Alibaba Cloud Table Store

Data channels

Issue: 20181016

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 intellectual 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.	 **Note:** 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 `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *{stand | slave}*

Contents

Legal disclaimer I
Generic conventions I
1 MaxCompute 1
1.1 Overview 1
1.2 Incremental synchronization (wizard mode) 2
1.3 Full export (script mode) 9
2 OSS 16
2.1 Overview 16
2.2 Full export (script mode) 18
2.3 Incremental synchronization (script mode) 20

1 MaxCompute

1.1 Overview

Table Store is a distributed NoSQL data storage service that is built on Alibaba Cloud Apsara distributed system. It uses data partitioning and load balancing techniques to seamlessly scale up data size and access concurrency, providing storage of, and real-time access to, massive structured data.

MaxCompute is a big data computing service that provides a fast and fully hosted PB-level data warehouse solution, allowing you to analyze and process massive data economically and efficientl y.

Scenarios

Table Store: Provides professional data-persistent storage service and user-oriented real-time read/write operations with high concurrency and low latency.

MaxCompute: Provides computing services, which are generally used for cleaning, correcting, and calculating data.

Activation

Activate Table Store

1. Go to the Table Store details page.
2. Click **Buy Now**.
3. In the *Table Store console*, create *instances* and *tables*.

> **Note:**
>
> - To use the incremental tunnel, you must activate the *Stream function* for tables. You can select 24 hours for the validity period.
> - Table Store supports the reserved CUs and additional CUs. If the reserved read and write CUs are both set to zero during table creation, then the additional read and write CU is used. You can adjust the reserved read/write CUs of each table at any time.
> - Table Store offers each registered account 25 GB of free storage per month.

Activate MaxCompute

1. Go to the MaxCompute details page.

2. Click **Buy now**.

> **Note:**
> Two billing methods are available, which are prepayment by CU cost and Pay-As-You-Go.

Data tunnel

- Real-time

  Direct read and write
- Offline
  - Incremental synchronization to MaxCompute

    Wizard mode
  - Full export to MaxCompute

    Script Mode
  - Full import to Table Store

    Script Mode

1.2 Incremental synchronization (wizard mode)

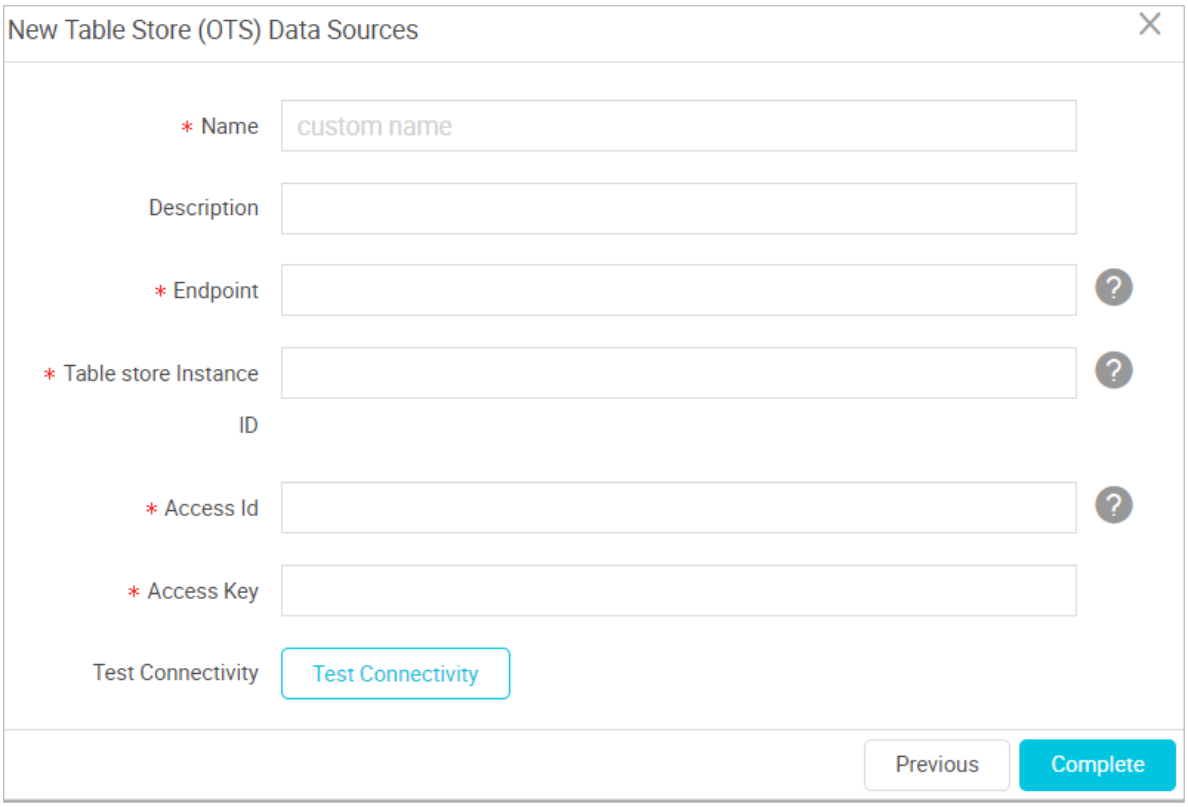
Data Integration supports data synchronization in wizard mode and script mode. The wizard mode is simpler while the script mode is more flexible.

This chapter describes how to synchronize incremental data (generated by the Put, Update, and Delete actions) from Table Store to MaxCompute through the Table Store feature in a near-real-time manner.

> **Note:**
> Because the offline synchronization mode is used, a latency of about 10 minutes exists.

Step 1. Create Table Store data source

1. Log on to the *Data IDE*.
2. If you are using Data Integration for the first time, you must first *create a Data Integration project*.
3. On the **Data Sources** page, click **New Source**.
4. Select Table Store as the data source.
5. Set parameters and click **test connectivity**.



The parameters are described as follows.

