

DataWorks

Quick Start

Quick Start

Guide description

This guide describes how to quickly perform data development and O&M operations.

NOTE:

If this is the first time you are using DataWorks, make sure that you have prepared an account and configured the project roles and project according to steps in **Preparation**. Then, go to the DataWorks console page and click **Enter Workspace** after a project to go to the **Data Development** page of DataWorks to start data development.

Generally, DataWorks project space data development and O&M involve the following operations.

Step 1: Upload a local file

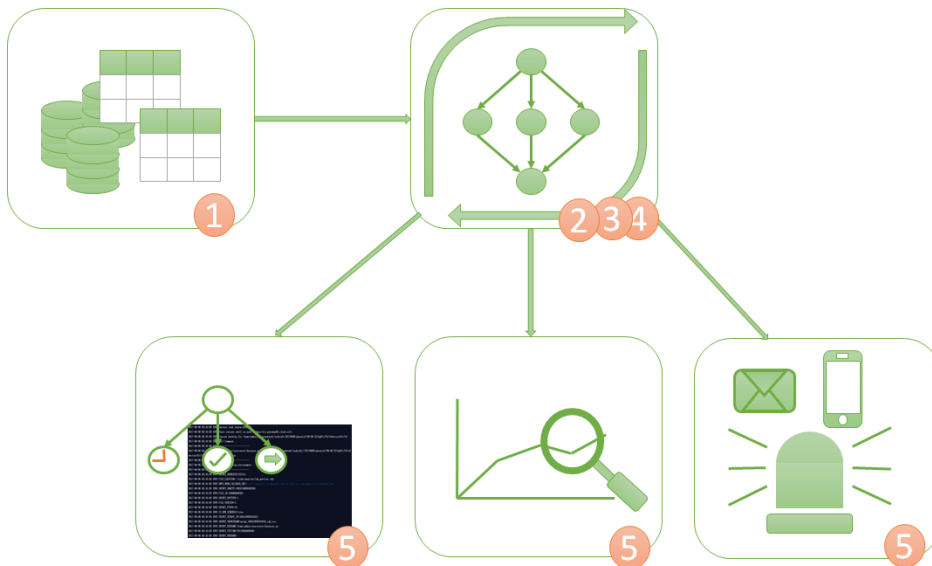
Step 2: Create a task

Step 3: Create a data sync job

Step 4: Scheduling and dependence settings

Step 5: Perform periodic O&M and view log troubleshooting results

The following is the general process illustration mainly based on the preceding steps.



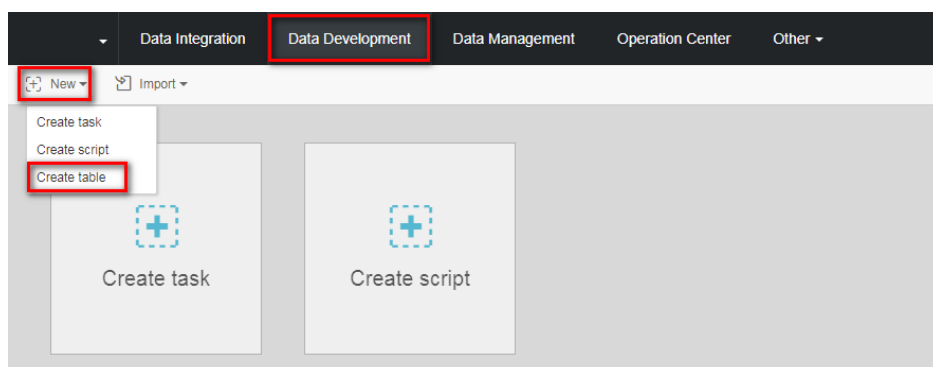
Upload a local file

In this article, we use creation of the tables `bank_data` and `result_table` as an example to describe how to create a table and upload data. The table of `bank_data` stores the business data, while the `result_table` stores the results after data analysis.

Procedure

Follow these steps to create `bank_data`.

Log on to the project and select **Data Development** > **New** > **Create Table**.



Enter the table creation statements, and click **OK**. For more information on table creation SQL syntax, see [MaxCompute-based table creation, view, and deletion](#).

The statements used for table creation in this example are as follows:

```
CREATE TABLE IF NOT EXISTS bank_data
(
age BIGINT COMMENT 'age',
job STRING COMMENT 'job type',
marital STRING COMMENT 'marital status',
education STRING COMMENT 'educational level',
default STRING COMMENT 'credit card ownership',
housing STRING COMMENT 'mortgage',
loan STRING COMMENT 'loan',
contact STRING COMMENT 'contact information',
month STRING COMMENT 'month',
day_of_week STRING COMMENT 'day of the week',
duration STRING COMMENT 'Duration',
campaign BIGINT COMMENT 'contact times during the campaign',
pdays DOUBLE COMMENT 'time interval from the last contact',
previous DOUBLE COMMENT 'previous contact times with the customer',
poutcome STRING COMMENT 'marketing result',
emp_var_rate DOUBLE COMMENT 'employment change rate',
cons_price_idx DOUBLE COMMENT 'consumer price index',
cons_conf_idx DOUBLE COMMENT 'consumer confidence index',
euribor3m DOUBLE COMMENT 'euro deposit rate',
nr_employed DOUBLE COMMENT 'number of employees',
y BIGINT COMMENT 'has time deposit or not'
);
```

After the table is created, click **Table Query** in the left-side navigation pane and enter the table name for search.

All projects [dropdown] [search] [refresh] [add]

ODPS Table

- bank_data
- odps_result
- result_table
- testodps2

Task development | Script development | Resource | Function | **Table query**

Columns | Partitions | Preview

Filter column [search]

<input type="checkbox"/>	Name	Type	Desc
<input type="checkbox"/>	age	BIGINT	age
<input type="checkbox"/>	job	STRING	jo...
<input type="checkbox"/>	marital	STRING	m...
<input type="checkbox"/>	educati...	STRING	ed...
<input type="checkbox"/>	default	STRING	cr...

Create result_table

Follow these steps to create result_table

Click **Data Development > New > Create Table**.

On the **Create Table** page, enter the table creation statements, and click **OK**. The statements used for table creation are as follows:

```
CREATE TABLE IF NOT EXISTS result_table
(
  education STRING COMMENT 'educational level',
  num BIGINT COMMENT 'number of people'
);
```

After the table is created, click **Table Query** in the left-side navigation pane and enter the table name for search.

Upload local data to bank_data

DataWorks supports the following operations:

Upload data in local text files to a table in the workspace.

Use the data integration module to import business data from multiple different data sources to the workspace.

