Alibaba Cloud
MaxCompute

Tools and Downloads

Issue: 20181128

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 disseminat ed 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, adjustment s, 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.	**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

1 Client

This article describes how to use the basic functions of the MaxCompute using client command line tool. Before using the MaxCompute client, you must *install and configure the client* .

> 📋 **Note:**
>
> - Do not perform the analysis operation based on the output format of the client. The output format of the client is not ensured for forward compatibility. Clients of different versions are different in their command formats and behaviors.
> - For more information about basic commands of the client, see *Basic commands*.
> - Download *MaxCompute console*
> - The client supports JDK 1.9 from the 0.28.0 version, and the previous version can only use JDK 1.8.
> - The client supports MaxCompute 2.0 from the 0.27.0 version *New data type*.

After the client is installed and configured, you can use a command line to perform the following operations.

Get Help

To view the help information of the console, the command format is as follows:

```
odps @>./bin/odpscmd-h;
```

You can also input `h;` or `help;`(case-insensitive) in an interactive mode.

The console also provides the `help [keyword];` command to get the command prompts related to the keyword. For example, input `help table;` to get command prompts related to the table operation as follows:

```
odps @ odps> help table;
Usage: alter table merge smallfiles
Usage: show tables [in]
      a list of tables | ls [-p,-project]
Usage: describe deserves mention | [.] [partition ()]
Usage: read [.]  [partition ()] [line_num]
```

Start parameters

When start the console, you can specify a series of parameters as follows:

```
Usage: odpscmd [option]...
```

```
where options include:
    --help (-h) for help
    --project= use project
    --endpoint= set endpoint
    -u-p user name and password
    -k will begining queries and start from specified position
    -r set retry times
    -f <"file_path;"> execute command in file
    -e <"command; [command;]..."> execute command, include sql command
    -C will display job counters
```

Take the -f parameter as an example, the operation is as follows:

1. Prepare the local script file script.txt. Suppose that the file is located in the disk D, and the

content is shown as follows:

```
DROP TABLE IF EXISTS test_table_mj;
CREATE TABLE test_table_mj (id string, name string);
DROP TABLE test_table_mj;
```

2. Run the following command:

```
odpscmd\bin>odpscmd -f D:/script.txt;
```

Interactive mode

Run the console to directly enter the interactive mode:

```
[admin: ~]$odpscmd
Aliyun ODPS Command Line Tool
Version 1.0
@Copyright 2012 Alibaba Cloud Computing Co., Ltd. All rights reserved.
odps@ odps> INSERT OVERWRITE TABLE DUAL SELECT * FROM DUAL;
```

Enter the command at the cursor position (use a semicolon as a statement terminator), and press

Enter to run.

Continuous running

- When using -e or -f option to run a command, if there are multiple statements, and you want to

   start running from a middle statement, you can specify the parameter -k, indicating to ignore

   the previous statements and to start running from the specified position. When the parameter

   <= 0 is specified, the execution starts from the first statement.

- Each statement separated by a semicolon is considered as a valid statement. The statements

   which run successfully or fail to run are printed out at runtime.

For example,

suppose there are three SQL statements in the file /tmp/dual.sql:

```
drop table dual;
create table dual (dummy string);
insert overwrite table dual select count(*) from dual;
```

To ignore the first two statements, and start running from the third statement, the command format
 is as follows:

```
odpscmd -k 3 -f dual.sql
```

Get current logon user

To get current logon user, the command format is as follows:

```
whoami;
```

Use example:

```
odps@ hiveut>whoami;
Name: odpstest@aliyun.com
End_Point: http://service.odps.aliyun.com/api
Project: lijunsecuritytest
```

Use the preceding command to get the current logon user Alibaba Cloud account, endpoint

configuration, and project name.

Exit

To exit the console, the command format is as follows:

```
odps@ > quit;
```

You can also use the following command to exit the console:

```
odps@ > q;
```

2 MaxCompute Studio

2.1 What is Studio

MaxCompute Studio is a big data integrated development environment (IDE) tool that is provided by the Alibaba Cloud MaxCompute platform and installed on the developer's client. It is a development plug-in based on the popular integrated development platform *IntelliJ IDEA*, helping users develop data conveniently. This article describes functional interfaces and common application scenarios of MaxCompute Studio.

Basic user interface

MaxCompute Studio is a plug-in on the IntelliJ IDEA platform, which shares basic development interfaces with IntelliJ IDEA. For more information about the IntelliJ IDEA interfaces, see *the Interface operation guide*.

Based on the IntelliJ IDEA interfaces, MaxCompute Studio provides the following functional interfaces.

- **SQL Editor**: Provides features such as SQL syntax highlighting, code complementing, real-time error prompting, local compilation, and job submission.

  Compiler View: Displays locally compiled prompts and error messages, and locates the code in the editor.

- **Project Explorer**: Connects to a MaxCompute project, and browses table structures, custom functions, and resource files in the project.

  Table Details View: Displays details and sample data of tables, views, and other resources.

- **Job Explorer**: Browses and searches for historical jobs of MaxCompute.

  - Job Details View: Displays running details of a job, including the execution plan and details of each execution task.

  - Job Output View: Displays output information of a running job.

  - Job Result View: Displays the output result of the SELECT job.

- **MaxCompute Console**: Integrates the *MaxCompute client*, on which MaxCompute client commands can be input and executed.

Connect to MaxCompute project

Before using most features of MaxCompute Studio, you must *Create a project connection*. After the project connection is created, you can view related data structures and resource information in

the **Project Explorer**. MaxCompute Studio automatically creates a local metadata backup task for each project to increase the access frequency to MaxCompute metadata and reduce the latency.

> **Note:**
>
> - You must specify the target project connection to modify SQL scripts, submit jobs, view job information, open the MaxCompute console, and implement other functions using MaxCompute Studio. Therefore, creating a connection to the MaxCompute project is necessary.
> - For more information about MaxCompute projects, see *Project*.
> - For more information about project management using MaxCompute Studio, see *Project space connection management*.

Manage data

You can use the **Project Explorer** of MaxCompute Studio to quickly browse table structures, custom functions, and resource files in the project. The tree control can be used to list data tables, columns, partition columns, virtual views, custom functions, function signatures, and resource files and types of all project connections. It also supports fast locating.

You can double-click a data table to open the **Table Details View** and view metadata, structure, and sample data of the data table. If you do not have the permission for a project, an error message is prompted.

MaxCompute Studio integrates *MaxCompute Tunnel* and supports local data upload and download. For more information, see *Import and export data*.

Write SQL scripts

You can easily compile a MaxCompute SQL script on MaxCompute Studio.

1. Open MaxCompute Studio and select **File** > **New** > **Project** or **File** > **New** > **Module…**.
2. Create a MaxCompute Studio project or module.
3. Select **File** > **New** > **MaxCompute Script** or right-click the menu and select **New** > **MaxCompute Script** , to create a maxcompute SQL script file.

   > **Note:**
   >
   > When a MaxCompute SQL script is created, MaxCompute Studio prompts you to select an associated MaxCompute project. You can also modify the associated project using the **project selector** on the right of the toolbar on the SQL editor. The editor automatically checks

metadata (such as the table structure) and reports errors of an SQL statement based on the project associated with the SQL script. The editor also sends the SQL statement to the associated project for execution when it submits the SQL statement for running. For more information, see *Compile an SQL script*.

SQL code intelligent prompt

After you enter the code, the SQL editor provided by MaxCompute Studio intelligently prompts the syntax errors, type matching errors, or warnings of SQL statements, and marks them on the code in real time. as shown in the following figure.
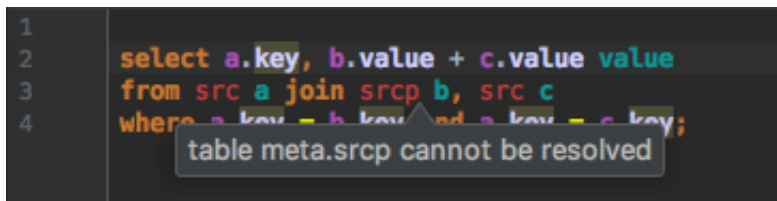
By using the code complementing function, MaxCompute Studio prompts you the name, table, field, function, type, and code keyword of a project based on the code context, and automatically complements the code based on your selections. as shown in the following figure.
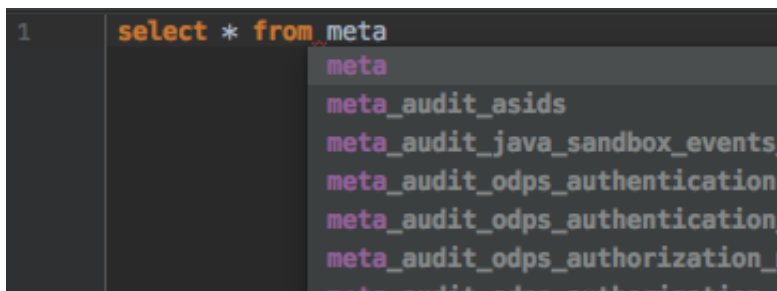
Compile and submit a job

- Compile a job

  Click the icon on the toolbar of the SQL editor to locally compile an SQL script. If syntax

  or semantic errors occur, the editor reports it.
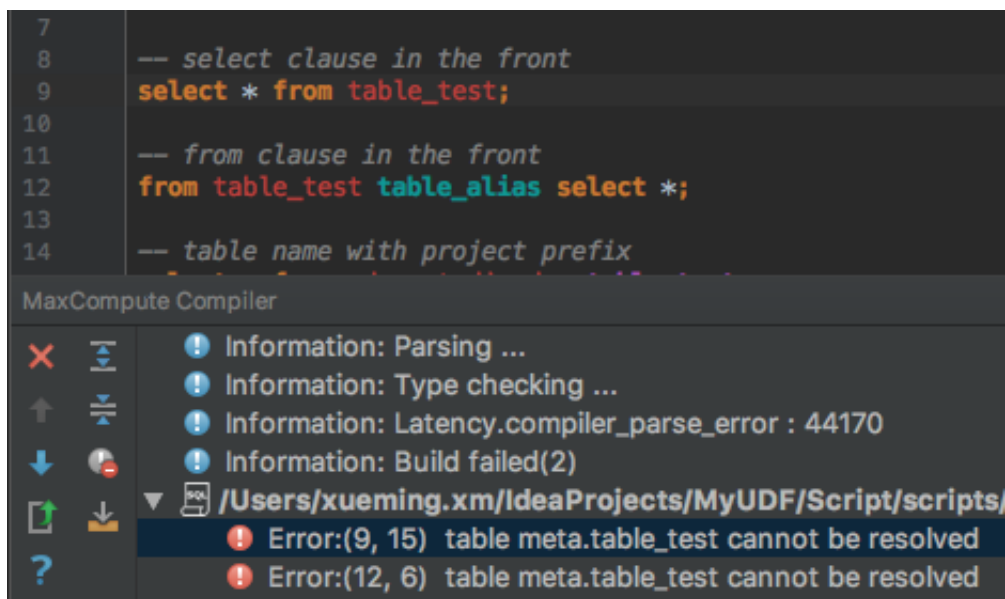
- Submit a job

  Click the icon on the toolbar of the SQL editor to submit an SQL script to the queue of

  the project specified by MaxCompute.

View history jobs

Open **Job Explorer** to view recently executed jobs in the specified project.

> **Note:**
> List only displays jobs submitted by the user ID of the current connection.

Double-click a job to view the job details. as shown in the following figure.

If you have the Log view URL of a job, you can select **MaxCompute** > **Open Logview** from the

menu to go to the details page of the job.

Develop a MapReduce program and UDF program

MaxCompute Studio also allows you to develop *MapReduce* and *Java UDF* programs.

Connect to a MaxCompute client

MaxCompute Studio is integrated with the *MaxCompute Client* of the latest version. Alternatively, you can specify the path of the locally installed MaxCompute client on the *Configuration page* of MaxCompute Studio.

On the **Project Explorer**, right-click a project and select **Open in Console** to open the **MaxCompute Console** window.
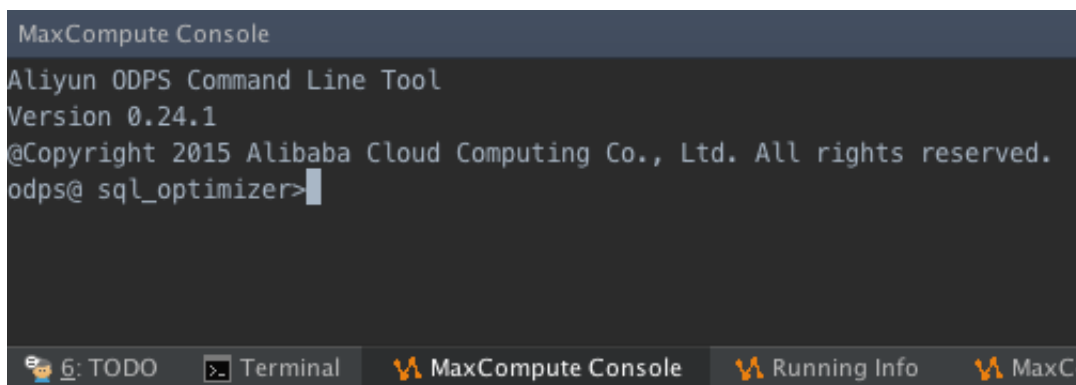
Next step

Now, you know the functional interfaces and common application scenarios of MaxCompute Studio. Continue to the next tutorial. In this tutorial, you will learn how to install MaxCompute Studio. For more information, see *Install IntelliJ IDEA*.

2.2 Tools Installation and version history

2.2.1 Install IntelliJ IDEA

This document describes how to install the basic platform IntelliJ IDEA of MaxCompute Studio.

Context

Procedure

Procedure

1. Click *here* to download the IntelliJ IDEA of the version corresponding to your operating system (Windows, macOS, or   Linux). The following assumes that the Windows operating system is used.

Download IntelliJ IDEA 14.1.4 or a later version. (The Ultimate version, PyCharm version, and free Community version are supported.)

2. After the download is complete, double-click the installation program to enter the installation page, and click **Next**, as shown in the following figure.

3. Specify the installation directory, and click **Next**, as shown in the following figure.
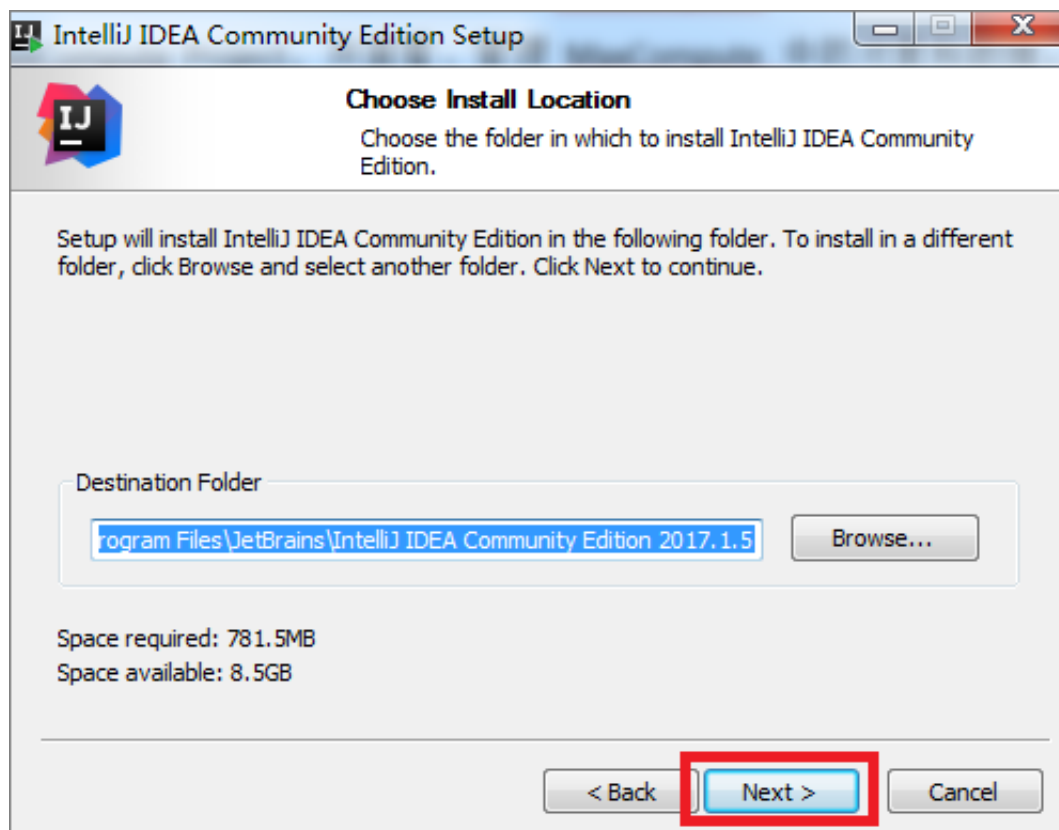
4. Select the 32-bit or 64-bit IntelliJ IDEA based on the version of the local operating system.

   **You can query the local operating system version by following these steps:**

   a) Open Windows Resource Manager, right-click **Computer** and select **Properties** from the shortcut menu. as shown in the following figure.

   b) In the displayed window, check the type of the operating system.

5. Select the corresponding system type and click **Next** , as shown in the following figure.
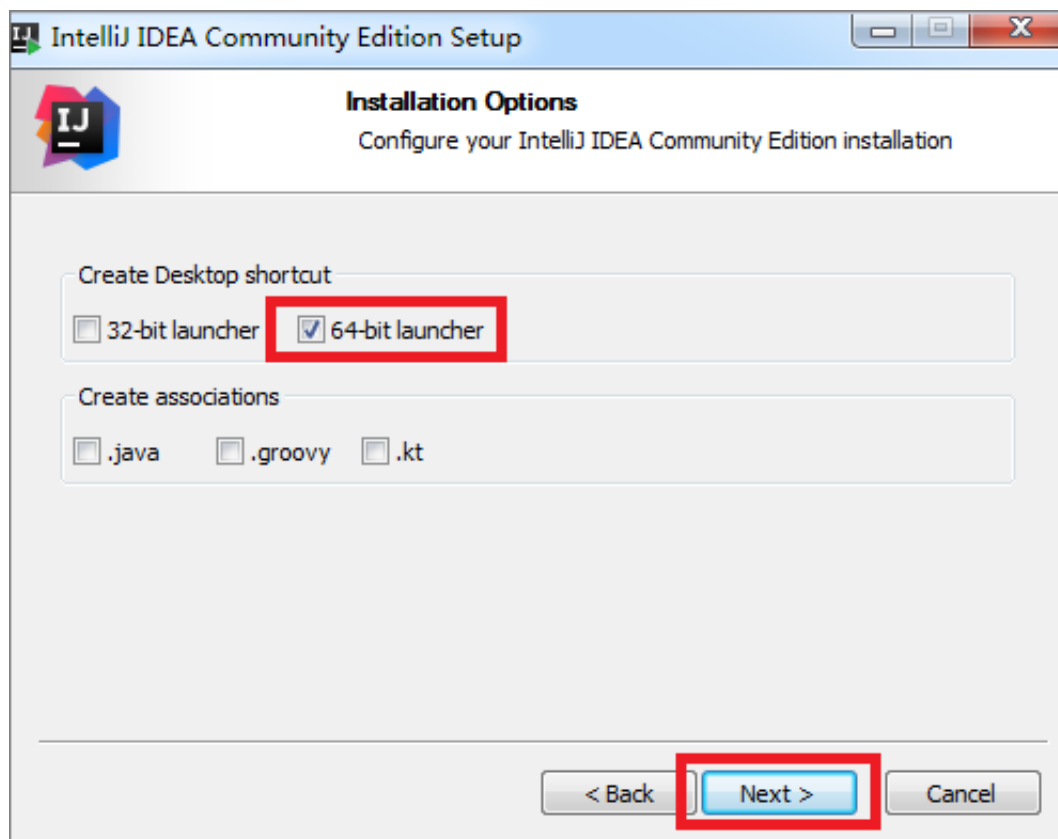
6. Click **Install** to start installation, as shown in the following figure.
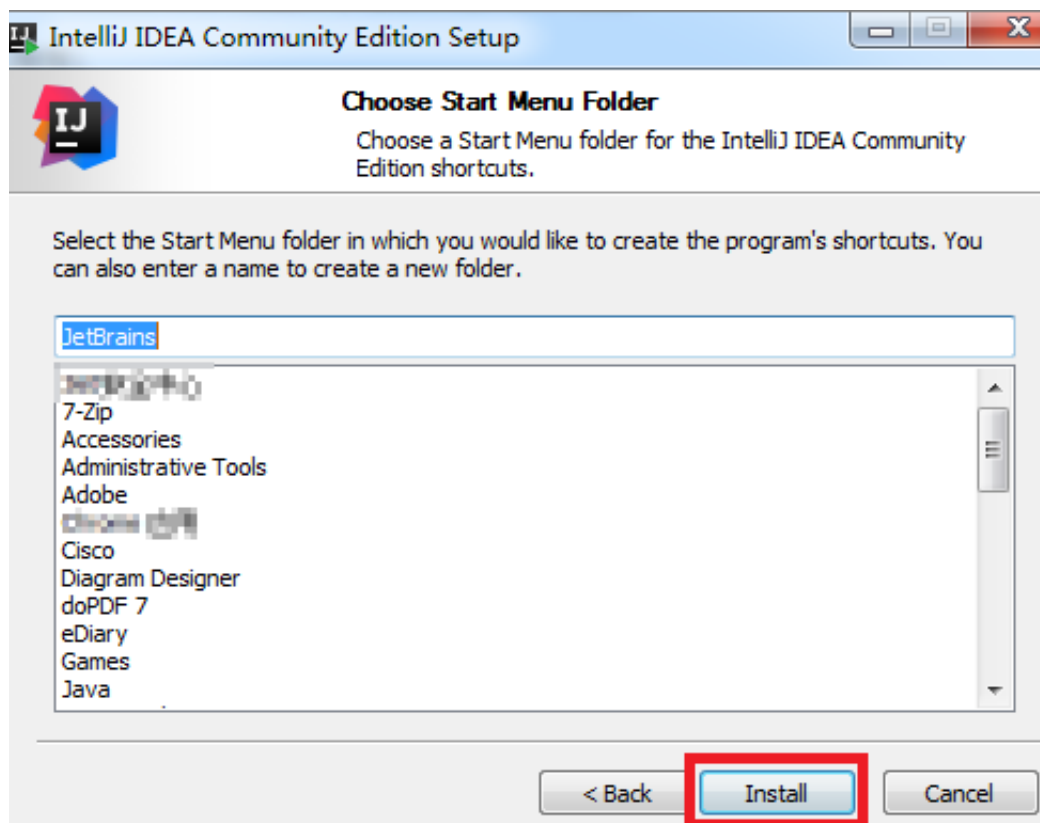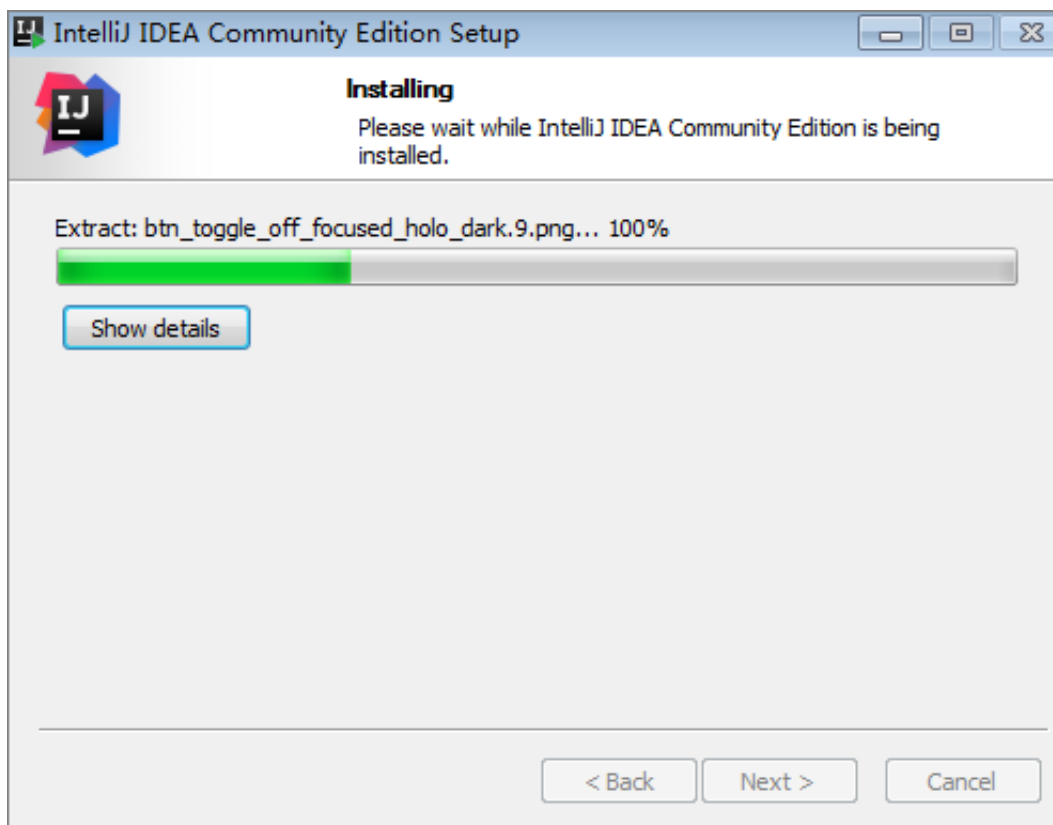
7. After the installation is complete, click **Finish**.