Parameter	Description
Name	Name of the Table Store data source. This example uses gps_data.
Description	Description of the data source.
Endpoint	Enter the instance address on the Table Store instance page. • If the Table Store instance is in the same region as the MaxCompute instance, enter the private network address. • If the Table Store instance is not in the same region as the MaxCompute instance , enter the public network address. **Note:** You cannot enter the VPC address.
Table Store ID	Name of the Table Store instance.
Access ID	AccessKeyID of the logon account.

Parameter	Description
Access Key	AccessKeySecret corresponding to the AccessKeyID of the logon account.

Note:
If the connectivity test fails, check whether the endpoint and instance name are correct. If the problem persists, *open a ticket*.

6. Click **complete**. Information about the Table Store data source is displayed on the **Data Sources** page.

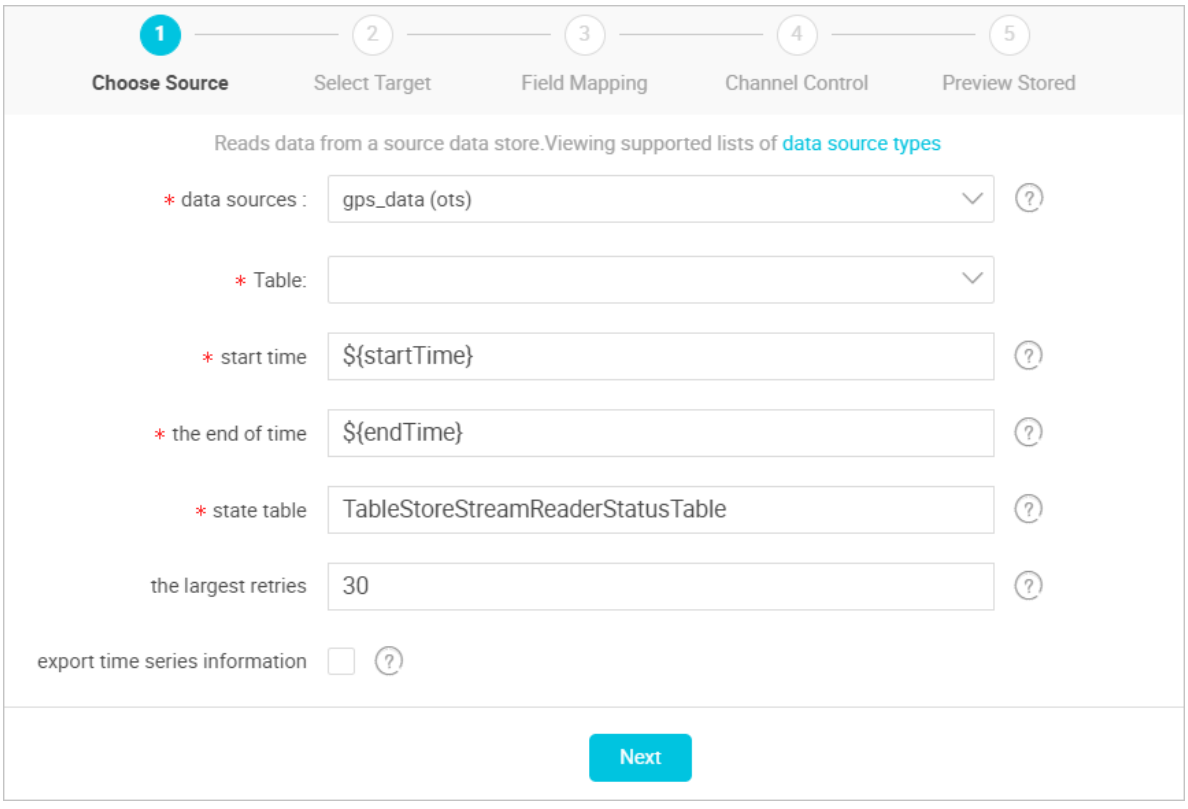
Step 2. Create MaxCompute data source

This operation is similar to Step 1. You only need to select MaxCompute as the data source.

In this example, the MaxCompute data source is named OTS2ODPS.

Step 3: Create an incremental real-time data tunnel

1. On the *Data IDE* page, click **Sync Tasks**.
2. At the right side of the page, click **Create a synchronization task**.
3. Select **Wizard mode**.
4. Select the Table Store data source created in Step 1.



The parameters are described as follows.

Parameter	Description
Data sources	The Table Store data source you created. In this example, gps_data is selected.
Table	Data Integration automatically obtains the latest data table from Table Store. Stream must be activated for the selected table. If Stream is not activated, click **Activate Stream in One Click** at the right side to activate Stream. The incremental data is valid for up to 24 hours.
Start time	Start time of incremental export. For a periodic task, the variable value is required. The default value is `${start_time}`.
End time	End time of incremental export. For a periodic task, the variable value is required. The default value is `${end_time}`.
Status table	It is used to store status values during incremental export. The default value is recommended.
Maximum number of retries	It indicates the maximum number of retries to perform during when the network is unstable. The default value is 30. You can set the value as needed.
Export time series information	It indicates whether the exported data contains the time information. It is not selected by default.

5. On the **Select Target** page, select the MaxCompute data source created in Step 2.

   The parameters are described as follows.

Parameter	Description
Data sources	The MaxCompute data source you created. In this example, OTS2ODPS is selected.

Parameter	Description
Table	Select a table in this data source. If no table is available, at the right side click **Create New Target Table** to create a table. In the dialog box that appears, replace **`your_table_name`** with the name of the table to be created, for example, ots_gps_data. (Because timestamp is a reserved field in MaxCompute and cannot be used in this box, ts can be used to represent timestamp if necessary.)*http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/pic/61034/cn_zh/1510908157657/61034-6.png*
Partition information	The default value is `${bdp.system.bizdate}`, indicating data in MaxCompute is partitioned by date.
Cleaning rule	Select **Clean Existing Data Insert Overwrite Before Writing**.

6. On the **Field Mapping** page, make sure the Table Store table maps the MaxCompute table.
7. On the **Channel Control** page, set the parameters.