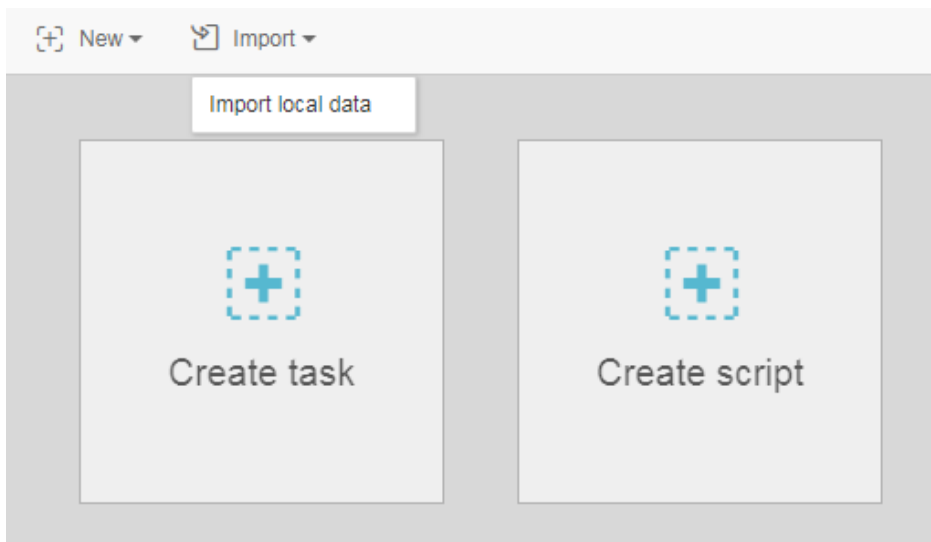
Note:

This section uses local files as the data source. Local text file uploads have the following limits:

- File type: Only .txt and .csv files are supported.
- File size: The file size cannot exceed 10 MB.
- Operation objects: Partition and non-partition tables can be imported, but Chinese partition values are not supported.

Using the import of the local file **banking.txt** to DataWorks as an example, the instruction is as follows:

Click **Import > Import Local Data**.



Select a local data file, configure the import information, and click **Next**.

Import local data ×

Selected files: banking.txt Only .txt, .csv and .log files are supported

Delimiter: Comma 自定义

Original character set: GBK

Import start line: 1

First line is title: Yes

44	blue-collar	married	basic.4y	unknown	yes	no	cellular	aug	thu	210	1	999	0	nonexistent
53	technician	married	unknown	no	no	no	cellular	nov	fri	138	1	999	0	nonexistent
28	management	single	university.degree	no	yes	no	cellular	jun	thu	339	3	6	2	success
39	services	married	high.school	no	no	no	cellular	apr	fri	185	2	999	0	nonexistent
55	retired	married	basic.4y	no	yes	no	cellular	aug	fri	137	1	3	1	success
30	management	divorced	basic.4y	no	yes	no	cellular	jul	tue	68	8	999	0	nonexistent
37	blue-collar	married	basic.4y	no	yes	no	cellular	may	thu	204	1	999	0	nonexistent
39	blue-collar	divorced	basic.9y	no	yes	no	cellular	may	fri	191	1	999	0	nonexistent

Enter at least two letters to search for the table by name. Select the table to which the data is to be imported, for example, bank_data.

To create a new table, click **Create Table**.

Import local data

Table: Create Table

Matching: Match by position Match by name

Target field	Source field

Prev Import Cancel

Select the field matching method ("Match by Position" is used in this example), and click **Import**.

Import local data

Table: Create Table

Matching: Match by position Match by name

Target field	Source field
age	empty column
job	empty column
marital	empty column
education	empty column
default	empty column
housing	empty column
loan	empty column

Prev Import Cancel

After the file is imported, the system displays a data import success or failure prompt.

Other data import methods

Create a data synchronization task

Applicability:

The data can be saved in multiple source types such as RDS, MySQL, SQL Server, PostgreSQL, MaxCompute, ApsaraDB for Memcache, DRDS, OSS, Oracle, FTP, dm, HDFS, and MongoDB.

For more information, see [Create a data synchronization task](#).

Upload a local file

Applicability:

The file size cannot exceed 10 MB, and only .txt and .csv files are supported. Only non-partitioned tables are supported.

For information on DataWorks local file uploads, see the [Upload local data to bank_data](#) section.

Use Tunnel commands to upload files

Applicability:

Local files and other resource files are larger than 10 MB.

Using the Tunnel commands provided by the MaxCompute Client to upload or download data, you can upload a local data file to a partitioned table.

For more information, see [Tunnel command operations](#).

Use DataX open-source tools

Applicability:

DataX can import local data in batches. The imported data must have a two-dimensional table structure. This method can be applied to some of the aforementioned scenarios as well.

For more information about DataX open-source tools, see [DataX open-source website](#).

Subsequent steps

You have learned how to create a table and upload data. You can go to the next tutorial for further study. This tutorial demonstrates how to create a flow for further data analysis and computing in the project space. For more information, see [Create a flow for data analysis](#).

Create a task

DataWorks offers a data development function that supports the graphic design of data analysis flows. It also processes data and forms mutual dependencies through flow tasks and inner nodes. Currently, it supports multiple task types such as ODPS_SQL, data synchronization, OPEN_MR, SHELL, machine learning, and virtual nodes. For more information about the use of each task type, see [Task type description](#).

Here, we use a creation of a flow task named “work” as an example to show how to create nodes in a flow, configure dependencies, and conveniently design and display steps and sequences for data analysis. This article explains how to use the data development function for further data analysis and computing in the workspace.

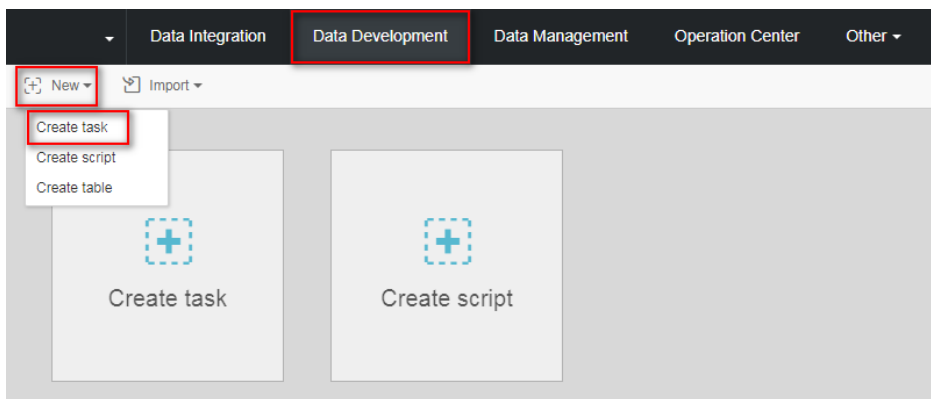
Prerequisites

You have prepared the business data table bank_data, the data it contains, and the result_table in the workspace according to Upload a local file instructions.

Procedure

Create a flow

Log on to the DTplus console, and click **Data Development > New > Create Task**.



Select the relevant content in the dialog box and specify the task type as **Flow task**.

