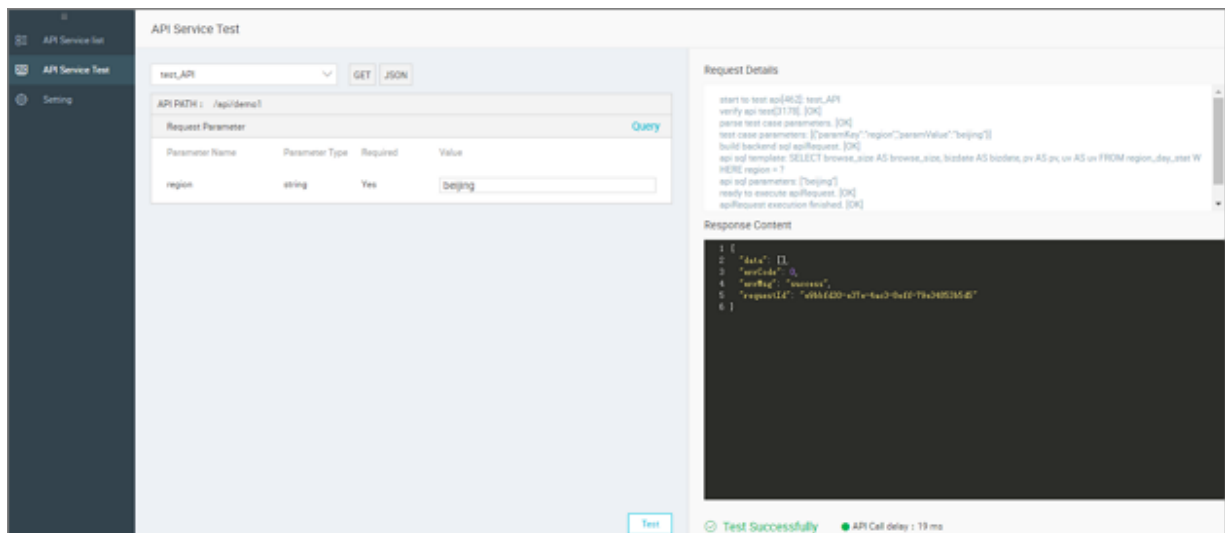
After completing the API test, click **Finish**. The data API is successfully created.

8.5 API service test

This article will show you how to test your API.

When creating and registering an API, you can test the API. For more information, see [Generate API in Wizard Mode](#).

The system also provides an independent API service test function for you to perform routine API tests online. You can choose **More > Test** in the Actions column of the API list to go to the API test page. Alternatively, you can click **API Service Test** in the left-side navigation pane, enter the API test page, and select the corresponding API.



Note:

The API service test page provides only the API online test function and does not allow update and storage of the API normal response examples. To update an API normal response example, click **Edit** in the API list, enter the API editing mode, and update the content of the normal response example in the API test process.

8.6 Publish an API

API Gateway is an API hosting service that provides full life cycle management covering API release, management, O&M, and sales. It provides you with a simple, fast, low-cost, and low-risk method to implement microservice aggregation, frontend-backend isolation, and system integration, and opens functions and data to partners and developers.

API Gateway provides permission management, traffic control, access control, and metering services. The service makes it easy for you to create, monitor, and secure APIs. Therefore, we recommend that you publish the APIs that have been created and registered in Data Service to API Gateway. Data Service and API Gateway are connected, which allows you to publish APIs to API Gateway easily.

Publish APIs to API Gateway



Note:

To release an API, you must first activate the API Gateway service.

After activating API Gateway, you can click **Publish** in the Actions column of the API service list to release the API to API Gateway. The system automatically registers the API to API Gateway

during the publish process. The system creates a group in API Gateway with the same name as the API group and releases the API to the group.

After the release, you can go to the API Gateway console to view the API information. You can also set the throttling and access control functions in API Gateway.

If your API needs to be called by your application, you must create an application in API Gateway, authorize the API to the application, and encrypt the signature call using the AppKey and AppSecret. For more information, see [API Gateway help documentation](#). At the same time, the API gateway also provides the SDK in the mainstream programming language, you can quickly integrate your API into your own applications, for more information, please refer to the [SDK download and user's guide](#).

Publish APIs to Alibaba Cloud API Marketplace

After your APIs from Data Service have been published to API Gateway, you can then publish them to Alibaba Cloud API Marketplace. This is an easy way to achieve financial gains for your company.

Before selling the API to the Ali cloud API market, first of all, it is necessary to enter the Ali cloud market as a service provider.



Note:

Select to enter API Marketplace as shown in the following figure. Note: only enterprise users are allowed to enter Alibaba Cloud API Marketplace.

Procedure

1. Enter the Ali cloud service provider platform.
2. Click **commodity management > publish the merchandise** and select the access type as the **API service**.
3. Select the API grouping that you want to list (one grouping corresponds to one API commodity).
4. Configure commodity information and submit audit.

Once your product has been successfully published to Alibaba Cloud API Marketplace, users can purchase it worldwide.

8.7 Delete API

Choose **More > Delete** in the Actions column of the API service list to delete an API.

**Note:**

- An API can be deleted only when it is in offline status. If it is online, deprecate the API and then delete it.
- The delete operation is irreversible. Delete an API with caution.

8.8 Call an API

This section describes how to call an API after this API is released on API Gateway.

API Gateway provides API authorization and the SDK for calling APIs. You can authorize yourself, your associates, or third parties to use APIs. If you want to call an API, perform the following operations.



Three elements for calling an API

To call an API, you need the following three elements:

- API: the API that you are about to call, which is clearly defined by the API parameters.
- app: Identity that you use to call the API. The AppKey and AppSecret are provided to authenticate your identity.
- Permission relationship between the API and app: When an app needs to call an API, the app must have the permission of this API. This permission is granted through authorization.

