# Oracle® Analytics

## User’s Guide for Oracle Analytics Desktop

E70158-29
September 2020

Oracle Analytics User’s Guide for Oracle Analytics Desktop,

E70158-29



Primary Author: Nick Fry

Contributing Authors: Pete Brownbridge, Stefanie Rhone

Contributors: Oracle Analytics development, product management, and quality assurance teams

# Contents

## 6 Explore, Visualize, and Analyze Data

## 7 Create and Apply Filters

## 10 Use Oracle Analytics Predictive Models and Oracle Machine Learning Models

## 11 Curate Your Data Using Data Flows

## E Data Preparation Reference



## F Expression Editor Reference



## G Oracle Analytics Desktop SDK Reference

# Preface

Learn how to explore data using Oracle Analytics Desktop.

**Topics**



## Audience

*User's Guide for Oracle Analytics Desktop* is intended for business users who use Oracle Analytics Desktop to upload and query data, analyze data within visualizations, work with their favorite projects, and import and export their projects.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

**Access to Oracle Support**

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

## Related Resources

These related Oracle resources provide more information.

- Oracle Analytics Product Information
- Oracle Community Forum
- Oracle Analytics Desktop Installation Download
- Oracle Analytics Library

# Conventions

Conventions used in this document are described in this topic.

### Text Conventions

| Convention | Meaning |
| --- | --- |
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

### Videos and Images

Your company can use skins and styles to customize the look of the application, dashboards, reports, and other objects. It is possible that the videos and images included in the product documentation look different than the skins and styles your company uses.

Even if your skins and styles are different than those shown in the videos and images, the product behavior and techniques shown and demonstrated are the same.

1

# Get Started with Oracle Analytics Desktop

This topic describes the benefits of using Oracle Analytics Desktop and how to get started using the samples provided.

Video

**Topics:**



## About Oracle Analytics Desktop

Oracle Analytics Desktop provides standalone data exploration and visualization in a per-user desktop download. Oracle Analytics Desktop is the tool for quick exploration of sample data from multiple sources or for analyses and investigation of your own local data sets.

Oracle Analytics Desktop enables you to visualize your data so you can focus on exploring data patterns. Just upload data files or connect to Oracle Applications or a database, select the elements that you're interested in, and let Oracle Analytics Desktop find the best way to visualize it. Choose from a variety of visualizations to look at data in a specific way.

Oracle Analytics Desktop also gives you a preview of the self-service visualization capabilities included in Oracle Analytics Cloud, Oracle's cloud analytics platform. Oracle Analytics Cloud extends the data exploration and visualization experience by offering secure sharing and collaboration across the enterprise, additional data sources, and a full mobile experience including proactive self-learning analytics delivered to your device.

## Get Started with Samples

Use the samples provided to discover all the capabilities of Oracle Analytics Desktop, and to learn the best practices.

Because these samples use business functions such as trending, binning, forecasting, and clustering, you can use them as a quick reference when you create your own visualization.

The sample data set is based on Sales Orders data and contains meaningful dimensions, distributions, examples of data wrangling, calculated columns, and more.

You can optionally download the samples during installation. If you didn't download the samples during installation, then you can still get them by uninstalling and then reinstalling Oracle Analytics Desktop. Your personal data isn't deleted if you uninstall and reinstall Oracle Analytics Desktop.

2

# Connect to Data Sources

Create data source connections to access the data that you want to analyze and explore. Examples of data sources are: a database, Oracle applications, or a snowflake data warehouse.

**Topics:**



## About Data Sources

A data source is any tabular structure. You get to see data source values after you load a file or send a query to a service that returns results.

A data source can contain any of the following:

- **Match columns** - These contain values that are found in the match column of another source, which relates this source to the other (for example, Customer ID or Product ID).
- **Attribute columns** - These contain text, dates, or numbers that are required individually and aren't aggregated (for example, Year, Category Country, Type, or Name).
- **Measure columns** - These contain values that should be aggregated (for example, Revenue or Miles driven).

You can analyze a data source on its own, or you can analyze two or more data sources together, depending on what the data source contains. If you use multiple sources together, then at least one match column must exist in each source. The requirements for matching are:

- The sources contain common values (for example, Customer ID or Product ID).
- The match must be of the same data type (for example, number with number, date with date, or text with text).

# Manage Connections to Data Sources

Manage connections to your data sources.

**Topics:**



## Create a Connection to a Data Source

You can create a connection to enable you to analyze data in that data source.