What's next

Now you know how to install IntelliJ IDEA. Continue to the next tutorial. In this tutorial, you will learn how to install the MaxCompute Studio plugin. For more information, see *Install the MaxCompute Studio plugin*.

2.2.2 Installation procedure

Environment requirements

IntelliJ IDEA can be installed on *Windows*, *Mac*, *Linux*. For more information about the hardware and system environment requirements, click *here* . IntelliJ IDEA-based MaxCompute Studio can also be installed on clients running these operating systems.

MaxCompute Studio has the following requirements on the your environment:

* A client running Windows, macOS, or Linux.

* IntelliJ IDEA 14.1.4 or a later version is installed. (The Ultimate version, PyCharm version, and free *Community version* are supported.)

* JRE 1.8 is installed. (JRE 1.8 has been bound to the latest IntelliJ IDEA.)

* JDK 1.8 is installed. (*Optional*: JDK is required if you need to develop and debug Java UDF.)

> **Note:**
>
> The client supports JDK 1.9 from the 0.28.0 version. The previous version only supports JDK 1.8.

Installation method

MaxCompute Studio is a plugin of IntelliJ IDEA, which can be installed using either of the following two methods:

* Online installation using the plugin library (recommended)

* Installation using a local file

Online installation (recommended)

MaxCompute Studio The MaxCompute Studio plugin has been opened for all users on the Internet . You can install MaxCompute Studio using the official IntelliJ IDEA plugin library.

Procedure

1. Open the plugin configuration page on IntelliJ IDEA. (If you are a Windows/Linux user, choose **File** > **Settings** > **> Plugins.** If you are a macOS user, **choose IntelliJ IDEA** > **Preferences** > **Plugins**).

2. Click **Browse repositories…** and search for `MaxCompute  Studio`.

3. On the MaxCompute Studio plugin page, click **Install**.

4. After the installation is confirmed, restart IntelliJ IDEA to complete installation.
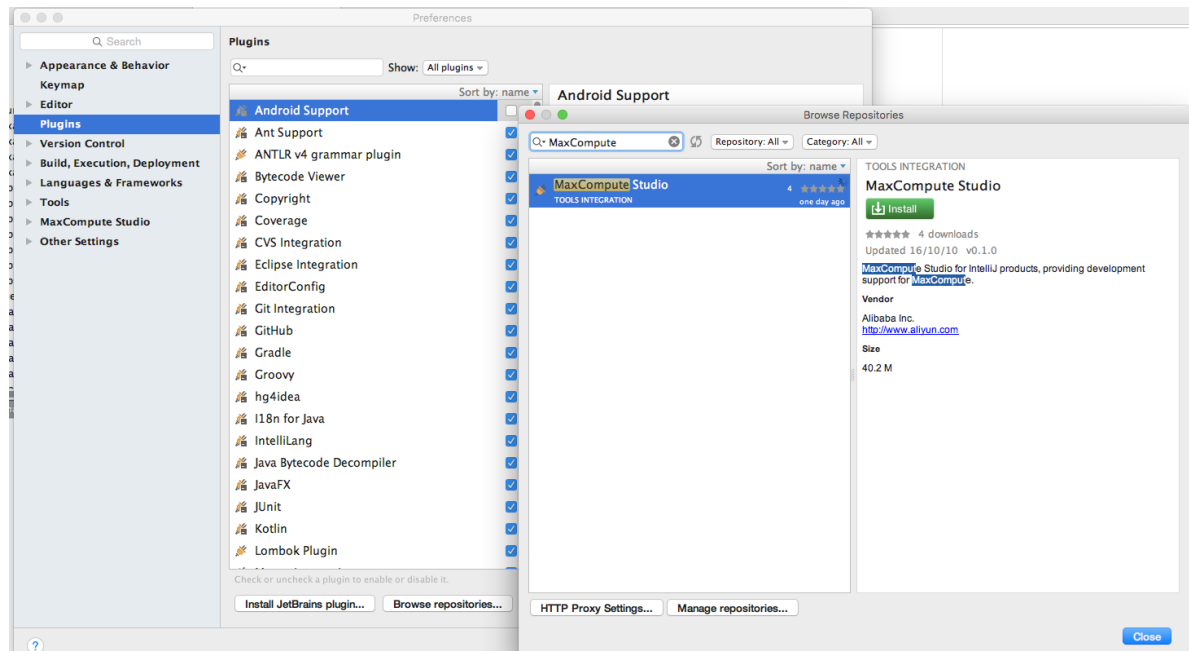

Local installation

MaxCompute Studio MaxCompute Studio can also be installed in a local environment.

Procedure

1. Go to the *MaxCompute Studio plugin page* to download the plugin package.

2. Run IntelliJ IDEA.

- If you access IntelliJ IDEA for the first time, a welcome page is displayed. Click **Configure** and select **Plugins** from the shortcut menu, as shown in the following figure.
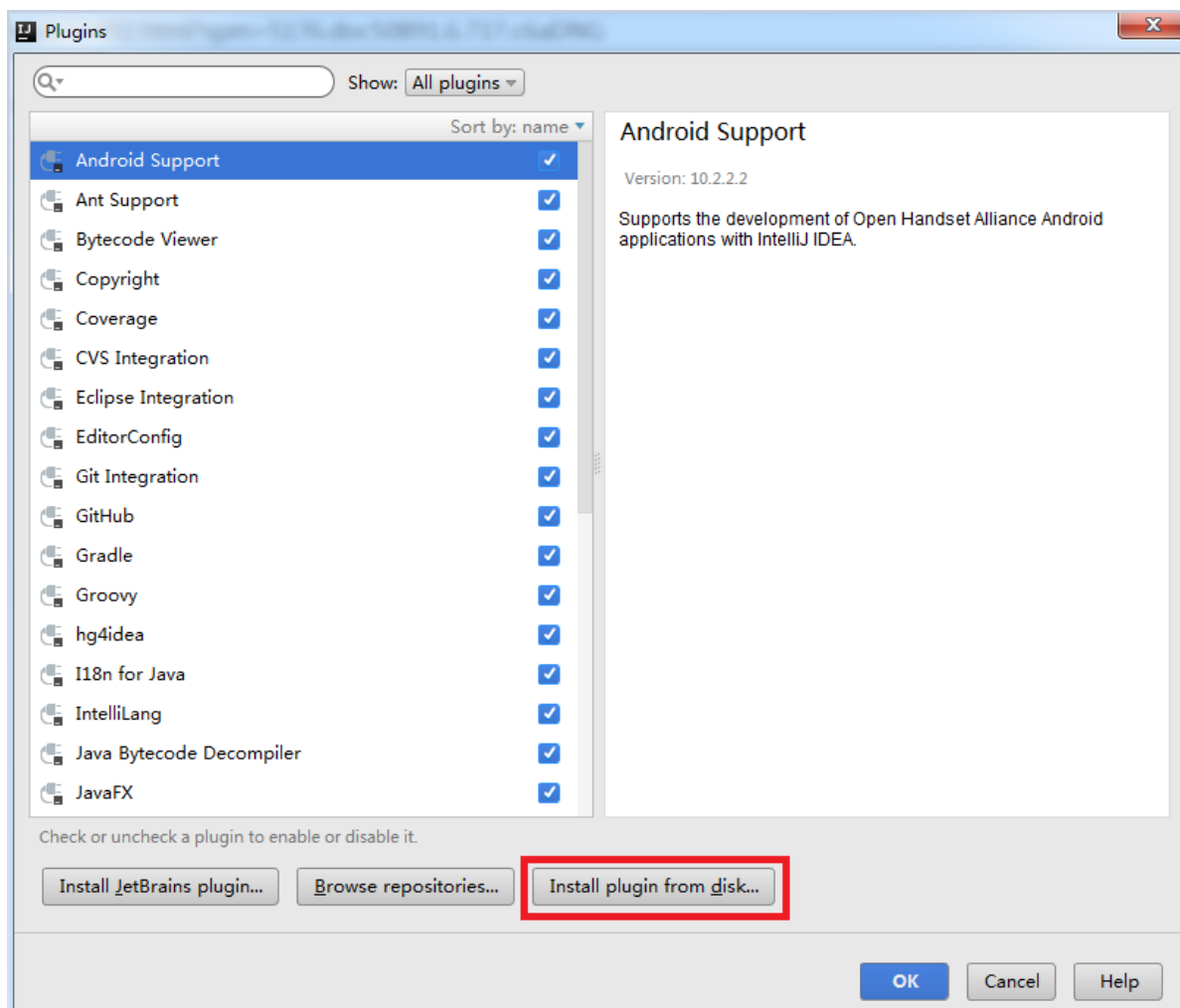
- If you have accessed IntelliJ IDEA before, choose **File** > **Settings** > **Plugins to** enter the same page, as shown in the following figure.
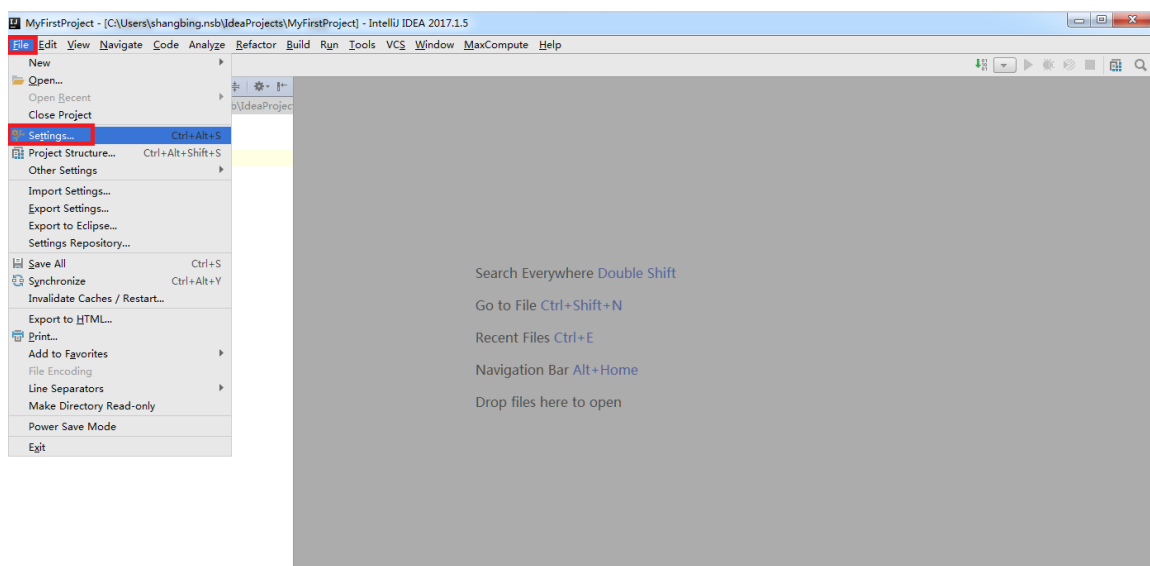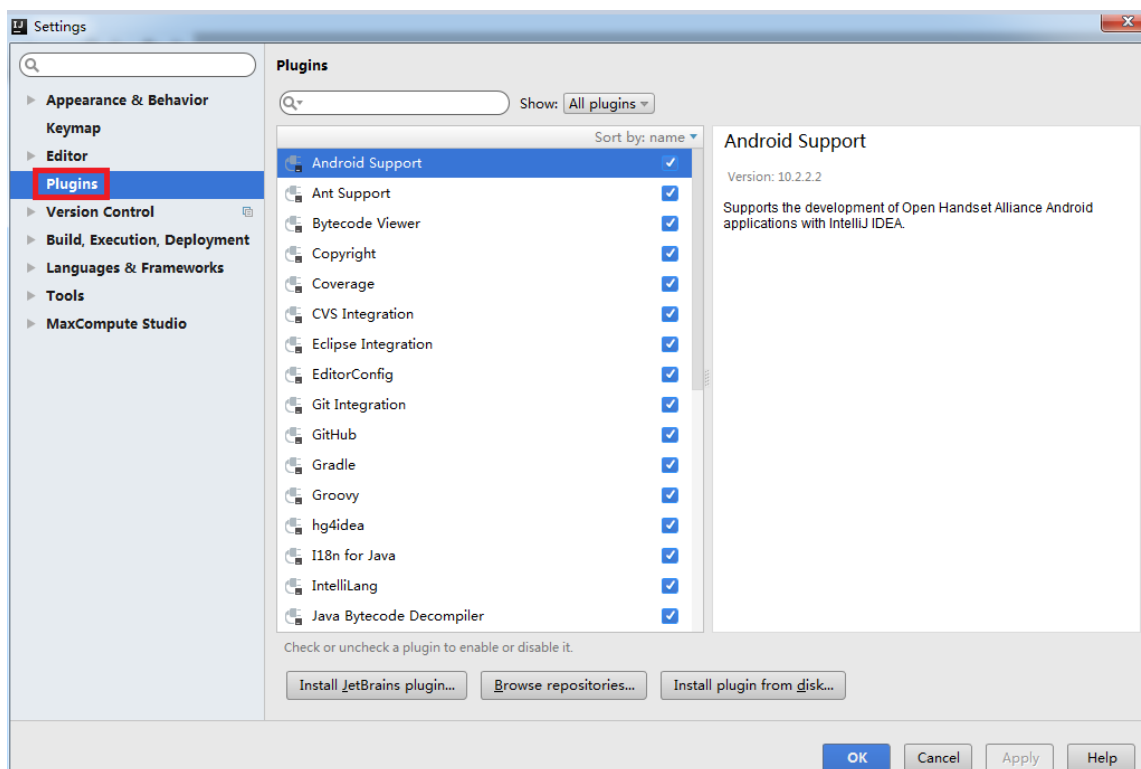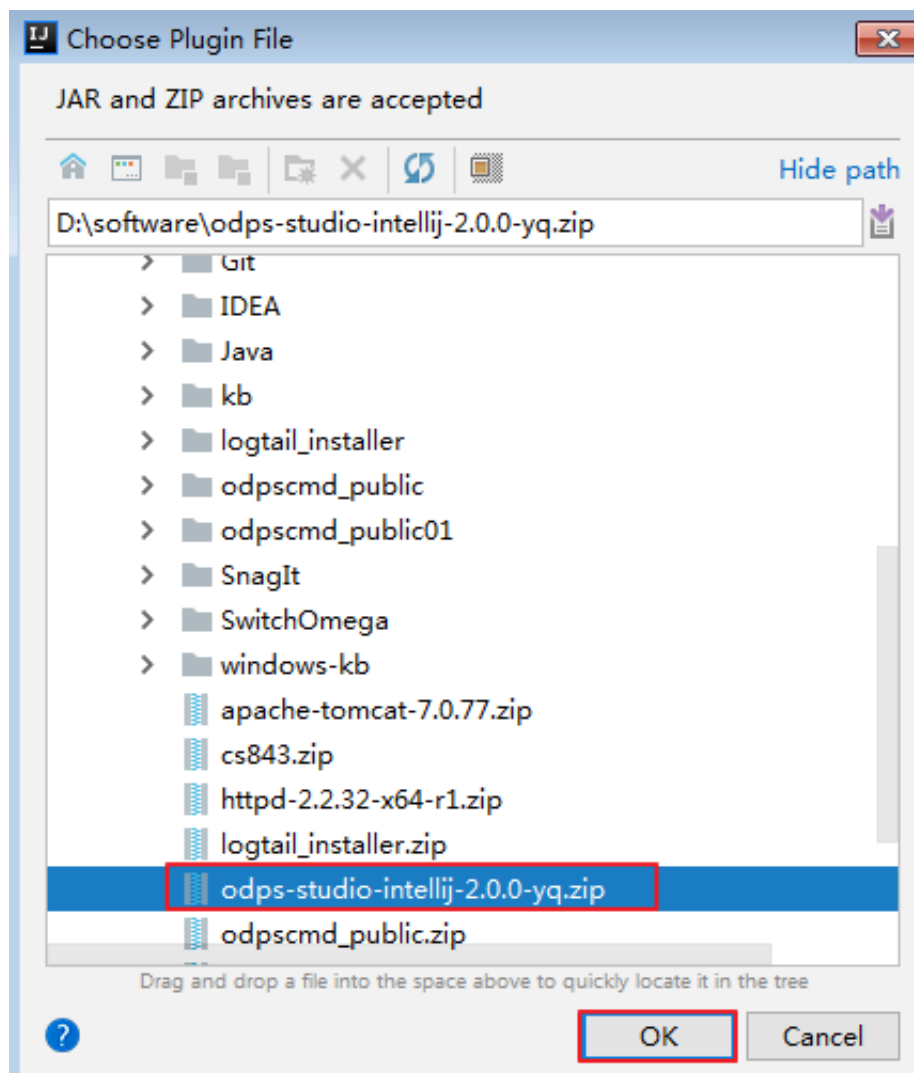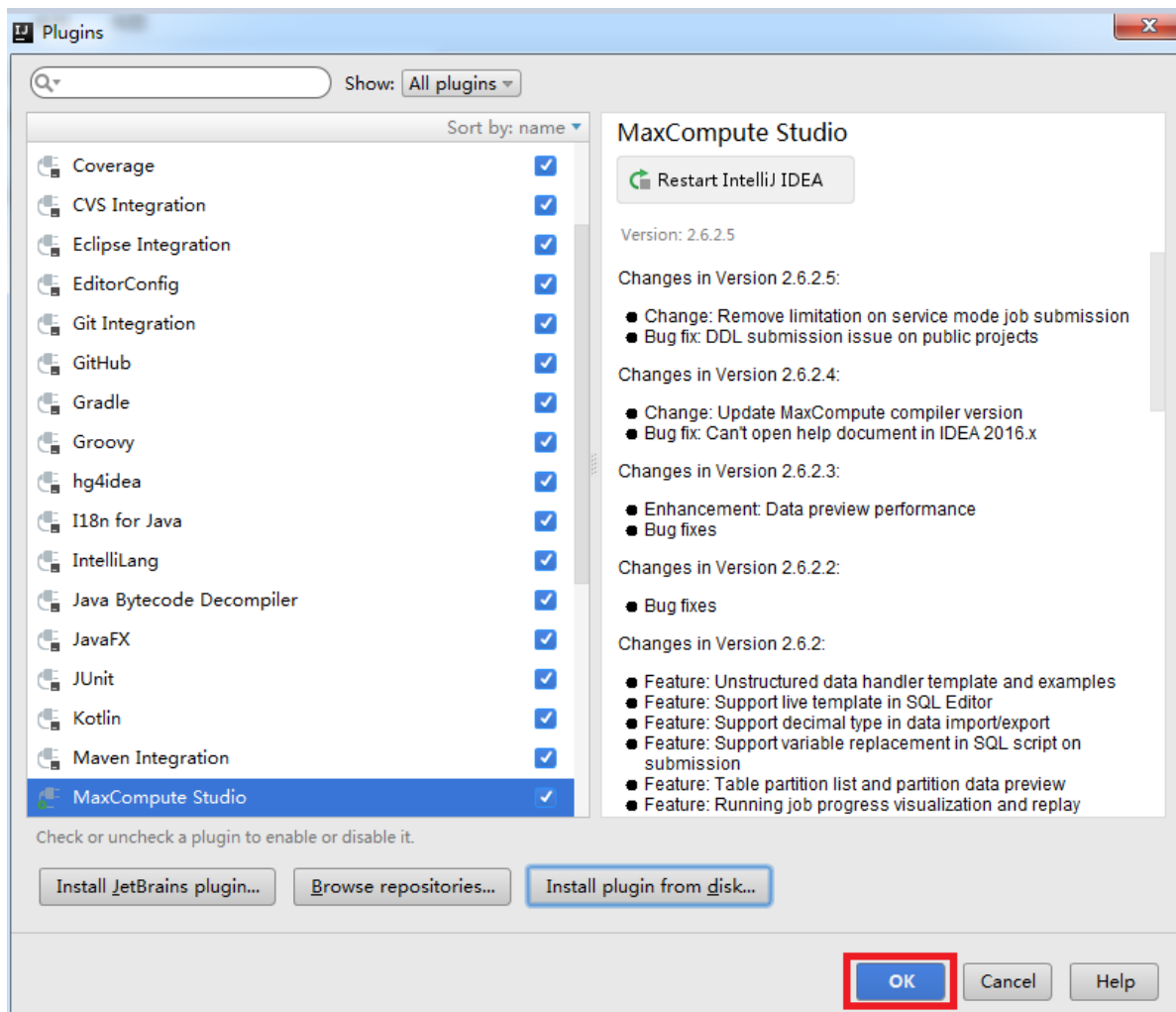
3. On the Plugins page, click **Install plugin from disk…**, as shown in the following figure.
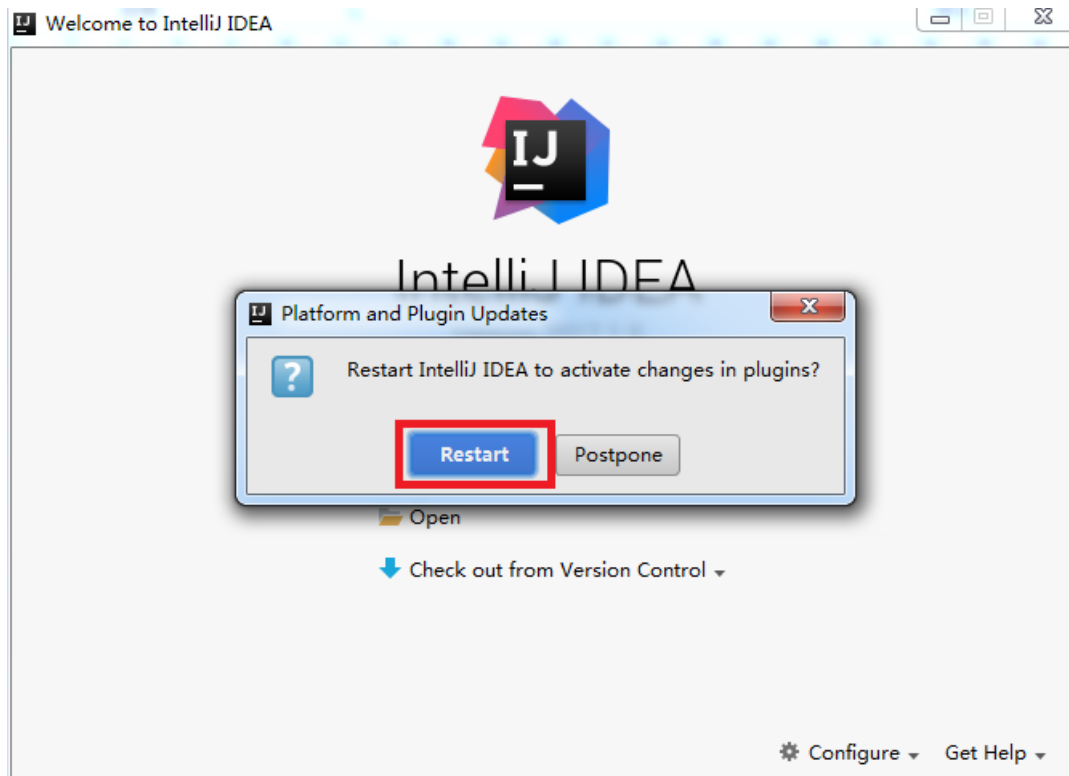
4. In the displayed window, click the gray icon before a directory for navigation, find the plugin file, select it, and click **OK**.
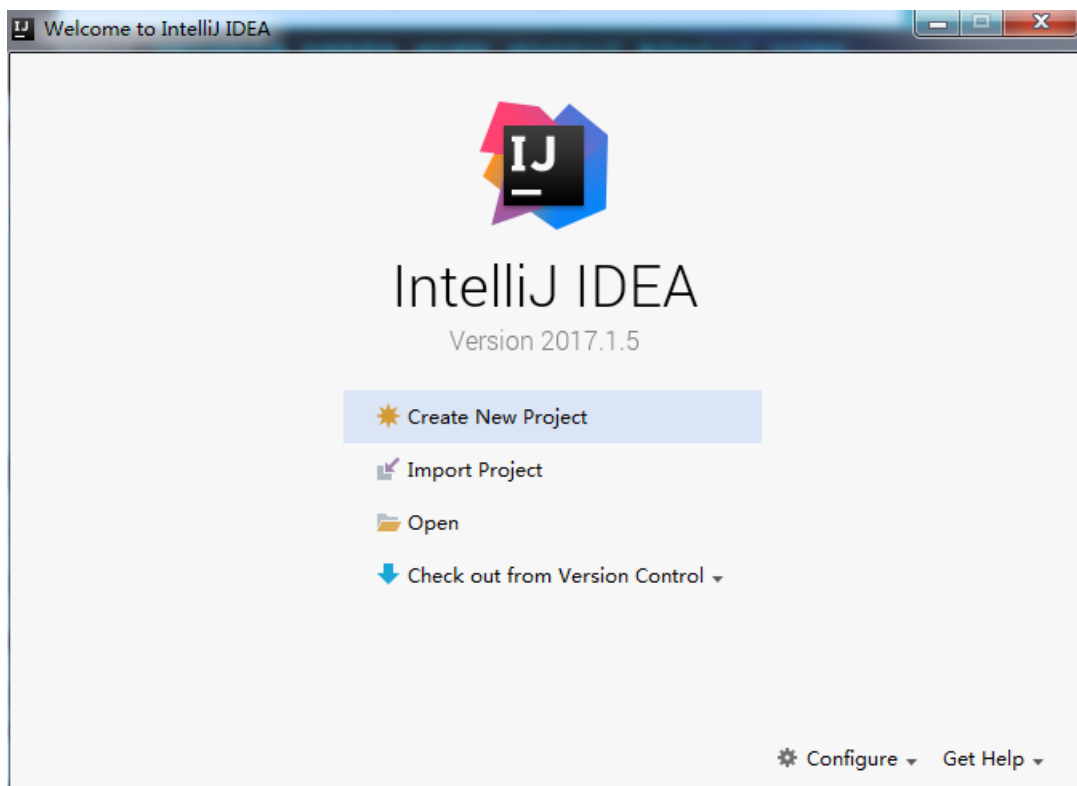
5. Return to the Plugins page and click **OK** to install the local plugin.