Procedure

1. Get the API documentation

The acquisition method varies according to the channel that you use to obtain the API. It is generally divided into API services purchased from the data market and not required to purchase, two ways are actively authorized by the provider. For more information, see [get API documentation](#).

2. Create a project

The app is the identity that you use to call an API. Each app has a set of AppKey and AppSecret, which are equivalent to an account and a password. For more information, see [creating an application](#).

3. Get the permission

Authorization means granting an app the permission to call an API. Your app must be authorized first to call an API.

The authorization method varies according to the channel that you use to obtain the API. For more information, see [obtaining authorization](#).

4. Call API

You can directly use the multi-language call sample provided by API Gateway Console, or use a self-compiled HTTP or HTTPS request to call the API. For more information, see [calling the API](#).

8.9 FAQ

- Q: Do I have to activate the API gateway?

A: API Gateway provides the API hosting service. If you plan to open your APIs to other users, the API Gateway service must be activated first.

- Q: Where can I configure the data sources?

A: To create a data source, select DataWorks > Data Integration > Data Sources. After the configuration, Data Service automatically reads the data source information.

- Q: What is the difference between a wizard-created API and a script-created API?

A: The script mode provides more powerful functions. For more information, see [Generate API in Script Mode](#).

- Q: What is an API group in Data Service? Is it the same as an API group in API Gateway?

A: An API group contains several APIs in a certain scenario. It is the minimum unit. In a word, the two are equivalent. When you publish an API group from Data Service to API Gateway, the gateway automatically creates an API group with the same name.

- Q: How can I configure an API group appropriately?

A: Typically, an API group includes APIs that provide similar functions or solve a specific issue. For example, the API for querying weather by city name and the API for querying weather by latitude and longitude can be put into an API group named "weather query".

- Q: How many API groups can be created?

A: An Alibaba Cloud account can create up to 100 API groups.

- Q: In what situations do I have to enable API response output pagination?

A: By default, an API outputs up to 500 records. To output more records, enable API response output pagination. When no API request parameters have been set, the API may output a large number of records, and the API response output pagination is automatically enabled.

- Q: Do APIs created by Data Source support POST requests?

A: Currently, a created API supports only the GET request.

- Q: Does Data Service support HTTP?

A: Currently, Data Service does not support HTTP. HTTP may be supported in later versions.

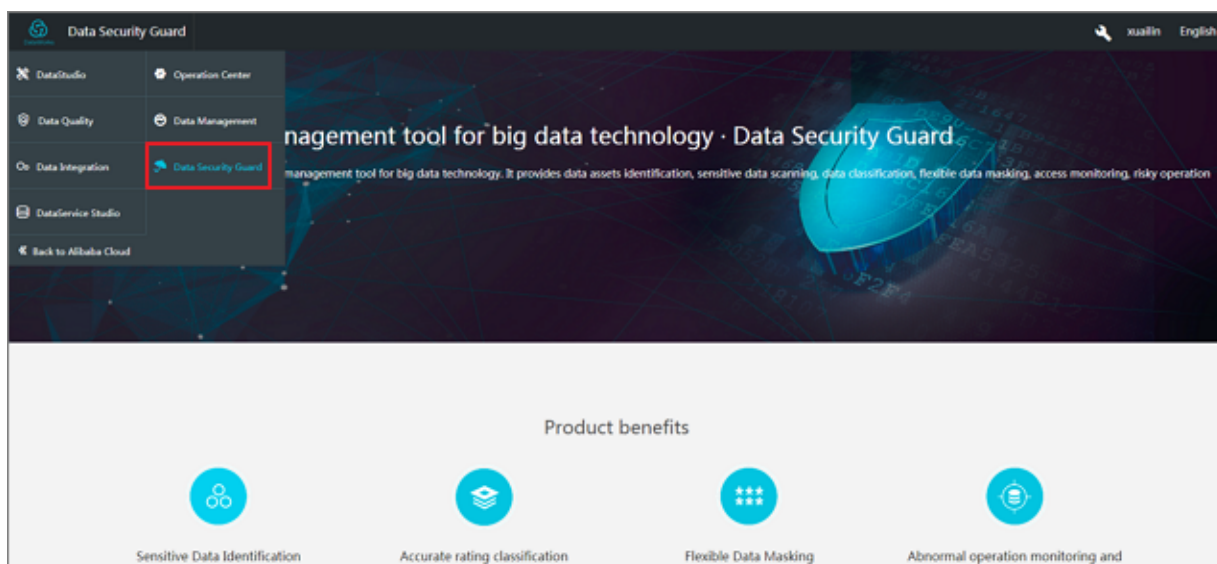
9 Data security guard

9.1 Enter Data Security Guard

Enter the start page

When you first enter the Data Security Guard, the Guide page appears, which introduces you to the core features and usage process of the data umbrella, help you get a basic understanding of the Data Security Guard.

Click **Try now** to enter the Data Security Guard authorization page (if the tenant Administrator has been authorized, then direct access to the Data Security Guard Home page).