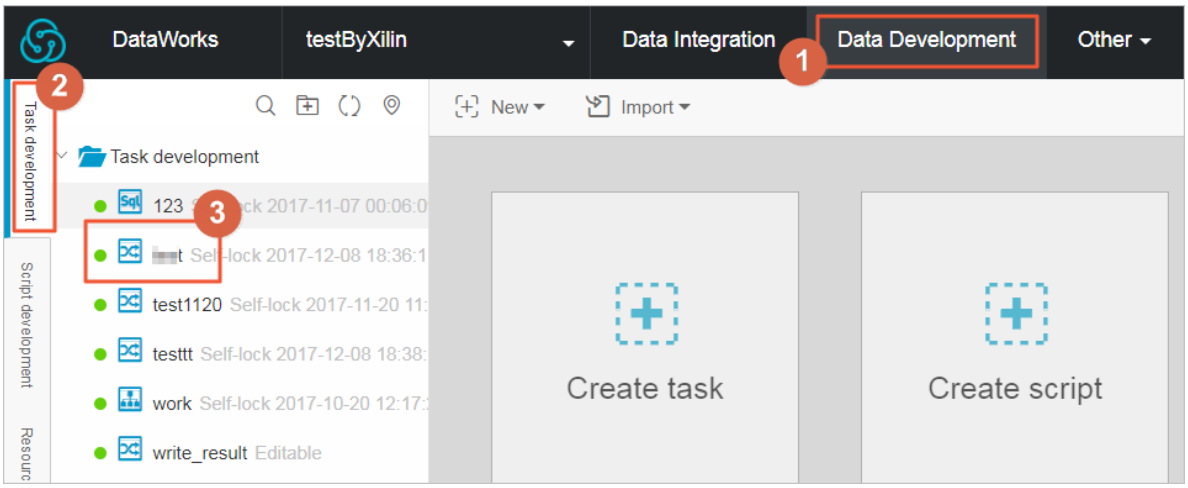
   The parameters are described as follows.

Parameter	Description
Job speed limit	Range: 1 MB/s to 20 MB/S. To request a higher job speed limit, *open a ticket*.
Number of concurrent jobs	The maximum value is 10. Maximum rate of a job = Task speed limit/Number of concurrent jobs
Number of error records	The task fails when the number of error records exceeds the value. The default value is 0.

8. On the **preview** page, check the configurations.
9. Click **Save**. In this example, the task name that is saved is OTStoODPS.

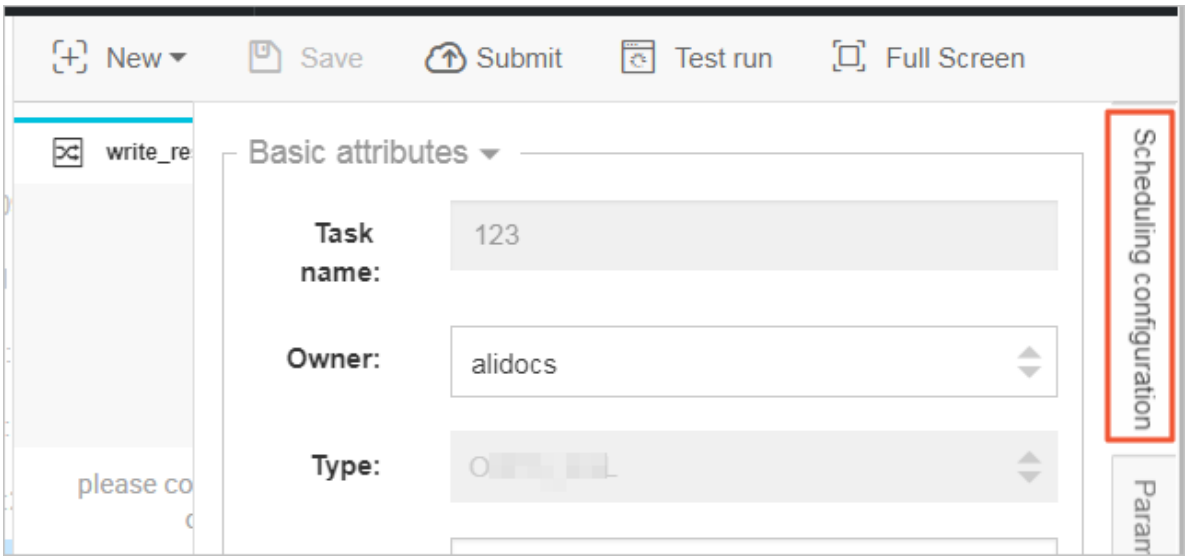
Step 4. Set scheduling parameters

1. At the top of the page, click **Data Development**.
2. On the **Task Development** tab, double-click the created task OTStoODPS.



3. Click **Scheduling configuration** to set the scheduling parameters.

   To set the task to run on the next day, configure the following parameters as shown.



   The parameters are described as follows.

Parameter	Description
Scheduling status	Indicates the running of the task. By default, it is not selected.
Error retry	We recommend that you select this parameter so that the system can retry if an error occurs.
Start date	The default value is recommended.
Scheduling cycle	Minute is used in this example.
Start time	It is set to 00:00 in this example.

Parameter	Description
Scheduling interval	The scheduling interval is set to 5 minutes in this example.
End time	It is set to 23:59 in this example.
Dependency attributes	Set the **Dependency Attribute** field based on your business needs, or retain the default value.
Cross-cycle dependency	Set the **Cross-Cycle Dependency** field based on your business needs, or retain the default value.