6. After the installation is complete, a dialog box is displayed, prompting you to restart IntelliJ
   IDEA. Click **Restart**.

7. After IntelliJ IDEA is restarted, the page is displayed as shown in the following figure.

Next step

Now, you know how to install the MaxCompute Studio plugin. Continue to the next tutorial. In the tutorial, you will learn how to configure a MaxCompute project connection to manage data and resources. For more information, see *Create a MaxCompute project connection*.

2.2.3 View and upgrade the version

View the MaxCompute Studio version

Perform the following steps to view the Studio version:

1. Go to the **Settings/Preferences** page (by pressing `Ctrl-Alt-S` in Windows or ⌘ in macOS).

2. Select **Plugins** on the left bar of the dialog box and search for *MaxCompute Studio*.

3. View the MaxCompute Studio version number and release information.

Alternatively, you can select **MaxCompute Studio** on the left bar of the **Settings** page to view the current version number.

Check new versions

By default, MaxCompute Studio automatically detects new versions. If a new version is available, MaxCompute Studio automatically notifies you.

After receiving an update prompt, you can select:

*   **Install**: Click the Install link in the update prompt. The new version is automatically downloaded and installed. After the installation is complete, restart IntelliJ IDEA.

*   **Configure**: Click the Configure link in the update prompt. You can configure whether to detect new versions automatically.

If the automatic update function is disabled, you can perform the following steps to check and install a new version of MaxCompute Studio.

1. Go to the **Settings/Preferences** page (by pressing `Ctrl-Alt-S` in Windows or ⌘ in macOS).

2. Select **MaxCompute Studio** on the left bar of the dialog box.

3. On the MaxCompute Studio configuration page, click **Check new versions**.

4. If a new available version is detected, Studio notifies you of the new version number. Click **Install new version** and restart IntelliJ IDEA to complete installation.

You can use the **Automatically checks for new version** check box to control the switch for automatic version update check.

Next step

Create a MaxCompute project connection

2.3 Manage data and resources

2.3.1 View tables and UDF

View tables and functionsView tables and functions

In the **Project Explorer window**, you can view tables, functions, and resources with connections added. For tables and functions to be viewed in the Project Explorer window, the MaxCompute project connections must be added, for more information, see *Add MaxCompute project connections*.

Browse tables and functions

To browse tables and functions in the project space, follow these steps.

1. Open the **Project Explorer window** and you can view the added Project node tree.
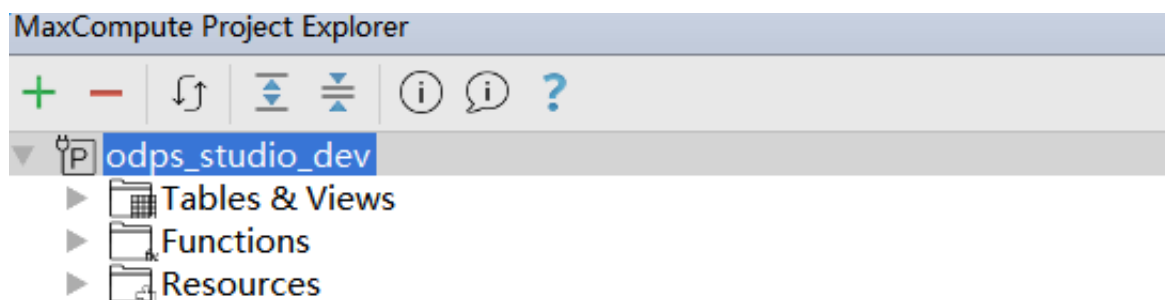
The toolbar is displayed at the top of the node tree, and includes:

- **Add Project**: Adds a connection to the MaxCompute project space.

- **Delete Project**: Deletes a connection from Project Explorer, which has no impact on the project space on the server end.

- **Update Metadata**: Updates metadata information from the project space on the server end and updates the locally buffered metadata.

- **Expand Node**: Expands all tree nodes.

- **Fold Node**: Folds all tree nodes.

- **User Feedback**: Submits user feedback.

- **Online Documentation**: Opens online documents.

2. Double-click the **Tables** node or click the drop-down arrow to expand the Tables node to list all tables in the project (including virtual views). The table name list serves the same purpose

as the show tables command. You must have the List Table permission in the project. The methods for the **Functions** and **Resources** nodes are similar to that of the Tables node.
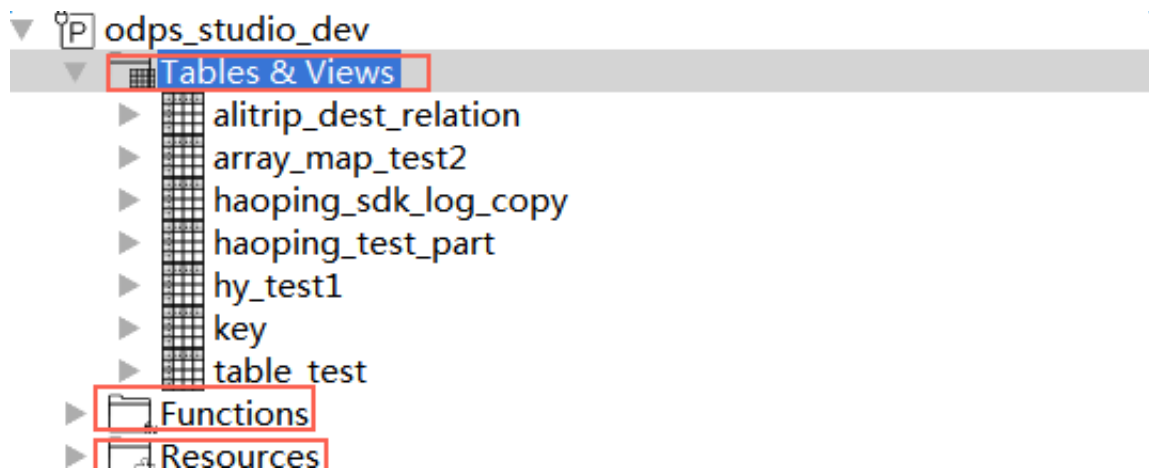
3. MaxCompute Studio downloads project metadata on the server to the local device. When metadata on the server end is updated, for example, a new table is added, you must manually trigger a refresh to reload changed metadata to the local device. The refresh can be performed at the Project or Table level. The procedure is as follows:
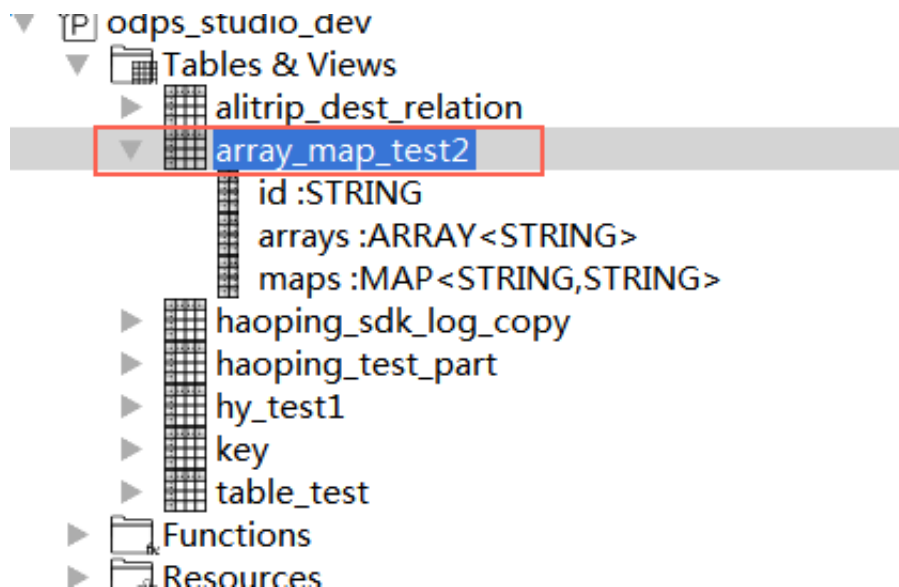
a. Select a node.

b. Click the Refresh icon on the toolbar or right-click the node and select Refresh meta.
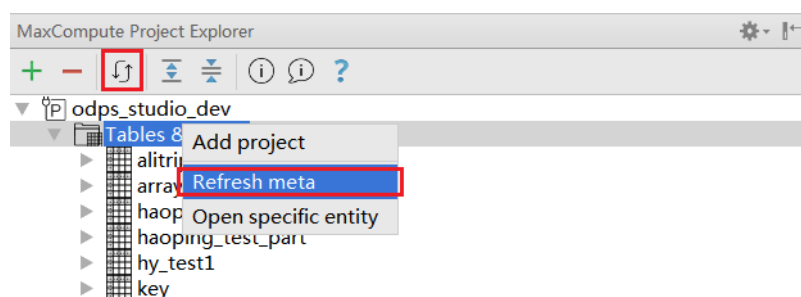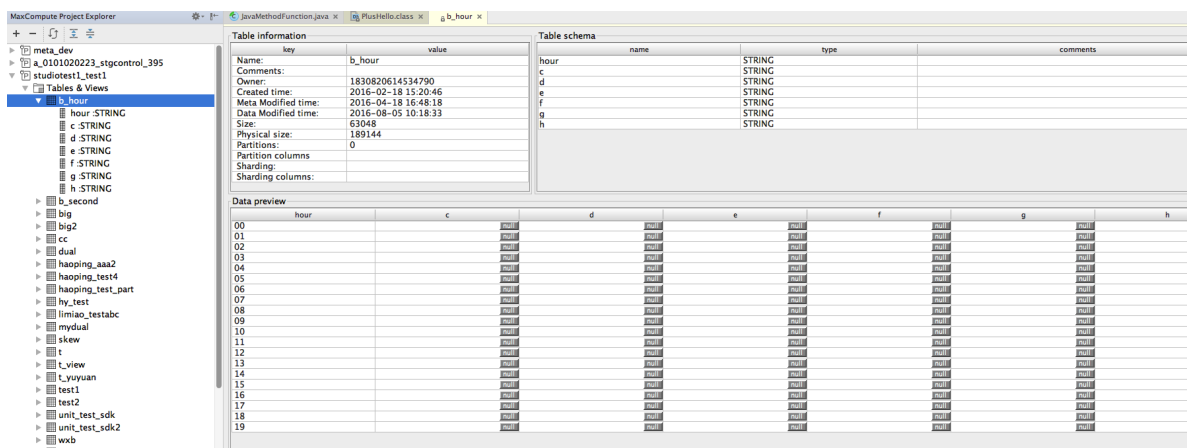

View table details

You can view data table information in **Table Details View** of MaxCompute Studio.

1. In the node tree, expand a table node to view the column name and type.

2. Double-click a table or right-click a table and choose **Show Table Detail** to view the table

   details. The table details include metadata, such as owner, size, and column, table structure

   information, and data preview.

3. Right-click **Tables & Views** and select **Open specific entity** to display the details of the

   specific table. Note that the complete table name must be specified. If you do not have the List

   permission on the project and only have the permission on a specific table, you can also view

   details of the table using this method. The methods for the Functions and Resources nodes

   are similar to that of the Tables node.
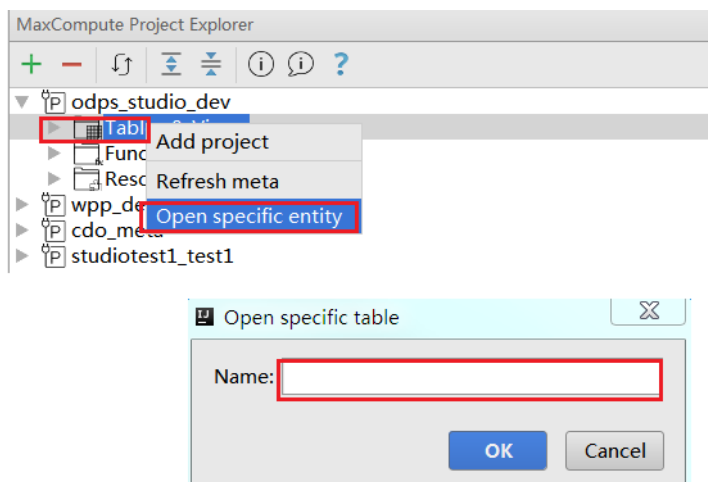
Intellij IDEA supports searching by default. After a table is expanded, you can directly press keys on the keyboard to perform fuzzy match.

4. MaxCompute Studio also supports quick search for the table, you can use the shortcut key (Windows: Ctrl + Alt + Shift + N, macOS: ⌥ + ⌘ + O) to call the navigation bar, then enter the name of the table and press Enter.
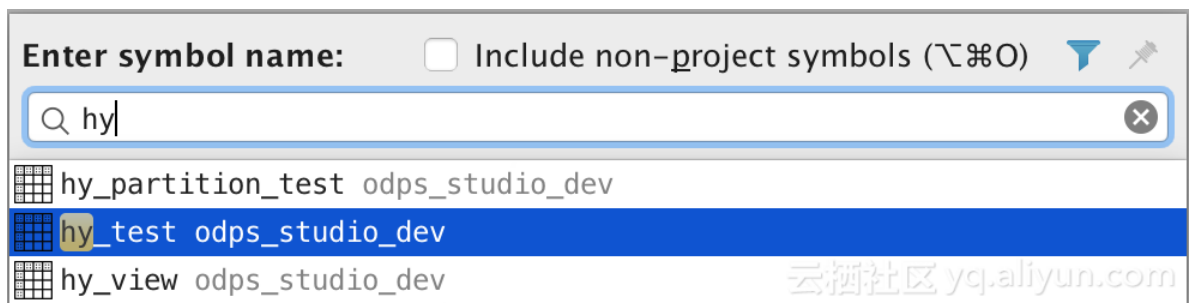

You can narrow the search by using the pre-keyword (table:, function:, or resource:). For example, to search for the function count, enter function:count.