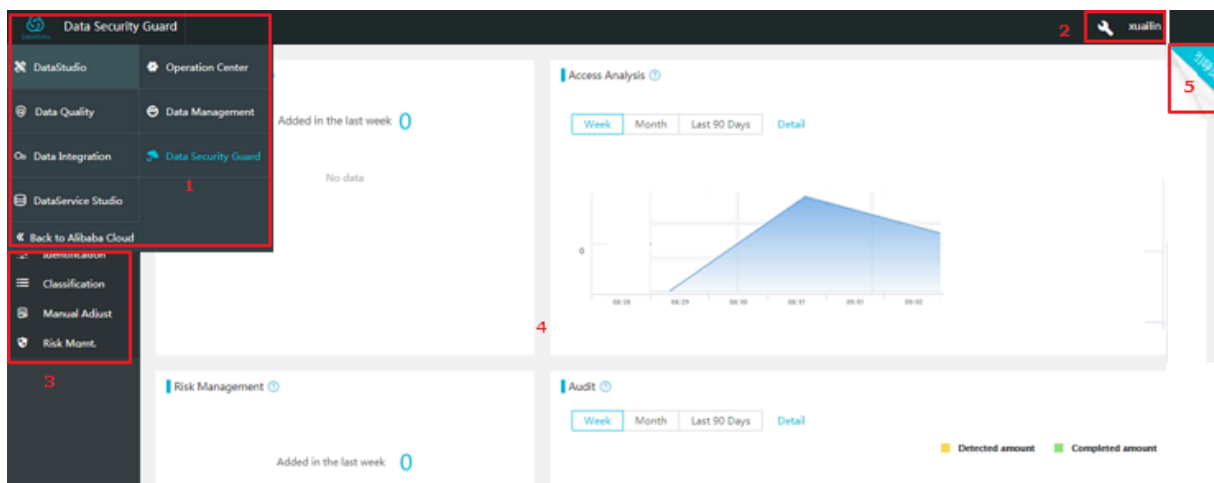


Enter the authorization page

Only the tenant Administrator can authorize the provision of Data Security Guard.

Logon Data Security Guard

Log in to the Data Security Guard, as shown in the following page:



Note:

No.	Name	Description
1	Function menu bar	The current user has the right to be visible to the function module, includes DataStudio, Data Quality, Data Integration, DataService Studio, Operation Center, Data Management and Data Security Guard.
2	User Information	Currently logged in, you can view and edit user information, including mailbox, phone, AccessKeyID, and AccessKeySecret.
3	Navigation Bar	Corresponding to the navigation bar of the function menu, different function modules correspond to different left navigation bars.
4	Home	<ul style="list-style-type: none"> The tenant has added data in the last week. All access data for nearly one week, nearly one month, nearly three months of access trends. New data risk nearly a week. The amount of discovery and completion of all risks for nearly one week, nearly one month, and nearly three months.
5	start page switch	Click start page to switch to the start page to view the product introduction information.

9.2 Data distribution

After the data security administrator completes the sensitive data rule configuration T + 1, you can view the data distribution in identifying the data distribution, it is divided into overall distribution, hierarchical distribution, and field details.

Depending on your query needs, filter your selections by project, rule name, rule type, risk level (that is, grading), and so on.

9.3 Access analysis

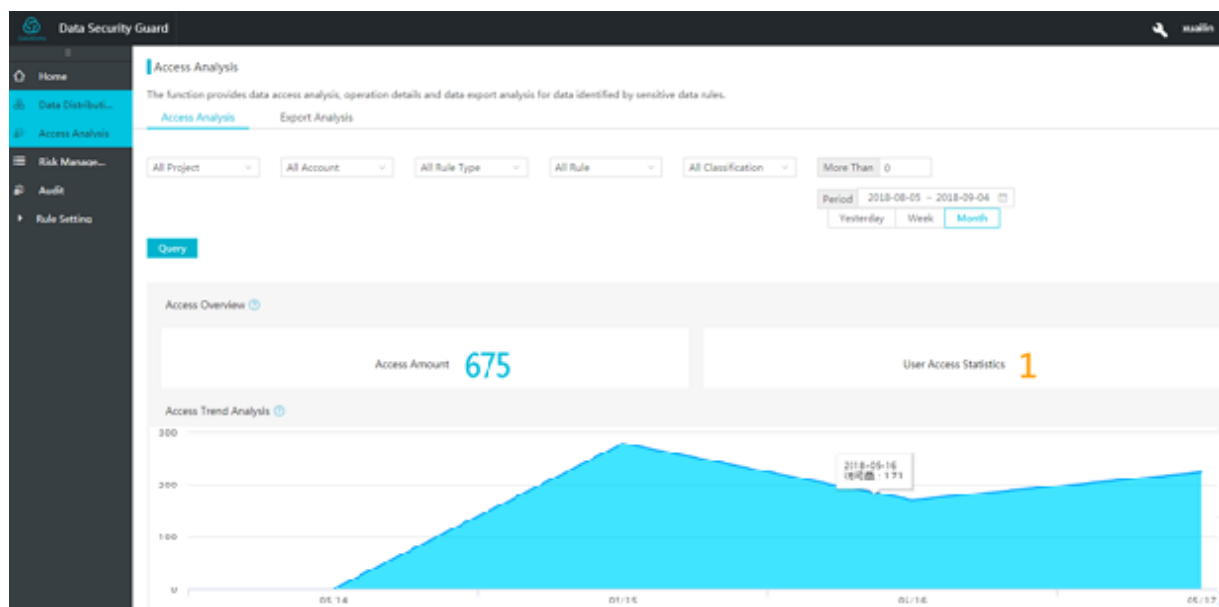
Data access includes both access behavior and export behavior.

- Access analysis: Contains create, insert operations, but does not include access failed behavior.
- Export analysis: the behavior that the data exports from MaxCompute.

Access analysis

After the data security administrator completes the sensitive data rule configuration T + 1, you can view data usage in the data access behavior, includes overview of access, access trends, and access details.

Depending on your query needs, by project, rule name, rule type, risk level (that is, grading), visitors, etc. for filtering selection.