Note: Once selected, the scheduling attribute cannot be changed.

Create task ✕

*Task type: Workflow task Node task

*Name:

*Schedule type: Manual scheduling Periodic scheduling

Description:

Select directory:
> Task development

+ New ▾ 📁 Save 📤 Submit 🖨 Test run 🖥 Full Screen 📄 Import ▾

📄 work ✕

Nodes
Data Process
<input type="button" value="OPEN_MR"/>
<input type="button" value="ODPS_SQL"/>
<input type="button" value="ODPS_MR"/>
<input type="button" value="Data SYNC"/>
<input type="button" value="Algorithm"/>
Script
<input type="button" value="SHELL"/>
Control
<input type="button" value="Virtual"/>

Create a node and dependency on the flow canvas

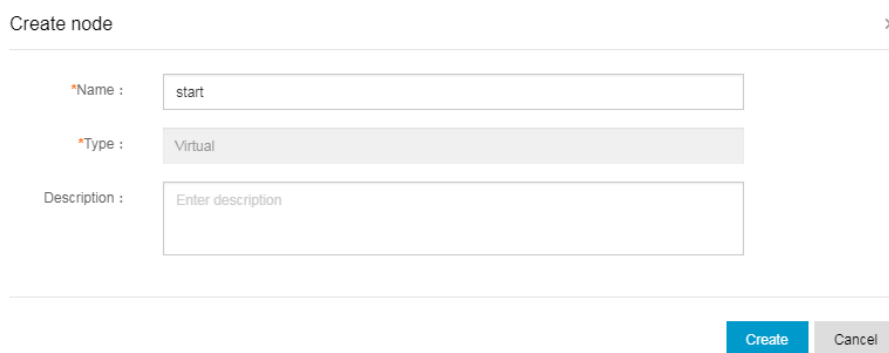
This section shows how to create a virtual node "start" and an odps_sql node "insert_data", and to configure "insert_data" to depend on "start".

Note:

- As a control-type node, the virtual node does not affect the data during flow operation and is only used for O&M control of downstream nodes.
- When a virtual node depends on the other nodes and its status is manually set to failure by the O&M personnel, its downstream nodes that have not run yet, cannot be triggered. This prevents further propagation of erroneous upstream data during the O&M process. For more information, see the section on virtual nodes in **Task type description**.

In a nutshell, we recommend that you create a virtual node as the root node to control the whole flow when designing a flow.

Double-click the virtual node, and enter the node name "start" .



Create node ×

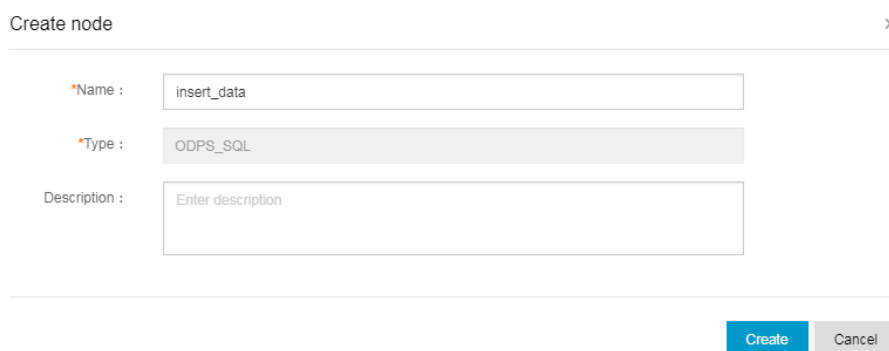
*Name :

*Type : Virtual

Description :

Create Cancel

Double-click **ODPS_SQL** and enter the node name "insert_data" .



Create node ×

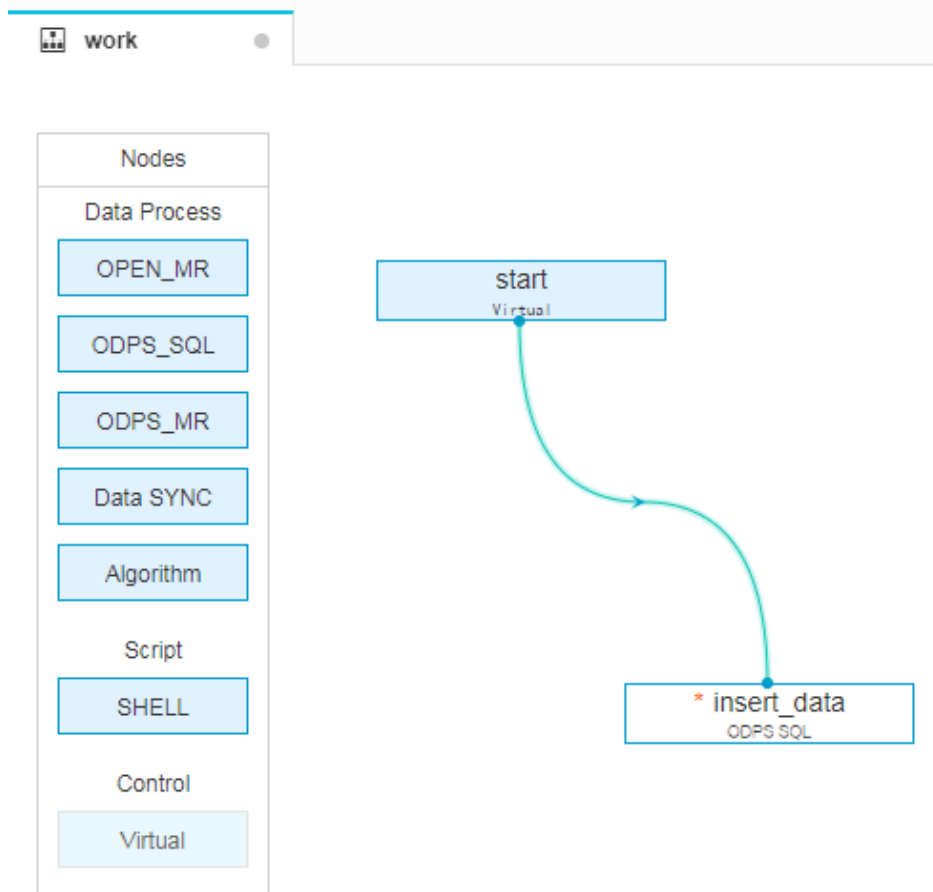
*Name :

*Type : ODPS_SQL

Description :

Create Cancel

Click the start node, and draw a line between start and insert_data to have insert_data dependent on start.