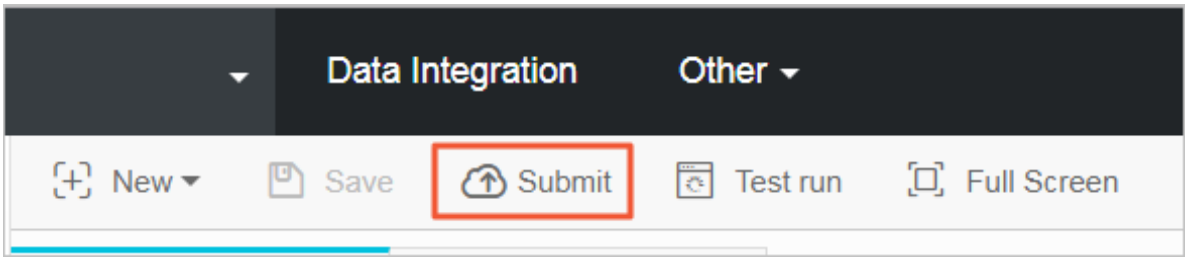
4. Click **Parameter Configuration**.

   The parameters are described as follows.

Parameter	Description
${bdp.system.bizdate}	It does not need to be configured.
startTime	It is the **Start Time** variable set in **Scheduling Configuration**. In this example, it is set to $[yyyymmddhh24miss-10/24/60], indicating a time equal to the scheduling task start time minus 10 minutes.
endTime	It is the **End Time** variable set in **Scheduling Configuration**. In this example, it is set to **$[yyyymmddhh24miss-5/24/60]**, indicating a time equal to the scheduling task start time minus 5 minutes.

Step 5. Submit the task

1. At the top of the page, click **Submit**.

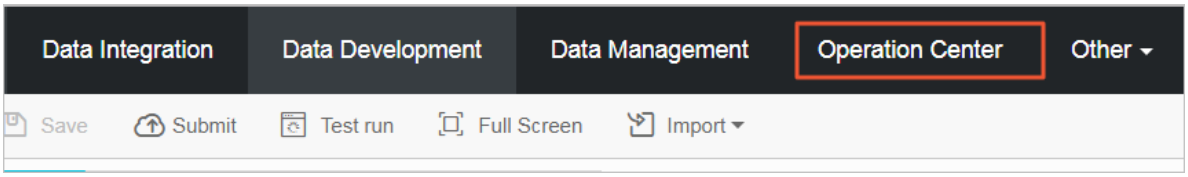


2. In the dialog box, click **Confirm Submission**.

After the task is submitted, the system prompts **The current file is read-only**.

Step 6. Check the task

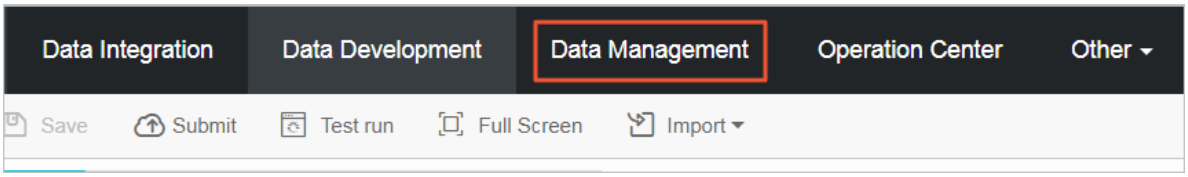
1. At the top of the page, click **Operation Center**.



2. In the left-side navigation pane, click **Task List** > **Cycle Task** to view the created task OTStoODPS.
3. The task starts running at 00:00 on the next day.
   - In the left-side navigation pane, click **Task O&M** > **Cycle Instance** to view scheduling tasks to be executed on the day. Click the instance name to view the details.
   - You can view the log when a task is running or after it is completed.

Step 7. View the data that has been imported to MaxCompute

1. At the top of the page, click **Data Management**.



2. In the left-side navigation pane, click **Query Data**. All the tables in MaxCompute are listed.
3. Find the table (ots_gps_data) to which the data is imported, and click the table to go to the table details page.
4. Click **Data Preview** to view the imported data.

1.3 Full export (script mode)

Data Integration supports data synchronization in wizard mode and script mode. Wizard mode is simpler while script mode is more flexible.

This topic describes how to export full data from Table Store (generated by the Put, Update, and Delete actions) to MaxCompute through Data Integration.

Step 1. Create a Table Store data source.

Note:

- Skip this step if a data source is already created.

- If you do not want to create the data source, you can specify the endpoint, instanceName, AccessKeyID, and AccessKeySecret on the subsequent configuration page.

For more information about how to create a data source, see *Create a Table Store data source*.

Step 2. Create a MaxCompute data source

This operation is similar to Step 1. You only need to select **MaxCompute** as the data source.

In this example, the data source is named OTS2ODPS.

Step 3. Create a full export tunnel

1. On the *Data IDE* page, click **Sync Tasks**.
2. Select **Script Mode**.
3. In the **Import Template** dialog box that appears, set **Source Type** to Table Store (OTS) and **Type of Objective** to MaxCompute (ODPS).
4. Click **OK** to go to the configuration page.
5. Set configuration parameters.

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
  "errorLimit": {
    "record": "0"    # Maximum number of errors allowed
  },
  "speed": {
    "mbps": "1",   # Maximum traffic, in Mbps.
    "concurrent": "1"  # Number of concurrent tasks.
  }
},
"reader": {
  "plugin": "ots",  # Name of the plugin read
  "parameter": {
    "datasource": "",  # Name of the data source
    "table": "",  # Name of the table
    "column": [  # Name of the column in Table Store that needs to
be exported to MaxCompute
      {
        "name": "column1"
      },
      {
        "name": "column2"
      },
      {
        "name": "column3"
      },
      {
        "name": "column4"
      },
      {
        "name": "column5"
```