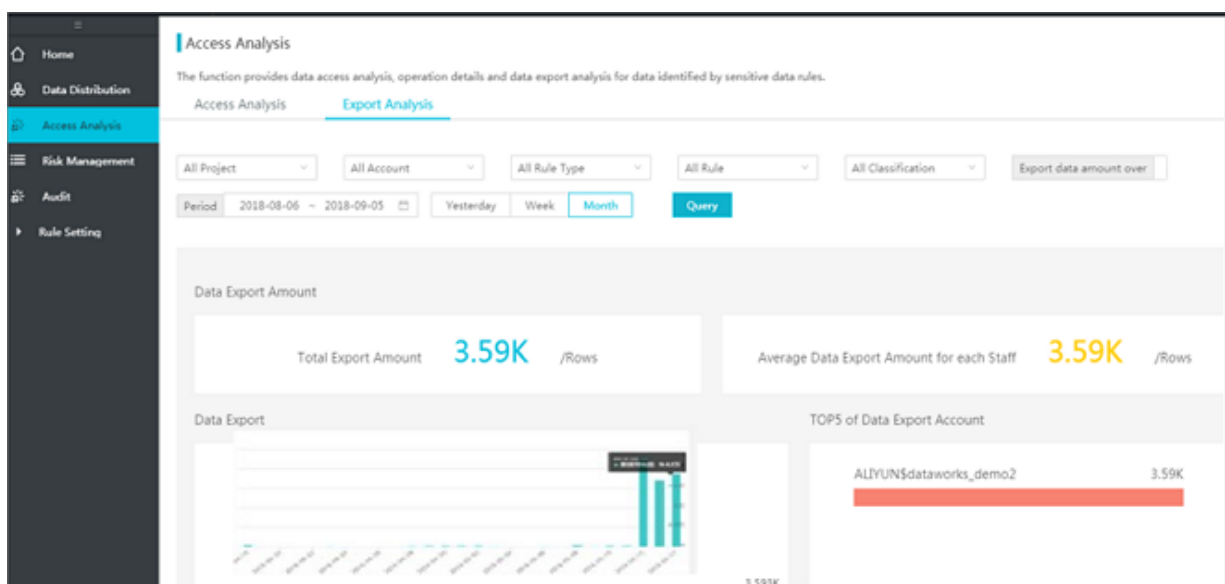


导出人(全部) ▾	导出账号(全部) ▾	导出ip(全部) ▾	IP所在地(全部) ▾	导出量 ▴	导出方式 ▴	导出时间 ▴	操作
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	200	tunnel下载	2018/08/10 22:03:01	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	399	tunnel下载	2018/08/10 22:03:00	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	399	tunnel下载	2018/08/10 22:03:00	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	399	tunnel下载	2018/08/10 22:02:59	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.99	数据保护全扫描任务	200	tunnel下载	2018/08/10 17:16:59	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.99	数据保护全扫描任务	399	tunnel下载	2018/08/10 17:16:57	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.99	数据保护全扫描任务	399	tunnel下载	2018/08/10 17:16:57	明细
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.99	数据保护全扫描任务	200	tunnel下载	2018/08/10 12:48:57	明细

Export analysis

After the data security administrator completes the sensitive data rule configuration T+1, you can see in the data export how the user exports the data from MaxCompute to the outside, includes total data export, top export users, and export details.

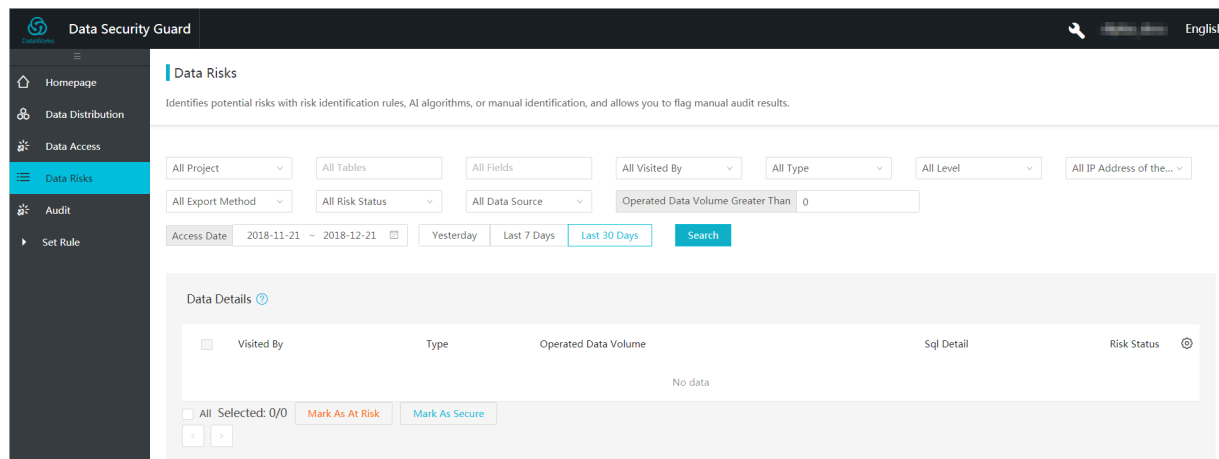
Depending on your query needs, filter your selections by rule name, rule type, export quantity, and so on.



Export user(all) ▾	Account(all) ▾	Export Ip(all) ▾	IP location(all) ▾	Export Amount ▴	Export Channel ▴	Period ▴	
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	200	Download By Tunnel	08/10/2018 22:03:01	Details
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	399	Download By Tunnel	08/10/2018 22:03:00	Details
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	399	Download By Tunnel	08/10/2018 22:03:00	Details
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.209	数据保护全扫描任务	399	Download By Tunnel	08/10/2018 22:02:59	Details
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.99	数据保护全扫描任务	200	Download By Tunnel	08/10/2018 17:16:59	Details
ALIYUN\$datamworks_demo2	ALIYUN\$datamworks_demo2	11.193.97.99	数据保护全扫描任务	399	Download By Tunnel	08/10/2018 17:16:57	Details

9.4 Data risks

Data Risks provides manual risk data identification, risk rule configuration identification and AI identification. It provides a list of risk data and the risk data can be audited for comments.