Edit the code in ODPS_SQL

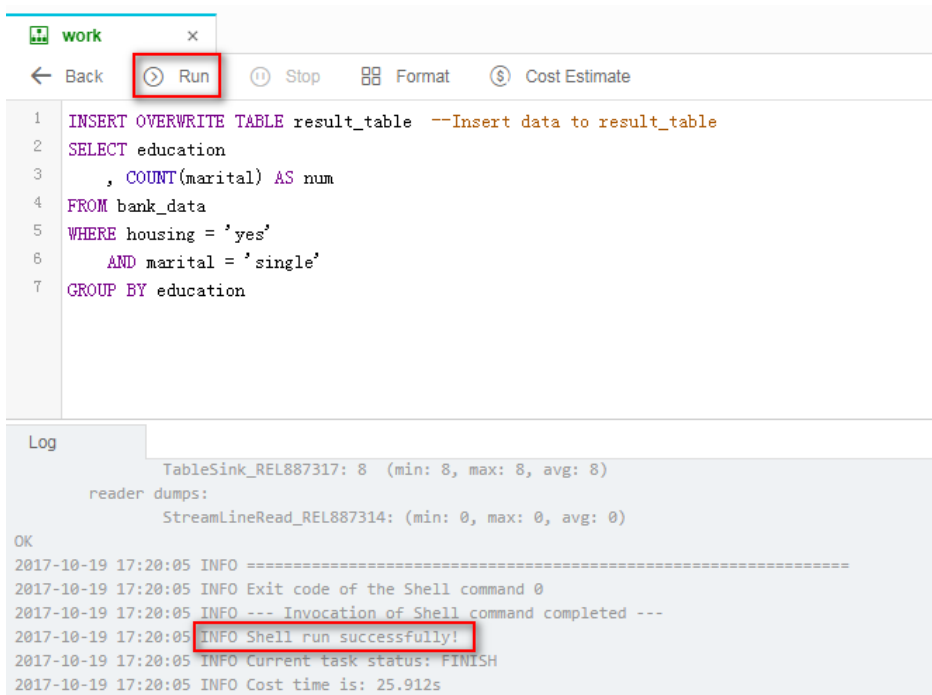
This section describes how to use SQL code in the ODPS_SQL node **insert_data** to query the quantity of mortgages available for individuals having different educational background and save results for analysis or display by the following nodes. For more information about the syntax, see the MaxCompute documentation. The SQL statements are as follows.

```
INSERT OVERWRITE TABLE result_table --Insert data to result_table
SELECT education
, COUNT(marital) AS num
FROM bank_data
WHERE housing = 'yes'
AND marital = 'single'
GROUP BY education
```

Run and debug ODPS_SQL

After editing the SQL statements in the insert_data node, click **Save** to prevent code loss.

Click **Run** to view operations logs and results.



The screenshot shows the DataWorks interface with a SQL query editor and a log window. The 'Run' button is highlighted with a red box. The SQL query is as follows:

```

1 INSERT OVERWRITE TABLE result_table --Insert data to result_table
2 SELECT education
3     , COUNT(marital) AS num
4 FROM bank_data
5 WHERE housing = 'yes'
6     AND marital = 'single'
7 GROUP BY education

```

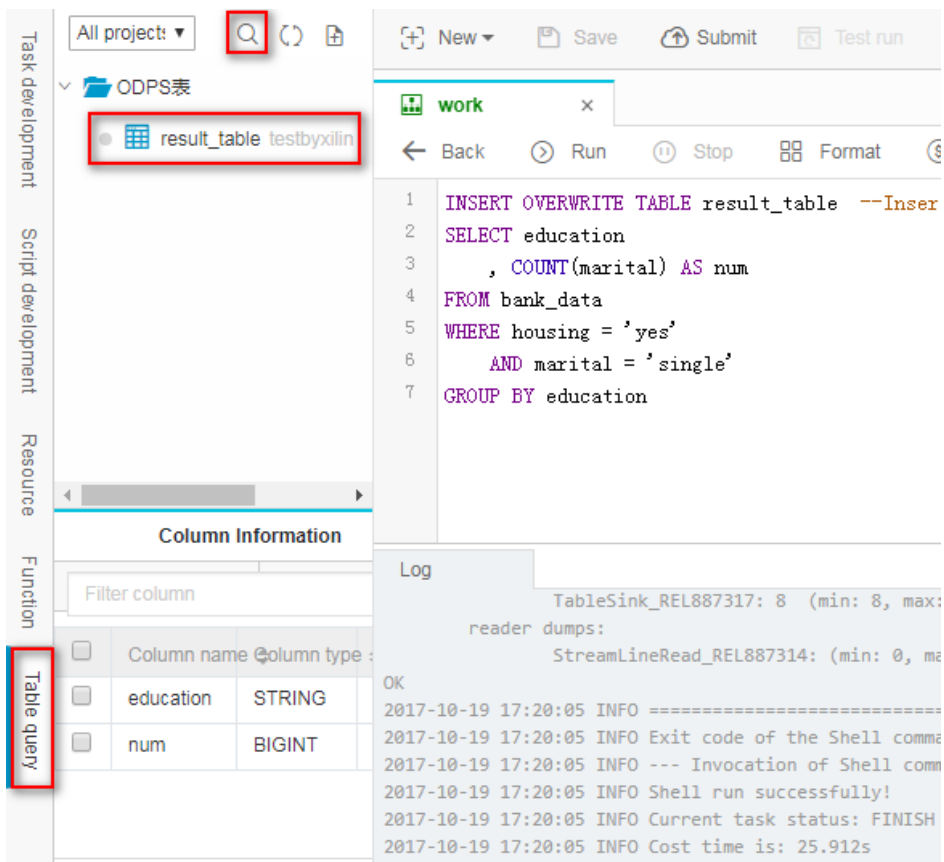
The log window shows the following output:

```

Log
    TableSink_REL887317: 8 (min: 8, max: 8, avg: 8)
    reader dumps:
    StreamLineRead_REL887314: (min: 0, max: 0, avg: 0)
OK
2017-10-19 17:20:05 INFO =====
2017-10-19 17:20:05 INFO Exit code of the Shell command 0
2017-10-19 17:20:05 INFO --- Invocation of Shell command completed ---
2017-10-19 17:20:05 INFO Shell run successfully!
2017-10-19 17:20:05 INFO Current task status: FINISH
2017-10-19 17:20:05 INFO Cost time is: 25.912s

```

Click **Table Query** in the left-side navigation pane, to query data in the table.