```
      }
    ],
    "range": {  # Range of the data to be exported. In full export
mode, the range is from INF_MIN to INF_MAX.
      "begin": [ # Start position of the data to be exported. The
minimum position is INF_MIN. The number of configuration items set
in "begin" must be the same as the number of primary key columns of
the table in Table Store.
        {
          "type": "INF_MIN"
        },
        {
          "type": "INF_MIN"
        },
        {
          "type": "STRING",  # Indicates that the start position in
the third column is begin1.
          "value": "begin1"
        },
        {
          "type": "INT",  # Indicates that the start position in the
 fourth column is 0.
          "value": "0"
        }
      ],
      "end": [  # End position of the data to be exported
        {
          "type": "INF_MAX"
        },
        {
          "type": "INF_MAX"
        },
        {
          "type": "STRING",
          "value": "end1"
        },
        {
          "type": "INT",
          "value": "100"
        }
      ],
      "split": [  # Indicates the partition scope, which is not
configured in normal cases. If performance is poor, you can open a
ticket to submit a query.
        {
          "type": "INF_MIN"
        },
        {
          "type": "STRING",
          "value": "splitPoint1"
        },
        {
          "type": "STRING",
          "value": "splitPoint2"
        },
        {
          "type": "STRING",
          "value": "splitPoint3"
        },
        {
          "type": "INF_MAX"
        }
```