1. On the Home page, click **Create**, then click **Connection**.
2. In the Create Connection dialog, click the icon for the connection type that you want to create a connection for (for example **Oracle Database**).
3. Enter the required connection information, such as host, port, username, password, and service name.
4. (Optional) When you connect to some database types, you might have to specify authentication options.

| Option | Description |
|---|---|
| Always use these credentials | The login name and password you provide for the connection are always used and users aren't prompted to log in. |
| Require users to enter their own credentials | Prompt users to enter their own user name and password for the data source. Users required to log in see only the data that they have the permissions, privileges, and role assignments to see. |

5. Save the details.

   You can now begin creating projects or data sets from the connection.

## Edit a Data Source Connection

You can update a data source's connection details.

1. In the Data page, click **Connections**.
2. Hover over the connection that you want to edit. To the right of the highlighted connection, click **Actions menu**, and select **Inspect**.
3. In the Inspect dialog, edit the connection details.

4. Click **Save**.

If you're editing an SSL connection to an Oracle Database and you need to use a new `cwallet.sso` file, in the **Client Wallet** field, click **Select** to browse for the `cwallet.sso` file. Ask your administrator for the location of the `cwallet.sso` file.

You must provide a unique **Connection Name**. If a connection with the same name already exists in your system, an error message is displayed. You can't see or edit the current password for your connection. If you need to change it, you must create a connection that uses the same password.

## Delete a Data Source Connection

You can remove a data source connection from Oracle Analytics Cloud. For example, you must delete a database connection and create a new connection when the database's password has changed.

If the connection contains any data sets, then you must delete the data sets before you can delete the connection.

1. Go to the Data page and select **Connections**.
2. Hover over the connection that you want to delete. To the right of the highlighted connection, click **Actions menu**, and select **Delete**.
3. Click **Yes**.

## Database Connection Options

When you specify connection details using the Create Connection dialog or Inspect dialog, some database types have extra options.

**General Options**

- When you create connections to Oracle Databases, you can connect in two ways using the **Connection Type** option:
  - **Basic** - Specify the **Host**, **Port**, and **Service Name** of the database.
  - **Advanced** - In the **Connection String** field specify the Single Client Access Name (SCAN) ID of databases running in a RAC cluster. For example:
    ```
    sales.example.com =(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=on)
    (FAILOVER=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.111)
    (PORT=1521)) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.222)
    (PORT=1521)) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.333)
    (PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=
    salesservice.example.com)))
    ```
- **Enable Bulk Replication** - If you're loading a data set for a project, then this option should be turned off and you can ignore it. This option is reserved for data analysts and advanced users for replicating data from one database to another database.

**Authentication Options**

| Option | Description |
|---|---|
| Always use these credentials | The login name and password you provide for the connection are always used and users aren't prompted to log in. |
| Require users to enter their own credentials | Prompt users to enter their own user name and password for the data source. Users required to log in see only the data that they have the permissions, privileges, and role assignments to see. |

## Share a Data Source Connection

You can assign access permissions to the data source connections that you create or administer.

1. From the Navigator on the Home page, click **Data**, then click **Connections**.
2. Hover over the connection that you want to share. To the right of the highlighted connection, click **Actions menu**, and select **Inspect**.
3. Click **Access**, and use the tabs to grant access:
   - **All** - Share the connection with individual users or roles.
   - **Users** - Share the connection with individual users.
   - **Roles** - Share the connection with application roles (for example, BI Consumer), so that all users with those roles can use the connection.

   When users next log in, they can use connections that you've shared to visualize data from this database.

# Create a Database Connection

You can create a connection to a database and use the connection to access data and build a data set.

1. On the Home page, click **Create**, then click **Connection**.
2. In the Create Connection dialog, click the icon for the connection type that you want to create a connection for (for example **Oracle Database**).
3. Enter a name for the connection, and then enter the required connection information, such as host, port, username, password, and service name.

   If you're creating an SSL connection to an Oracle Database, in the **Client Wallet** field, click **Select** to browse for the `cwallet.sso` file. Ask your administrator for the location of the `cwallet.sso` file.
4. (Optional) When you connect to some database types, you might have to specify the following authentication options on the Create Connection and Inspect dialogs:

| Option | Description |
|---|---|
| Always use these credentials | The login name and password you provide for the connection are always used and users aren't prompted to log in. |
| Require users to enter their own credentials | Prompt users to enter their own user name and password for the data source. Users required to log in see only the data that they have the permissions, privileges, and role assignments to see. |

5. Click **Save**.

   You can now begin creating data sets from the connection.
   You can't use remote connections to save a data set from a Data Flow.

# Connect to Oracle Applications

You can create connections to Oracle Applications to help you visualize, explore, and understand the data in your Oracle Fusion Applications with Oracle Transactional Business Intelligence and Oracle BI EE subject areas and analyses.

**Topics:**



## Create an Oracle Applications Connection

You can create a connection to Oracle Applications and use the connection to access your data.