The screenshot shows the DataWorks interface with the 'Table Query' option selected in the left-side navigation pane. The SQL query editor and log window are also visible. The 'Table Query' option is highlighted with a red box. The SQL query is as follows:

```

1 INSERT OVERWRITE TABLE result_table --Inser
2 SELECT education
3     , COUNT(marital) AS num
4 FROM bank_data
5 WHERE housing = 'yes'
6     AND marital = 'single'
7 GROUP BY education

```

The log window shows the following output:

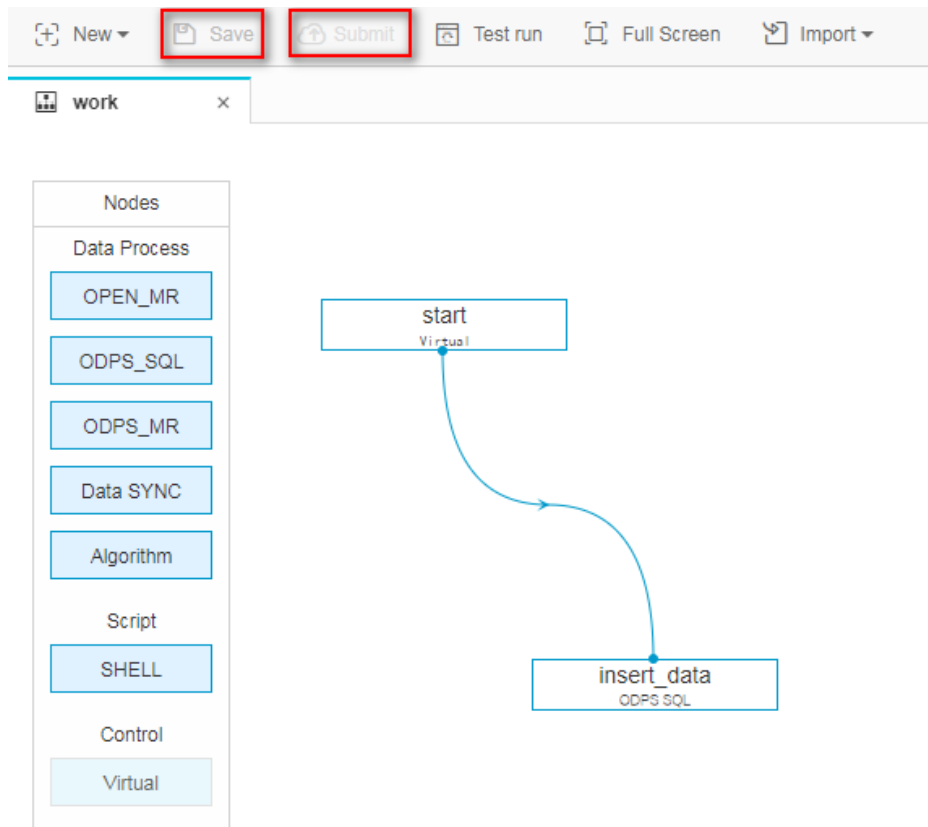
```

Log
    TableSink_REL887317: 8 (min: 8, max:
    reader dumps:
    StreamLineRead_REL887314: (min: 0, ma
OK
2017-10-19 17:20:05 INFO =====
2017-10-19 17:20:05 INFO Exit code of the Shell comm
2017-10-19 17:20:05 INFO --- Invocation of Shell comm
2017-10-19 17:20:05 INFO Shell run successfully!
2017-10-19 17:20:05 INFO Current task status: FINISH
2017-10-19 17:20:05 INFO Cost time is: 25.912s

```

Save and submit the flow

After running and debugging the ODPS_SQL node "insert_data", return to the flow page. Click **Save** and **Submit** the whole flow.

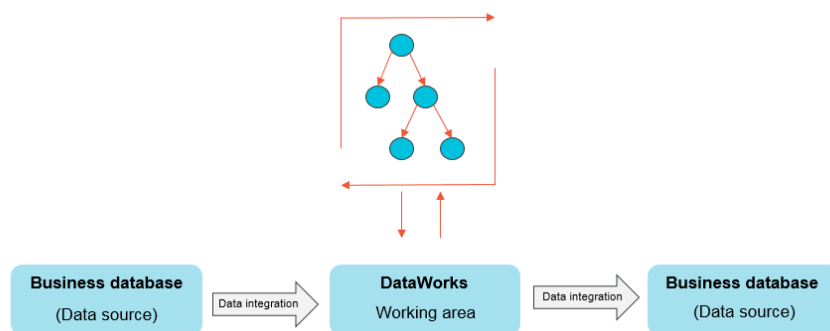


Subsequent steps

Now you have learned how to create, save, and submit the flow. You can proceed with the next tutorial that demonstrates how to create a synchronization task to export data to the different types of the data sources. For more information, see [Create a synchronization task to export results](#).

Create a data sync job

The data integration function allows to periodically import business data generated in your system to the workspace and periodically export the flow computing results to the data source you specify for further display or operation.



Currently, data from the following data sources can be imported to or exported from the workspace through the data integration function: RDS, MySQL, SQL Server, PostgreSQL, MaxCompute, ApsaraDB for Memcache, DRDS, OSS, Oracle, FTP, DM, Hdfs, MongoDB, and so on. For more information, see [Supported data source types](#).

This section uses MySQL as an example to show how to export data in MaxCompute to MySQL through the data integration function.

Prerequisites

If your database is a self-built database on ECS or a RDS/MongoDB data source, you must add the data synchronization machine IP address whitelist to your ECS security group or RDS/MongoDB whitelist. For more information, see [Add whitelist and security group](#).

Note:

If you use a custom resource group to schedule RDS data synchronization tasks, you must add the machine IP address of the custom resource group to the RDS whitelist.

Procedure

Add a data source

Note:

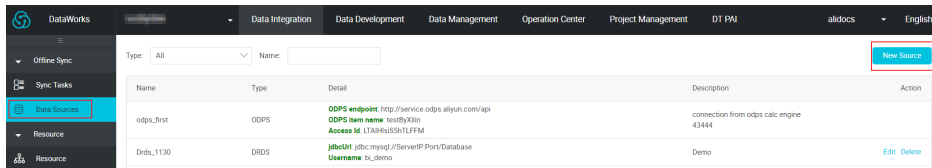
Only the project administrator can create a data source. Other roles can only view the data source.

Log on to the DataWorks console as an administrator and click **Enter Project** in the operations column of the relevant project in the **Project List**.

Click **Data Integration** from the upper menu, and click **Data Sources** in the left-side

navigation pane.

Click **New Source** in the upper-right corner, as shown in the following figure.



Enter the configuration items in the create data source dialog box, as shown in the following figure.

The 'New MySQL data sources' dialog box contains the following fields and elements:

- Type:** A dropdown menu with the selected value 'there are public ip'.
- Name:** A text input field containing 'custom name'.
- Description:** An empty text input field.
- JDBC URL:** A text input field containing the format 'format: jdbc:mysql://ServerIP:Port/Database'.
- username:** An empty text input field.
- password:** An empty text input field.
- test connectivity:** A button labeled 'test connectivity'.
- Warning:** A red warning icon followed by the text: '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', and 'ensure that the database has been launched'.
- Navigation:** 'previous' and 'complete' buttons at the bottom right.

Type: The network type of data sources.

Name: The name must contain letters, numbers, and underscores (), *but cannot begin with a number or an underscore ()*, for example, abc_123.

Description: The description cannot exceed 80 characters.

JDBC URL: jdbc:mysql://host:port/database

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

For configurations of different types of data sources, see the articles under **Data Source Config**.

Click **Test Connectivity**.