5. To know the scripts in which the table is used, right-click the table and select Find usages.
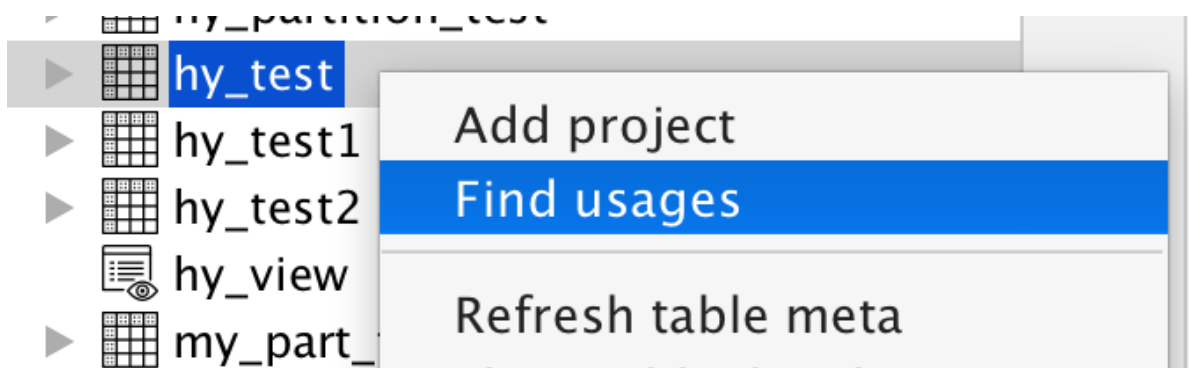
View function details

1. Expand a function node under the UserDefined node of **Functions** to display the method signature of this function. Double-click a function node under the Functions node. Alternatively, double-click the source code resource of the function under the **Resources** node. In this case, codes of this function are displayed.
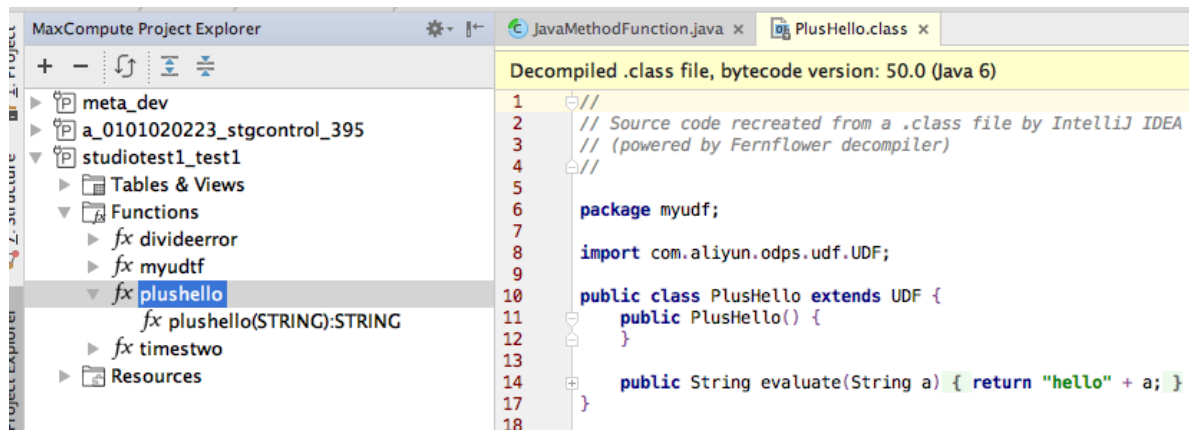
📋 **Note:**

The Java code is obtained by decompiling JAR, which is not the source code. To enable the Python UDF to parse the signature, install PyODPS (MaxCompute Python SDK) first. Install pip: sudo/usr/bin/python get-pip.py (Download get-pip.py from Google manually) and then PyODPS: sudo/usr/bin/python -m pip install PyODPS. Note that the Mac operating system has Python, which is stored in /usr/bin/python. Install PyODPS in this directory.

2. The classification under the BuiltIn node of **Functions** shows the built-in functions of the system, expand it to display signature and double-click it to display function document.
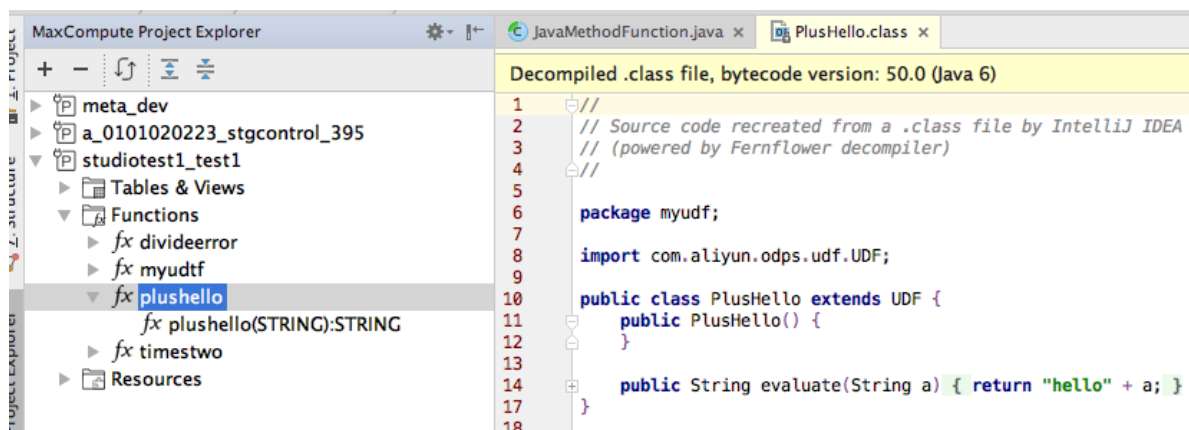
2.3.2 Import and export data

MaxCompute Studio can import local data files in CSV or TSV format to MaxCompute tables and export MaxCompute table data to local files. MaxCompute Studio completes data import and export by using Batch data Tunnel provided by the MaxCompute platform.

Usage instructions

- The MaxCompute Tunnel service must be used for data import and export. Therefore, the MaxCompute project added in Studio must be configured with the Tunnel service.

- Related permissions must be granted for table import and export.

Import data

1. Open the **Project Explorer** window, right-click a table name or a field attribute in Data preview of Table details and select **Import Data Into Table**.
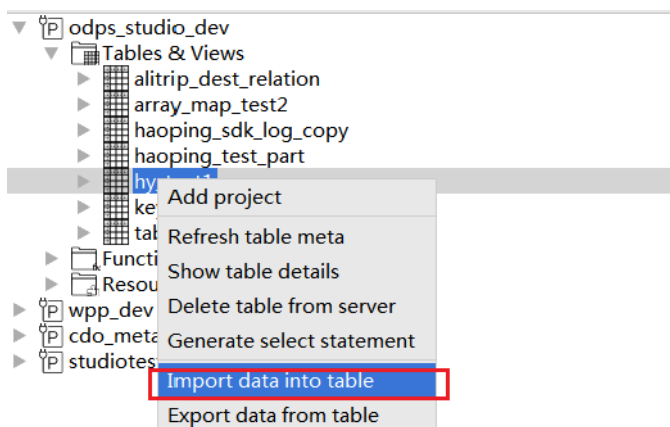
2. In the **Import Data** dialog box that appears, select the path of the imported data file, column separator, size limitation, and number of lines for an error tolerance, and click **OK** .
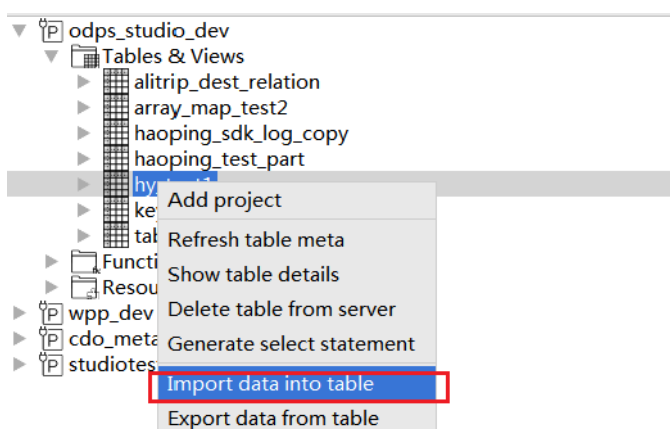
3. If **Import Data Success** is displayed, data import is successful and imported data can be viewed in the table.

Export data

1. Two methods are provided for table data export.

- Right-click a table name and select **Export Data From Table**.

- Right-click a field attribute in **Data Preview** of Table details and select **Export Data From Table**.

2. In the **Export Data** dialog box that appears, select the path for saving the exported data file, column separator, size limitation, and number of lines for an error tolerance, and click OK.
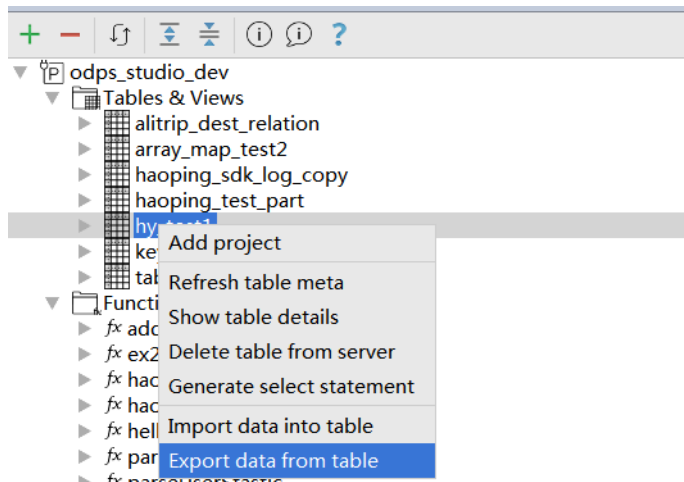
3. If **Export Data Success** is displayed, data export is successful and exported data can be
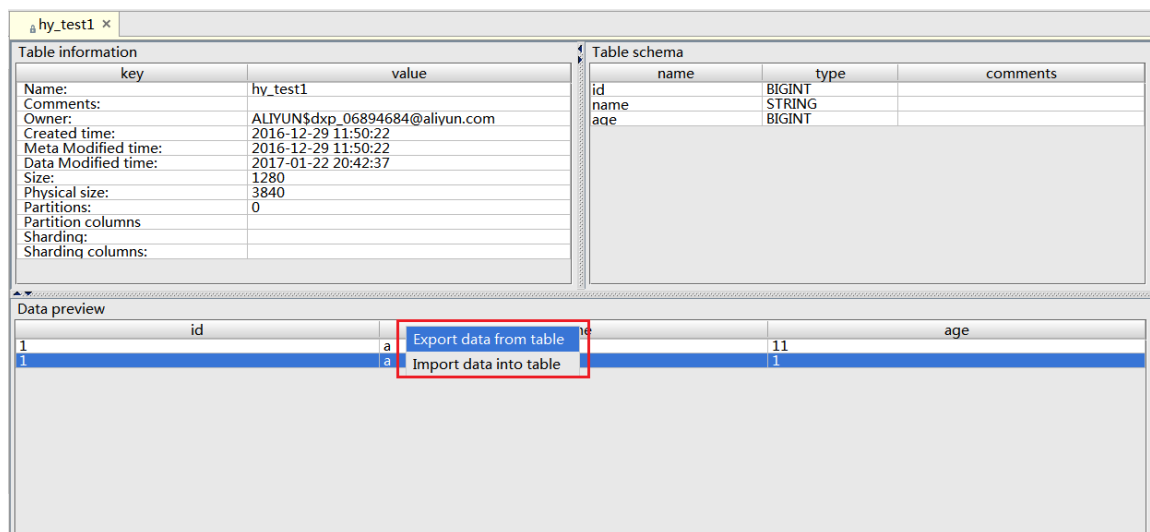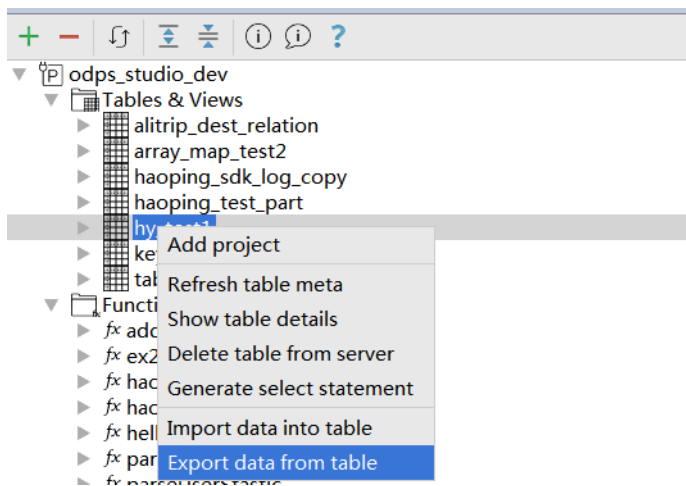
   viewed in the target file.

   You can also right-click **Data Preview** of Table and choose **Export Grid Data** to export data.
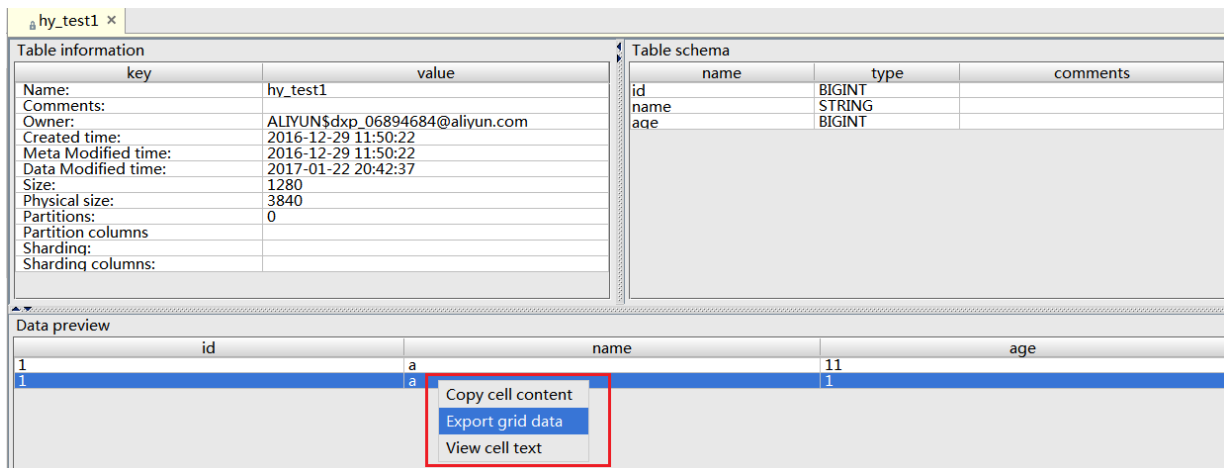
> 📋 **Note:**
>
> The **Data Preview** function in Data preview is used only to export data displayed in Data sample
> instead of all data in the table.

New Type Import Export

Simply generate text in the agreed format and store it in CSV or TSV format, you can import to
table through studio.

Conversion rules for each data type are described in detail below.

Basic Type

1. Tinyint, smallint, Int, bigint is stored directly as an integer string, and the numeric value exceeds

   the type boundary is reported as wrong

2. Float, double to store fractional strings or floating-point forms, such as: 2.342 1x + 7

3. Varchar is stored directly as a string, which is automatically truncated above the upper limit, and no errors will be reported

4. String is stored directly as a string

5. Decimal string that supports shaping or floating-point

6. Binary needs to encode binary data to base64 string

7. Datetime date time requires that the format specified in import dialog is consistent and that the format mismatches will be reported incorrectly

8. Timestamp timestamp needs to follow yyyy-[m] M-[d] d hh \: mm \: ss [. f...] Format is stored as a string

9. Boolean true or false string?

Composite Type

1. Array needs to be stored as a JSON array. The array elements are converted to strings according to the rules agreed in this article, array elements support any type.

2. Map needs to be stored as a JSON object, and map key, value are converted to a string according to the rules agreed in this article, value supports any type of nesting.

3. Struct needs to be stored as a JSON object, the struct field name is string, and the key converted to a JSON object, strect field values converted to JSON The value of the object, the value of the field that defines the rule transformation in this article.

Example

Array type

For Table structures as follows:

Column name	Column data types
c_1	ARRAY<TINYINT>
c_2	ARRAY<INT>
c_3	ARRAY<FLOAT>
c_4	ARRAY<DATETIME>
c_6	ARRAY<TIMESTAMP>
c_7	ARRAY<STRING>

You can import data in the CSV format shown below:

```
c_1,c_2,c_3,c_4,c_6,c_7
"[" "1" "," "2 "", "3" "]", "[" 1 "", "2", "3", "4"], "[" "1.2" "," 2.
0 ""] ", "[" "3-00:00:00", "3-5-00:00:00 "", "00:00:00", "[" At 00:00:
00. 123456789 "", "At 00:00:00. 123456789 "", "At 00:00:00. 123456789
 ""] "," ["AAA" "," Steamboat "", "4C" "]"
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00:00
.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb"",""
ccc""]"
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00:00
.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb"",""
ccc""]"
```

📋 **Note:**

The CSV format needs to escape double quotes, which are expressed by two double quotes, you can refer specifically to the CSV format specification.

Map Type

For Table structures as follows:

Column name	Column data types
c_1	MAP<TINYINT,STRING>
c_2	MAP<STRING,INT>
c_3	MAP<FLOAT,STRING>
c_4	MAP<STRING,DATETIME>
c_5	MAP<STRING,STRING>
c_6	MAP<TIMESTAMP,STRING>

You can import data in the CSV format shown below:

```
c_1,c_2,c_3,c_4,c_5,c_6
"{1:" "2345" "}", "{" 123 "": "2", "3": "4 ""}", "{2.0:" "223445"
", 1.2:" 1111 ""}", "{" "AAA" ":" "hub11 00:00:00 "", "4C" ":" China
 "11 00:00:00" "," Steamboat "": "00:00:00} "," ckey "": "cvalue}
 "," {"" hub11 01:00:00. 123456789 "": "dddd" "," "hub11 00:00:00.
123456789 "": "AAA" "," 027 11 00:01:00. 123456789 "": "DDD ""}"
"{1:" "2345" "}", "{" 123 "": "2", "3": "4 ""}", "{2.0:" "223445"
", 1.2:" 1111 ""}", "{" "AAA" ":" "hub11 00:00:00 "", "4C" ":" China
 "11 00:00:00" "," Steamboat "": "00:00:00} "," ckey "": "cvalue}
 "," {"" hub11 01:00:00. 123456789 "": "dddd" "," "hub11 00:00:00.
123456789 "": "AAA" "," 027 11 00:01:00. 123456789 "": "DDD ""}"
```

```
"{1:""2345""}","{""123"":""2"",""3"":""4""}","{2.0:""223445"",1.2
:""1111""}","{""aaa"":""2017-11-11 00:00:00"",""ccc"":""2017-11-11
 00:00:00"",""bbb"":""2017-11-11 00:00:00""}","{""ckey"":""cvalue
""}","{""2017-11-11 01:00:00.123456789"":""dddd"",""2017-11-11 00:00:
00.123456789"":""aaa"",""2017-11-11 00:01:00.123456789"":""ddd""}"
```

Struct Type

For Table structures as follows:

Column name	Column data types
C_struct	<RUCT<x:INT,y:VARCHAR(256),z:STRUCT<a:TINYINT,b:STRING>>

You can import data in the CSV format shown below:

```
c_struct
"{""x"":""1000"",""y"":""varchar_test"",""z"":{""a"":""123"",""b"":""
stringdemo""}}"
"{""x"":""1000"",""y"":""varchar_test"",""z"":{""a"":""123"",""b"":""
stringdemo""}}"
```
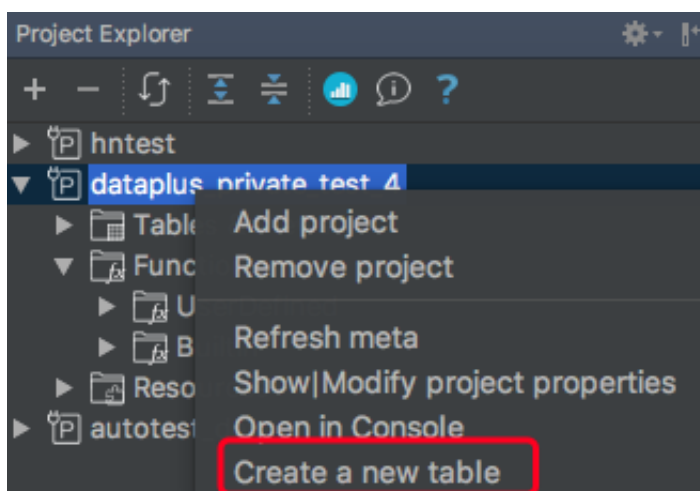
2.3.3 Visualization of operating the tables