```
      ]
    }
  }
},
"writer": {
  "plugin": "odps",  # Name of the plugin written by MaxCompute
  "parameter": {
    "datasource": "",  # Name of the MaxCompute data source
    "column": [],  # Name of the column in MaxCompute. The column
name sequence corresponds to that in Table Store.
    "table": "",  # Name of a table in MaxCompute. It must be
created first; otherwise, the task may fail.
    "partition": "",  # It is required if the table is partitioned.
For non-partition tables, do not set this parameter. The partition
information of the data table must be written. Specify the parameter
 until the last-level partition.
    "truncate": false  # Indicates whether to clear the previous
data
  }
}
}
}
```

6. Click **Save**.

Step 4. Run the task (test)

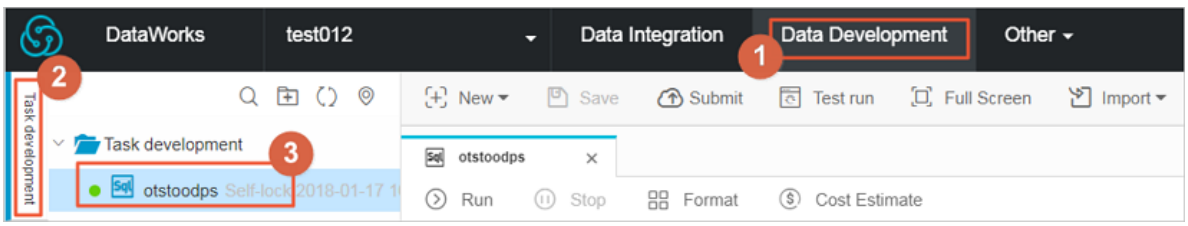
1. At the top of the page, click **operation**.

   If no variable is included in the configurations, the task is executed immediately. If a variable exists, you must enter the actual value of the variable, and then click **OK**. Then, the task starts running.

2. After running the task, you can check whether the task is successful, and view the number of exported data rows in the log.

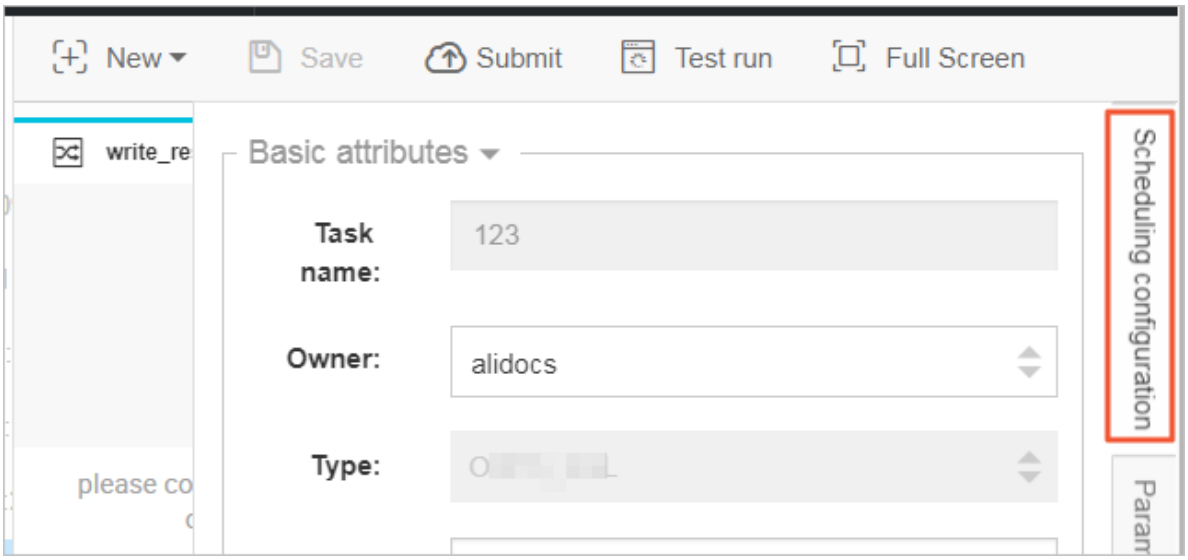
Step 5. Set scheduling parameters

1. At the top of the page, click **Data Development**.
2. On the **Task Development** tab, double-click the created task OTStoODPS.



3. Click **Scheduling Configuration** to set the scheduling parameters.

   To set the task to start running on the next day, configure the following parameters as shown.



The configurations are described as follows:

Parameter	Description
Scheduling status	It is not selected by default, indicating running the task.
Auto retry	We recommend that you select this parameter so that the system can retry after an error occurs.
Activation date	The default value is recommended.
Scheduling period	Minute is used in this example.
Start time	It is set to 00:00 in this example.
Interval	The scheduling interval is set to 5 minutes in this example.
End time	It is set to 23:59 in this example.
Dependency attribute	Set the **`Dependency Attribute`** based on your business needs, or retain the default value.
Cross-cycle dependency	Select **Self-dependent; operation can continue after the conclusion of the previous scheduling period**.

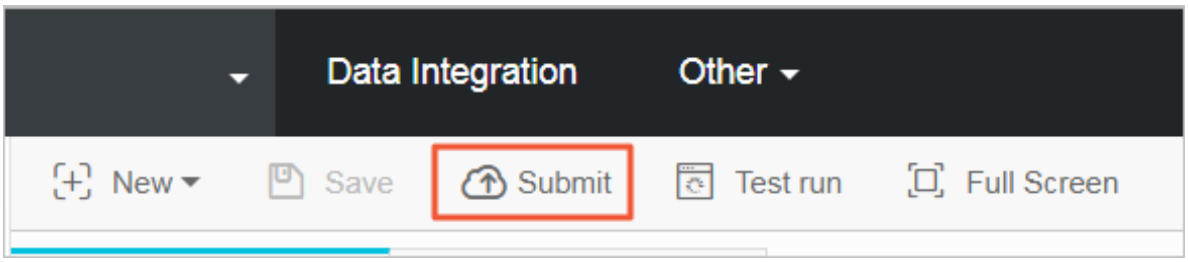
4. Click **Parameter Configuration** to set the parameters.

   The parameters are described as follows.

Parameter	Description
${bdp.system.bizdate}	It does not need to be configured.
startTime	It is the **Start Time** variable set in **Scheduling Configuration**. In this example, it is set to **$[yyyymmddhh24miss-10/24/60]**, indicating a time equal to the scheduling task start time minus 10 minutes.
endTime	It is the **End Time** variable set in **Scheduling Configuration**. In this example, it is set to **$[yyyymmddhh24miss-5/24/60]**, indicating a time equal to the scheduling task start time minus 5 minutes.

Step 6. Submit the task

1. At the top of the page, click **Submit**.

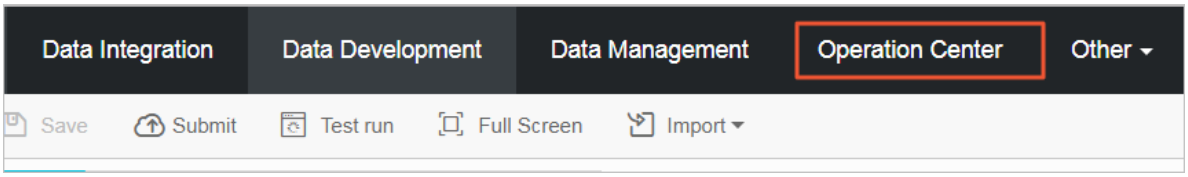


2. In the displayed box, click **Confirm Submission**.

   After the task is submitted, the current file is read-only.

Step 7. Check the task

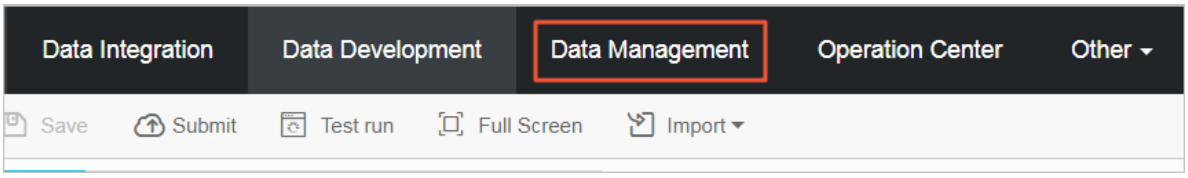
1. At the top of the page, click **Operation Center**.



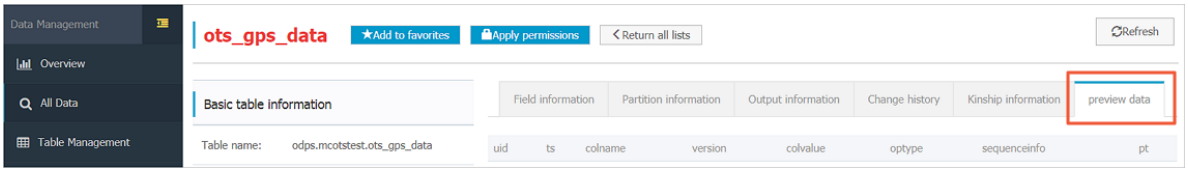
2. In the left-side navigation pane, click **Task List** > **Cycle Task** to view the newly created task OTStoODPS.
3. The task starts running at 00:00 on the next day.
   - In the left-side navigation pane, click **Task O&M** > **Cycle Instance** to view scheduling tasks to be executed on the day. Click the instance name to view the details.
   - You can view the log when a task is running or after it is completed.

Step 8. View the data that has been imported to MaxCompute

1. At the top of the page, click **Data Management**.



2. In the left-side navigation pane, click **All Data**.
3. Find the table (ots_gps_data) to which the data is imported, and click the table to go to its corresponding details page.
4. At the right-side, click the **preview data** tab to view the imported data.



2 OSS

2.1 Overview

Table Store is a distributed NoSQL data storage service that is built on Alibaba Cloud Apsara distributed system. It uses data partitioning and load balancing techniques to seamlessly scale up data size and access concurrency, providing storage of, and real-time access to, massive structured data.

Object Storage Service (OSS) is a massive-volume, secure, low-cost, and highly-reliable cloud storage service. It provides 99.999999999% data reliability. You can use RESTful API for storage and access in any place on the Internet. Its capacity and processing capability can be elastically scaled, and multiple storage modes are provided, comprehensively optimizing the storage cost.

Scenarios

Table Store: Provides professional data-persistent storage service and user-oriented real-time read/write operations with high concurrency and low latency.

OSS: Supports backup at an extremely low cost.

Usage

- Write

  Data can be directly written to Table Store.
- Read

  Data can be directly read from Table Store.
- Back up

  Automatic backup is supported.
- Restoration

  Data can be re-written to Table Store through Data Integration (OSSReader and OTSWriter).

Constraints

- Write by whole row

  Table Store Stream requires that a whole row of data be written to Table Store each time. Currently, the whole-row data write mode is applied to the writing of time sequence data such as IoT data. Therefore, data cannot be modified subsequently.

- Synchronization latency

  Currently, periodic scheduling is used and the scheduling interval is 5 minutes. The plugin has a latency of 5 minutes and the total latency of a synchronization task is 5 to 10 minutes.

Activation

- Activate Table Store

  1. Go to the Table Store product details page.
  2. Click **Buy Now**.
  3. In the *Table Store console*, create *instances* and *data tables*.

  > **Note:**
  > - To use the incremental tunnel, you must activate the *Stream function* for data tables. You can select 24 hours for the validity period.
  > - Table Store supports the reserved CUs and additional CUs. If the reserved read and write CUs are both set to zero during table creation, then the additional read and write CU is used. You can adjust the reserved read/write CU of each table at any time.
  > - Table Store offers each registered account 25 GB of free storage per month.

- Activate OSS

  1. Go to the OSS product details page.
  2. Click **Buy Now**.

Data tunnel

Offline

- Export the full data to OSS.

  — *Script mode*

- Synchronize data to OSS in incremental mode.

  — *Script mode*

- Fully import data into Table Store.

  — Script mode

2.2 Full export (script mode)

Data Integration supports data synchronization in wizard mode and script mode. Wizard mode is simpler, while script mode is more flexible.

This section describes how to export full data in Table Store to OSS using the script mode of Data Integration, so that you can download the data as needed or save it as backup data of Table Store to OSS.

Channels

Script mode of Data Integration:

- Reader: OTSReader
- Writer: OSSWriter

Step 1. Create a Table Store data source

> **Note:**
> Skip this step if you have created a Table Store data source.

For more information about how to create a data source, see *Create a Table Store data source*.

Step 2. Create an OSS data source

This operation is similar to Step 1. You only need to select OSS as the data source.

> **Note:**
> During parameter configuration of the OSS data source, Endpoint does not contain bucketName.

Step 3. Create an export task

1. Log on to the *Data Integration console*.
2. On the **Sync Tasks** page, select **Script Mode**.
3. In the **Import Template** dialog box, set **Source Type** to Table Store (OTS) and **Type of Objective** to OSS.
4. Click **OK** to go to the configuration page.

Step 4. Set configuration items

1. On the configuration page, templates for OTSReader and OSSWriter are provided. Complete the configurations by referring to the following annotations.