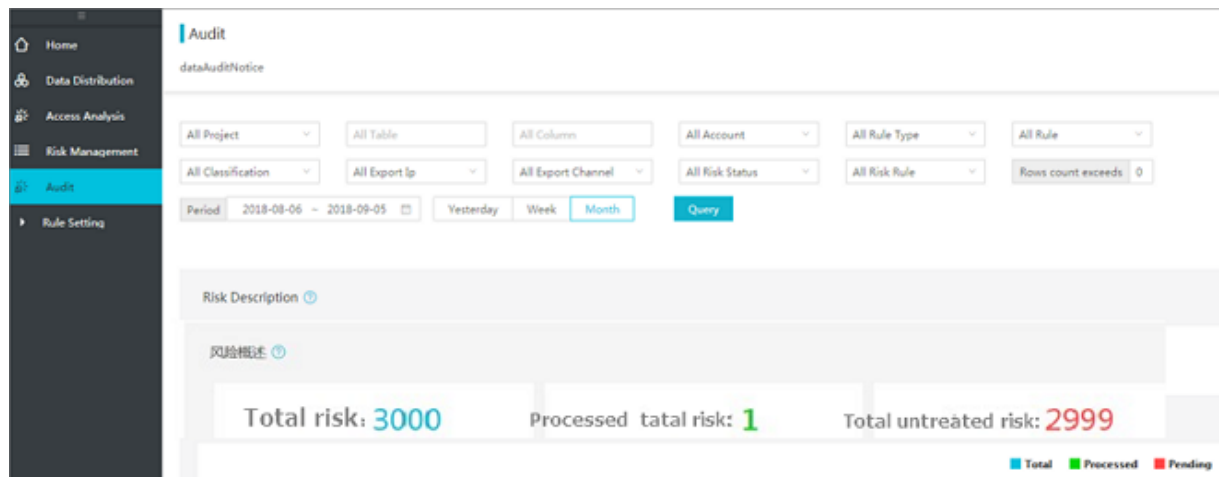


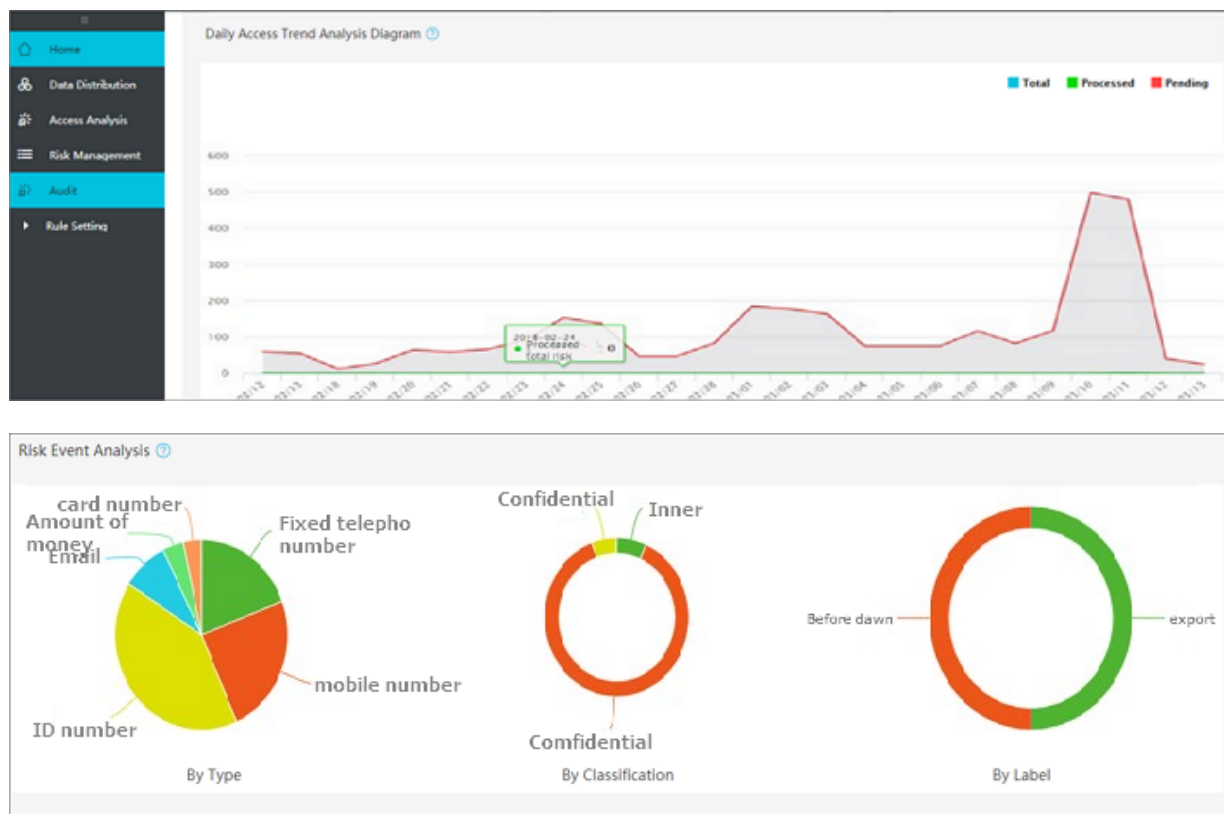
The page description is as follows:

- To query risk data conditions: the conditions available for filtering include project, table name, field, rule type, rule name, grade, export IP, export risk, risk status, and risk data type.
- Risk data details: you can select an audit comment in the **Settings** button at the title bar according to the need to view the metrics. It supports adding labels, adding detailed notes, and information.
- Bulk audit processing: divided into batch/risk free dimensions and detailed information notes.