The Project Explorer of MaxCompute Studio provides the visualized table structure editor used to create and modify tables.
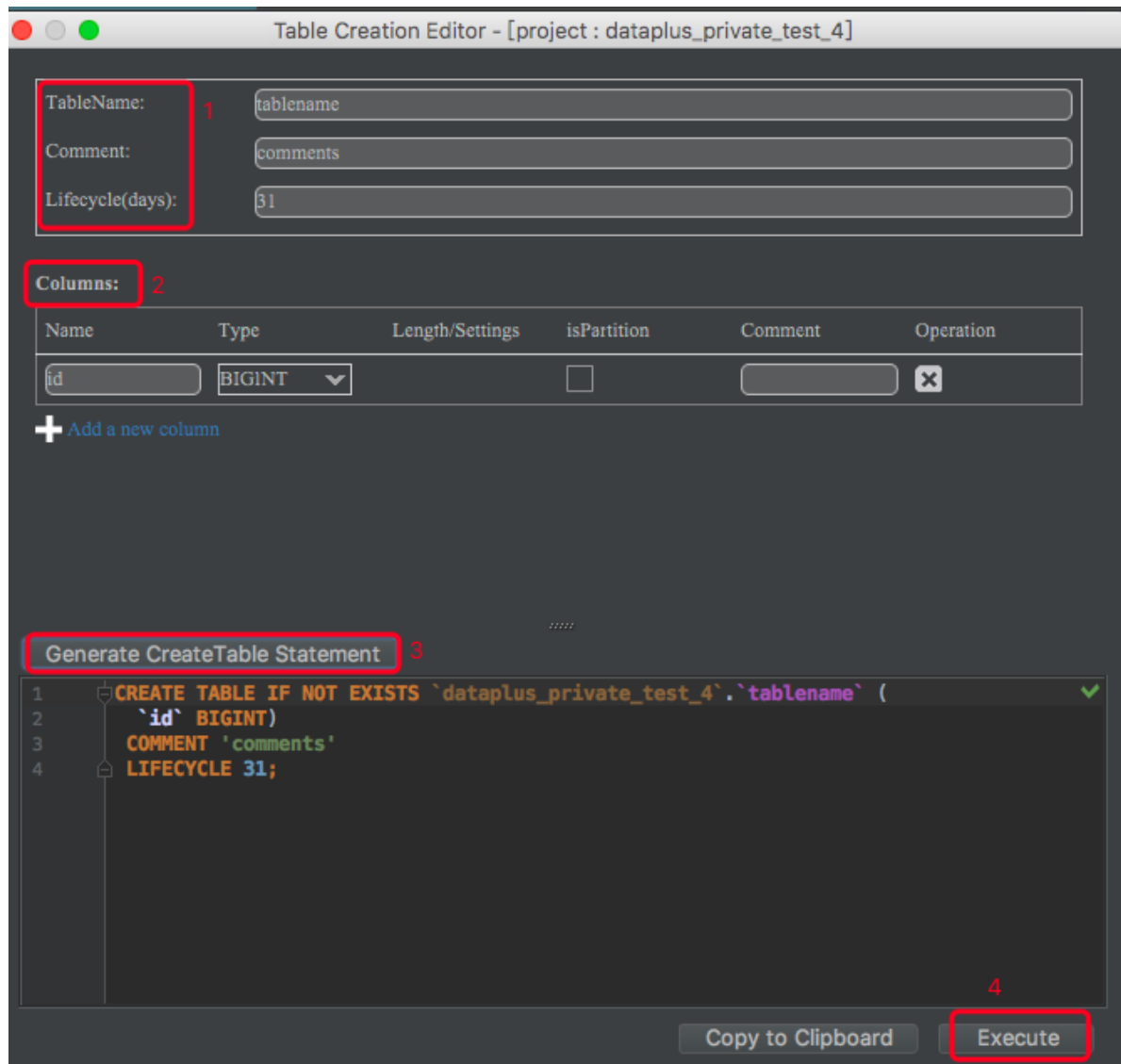
Visualization of creating a table

Procedure

1. Right-click the project that you want to create the table, and select **create a new table**:

2. In the dialog box that appears, enter a table name and column information. Click **Generate CreateTable Statement** generates the corresponding pant statement, click **Execute** to execute the build table.


When you set the table name, column name, type, and lifecycle, observe the related requirements of MaxCompute. For more information, see *Table Operations*.

3. After the table is created, view the table metadata in table&view of the Project Explorer. If no metadata is displayed, refresh the list.
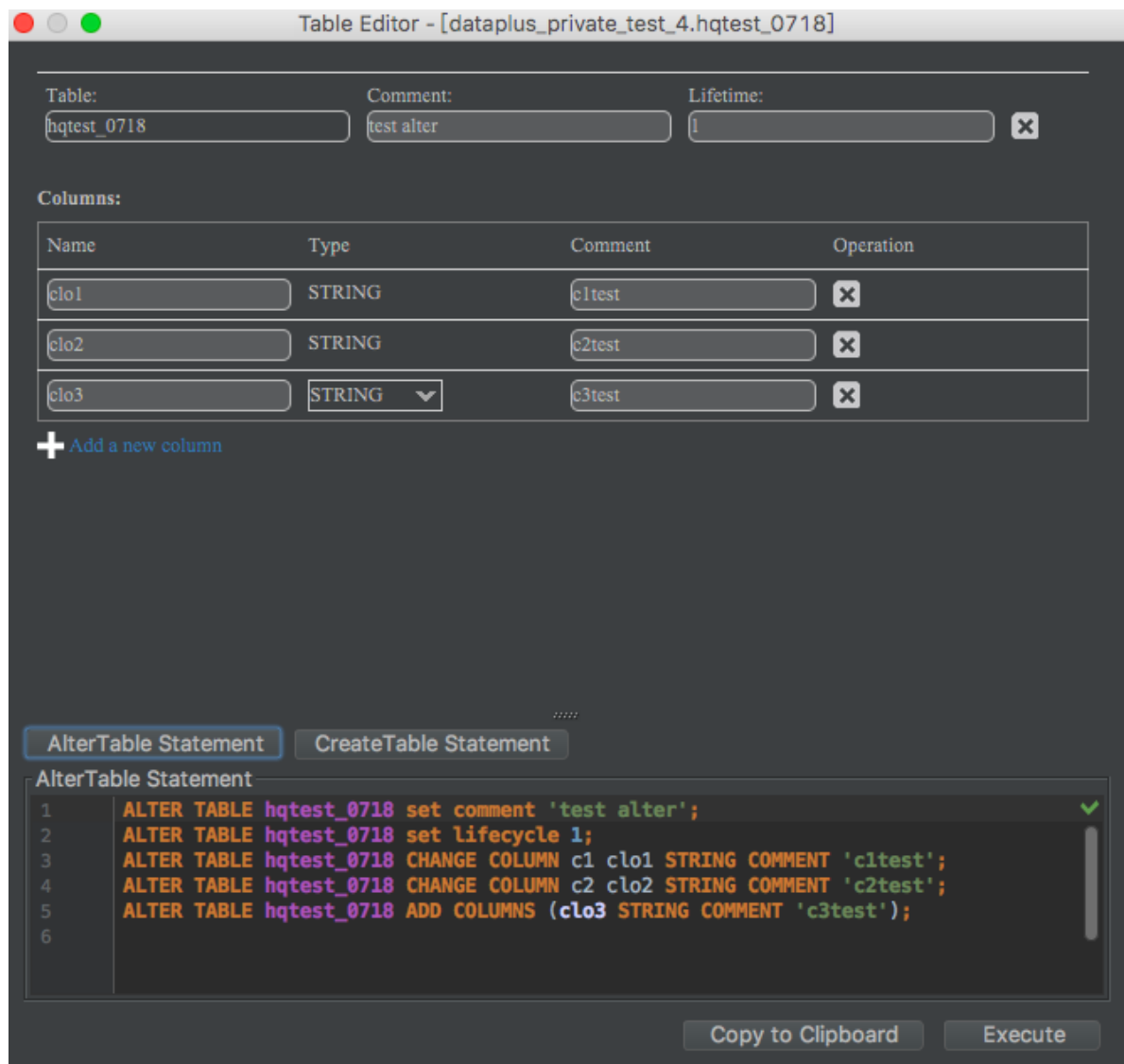
Visualization of modifying a table

Procedure

1. In table&view of the Project Explorer, right-click the expected table and select **Open table Editor**:
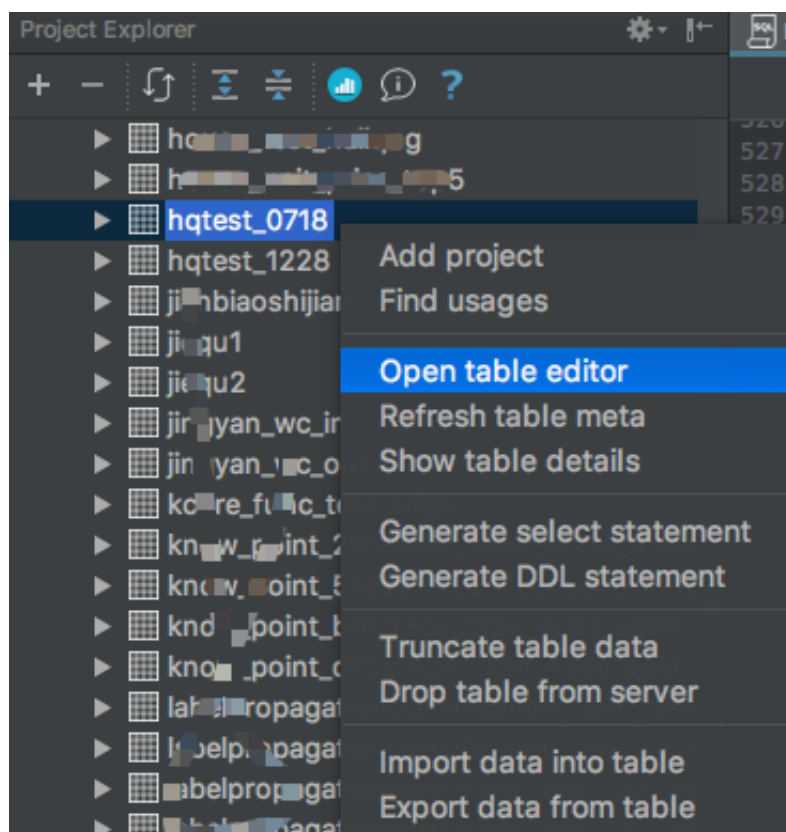
2. In the dialog box that appears, edit the table. You can modify the table comments, table

lifecycle, column name, and column description, and add columns. Specific rules follow the

MaxCompute table requirements and can be found in *Table Operations*.

3. After completing the modifications, click **Alter Table Statement** Generate a specific alter statement and click **Execute**to perform the table modify operation. After successful execution, view the table metadata.

Visualization of deleting a table

In table&view of the Project Explorer, right-click the expected table and select **Drop table from server**:
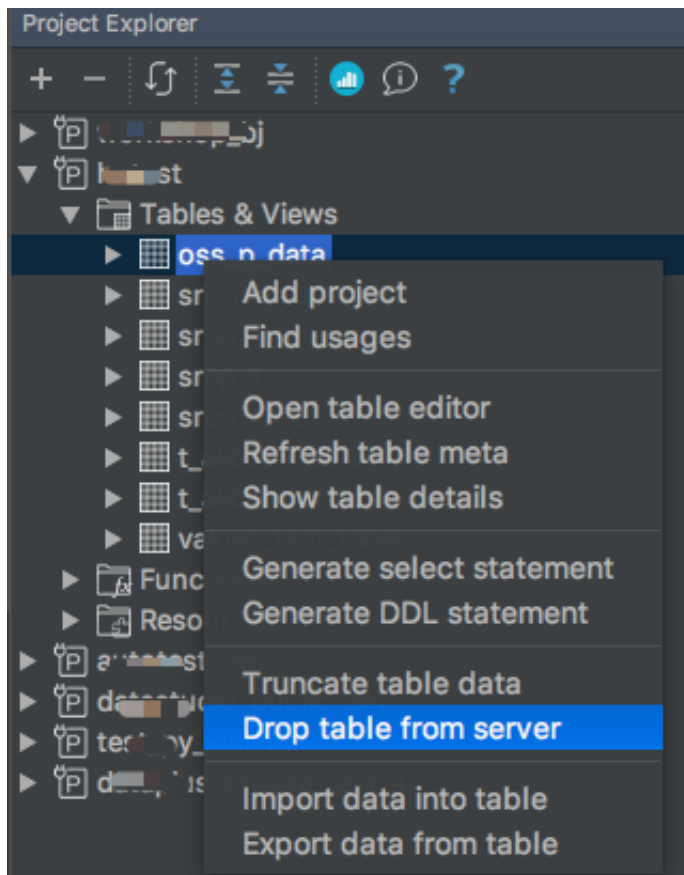
Select **OK** in the bullet box to remove the table from the maxcompute service.

2.4 Developing Java

2.4.1 Create MaxCompute Java Module

MaxCompute Studio supports Java user-defined function (UDF) and MapReduce development. First, a MaxCompute Java module must be created.
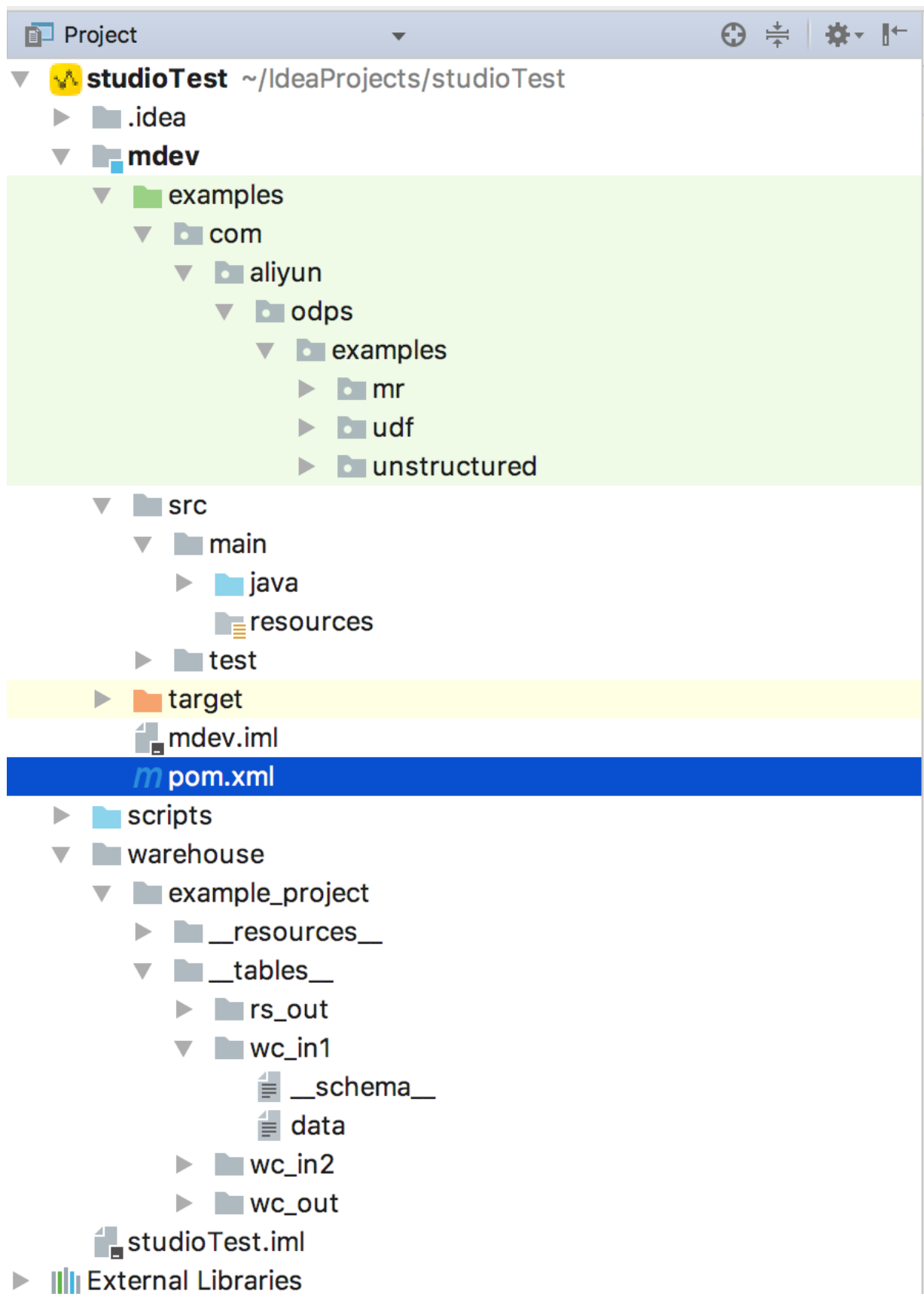
Create a module

Choose `File > New > Module`, set the module type to `MaxCompute Java`, and configure `Java JDK`. Click **Next**, enter a module name, and click **Finish**. MaxCompute Studio automatically creates a Maven module and introduces MaxCompute dependencies.

Module structure

So far, a module for developing a MaxCompute Java program has been established, that is the mDev shown in the following figure. Its main directories include:

• src/main/java: Source code for Java program development.

- examples: Sample code, including unit test (UT) examples. You can see the examples to develop or compile UT.

- warehouse: Schema and data required for running locally.

📑 Project ▾　　　　　　　　　　　　　⊕ ⊹ ❈▾ ⊩

▾ 🟡 **studioTest** ~/IdeaProjects/studioTest
　▶ 📁 .idea
　▾ 📁 **mdev**
　　▾ 🟩 examples
　　　▾ 📁 com
　　　　▾ 📁 aliyun
　　　　　▾ 📁 odps
　　　　　　▾ 📁 examples
　　　　　　　▶ 📁 mr
　　　　　　　▶ 📁 udf
　　　　　　　▶ 📁 unstructured
　　▾ 📁 src
　　　▾ 📁 main
　　　　▶ 📁 java
　　　　　📁 resources
　　　▶ 📁 test
　　▶ 📁 target
　　　📄 mdev.iml
　　　_m_ **pom.xml**
　▶ 📁 scripts
　▾ 📁 warehouse
　　▾ 📁 example_project
　　　▶ 📁 __resources__
　　　▾ 📁 __tables__
　　　　▶ 📁 rs_out
　　　　▾ 📁 wc_in1
　　　　　📄 __schema__
　　　　　📄 data
　　　　▶ 📁 wc_in2
　　　　▶ 📁 wc_out
　　📄 studioTest.iml
▶ 📊 External Libraries
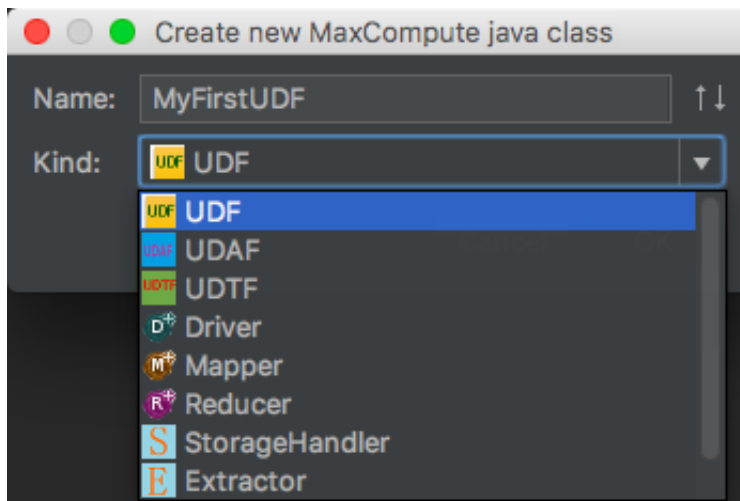
2.4.2 Develop and debug UDF

Once the *MaxCompute Java Module* has been created, udfs can be developed.

Procedure

1. Expand the MaxCompute Java Module Directory that you created and navigate to **src** > **main** > **java** > **new**, and click **MaxCompute Java** as shown in the following figure.
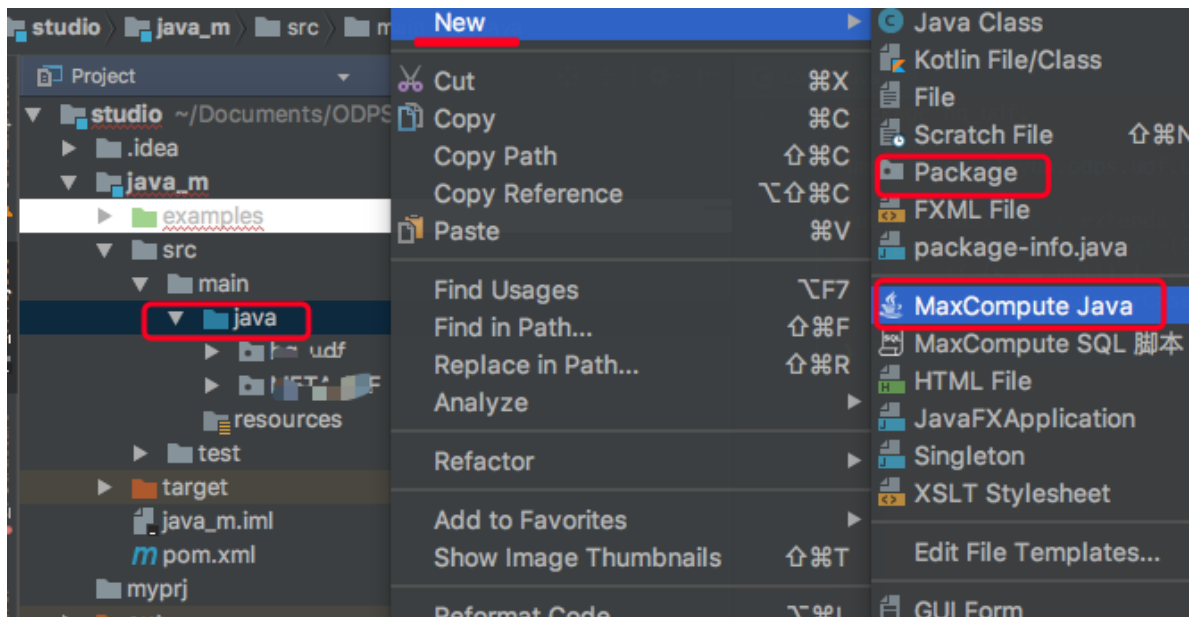

2. Set **Name** and **Kind**, and click **OK**. as shown in the following figure.

- Name: Specifies the name of the MaxCompute Java Class. If you have not created a package, you can enter packagename.classname to automatically create a package.