```
{
"type": "job",    # It cannot be modified.
```

```
"version": "1.0",  # It cannot be modified.
"configuration": {
 "setting": {
   "errorLimit": {
     "record": "0"  # The import task fails when the number of error
 records exceeds the value.
   },
   "speed": {
     "mbps": "1",  # Import speed, in Mbps.
     "concurrent": "1"  # Concurrency.
   }
 },
 "reader": {
   "plugin": "ots",  # It cannot be modified.
   "parameter": {
     "datasource": "",   # Name of the data source in Data
Integration, which must be set in advance. You can configure data
source or write authentication information such as the AccessKeyID
in plaintext. We recommend that you configure data source.
     "table": "",    # Table name in Table Store.
     "column": [   # Name of the column that needs to be exported
 to OSS. If all the columns need to be exported to OSS, set this
parameter to an empty array.
       {
         "name": "column1"    # Name of the column in Table Store,
which needs to be imported to OSS
       },
       {
         "name": "column2"   # Name of the column in Table Store,
which needs to be imported to OSS
       }
     ],
     "range": {
       "begin": [
         {
           "type": "INF_MIN"   # Start position of the first primary
 key column in Table Store. If you want to export full data, set
this parameter to INF_MIN. If you want to export a portion of the
 data, set this parameter as needed. The number of configuration
 items in "begin" must be the same as the number of primary key
columns.
         }
       ],
       "end": [
         {
           "type": "INF_MAX"   # End position of the first primary
key column in Table Store. If you want to export full data, set this
 parameter to INF_MAX. If you want to export a portion of the data,
set this parameter as needed.
         }
       ],
       "split": [  # Used to configure partition information about
the Table Store table, which can accelerates the export. In the next
 version, this configuration is automatically processed.
       ]
     }
   }
 },
 "writer": {
   "plugin": "oss",
   "parameter": {
     "datasource": "",  # Name of the OSS data source
```

```
      "object": "",  # Prefix of the object excluding the bucket name
, for example, tablestore/20171111/. If the export is scheduled, a
variable, for example, tablestore/${date}, must be used, and ${date
} must be configured when the scheduling parameters are set.
      "writeMode": "truncate", # truncate, append, and nonConflict
 are supported. truncate is used to clear existing files with the
same name, append is used to add the data to existing files with the
 same name, and nonConflict is used to return an error when files
with the same name exist. truncate is used during full export.
      "fileFormat": "csv", # CSV and TXT are supported.
      "encoding": "UTF-8",  # Encoding mode
      "nullFormat": "null", # Defines a string identifier that
represents the null value. It can be an empty string.
      "dateFormat": "yyyy-MM-dd HH:mm:ss",  # Time format
      "fieldDelimiter": "," # Delimiter of each column
    }
  }
}
}
```

2. Click **Save** to save the task.

Step 5. Run the task

1. Click **operation** to run the task.

   If the configurations contain variables, for example, *${date}*, the variable setting page is displayed. You can set only specific values.

2. View logs in the lower part of the page.

   If no error is logged, the task is successfully executed, and you can check the data in the target OSS instance.

   Note:
   Full export is generally a one-time task, and thus you do not need to set automatic scheduling parameters. For more information about how to set the scheduling parameters, see *Incremental synchronization*.

Step 6. Check the data exported to OSS

1. Log on to the *OSS console*.
2. Select the bucket and file name, and verify its contents.

2.3 Incremental synchronization (script mode)

Data Integration supports data synchronization in wizard mode and script mode. Wizard mode is simpler while script mode is more flexible.

This section describes how to synchronize incremental data in Table Store to OpenSearch using the script mode of Data Integration.

Channels

Script mode of Data Integration

- Reader: OTSStream Reader
- Writer: OSSWriter

Configure Table Store

No prior configurations required.

Configure OSS

No prior configurations required.

Configure Data Integration

1. Create a Table Store data source.

   Note:

   - If you have already created a Table Store data source, skip this step.
   - If you do not want to create a data source, you can specify the endpoint, instanceName, AccessKeyID, and AccessKeySecret on the subsequent configuration page.

   For more information about how to create a data source, see *Create a Table Store data source*.

2. Create an OSS data source.

   This step is similar to Step 1. You only need to select OSS as the data source.

   Note:
   During parameter configuration of the OSS data source, Endpoint does not contain bucketName.

3. Create a synchronization task.

   a. Log on to the *Data Integration console*.

   b. On the **Sync Tasks** page, select **Script Mode**.

   c. In the **Import Template** dialog box that appears, set **Source Type** to Table Store Stream (OTS Stream) and **Type of Objective** to OSS.

   d. Click **OK** to go to the configuration page.

4. Set configuration items.

   a. On the configuration page, templates of OTSStreamReader and OSSWriter are provided.

   Complete the configurations by referring to the following annotations.