9.5 Audit

The **Audit** page is a summary of Data risk statistics, includes an overview of risk data, daily risk trends, and risk dimension analysis.





9.6 Rule setting

Defining sensitive data

The steps for the data security administrator are as follows:

1. Logon Data Protection Platform.
2. Navigate to **Rules Setting > identification**, and click **New**.
3. Complete the basic information in the dialog box, and click **Next**. Configurations:
 - Data Type: that is, the classification to which the rule belongs, which supports adding by template or custom adding.
 - Data name: 11 Sensitive data identification definition templates are built into the system, ID card, banking card number, mailbox, mobile phone number, IP, MAC address, fixed phone, license plate number, identification of company, address and name, user-defined rules are also provided.
 - Owner: the rule sets the person information.
 - Note: set additional information descriptions for this rule.
4. Complete the configuration rules in the dialog box, and click **Next**.

Configurations:

- **Classification:** rank the configured data, and if the existing level does not meet the requirements, please set up in the Grading Information Management Service.
 - **Content scan:** One of the Data Recognition Methods provided, each of the 11 Data Recognition templates in the system is content scanned.
 - If you select a template, you cannot change the recognition rule, but you are provided with a channel to verify the accuracy of the rule, at the same time, the recognition of the situation can be manually corrected.
 - If you select regular match, the recognition rules are customized.
 - **Meta Scan:** Provides the exact matching of Field Names and Fuzzy Matching methods to support multiple field matches, the relationship between the fields is or.
5. When the settings are complete, click **Next** and save.
 6. If you need to modify an existing rule, you can click the **Configuration rules** that you want to manipulate, configure and modify advanced information.
 7. When the rule configuration is complete, click **Save**.
 8. After saving the rule is invalid, the change status takes effect after the confirmation rule is correct.

**Note:**

When defining sensitive data, follow these rules.

- The rule name must be unique.
- Content or field scans for different rules must be unique.
- Rules identify data, T + 1 is displayed in the report.

Sensitive data defined

If you have defined sensitive data, jump directly to identify data distribution, data access behavior, and data export module features.


9.7 Classification management

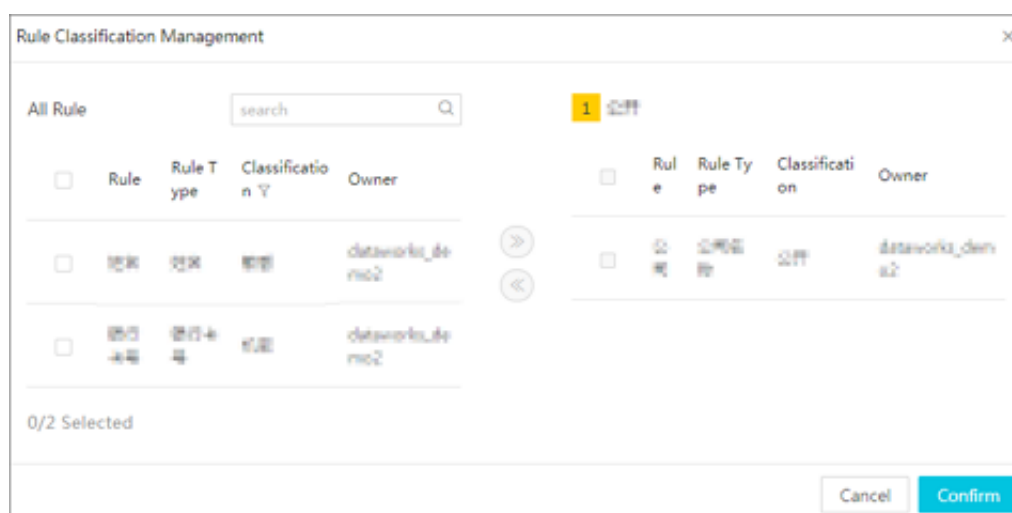
When the rated selection in the rule configuration does not meet your needs, you can set up in rated Page Management, this page provides the ability to create new grading, delete grading, grading priority adjustment, and rule grading adjustment.


Data Classification	Name	Owner	Operation time	Rules
	公开	lu	2018/06/14 19:57:00	1
	敏感	lu	2018/06/14 19:56:42	1
	机密	lu	2018/06/14 19:56:28	1


The page description is as follows:

- Create Classification: Click **New** to add a new classification, fill in the name and operator.

- : Adjusts rule grading for rule selection and adjustment when clicked.

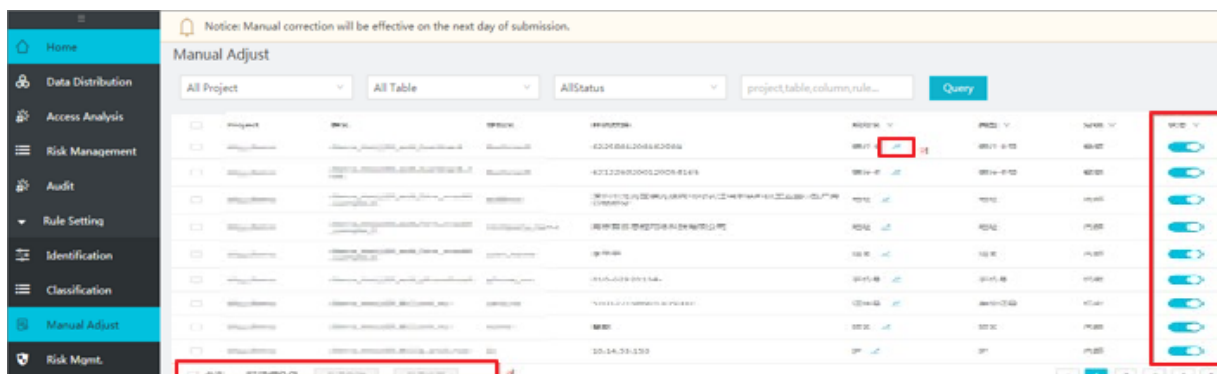


- : Adjust the grading priority, click Next (lower priority), or drag up (increase priority).