If the connectivity test is successful, click **Complete**.

Note: Make sure that the target MySQL database contains tables.

Create the table `odps_result` in the MySQL database. The statements used for table creation are as follows.

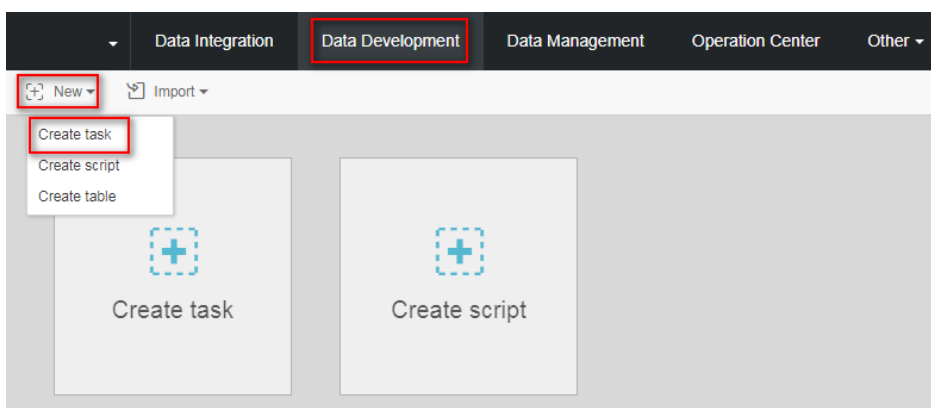
```
CREATE TABLE `ODPS_RESULT` (  
  `education` varchar(255) NULL ,  
  `num` int(10) NULL  
)
```

After the table is created, you can run `desc odps_result;` to view the table details.

Create and configure a synchronization node

This section shows how to create and configure the synchronization node `write_result`, and write data from `result_table` to the MySQL database. The specific steps are as follows.

Create the node `write_result`, as shown in the following figure.



Task type: Workflow task Node task

*Type: Data SYNC

*Name: write_result

*Schedule type: Manual scheduling Periodic scheduling

Description:

Select directory: /
> Task development

Create Cancel

Select the source.

Select the MaxCompute data source and the source table **result_table** and click **Next**, as shown in the following figure.

write_result

1 Select Source 2 Select Target 3 Field Mapping 4 Channel Control 5 Preview & Save

You may need to select the source type of data, it can be your own independent database server, or RDS in Alibaba Cloud, see [support data source type](#)

* Data Source: odps_first (odps)

* Table: result_table

Partition: No Partition ?

Preview Data

Next

Select the target.

Select the MySQL data source and the target table **odps_result** and click **Next**, as shown in the following figure.

write_result

1 Select Source 2 **Select Target** 3 Field Mapping 4 Channel Control 5 Preview & Save

You may need to select the destination type of data, it can be your own independent database server, or RDS in Alibaba Cloud, see support the [data target type](#)

* Data Source: mysql_source (mysql)

* Table: odps_result

Pre-import statement: Enter the sql script executed before importing the data...

Prepare statements after import: Enter the sql script after importing the data...

Key Conflict: Replace Into

Previous Next

Map the fields.

Select the mapping between fields. You must configure the field mapping relationships. The **Source Table Fields** on the left correspond (one-to-one) with the **Target Table Fields** on the right.

1 choose source 2 select target 3 **field mapping** 4 channel control 5 preview stored

would you like to source configuration table strap with goals mapping, via audio link will be synchronized linked around the field, also can through peer innuendo volume complete mapping. [data synchronization files](#)

source table field	type	the destination tabl_	type
education	STRING	education	STRING
num	BIGINT	num	BIGINT

[peer mapping automatic typesetting](#)

previous Next

Control the channel.

Click **Next** to configure the maximum job rate and dirty data check rules, as shown in the following figure.

The screenshot shows the 'Channel Control' step in a task configuration wizard. The progress bar at the top indicates that steps 1 through 4 are completed, and step 4 is the current active step. The configuration options are:

- Maximum Speed Rate:** 1MB/s
- Incorrect records more than:** Dirty data number range, allow dirty data default

Buttons for 'Previous' and 'Next' are visible at the bottom.

Preview and store.

After configuration, you can scroll up or down to view the task configurations. If no errors are found, click **Save**.

The screenshot shows the 'Preview & Save' step in the task configuration wizard. The progress bar indicates that all steps are completed. The configuration options are:

- Data Source:** mysql_source
- Table:** odps_result
- Pre-import statement:** Unfilled
- Prepare statements after import:** Unfilled

Buttons for 'Previous' and 'Save' are visible at the bottom.

Submit a data synchronization task

Once you save a synchronization task click **Submit**, and the synchronization task is submitted to the scheduling system. The scheduling system automatically and periodically runs the task from the second day according to the configuration attributes.

Subsequent steps

Now, you know how to create a synchronization task and export data to data sources of different types. Continue to the next tutorial for further study. This tutorial shows you how to set the scheduling attribute and dependency for a synchronization task. For more information, see [Set task scheduling attribute and dependency](#).

Scheduling and dependence settings

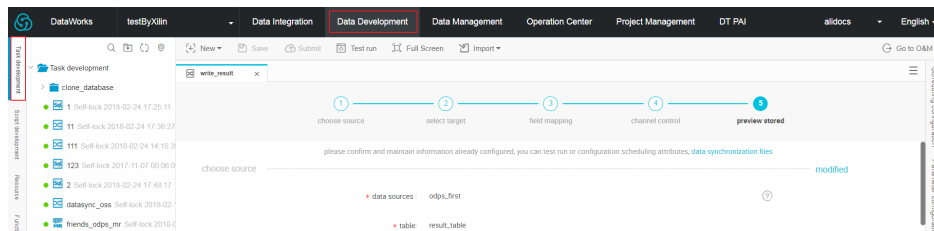
DataWorks provides powerful scheduling capabilities including time-based or dependency-based task trigger functions to perform **tens of millions** of tasks accurately and timely each day, based on DAG relationships. It supports scheduling by minute, hour, day, week, and month. For more information, see [Scheduling configuration](#).

This section uses `write_result` created in [Create a data sync job](#) as an example and configures the scheduling period to weekly, to explain the scheduling configurations and task O&M functions of DataWorks.

Procedure

Configure the scheduling attribute of a synchronization task

Select **Data Development > Task Development**. The task development list is displayed on the left-side of the page.



Double-click any synchronization task that you want to configure, for example, the `write_result` task.

Click **Scheduling Configuration** to configure the **Scheduling attribute** of the task. See the following figure.

- Basic attributes ▶

Scheduling attribute ▼

Scheduling status: Frozen

Auto retry: open ?

Activation date: 1970-01-01 [calendar icon] to 2116-10-20 [calendar icon]

*Scheduling period: Day [dropdown arrow]

*Specific time: 00 [dropdown arrow] : 00 [dropdown arrow]

Scheduling configuration

Parameter configuration

The configuration parameters are described as follows.