- Kind: Specifies the type. Supported types include custom functions (UDF/UDAF/UDTF), MapReduce (Driver/Mapper/Reducer), and non-structural development (StorageHandler/Extractor).

3. After the creation is successful, the Java program can be developed, modified, and tested.
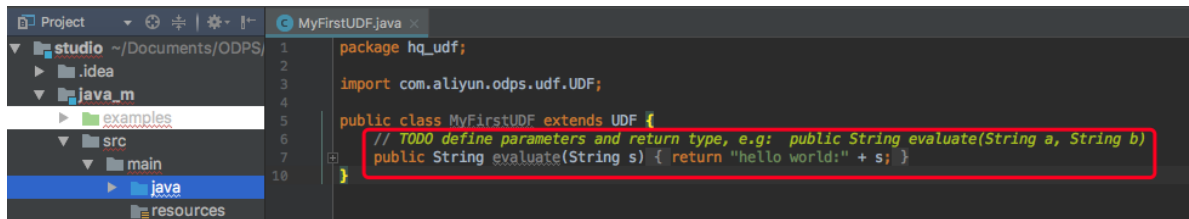
> **Note:**
>
> Here's a code template that can be customized in Intellij. You can define it in **Preference** >
>
> **Editor** > **File** > **Code Templates**. Then look for the corresponding template in the **Code** tab.

For detailed development steps, see*JAVA UDF development*.

Normally, the development of JAVA UDF can be done in the following ways:

- Use MaxCompute Studio to complete the whole process of JAVA UDF development.

- Use *Develop and debug JAVA UDF using the Eclipse plug-in*, export the Jar package, then *Add resources* through commands or DataWorks, and *Register the function*.
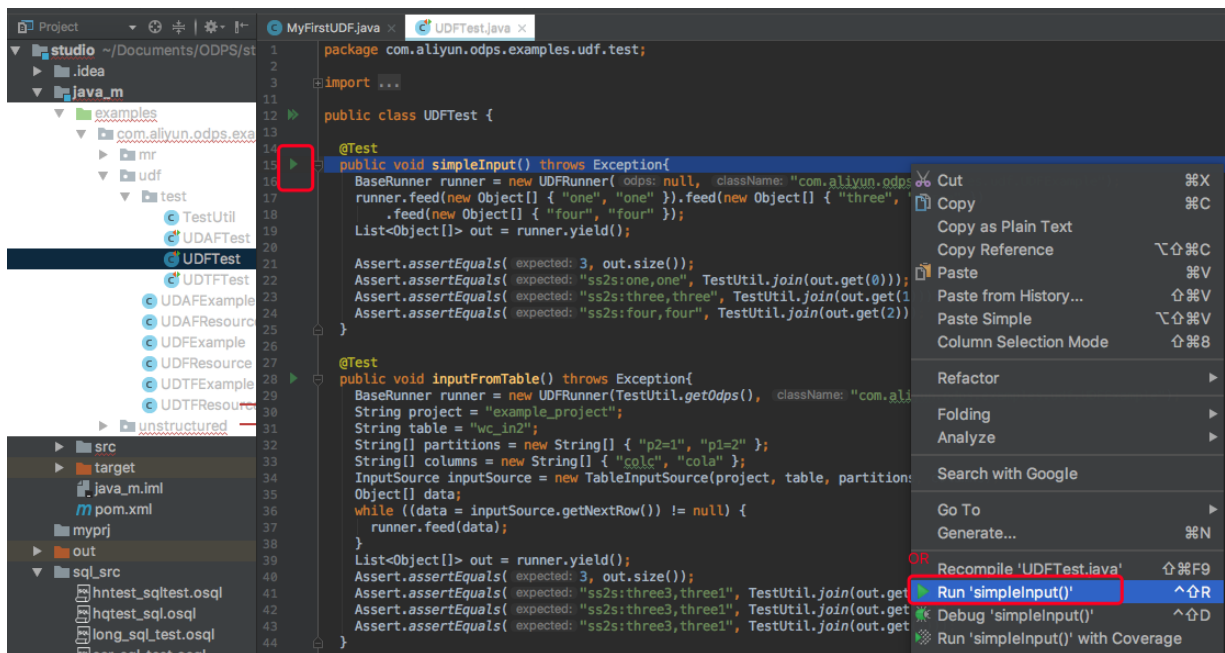
For detailed development steps, see*JAVA UDF Development*.

Debug the UDF program

After the UDF program is developed, it can be tested using unit test (UT) or local running to check whether it meets expectations.

Unit Testing

There are various UT examples in the examples directory and you can refer to them to compile your UT.

Run locally

During local running of the UDF program, the running data source must be specified. The following two methods are provided to set the test data source:

- MaxCompute Studio uses the Tunnel Service to automatically download table data of a specific project to the warehouse directory.

- The mock project and table data are provided. You can see example_project in warehouse to set it by yourself.

Procedure

1. Right-click UDF Class and select **Run UDF class.main()**. The Run Configuration dialog box is displayed. In normal cases, UDF/UDAF/UDTF data is used as columns in tables of a select sub-statement. The MaxCompute project, table, and column need to be configured. (**The metadata is from the mock project under project explorer and warehouse**.) Debugging for complex types is also supported, as shown in the following figure:
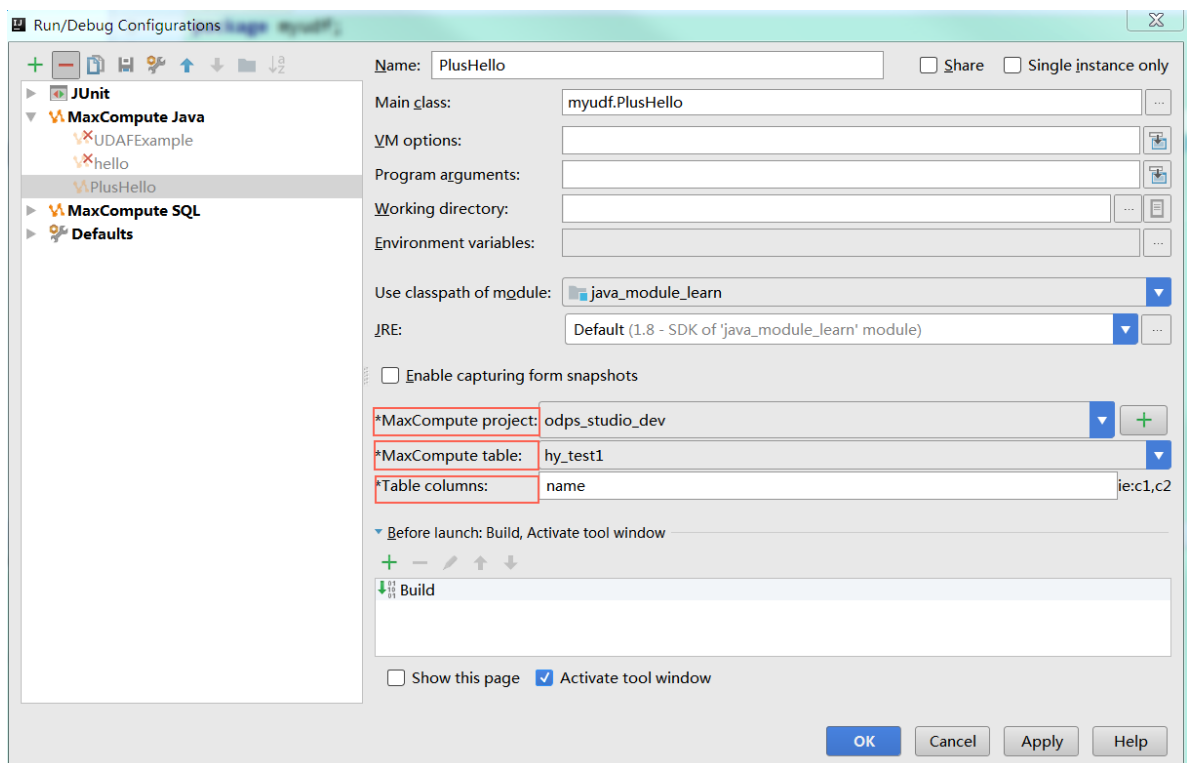
2. Click **OK**.
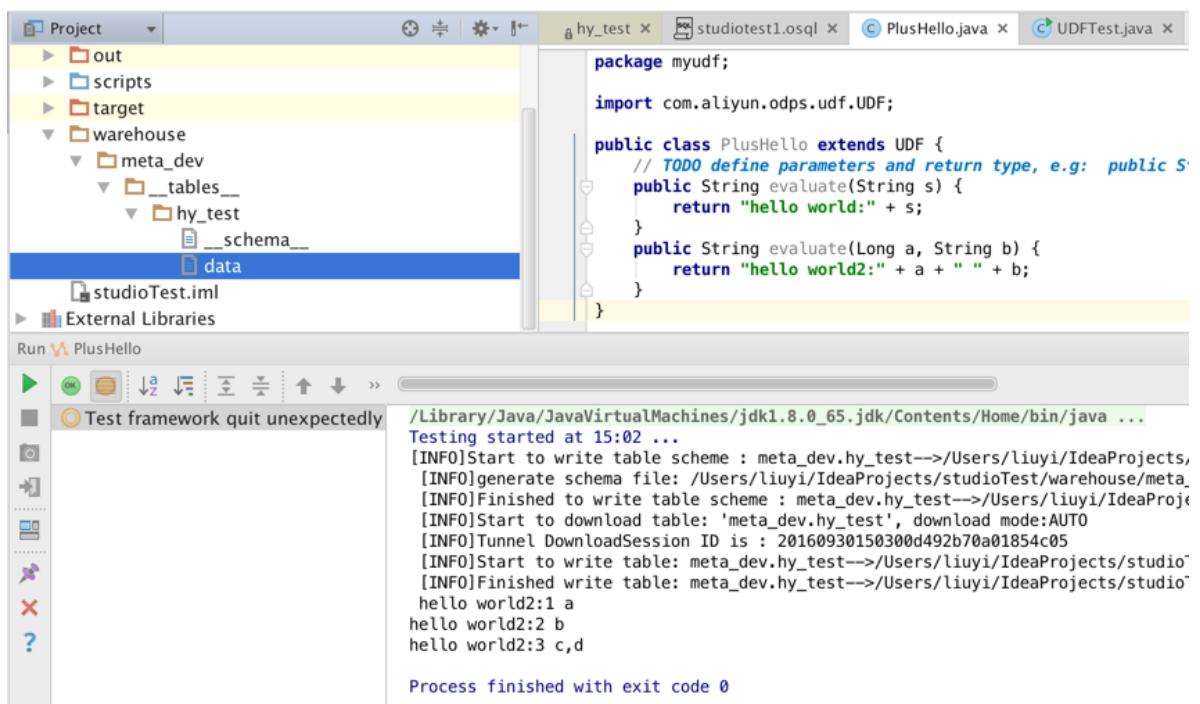
Note:

- If the table data under the specified project is not downloaded into glashourse, You need to download the data first, default download 100 entries. If more data is required, use the Tunnel Command of the console or table downloading function of Studio.
- If the mock project is used or the table data is downloaded, directly run the program.
- The UDF local run framework uses data in specific columns in warehouse as the UDF input and run the UDF program locally. You can view log output and result display on the console.

Local warehouse directory

The local warehouse directory is used to store tables (including meta and data) or resources for local UDF or MR running. The following figure shows the warehouse directory.
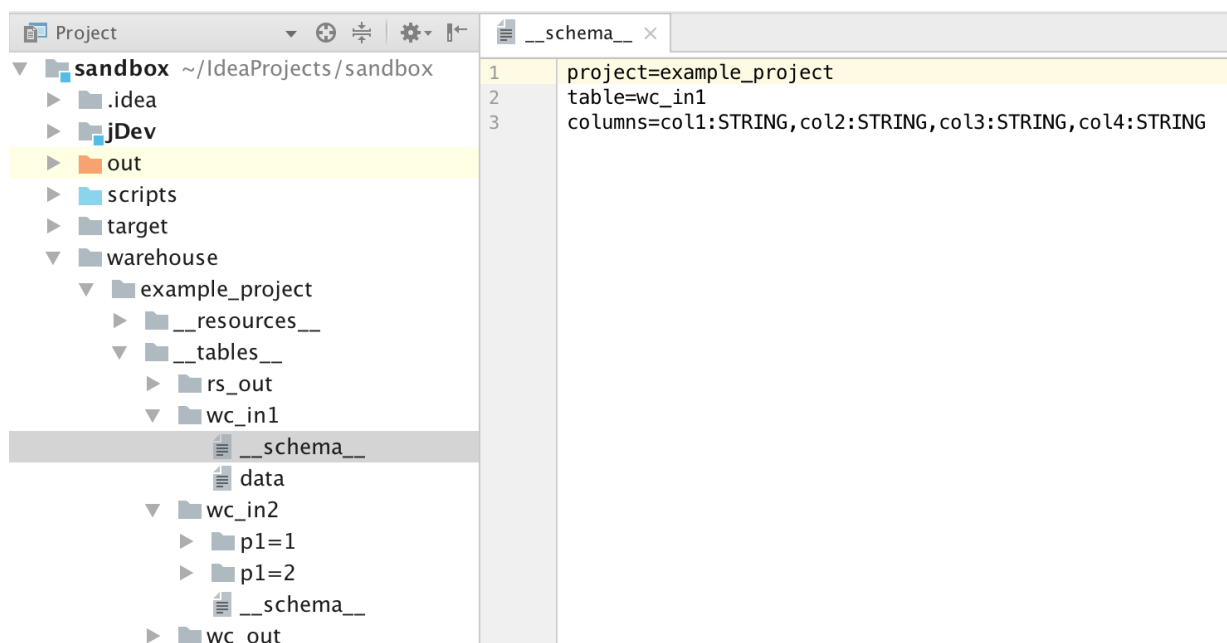
📋 **Note:**

- The project name, tables, table name, table scheme, and sample data are under the warehouse directory in sequence.
- The schema file is configured with the project name, table name, and column name and type (separated using a colon) in sequence. For a partition table, the partition column also needs to be configured. (For a non-partition table, refer to wc_in1. For a partition table, refer to wc_in2).
- The data file uses the standard CSV format to store table sample data.

  — Special characters include comma, double quotation marks, and line feed (\n or \r\n).

 — The column separator is comma and the line separator is \n or \r\n.

- If the column content includes special characters, double quotation marks (") must be
  added before and after the column content. For example, if the column content is 3,No, it is
  changed to "3, No".

- If the column content includes double quotation marks, each double quotation mark is
  converted to two double quotation marks. For example, if the column content is a"b"c, it is
  changed to "a""b""c".

- \N indicates that a column is null. If the column content (string type) is \N, it must be
  converted to """\N""".

- The file character code is UTF-8.

2.4.3 Develop MapReduce

After the *MaxCompute Java module* is created, *MR* can be developed.

Develop the MR program

1. Right-click the module source code directory src > main, select **New**, and select **MaxCompute
   Java**.

2. Create Driver, Mapper, and Reducer.
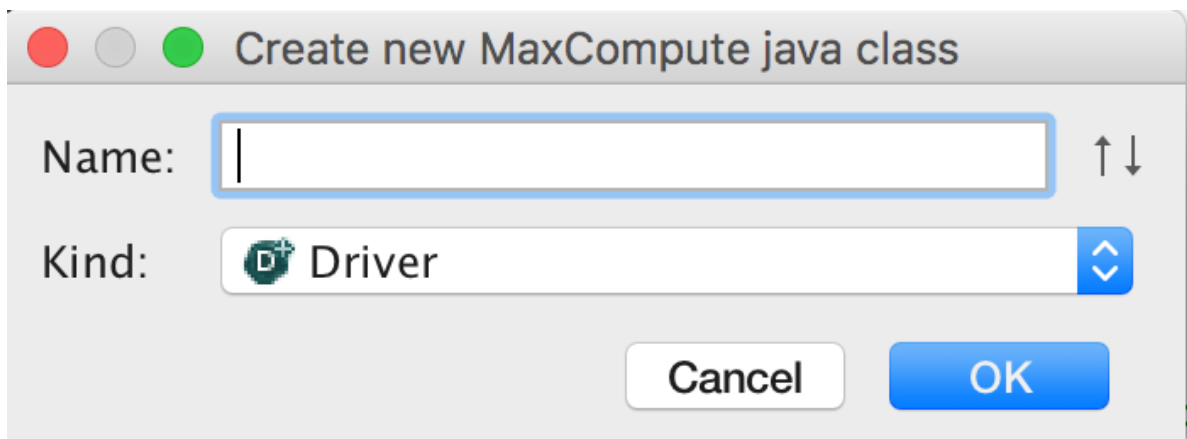
3. Set the input/output table and Mapper/Reducer class. The framework code is automatically
   filled in the template.

```
  HelloDriver.java  ×

        HelloDriver
1       package mymr.myudf;
2
3     ⊞import ...
11          💡
12  ▶   public class HelloDriver {
13
14  ▶       public static void main(String[] args) throws OdpsException {
15
16               JobConf job = new JobConf();
17
18               // TODO: specify map output types
19               job.setMapOutputKeySchema(SchemaUtils.fromString( ?));
20               job.setMapOutputValueSchema(SchemaUtils.fromString( ?));
21
22               // TODO: specify input and output tables
23               InputUtils.addTable(TableInfo.builder().tableName( ?).build(), job);
24               OutputUtils.addTable(TableInfo.builder().tableName( ?).build(), job);
25
26               // TODO: specify a mapper
27               job.setMapperClass( ?);
28               // TODO: specify a reducer
29               job.setReducerClass( ?);
30
31               RunningJob rj = JobClient.runJob(job);
32               rj.waitForCompletion();
33
34           }
35
36  }
```
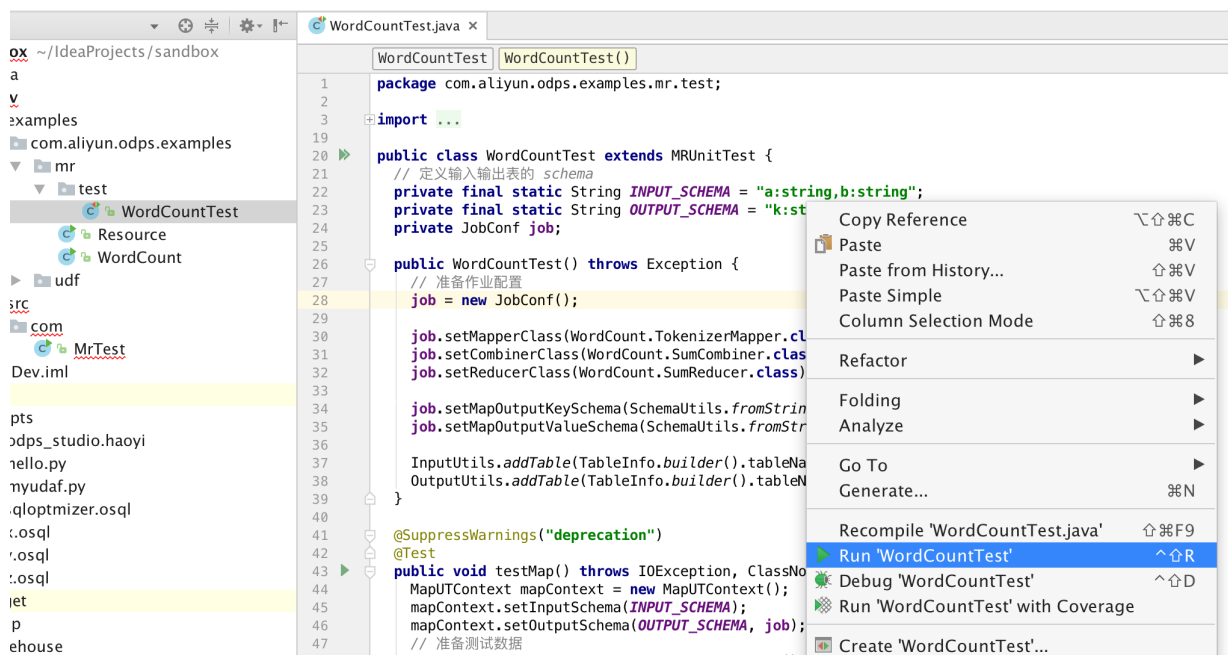
For details of developing MR, see *To write MapReduce*.

Debug the MR program

After the MR program is developed, test your code and check whether it meets the expectations. The following two methods are supported:

Unit test (UT): There are WordCount UT examples in the examples directory. You can refer to them to compile your UT.

Local MR running: During local running, the running data source must be specified. The following two methods are provided to set the test data source:

- MaxCompute Studio uses the Tunnel Service to automatically download table data of a specific MaxCompute project to the warehouse directory. By default, 100 data records are downloaded. If more data is required for testing, use the Tunnel Command of the console or table downloading function of MaxCompute Studio.