1. On the Data page or Home page, click **Create**, then click **Connection**.
2. Click the **Oracle Applications** icon.
3. Enter the connection details.
4. For **URL**, enter the URL for Oracle Fusion Applications with Oracle Transactional Business Intelligence or Oracle BI EE.
5. Select an **Authentication** option.
   - Select **Always use these credentials**, so that the login name and password that you provide for the connection are always used and users aren't prompted to log in.
   - Select **Require users to enter their own credentials** when you want to prompt users to enter their user name and password to use the data from the Oracle Applications data source. Users must log in to access only the data that they have the permissions, privileges, and role assignments to access.
6. Save the details.

   You can now create data sets from the connection.

The connection is visible only to you (the creator), but you can create and share data sets for it.

## Edit an Oracle Applications Connection

You can edit an Oracle Applications connection. For example, you must edit a connection if your system administrator changed the Oracle Applications login credentials.

1. In the Data page, click **Connections**.
2. In the Edit Connection dialog, edit the connection details. Note that you can't see or edit the password that you entered when you created the connection. If you need to change the connection's password, then you must create a new connection.
3. Click **Save**.

## Delete an Oracle Applications Connection

You can delete an Oracle Applications connection.

1. Delete any data sets that use the connection you need to delete.
2. In the Data page, click **Connections**.
3. Click **Yes**.

# Create a Connection to Oracle Essbase

You can create a connection to Oracle Essbase 11g and use the connection to access source data.

1. Click **Create**, and then click **Connection**.
2. Click **Oracle Essbase**.
3. Enter the connection details.
4. For **DSN** (data source name), enter the agent URL for your data source.

   If you want to connect to an Oracle Essbase 11g database, enter the hostname and agent port number on which Oracle Essbase is running. Use the format:

   *`hostname`*`:`*`port`*

   For example: `essbase.example.com:1423`

   The default port is 1423.

   Your Essbase administrator must open agent port 1423 and server ports in the range 30000-34000 to allow the connection.
5. For **Username** and **Password**, enter user credentials with access to the Essbase data source.
6. Select the **Authentication** option:

- **Always use these credentials**: The username and password you provide for the connection are always used. Users aren't prompted to sign in to access the data available through this connection.
- **Require users to enter their own credentials**: Users are prompted to enter their own username and password if they want access to this data source. Users see only the data that they have the permissions, privileges, and role assignments to see.

7. Save the details.

   You can now create data sets from the connection.

# Connect to Dropbox

You can create a connection to Dropbox and use the connection to access data.

1. If needed, set up an application in Dropbox:
   a. Sign into your Dropbox account, and then go to the Developer's Area.
   b. Click **Create app** to create and save an application.
   c. Open the application's Settings, paste the redirect URL provided by Oracle Analytics, and copy the App key and App secret.

      Read the Dropbox documentation for more information about how to perform these tasks.
2. On the Data or Home page, click **Create**, then click **Connection** to display the Create Connection dialog.
3. Click **Dropbox**.
4. Enter the connection details:

| Field | Description |
|---|---|
| Redirect URL | Confirm that the Dropbox application is open and its Settings area is displaying. Copy the URL in the **Redirect URL** field and paste it into the Dropbox application's **OAuth 2 Redirect URIs** field and then click **Add**. |
| Client ID | Go to the Dropbox application, locate the **App key** field, and copy the key value. Go to Oracle Analytics and paste this value into the **Client ID** field. |
| Client Secret | Go to the Dropbox application, locate the **App secret** field, click **Show** to reveal the secret, and copy the secret value. Go to Oracle Analytics and paste this value into the **Client Secret** field. |

5. Click **Authorize**. When prompted by Dropbox to authorize the connection, click **Allow**.

   The Create Connection dialog refreshes and displays the name of the Dropbox account and associated email account.
6. Save the details.

   You can now create data sets from the Dropbox connection.

# Connect to Google Drive or Google Analytics

You can create a connection to Google Drive or Google Analytics and use the connection to access data.

1. If needed, set up an application in Google:
   a. Sign into your Google account, and go to the Developer's Console.
   b. Create a project, then go to the API Manager Developers area of the Google APIs site and click **Create app** to create and save an application.
   c. Enable the application and create credentials for the application by accessing the Analytics API.
   d. Open the page displaying the credential information, and paste the redirect URL provided by Oracle Analytics, and copy the Client ID and Client secret.