Scheduling status: When this parameter is selected, the task is paused.

Error retry: When this parameter is selected, error retry is enabled.

Start date: The date on which the task takes effect, which can be set based on actual needs.

Scheduling period: The operating period of the task, which can be set by month, week, day, hour, and minute. For example, a task can be scheduled weekly.

Specific time: The specific operating time of the task. For example, you can set up the task to run at 02:00 every Tuesday.

Configure the dependency attribute of a synchronization task

After configuring the scheduling attribute of a task, you can configure its dependency attribute. See the following figure.

Dependency attribute ▾

Project:

Upstream task:

Project name	Task name	Owner	Actions
	work	shu...	Delete

Cross-cycle dependency ▾

- Not dependent on the previous scheduling period
- Self-dependent; operation can continue after the conclusion of the previous scheduling period
- Operation can continue after the conclusion of the previous downstream task scheduling period
- Operation can continue after the conclusion of the previous custom task scheduling period

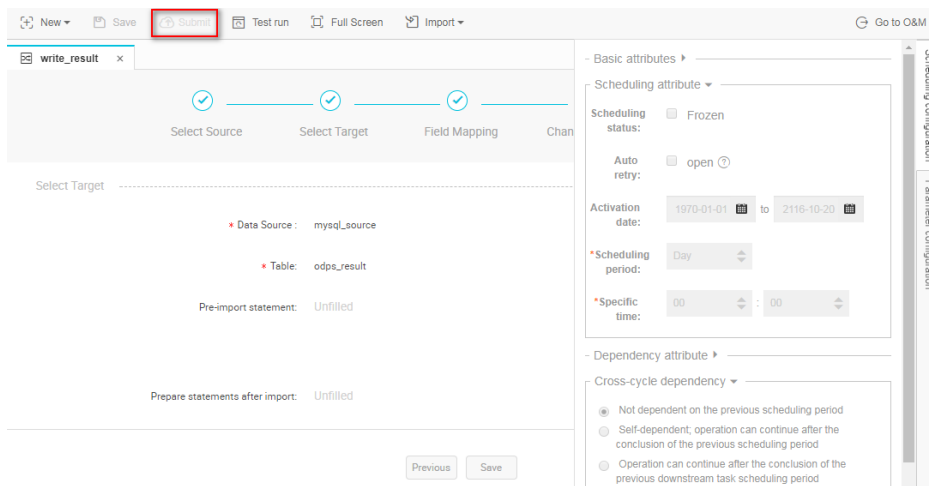
You can configure an upstream dependency for a task. In this way, even if the scheduled time of an instance of the current task is reached, the task can run only after the instance of its upstream task is completed.

The configuration in the preceding figure indicates that instances of the current task are triggered only after the instance of the upstream task `write_result` is finished. You can enter **work** in the upstream task to configure an upstream task for `write_result`.

If no upstream task is configured then, by default the current task is triggered by the project . Therefore, by default, the upstream task of the current task is `project_start` in the scheduling system. By default, a `project_start` task is created as a root task for each project.

Submit a synchronization task

Save the synchronization task `write_result`, and click **Submit** to submit it to the scheduling system. See the following figure.



The system automatically generates an instance for the task at each time point according to the scheduling attribute configuration and periodically runs the task from the second day only after a task is submitted to a scheduling system.

Note: If a task is submitted after 23:30, the scheduling system automatically generates instances for the task and periodically runs the task from the third day.

Subsequent steps

Now you know how to set the scheduling attribute and dependency of a synchronization task. Continue to the next tutorial for further study. This tutorial shows you how to perform periodic O&M for the submitted tasks and view the log troubleshooting results. For more information, see [Perform periodic O&M and view log troubleshooting results](#).

Perform periodic O&M and view log troubleshooting results

In the previous operations, you have set a synchronization task to run at 02:00 every Tuesday. After the task is submitted, you can view the automatic operation results in the scheduling system from the next day.

Now, how can we check whether the instance schedule and dependency are as expected? To work this out, DataWorks provides three triggering methods: test run, data population, and periodic running, which are described as follows:

Test run: The task is triggered manually. If you must check the timing and operation of a

single task, test run is recommended.

Data population: The task is triggered manually. This method applies if you must check the timing and dependencies of multiple tasks or re-execute data analysis and computing from a root task.

Periodic running: The task is triggered automatically. After successful submission, the scheduling system automatically generates task instances at different time points starting from 00:00 of the next day. It checks whether upstream instances of each instance have run successfully according to the scheduled time. If all the upstream instances have run successfully at the scheduled time, the current instance runs automatically without manual intervention.

Note:

The scheduling system periodically generates instances based on the same rules that apply to both manual and automatic triggering modes.

The period can be set to monthly, weekly, daily, hourly, or even by minute. The scheduling system always generates an instance for the task on a specified day or at a specified time.

The scheduling system regularly runs the instance on a specified date and generates operation logs.

Instances rather than on a specified date does not run, and their statuses are directly changed to “Successful” if the running conditions are met. Therefore, no running logs are generated.

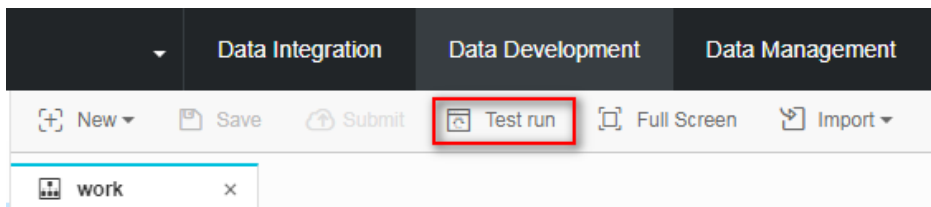
Procedure

The following procedures show how to configure these three triggering methods.

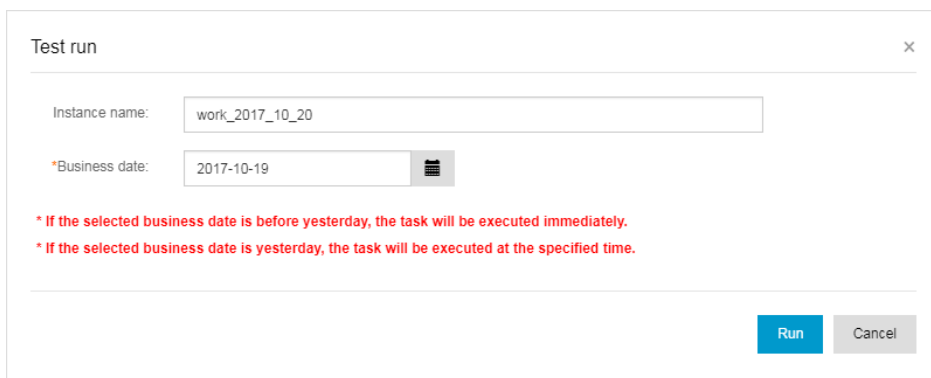
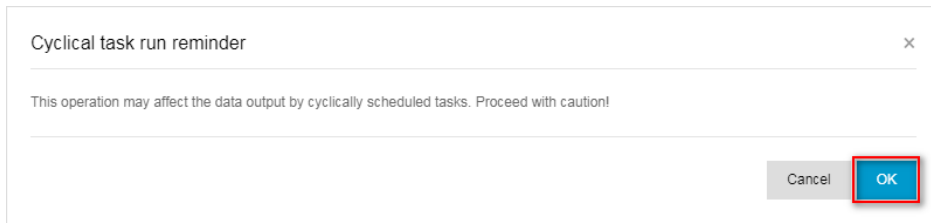
Test run