- Provide the mock project (example_project) and table data. You can see example_project in warehouse to set it by yourself.

1. Run the MR program. Right-click the Driver class and select **Run**. In the displayed **Run Configuration** dialog box, configure the MaxCompute project on which the MR program runs.
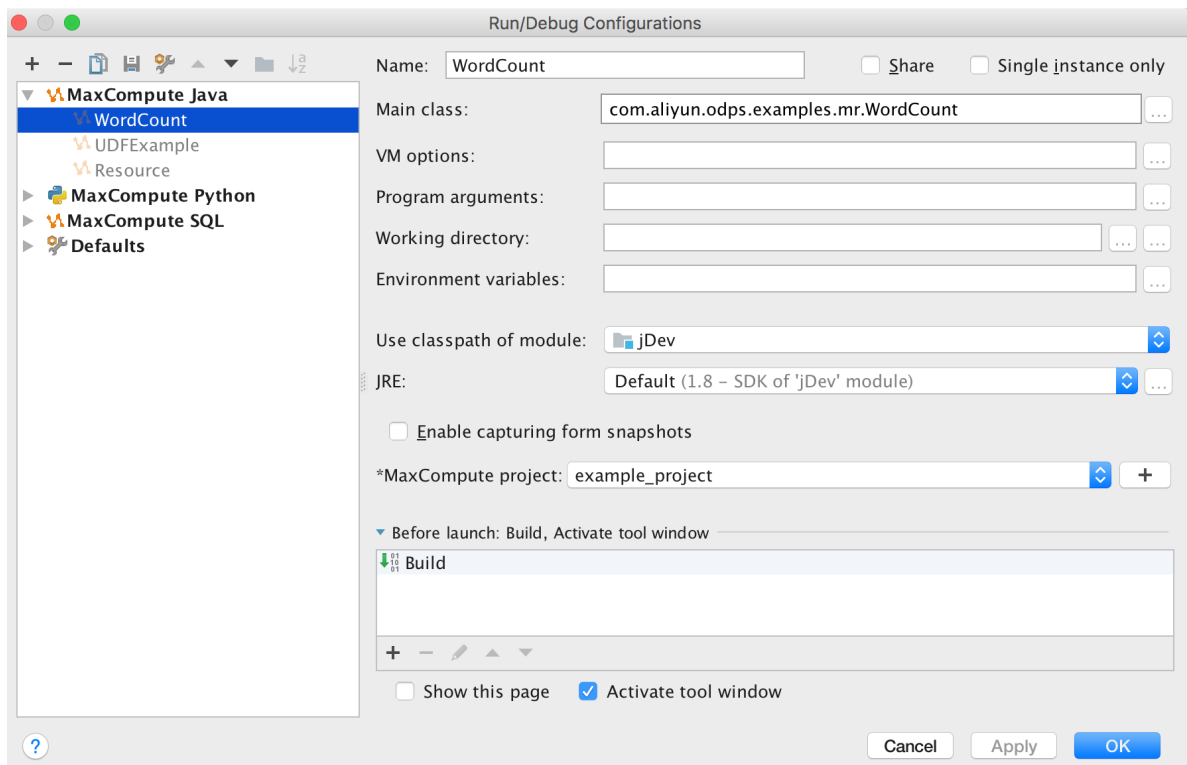
2. Click **OK**. If table data of the specified MaxCompute project is not downloaded to warehouse, download data first. If a mock project is used or the MaxCompute project table data is downloaded, skip this step. Then, the MR local run framework reads specified table data in warehouse as the MR input and runs the MR program locally. You can view log output and result display on the console.
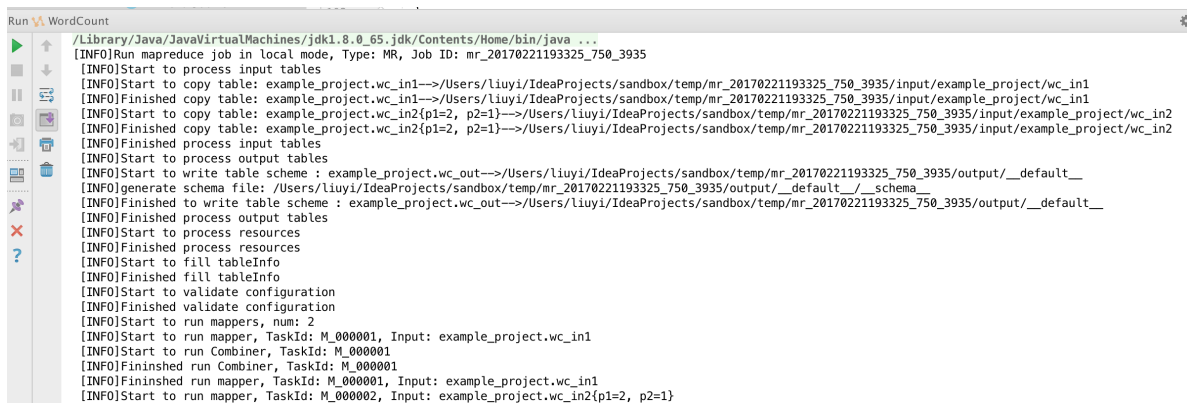
Run the MR program in the production environment

After local debugging is complete, release the MR program to the server and run it in the MaxCompute distributed environment.

1. Package the MR program to a JAR package and release it to the server. *For more information , see How to package and release MR.*

2. Use the MaxCompute console integrated with MaxCompute Studio in seamless mode, that is, in the Project Explorer window, right-click Project and select Open in Console, and input the commands similar to the following *JAR command* in the console command line:

```
jar -libjars wordcount.jar -classpath D:\odps\clt\wordcount.jar com.
aliyun.odps.examples.mr.WordCount wc_in wc_out;
```

2.4.4 Unstructured development

An *unstructured data processing framework* is added for MaxCompute 2.0, supporting access to the OSS and Table Store using external tables. Studio provides some code templates for the framework, facilitating users' fast development.
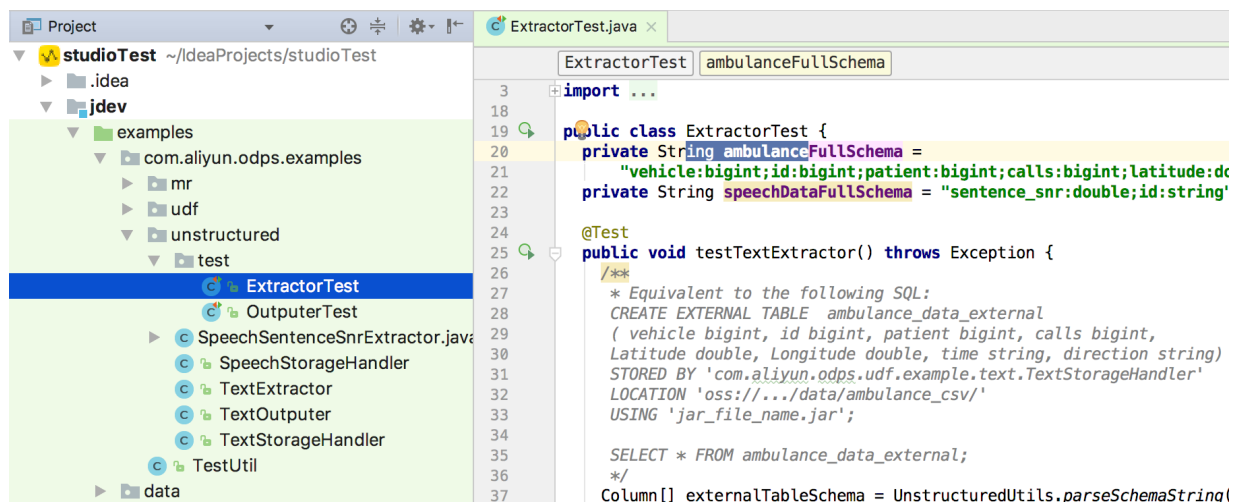
Compile StorageHandler/Extractor/Outputter

1. After the *MaxCompute Java Module* (Sample code is provided in the unstructured folder of the examples directory for your reference).

2. Right-click the module source code **directory src >** > **main,** select **new**, and select **MaxCompute Java**.

3. Specify Name and Kind. For example, set Name to myun.MyExtractor and Kind to Extractor. Click **OK**.
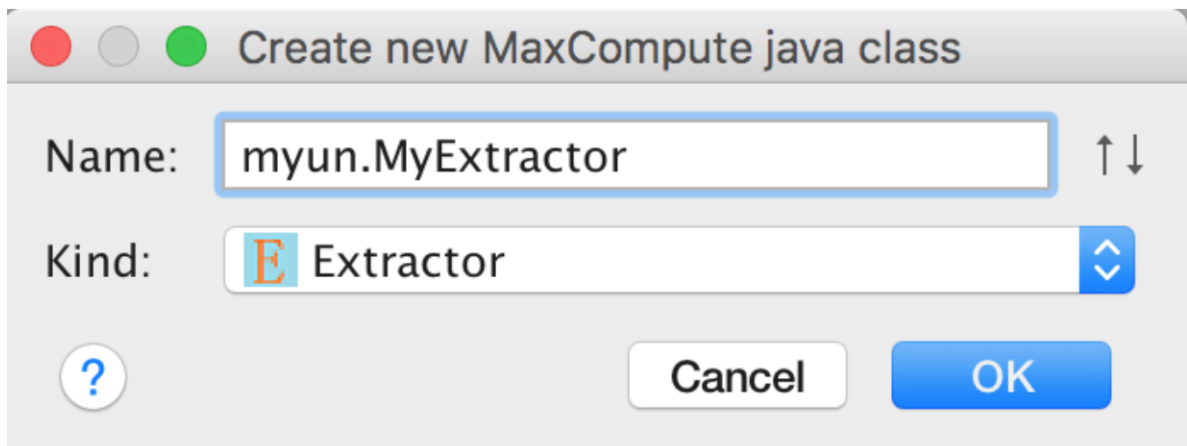
4. The framework code has been automatically filled in the template. Compile your logic code.

5. Compile Outputter and StorageHandler by following the preceding steps.

Unit Testing

You can compile the unit test (UT) by following the examples in the examples directory to test your Extractor/Outputter.
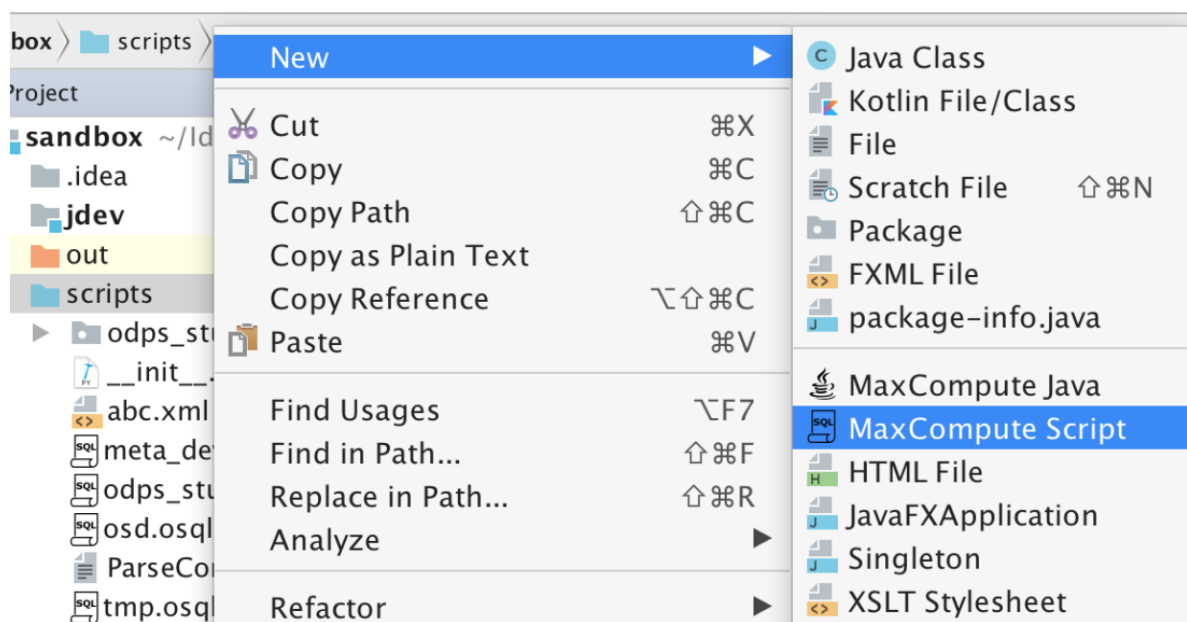
Package and upload

After StorageHandler/Extractor/Outputter is compiled, compress the completed Java program to a JAR package, and upload the package as a resource to the server, see *Package and release*.

Create External Table

1. Right-click **scripts** and select **new** > **MaxCompute Script**.

2. Enter the SQL script name. Select the MaxCompute project in which the script is to be executed for Target Project and click **OK**.

3. Select create external table live template in the editor to rapidly insert the script template for creating an external table.
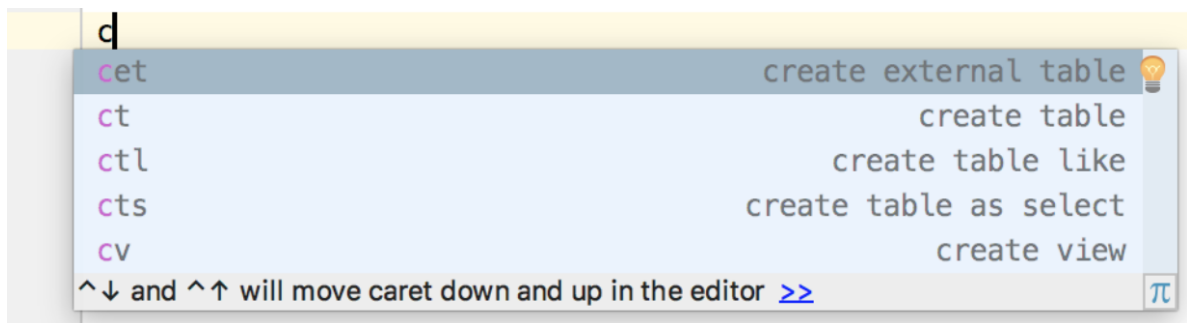
Modify the external table name, column, type, StorageHanlder class path, configuration parameter, external path, and JAR name. Click Run MaxCompute SQL Script to create the external table.

4. Query the created external table, such as:

2.4.5 Package/Upload/Register

After a *user-defined function* or *MapReduce* is developed, you must package and release it to the MaxCompute system.

Package a UDF or MapReduce

To release a UDF or MapReduce to the MaxCompute server for production use, you must complete **packaging** > **uploading** > **registration** in sequence. You can use the one-click release function to complete these procedures. MaxCompute Studio runs the mvn clean package command, uploads a JAR package, and registers the UDF in one stop. To use this function, right-click the UDF or MapReduce and select Deploy to server…. Make sure that the target class is in the **src** > **main** > **java** subdirectory and is successfully compiled on the Maven module. The dialog box shown in the following figure appears. Select the MaxCompute project to be deployed and enter a resource name and a function name. Click OK and wait until the operation in the background is complete.
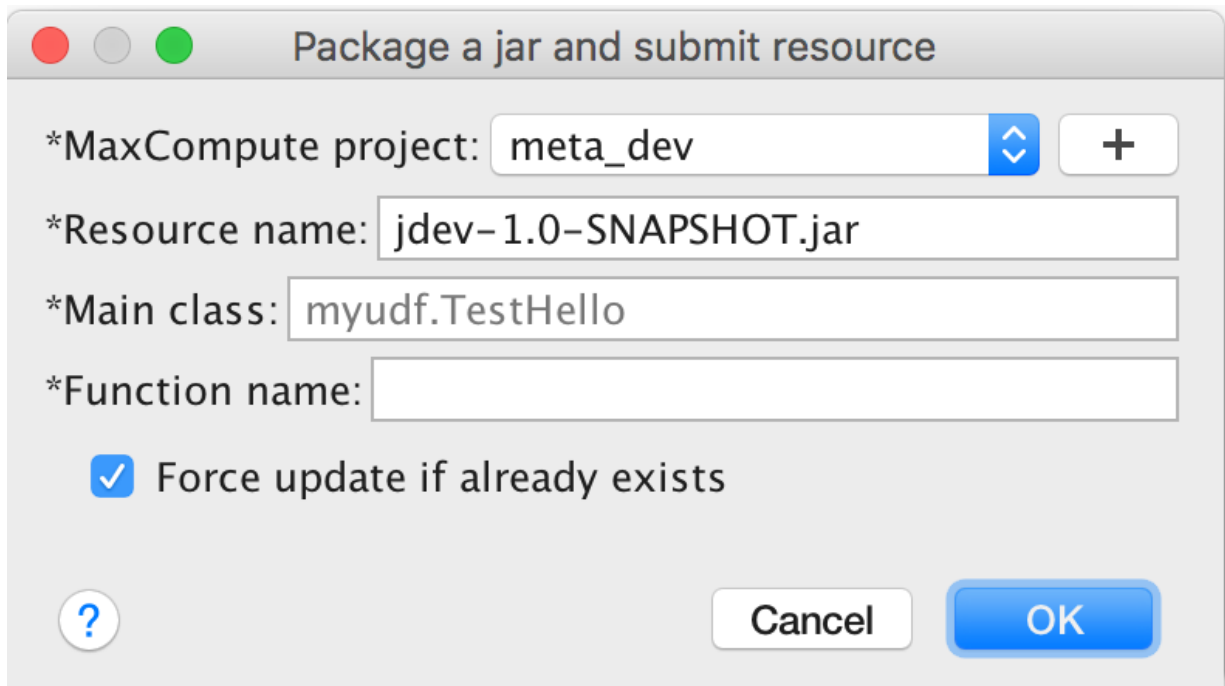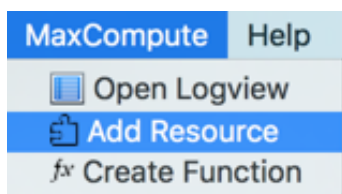
> **Note:**
>
> If you require special packaging, you can modify relevant settings in the pom.xml file. After
>
> packaging, follow these steps to upload the JAR package and register the UDF.

Upload the JAR package

After the JAR package is prepared, upload it to the MaxCompute server.
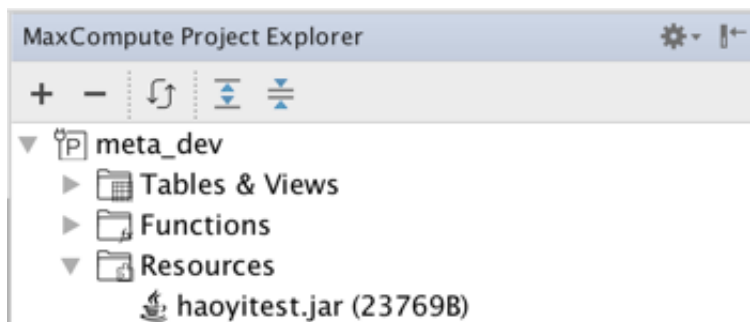
1. Select Add Resource from the MaxCompute menu.

2. Select the MaxCompute project you want to upload the resource to, the JAR file path, and the
resource name you want to register. Determine whether to force update when the resource or
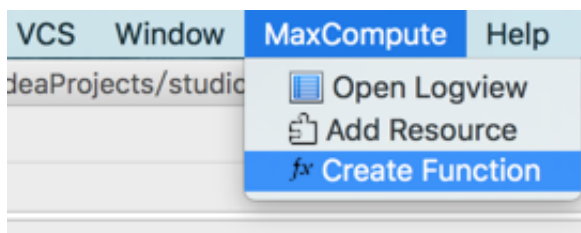function already exists. Then click OK.
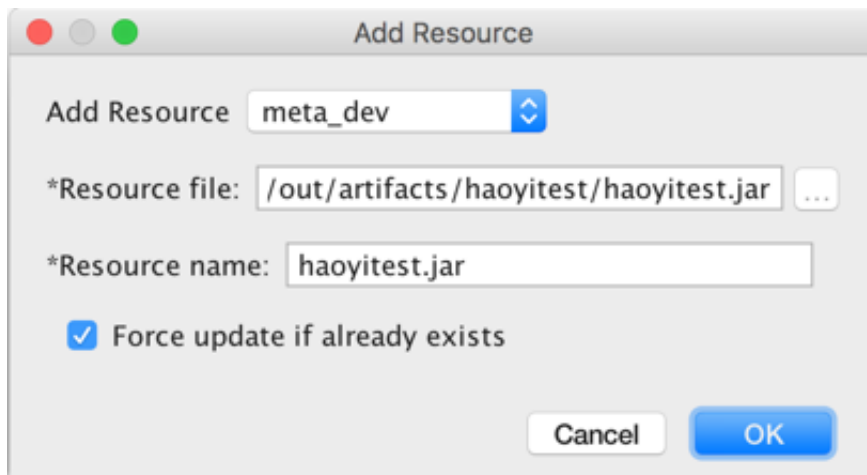
3. After uploading is successful, you can view the resource under the Resources node of the Project Explorer window.

Register the UDF

After the JAR package is uploaded, register the UDF.
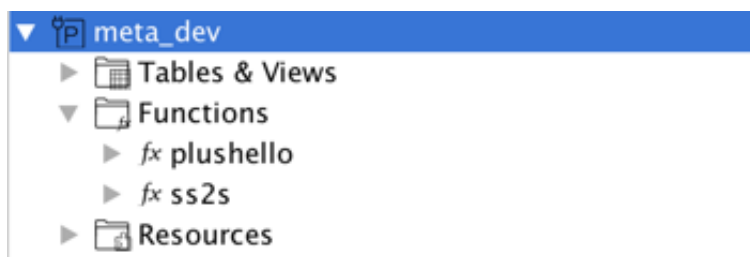
1. Select **Create Function** from the MaxCompute menu.