- : Delete the grading, And you can delete unwanted grading after you click.

9.8 Manual ajust

The manual remediation page provides the ability to manually correct situations where sensitive data is not accurate for rule recognition, includes removing identifying error data, changing identifying data types, and bulk processing.



The page description is as follows:

- Remove the recognition error data: the button under the sliding **Status** column changes to the removed state, the data that has been eliminated can be recovered.
- : Change the identification data type. If you recognize as a mailbox and are actually a license plate number, click make changes to the right  of the mailbox, only Configured Rule names can be selected.
- Bulk processing: includes bulk removal and bulk recovery, selecting data for operation, click the check box on the left side of the data, and then click the appropriate action.

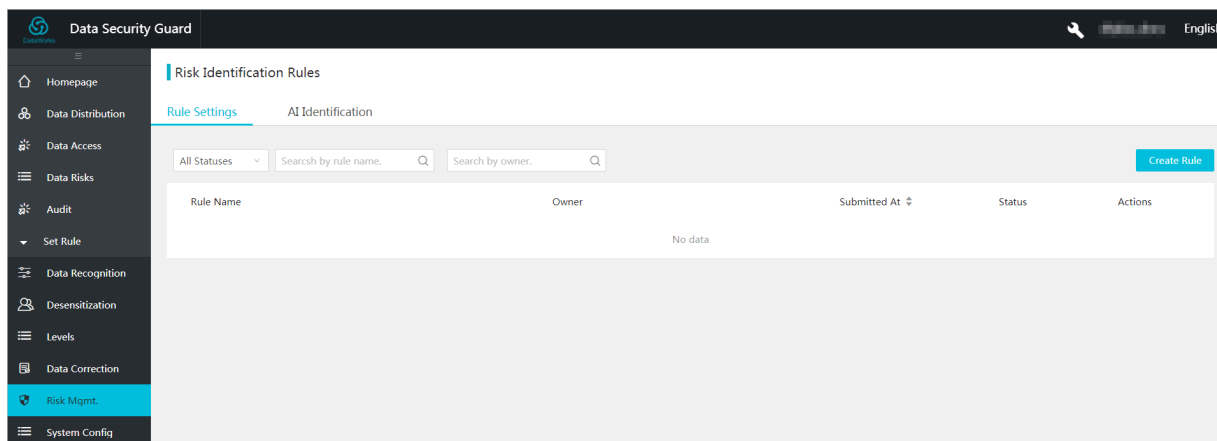


Note:

Manually correcting data requires following exit and changing the data name type T + 1 to be in effect for identifying data distribution, data access behavior, rules for data export pages.

9.9 Risk Mgmt

The **Risk Mgmt** page provides the risk data rule configuration, you can identify risks in your daily visits and start AI to identify data risks automatically. The identified risk data is displayed on the [Data risk page](#) and audited, it also marks the data at the data access page.



The page description is as follows:

- Risk identification management: divided into risk rule configuration and AI identification. AI-aware pages include personal information queries, similar SQL queries, and identification descriptions of these two pieces. You only need to start it in the Status column. It can also be turned off after startup (no previously identified data is deleted).
- Risk Rule Configuration _ new rule: after you enter the rule name, owner, and rule note information in the dialog box, the rule basic information is created.
- Risk Rule Configuration _ actions: provides the ability to copy rules, edit risk rule entries, and delete rules.
- Risk Rule Configuration _ rule item configuration: provides project (Multi-select Enabled), type (Multi-select supported), rules (Multi-selection support), grading (Multi-selection support), export method (Multi-selection Support), tables (supports fuzzy/exact matching), fields (supports fuzzy/precise matching), accessor (supports fuzzy/exact matching), the amount of operation data, and the access time condition configuration.
- Risk Rule Configuration _ Status: After you have configured the rule, you need to take effect after the Status column starts the rule.

**Note:**

Risk identification management data needs to follow the rules configured as well as AI identification, data takes effect on the page by t+1.

10 MaxCompute manager

10.1 MaxCompute Manager

The MaxCompute Manager provides system status monitoring, resource group allocation, and task monitoring for system operators. This article introduces how to use the MaxCompute Manager.

Prerequisite

- You should already have purchased MaxCompute Subscription CU resources and a quantity of 60 CUs or more.