Manually trigger the test run

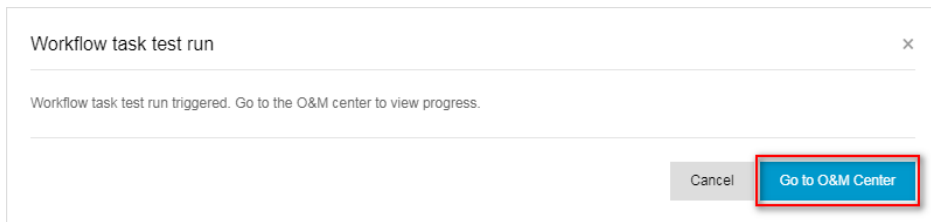
Click the **Test Run** button on the flow page.



As prompted on the page, click **OK** and **Run**.



Click **Go to O&M Center** to view the task operation status.



View the information and operation logs of the test instance

Click the task name to view the instance DAG. In the instance DAG view, right-click an instance to view its dependencies and more information. Also, you can terminate or re-run the instance. In the instance DAG view, double-click an instance and a dialog box appears, showing the task attributes, running logs, operation logs, and code.

Note:

In test run mode, the task is triggered manually. The task runs immediately if the set

time is reached, regardless of the instance' s upstream dependencies.

According to the previously mentioned instance generation rules, set up the task write_result to run at 02:00 every Tuesday. If the business date of test run is Monday (business date = running date -1), the instance runs at 02:00. If not, the instance status is changed to "Successful" at 02:00 and no logs are generated.

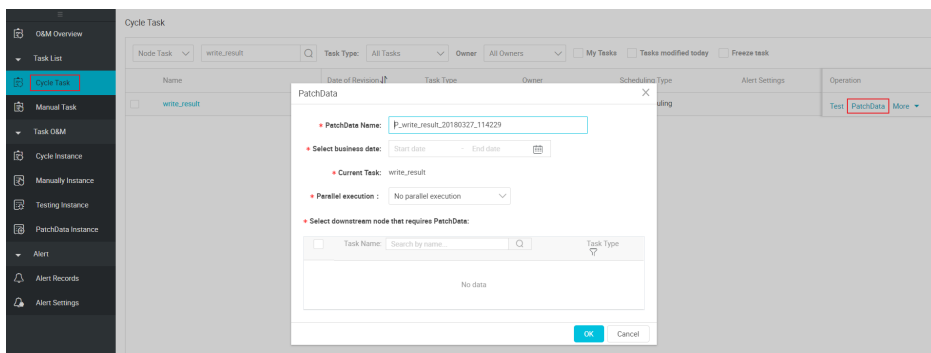
Data population

Manually trigger data population

If you must check the timing and dependency of **multiple tasks** or re-execute data analysis and computing from a root task, go to the **O&M Center > Task List > Cycle Task** page and click **PatchData** to run multiple tasks of a specific period of time.

Log on to the **O&M Center > Cycle Task** and enter the task name.

Select the task query results and click **PatchData**. See the following figure.



Set the business date of the data population as May 11, 2017 to May 12, 2017, select the insert_data and write_result node tasks, and click **OK**.

Click **PatchData Instance**. See the following figure.

Node Task	Name of Workflow/Node Ta	Task Type	All Tasks	Owner	alldocs	<input checked="" type="checkbox"/> My Tasks	<input type="checkbox"/> Tasks modified today	<input type="checkbox"/> Freeze task
	1	2018-02-24 17:25:10	ODPS_SQL	alldocs	Daily Scheduling			
	ftp_cdp	2018-02-24 15:02:14	Data Synchronization	alldocs	Daily Scheduling			
	mysql2odps_a12	2018-02-23 16:38:13	Data Synchronization	alldocs	Daily Scheduling			
	mysql2odps_a11	2018-02-23 16:38:11	Data Synchronization	alldocs	Daily Scheduling			
	mysql2odps_a10	2018-02-23 16:38:09	Data Synchronization	alldocs	Daily Scheduling			
	mysql2odps_a1	2018-02-23 16:38:07	Data Synchronization	alldocs	Daily Scheduling			
	mysql_source_virtual	2018-02-23 16:38:07	Virtual node	alldocs	Daily Scheduling			
	write_result	2017-10-20 11:48:45	Data Synchronization	alldocs	Daily Scheduling			
	project_enf_start	2017-10-18 11:22:27	Virtual node	alldocs				

View information and operation logs of patchdata instance

On the **PatchData Instance** page, find the task instance: Click the task name to view the instance DAG. In the instance DAG view, right-click an instance to view its dependencies and more information. Also, you can terminate or re-run the instance. In the instance DAG view, double-click an instance and a dialog box appears, showing the task attributes, running logs, operation logs, and code.

Note:

Data population task instances depends on the previous day instances. For example, for a patchdata task within the period from September 15, 2017 to September 18, 2017, if the instance on the 15th fails to run, the instance on the 16th is not run.

According to the previously mentioned instance generation rules, set up the task write_result to run at 02:00 each Tuesday. If the business date selected during patchdata is Monday (service date = running date -1), the instance runs at 02:00. If not, the instance status is changed to "Successful" at 02:00 and no logs are generated.

Periodic automatic run

In periodic automatic run mode, the scheduling system automatically triggers tasks according to all task scheduling configurations. Therefore, no operation portal is provided. You can view the instance information and operation logs by using either of the following methods.

Go to the **O&M Center > Cycle Task** page, select parameters such as service date or running date, search instances corresponding to the task write_result, and then right-click an instance to view its information and operation logs.

Click the task name to view the instance DAG. In the instance DAG view, right-click an instance to view its dependencies and more information. Also, you can terminate or re-run the instance. In the instance DAG view, double-click an instance and a dialog box appears,

showing the task attributes, running logs, operation logs, and code.

Note:

If the initial status of a task instance is “Not Run” , when the scheduled time is reached, the scheduling system checks whether all the upstream instances are successful or not.

The instance is triggered only when all of its upstream instances are successful and its scheduled time is reached.

For an instance in a “Not Run” status, check that all its upstream instances are successful and its scheduled time is reached.