2. Select the required resource JAR and JAR main class, and enter the function name. Click **OK**.
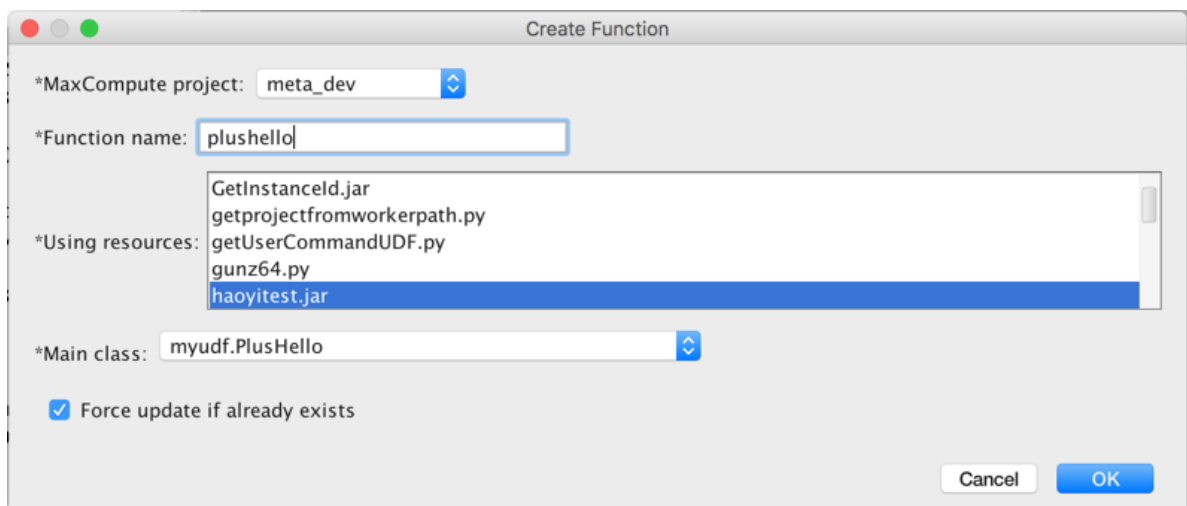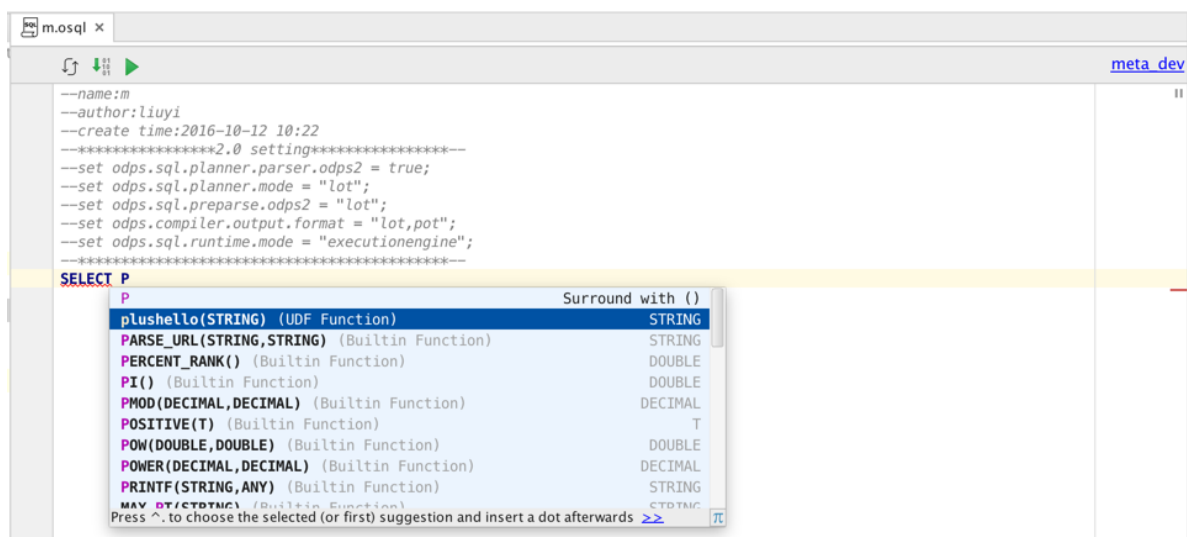
3. After the registration is successful, you can view the function under the Functions node of the Project Explorer window.

Apply the UDF

- Apply the UDF in SQL to complete subsequent development.

2.4.6 Develop Graph

After the *MaxCompute Java module* is created, *Graph* can be developed.

Sample Code

There are some code examples of Graph in the examples directory, and you can refer to the example to get familiar with the structure of the Graph program.
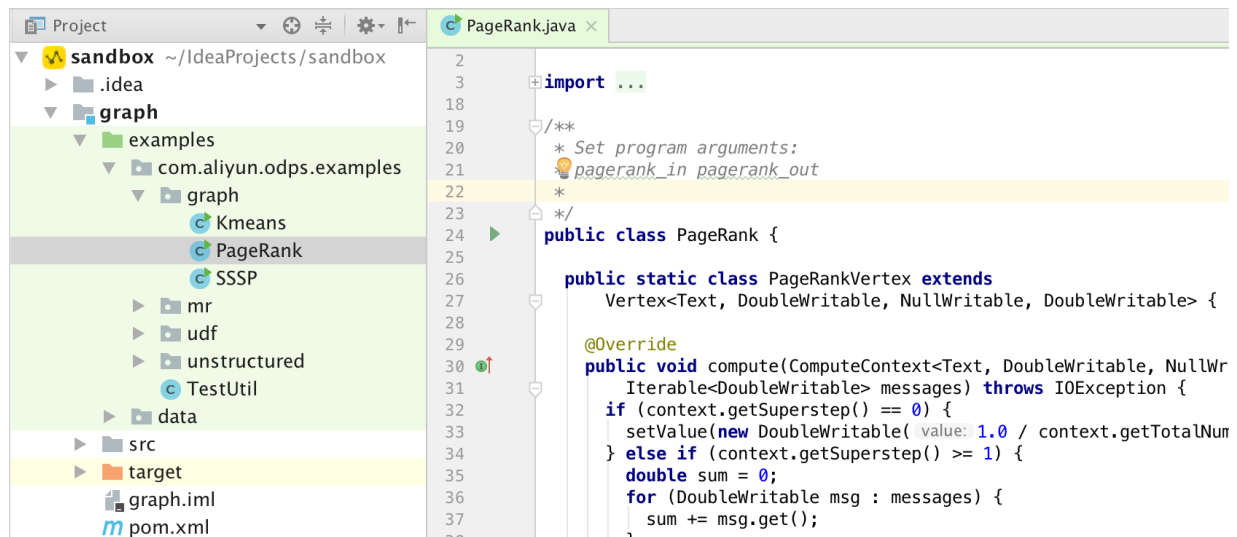
Develop a Graph Program

1. Right-click the module source code **directory src >** > **main,** select **new**, and select **MaxCompute Java**.

2. Select the GraphLoader/Vertex type and enter the class name (package name is supported) in the **Name** text box. Click **OK**, and the frame code will be automatically filled in by the template, you can continue to modify.
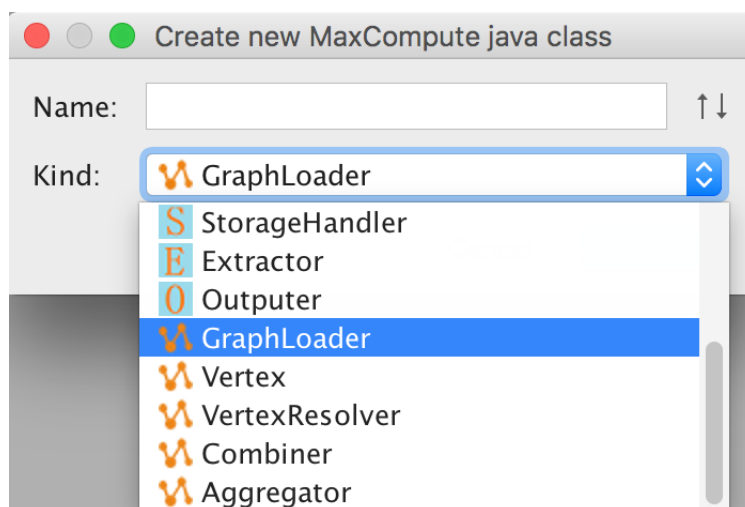
Debug Graph Locally

After the Graph program is developed, test your code and check whether it meets the expectations. You can run the Graph code locally.

1. Run the Graph program: Right-click the Driver class and select **Run**. In the displayed **Run Configuration** dialog box, configure the MaxCompute project on which the Graph program runs.

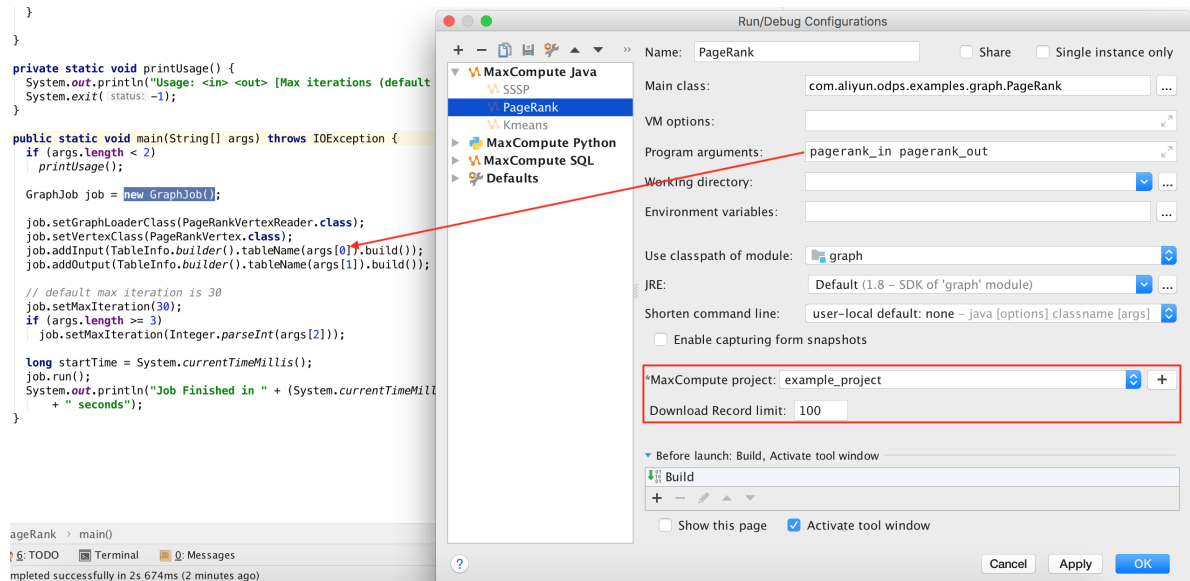2. Click **OK**. If table data of the specified MaxCompute project is not downloaded to warehouse, download data first. If a mock project is used or the MaxCompute project table data is downloaded, skip this step. Then, the graph local run framework reads specified table data in warehouse as the Graph input and runs the Graph program locally. You can view log output and result display on the console. Each time you debug locally , a new temporary directory is created under the Intellij directory, as shown in the following figure:
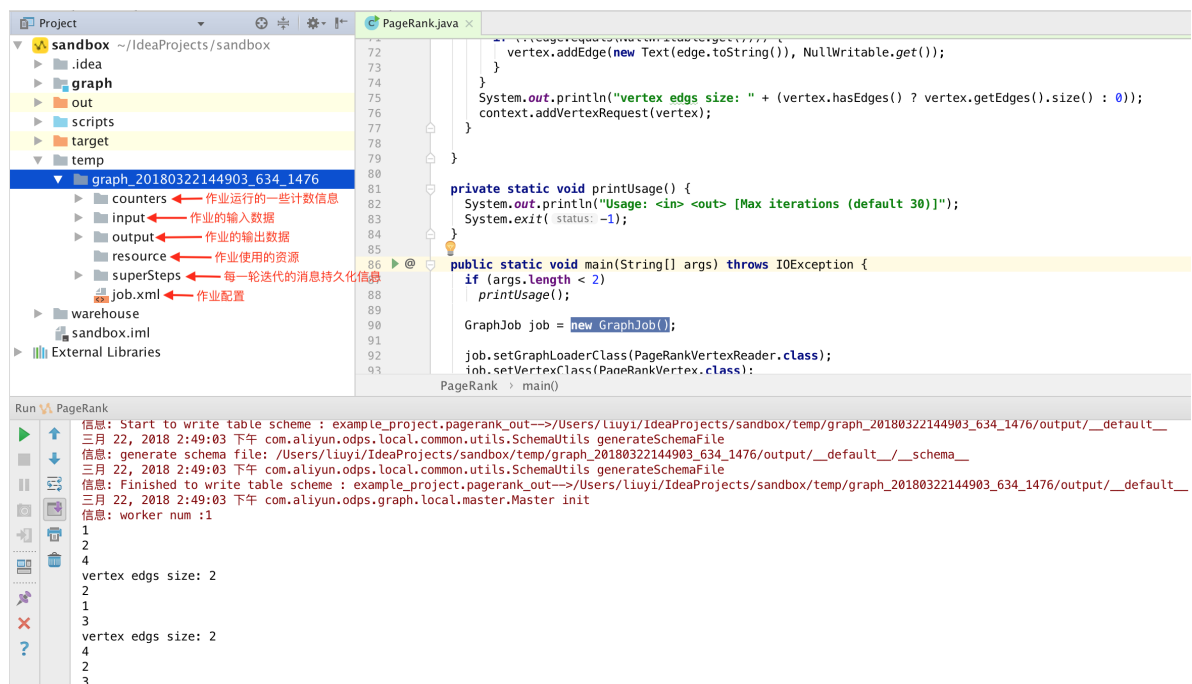
> **Note:**
>
> For a detailed introduction to warehouse, see   Local Warehouse Directory section in *Develop UDF*.

Run the Graph Program in the Production Environment

After local debugging is complete, release the Graph program to the server and run it in the MaxCompute distributed environment.

1. Package the Graph program to a JAR package and release it to the server. *For more information, see How to package and release Graph.*

2. Use the MaxCompute console integrated with MaxCompute Studio in seamless mode, that is, in the Project Explorer Window, right-click Project and select Open in Console, and input the commands similar to the following *JAR command* in the console command line:

```
jar -libjars xxx.jar -classpath /Users/home/xxx.jar com.aliyun.odps.
graph.examples.PageRank pagerank_in pagerank_out;
```

2.5 Configure options

2.5.1 Configure MaxCompute Studio

After the MaxCompute Studio plug-in is installed, you can find configuration items of MaxCompute Studio on the left bar of the Settings page of IntelliJ IDEA. For more information about how to open the IntelliJ IDEA configuration page, see *IntelliJ IDEA Documentation*.

MaxCompute Studio configuration option page

The MaxCompute Studio configuration option page provides the following configuration items:

1. **Path for storing the local metadata base**

   Specifies the path for locally storing metadata of a MaxCompute project. On MaxCompute Studio, the metadata is stored in the hidden directory `.odps.studio/meta` of the local user directory by default.

2. **Version update options**

   - You can use the **Automatically checks for new version** check box to control whether MaxCompute Studio automatically checks for new version updates.

   - You can use the **Check new versions** button to manually check new versions. After you click this button, if a new version is available, the **Install new version** button is displayed. You can click this button to install the new version, and restart IntelliJ IDEA after the installation is complete.

SDK and Console configuration option page

The SDK and Console configuration option page provides the following configuration items:

1. **Path for installing a MaxCompute client**

   Specifies the path for local installation of MaxCompute client. MaxCompute Studio detects the version of the MaxCompute client installed in the path. If detection fails, an error message is prompted.

   > 📋 **Note:**
   >
   > MaxCompute Studio later than the 2.6.1 version provides the latest MaxCompute client. You do not need to specify the path. If you must use a MaxCompute client of a specific version, you can specify the path.

MaxCompute SQL configuration option page

The MaxCompute SQL configuration option page provides the following configuration items:

1. **Enable syntax coloring**

Select **Enable syntax coloring** to enable the syntax highlighting feature.

2. **Enable code completion**

   Select **Enable code completion** to enable the automatic code complementing feature.

3. **Enable code formatting**

   Select **Enable code formatting** to enable the code formatting feature.

4. **Compiler options**

   These are global default compiler options. The following options can be separately set for each file on the toolbar of the SQL compiler.

   - **Compiler Mode**

     — Statement Mode: In this mode, the compiler compiles and submits a single statement of an SQL file as a unit.

     — Script Mode: In this mode, the compiler compiles and submits an entire SQL file as a unit. *NOTE: Script Mode enables the compiler and optimizer to optimize the execution plan and improve the overall execution efficiency. This mode is in the test phase now.*

   - **Type System**

     — Legacy TypeSystem: Indicates the type system of original MaxCompute.

     — MaxCompute TypeSystem: Indicates the new type system introduced by MaxCompute 2 .0.

     — Hive Compatible TypeSystem: Indicates the type system in Hive compatibility mode introduced by MaxCompute 2.0.

   - **Compiler Version**

     — Default Version: Indicates the default version of the compiler.

     — Flighting Version: Indicates the experimental version of the compiler, which includes new features of the compiler being tested.

Account configuration option page

You can add or manage accounts used to access MaxCompute on the Account configuration option page. For more information, see *User authentication*.

You must specify an account on MaxCompute Studio to access a MaxCompute project and run or submit jobs. MaxCompute Studio currently supports the following account type:

- Alibaba Cloud account (AccessKey)

Add an account

On the Account configuration option page, follow these steps:

1. Click **+** or press `Ctrl-N`.

2. Select the account type **Aliyun Account by AccessKey**.

3. In the displayed **Add Account** window, set the following items:

   - `Account Name`: Indicates the name of the account on MaxCompute Studio.

   - `Using properties file`: Read the AccessKey ID and AccessKey Secret from the

     configuration file.

     — Select the configuration file `conf/odps_config.ini` after you process *User*

       authentication.

   - `Using properties`: Manually enter the AccessKey ID and AccessKey Secret.

     — `Access Id`: Enter the AccessKey ID of your Alibaba Cloud account.

     — `Access Key`: Enter the AccessKey Secret of your Alibaba Cloud account.
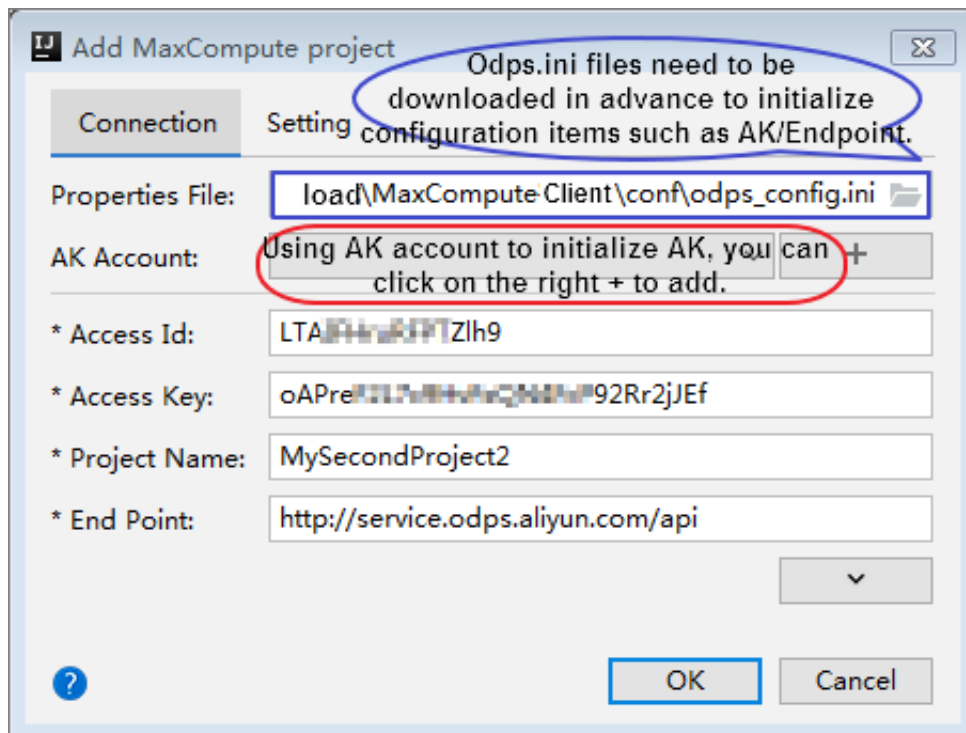
4. Click **OK** to complete addition. Then, the account will be displayed in the Account list on the

   Account configuration option page.

Delete an account

On the Account configuration option page, follow these steps: (This operation only deletes the

account configuration on Studio configuration, which does not affect your account.)

1. Select the account to be deleted in the Account list.

2. Click **-**.

3. In the displayed dialog box, click **OK**.

Modify the AccessKey of an account

On the Account configuration option page, follow these steps:

1. Select the account to be deleted in the Account list.

2. Click the pencil icon.

3. In the displayed **Edit Account** window, modify the account information. The content is similar to that in the preceding section **Add Account**.

View the opening and connection of MaxCompute Region and the settings of Endpoint, see

Endpoints and Data Centers.

3 Downloads

This document provides you with the download address of the relevant tools and plugins.

- SDK Downloads: Maven users can search odps-sdk from Maven library to get different versions of the Java SDK.
- *MaxCompute console*
- *Eclipse plugin*
- *Intelij plugin*, *Studio*