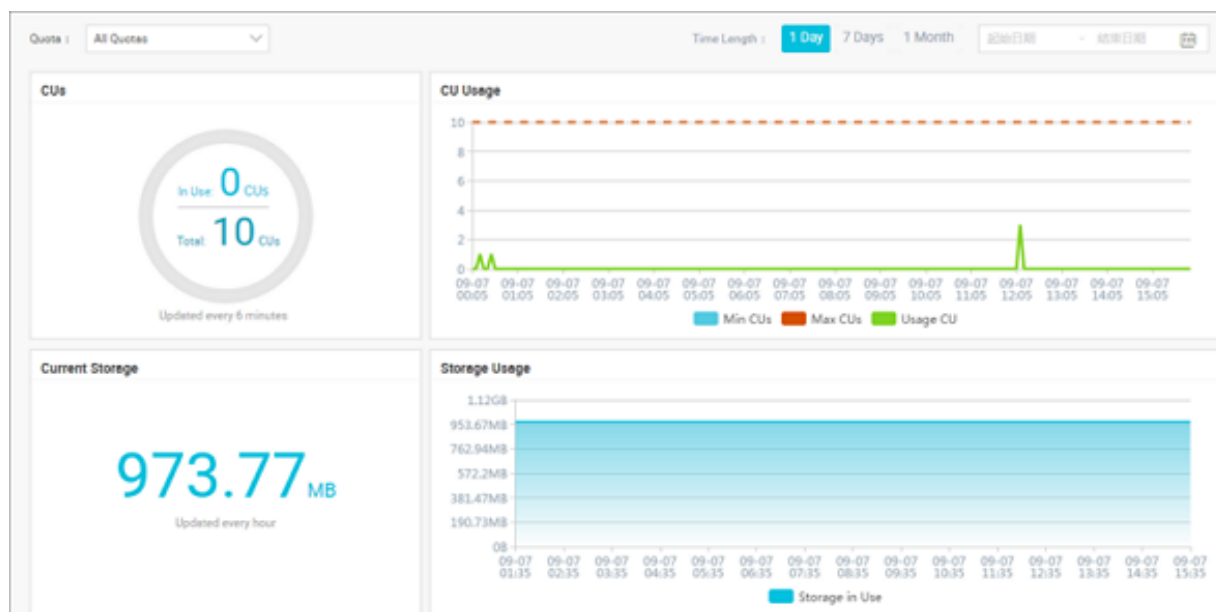
Note:

You can only take complete advantage of computing resources and MaxCompute Manager when you have sufficient CUs. If you disable the AK for the master account, it will result in the failure to use MaxCompute Manager with the corresponding sub-account.

You can log on the [DataWorks management console](#), click **CU Manage**.

System Status

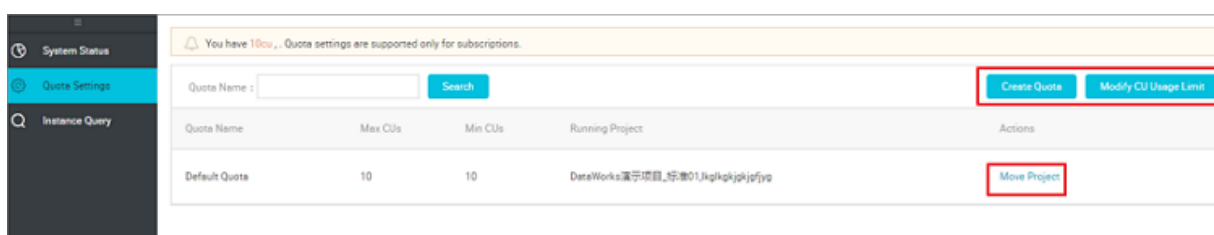
On System Status page, you can see the consumption of CU computing resources and current storage.



- **Quotas:** You can select the resource group you want to view and find its consumption information and current storage.
- **Time Length:** You can select the time periods for the selected resource group. With different time periods, resource group data are displayed with different time granularities.

Quota settings

A quota refers to a resource group. For example, if you purchased 100 CUs, you have a total quota of 100 CUs. You can create a new quota using MaxCompute Manager. Operators can easily isolate the resources of each project to ensure that the calculation resources of the important projects are sufficient.



- **Create Quota:** Create a quota, and assign projects to it. Created quotas cannot be deleted if there is an project under the current quota.
- **Modify CU Usage Limit:** You can modify the minimum CUs used by a quota.
- **Move Project:** You can move projects under the current quota to another quota.
- **Delete:** The quota cannot be deleted if there is an project under the current quota.

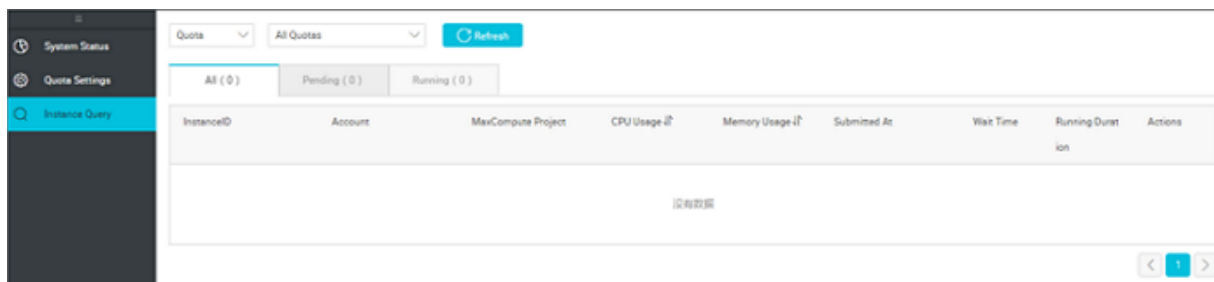


Note:

Max is the largest assigned resource, and Min is the smallest guaranteed resource.

Instance Query

You can view the current task queuing status, such as which task has occupied the resource. Then you can analyze your task and decide if you want to stop it.



You can specify the quota and the project name to filter the tasks.

- **Instance ID:** Each MaxCompute task has an instance. You can jump to the logview page by clicking instance ID, view specific task progress.
- **Account:** Based on this account information, you can find the person responsible for the task.
- **MaxCompute Project:** The project to which the instance belongs.
- **CPU Usage:** CPU used by the quota.
- **Memory Usage:** Memory used by the quota.
- **Submitted At:** The commit time of the current instance.
- **Waiting Time:** How much time spent on waiting for resources.
- **Actions:** You can check the status of the instance. Both the current status and historical status are displayed.