```
{
"type": "job",
"version": "1.0",
"configuration": {
"setting": {
"errorLimit": {
 "record": "0"  # Allowed number of errors. If the number of
errors exceeds the value, the synchronization task fails.
},
"speed": {
 "mbps": "1",  # Maximum traffic of each synchronization task.
 "concurrent": "1"   # Number of concurrent synchronization tasks
each time.
}
},
"reader": {
"plugin": "otsstream",  # Name of the Reader plugin.
"parameter": {
 "datasource": "", # Name of the Table Store data source. If this
 parameter is set, you do not need to set endpoint, accessID,
accessKey, and instanceName.
 "dataTable": "", # Name of the table in Table Store.
 "statusTable": "TableStoreStreamReaderStatusTable", # Table that
 stores the Table Store Stream status; using the default value is
recommended
 "startTimestampMillis": "",  # Start time of the export. In
incremental export mode, the task needs to be executed cyclically
, and the start time is different at each execution. Therefore,
you must set a variable, for example, ${start_time}.
 "endTimestampMillis": "",  # End time of the export. You must set
 a variable, for example, ${end_time}.
 "date": "yyyyMMdd",  # Date from which data is exported. This
 parameter is the same as startTimestampMillis and endTimesta
mpMillis, and therefore must be deleted.
 "mode": "single_version_and_update_only", # Format of the data
exported from Table Store Stream. Currently, the parameter must be
 set to single_version_and_update_only. Add this parameter if it
is not in the configuration template.
 "column":[  # Names of the columns to be exported from Table
Store to OSS. Add this parameter if it is not in the configuration
 template. Set this parameter as needed.
          {
             "name": "uid"  # Name of the column. It is the
primary key column in Table Store.
          },
          {
             "name": "name"  # Name of the column. It is an
attribute column in Table Store.
           },
 ],
 "isExportSequenceInfo": false, # This parameter can only be set
to false in single_version_and_update_only mode.
 "maxRetries": 30 # Maximum number of retry times.
}
},
```

```
"writer": {
"plugin": "oss", # Name of the Writer plugin
"parameter": {
 "datasource": "", # Name of the OSS data source
 "object": "",  # Prefix of the name of the last file to be backed
 up to OSS. The recommended value is the Table Store instance name
, table name, or date, for example, "instance/table/{date}".
 "writeMode": "truncate", # truncate, append, and nonConflict are
 supported. truncate is used to clear existing files with the same
 name, append is used to add the data to existing files with the
same name, and nonConflict is used to return an error when files
with the same name exist.
 "fileFormat": "csv", # File format
 "encoding": "UTF-8", # Encoding mode
 "nullFormat": "null", # Mode of representation in a TXT file
under control
 "dateFormat": "yyyy-MM-dd HH:mm:ss", # # Time format
 "fieldDelimiter": "," # Delimiter of each column
}
}
}
}
```

b. Click **Save**.

5. Run the task.

a. Click **operation**.

b. In the dialog box that appears, set the variable parameters.

c. Click **OK**.

d. After the task is completed, log on to the *OSS console* to verify whether files are backed up.

6. Configure scheduling.

a. Click **Submit**.

b. In the dialog box that appears, set the scheduling parameters.

The parameters are described as follows.

Parameter	Description
Scheduling type	Select **cycle control**.
Automatically re-run	This parameter indicates that the task reruns for three times at an interval of 2 minutes if the task fails.
Start date	The default value is recommended, which is from January 1, 1970 to 100 years later.
Scheduling cycle	Select Minute.
Start Time	Select “00:00 to 23:59”, which indicates that scheduling is required for a full day.

Parameter	Description
Interval	Select 5 Minutes.
start_time	Enter `$[yyyymmddhh24miss-10/24/60]`, which indicates the time of the scheduling task minus 10 minutes.
end_time	Enter `$[yyyymmddhh24miss-5/24/60]`, which indicates the time of the scheduling task minus 5 minutes.
date	Enter `${bdp.system.bizdate}`, which indicates the scheduling date.
Dependency attributes	Set this parameter if a dependency exists . If no dependency exists, do not set this parameter.
Cross-cycle dependency	**Self-dependent**: The operation can continue only after the previous scheduling cycle is completed.

c. Click **OK**.

The periodic synchronization task is configured, the configuration file status is Read-only.

7. Check the task.

a. At the top of the page, click **Operation Center**.

b. On the left-side navigation pane, click **Task List** > **Cycle Task** to view the created synchronization task.

c. The new task begins running at 00:00 on the next day.

- In the left-side navigation pane, choose **Task O&M** > **Cycle Instance** to view each pre-created synchronization task of the day. The scheduling interval is 5 minutes and each task processes data from the past 5 to 10 minutes.
- Click the instance name to view its details.

d. You can view the log when a task is running or after it is completed.

8. Check the data exported to OSS.

Log on to the *OSS console* to check whether a new file is generated and whether the file content is correct.

Once the preceding settings are completed, data in Table Store can be automatically synchronized to OSS at a latency of 5 to 10 minutes.