      Read the Google documentation for more information about how to perform these tasks.
2. On the Data or Home page, click **Create**, then click **Connection** to display the Create Connection dialog.
3. Click **Google Drive** or **Google Analytics**.
4. Enter connection details.

| Field | Description |
|---|---|
| Redirect URL | Confirm that the Google application is open and its Credentials area is displaying. Copy the URL in the **Redirect URL** field and paste it into the Google application's **Authorized redirect URIs** field. |
| Client ID | Go to the Google application's Credentials area, locate the **Client ID** field, and copy the key value. Go to Oracle Analytics and paste this value into the **Client ID** field. |
| Client Secret | Go to the Google application's credential information, locate the **Client secret** field and copy the secret value. Go to Oracle Analytics and paste this value into the **Client Secret** field. |

5. Click **Authorize**.
6. When prompted by Google to authorize the connection, click **Allow**.

   The Create Connection dialog refreshes and displays the name of the Google account, and its associated email account.
7. Save the details.

   You can now create data sets from the Google Drive or Google Analytics connection.

# Create Generic JDBC Connections

You can create generic JDBC connections to databases and use those connections to access data sources. For example, to connect to databases that aren't listed with the default connection types.

This method enables you to use drivers in a JDBC JAR file to connect to specific databases.

The JDBC driver version must match the database version. A version mismatch can lead to false errors during the data load process. Even using an Oracle database, if the version of the JDBC driver doesn't match that of the database, then you must download the compatible version of the JDBC driver from Oracle's website and place it in the `\lib` directory.

1. Confirm that you've copied the required JDBC driver's JAR file into the `Oracle Analytics Desktop\lib` directory.

   For example, `C:\Program Files\Oracle Analytics Desktop\lib`.

2. On the Data or Home page, click **Create**, then click **Connection**.
3. In the Create Connection dialog, locate and click the **JDBC** icon.
4. Enter the connection criteria. Note the following:
   - Avoid using instance-specific connection names such as host names, because the same connection can be configured against different databases in different environments (for example, development and production).
   - Check the driver documentation and the JAR file for specifying the URL of your JDBC data sources.
   - Find the driver class name in the JAR file or from wherever you downloaded the JAR file.
5. Click **Save**.

   You can now create data sets from the connection.

If you import a project containing a JDBC connection into an Oracle Analytics Desktop installation where the JDBC driver isn't installed, the import still works. However, the connection doesn't work when you try to run the project or Data Flow. You must recreate the JDBC connection and JDBC driver to a suitable data source.

# Create Generic ODBC Connections

You can create generic ODBC connections to databases and use the connections to access data sources. For example, to connect to databases and database versions that aren't listed with the default connection types.

You can only use generic ODBC connections to connect on Windows systems.

1. Confirm that the appropriate database driver is installed on your computer.

   You must have the required database driver installed on your computer to create an ODBC Data Source Name (DSN). If you need to install a database driver, use installation instructions provided by the organization that supplies the database driver.

2. Create the new ODBC data source in Windows.

   a. In Windows, locate and open the ODBC Data Source Administrator dialog.
   b. Click the **System DSN** tab, and then click **Add** to display the Create New Data Source dialog.
   c. Select the driver appropriate for your data source, and then click **Finish**.
   d. The remaining configuration steps are specific to the data source you want to configure.

      Refer to the documentation for your data source.

3. Create the generic ODBC data source.

   a. On the Data or Home page, click **Create**, then click **Connection**.
   b. In the Create Connection dialog, locate and click the **ODBC** icon.

c. Enter the connection criteria. Note the following:
   - **Name** - Any name that uniquely identifies the connection.
   - **DSN** - The name of the system DSN that you set up on your computer.

d. Click **Save**.

   You can now create data sets from the connection. See Create Data Sets from Databases.

If you import a project containing an ODBC connection into an Oracle Analytics Desktop installation where the ODBC DSN doesn't exist, and the ODBC driver isn't installed, the import still works. However, the connection doesn't work when you try to run the project or Data Flow. You must recreate the ODBC connection, and recreate the ODBC DSN, and ODBC driver to a suitable data source.

# Connect to Oracle Autonomous Data Warehouse

You can create a connection to Oracle Autonomous Data Warehouse and use the connection to access data.

Tutorial

1. Before you create connections to Oracle Autonomous Data Warehouse, you must obtain the client credentials zip file containing the trusted certificates that enable Oracle Analytics to connect to Oracle Autonomous Data Warehouse.

   See Download Client Credentials (Wallets) in *Using Oracle Autonomous Data Warehouse on Shared Exadata Infrastructure*.

   The credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Data Warehouse. The wallet file (for example, `wallet_ADWC1.zip`) that you upload must contain SSL certificates, to enable SSL on your Oracle Database Cloud connections.

2. To create a connection to Oracle Autonomous Data Warehouse:
   a. On the Home page, click **Create** then click **Connection**.
   b. Click **Oracle Autonomous Data Warehouse** to display the fields for the connection.
   c. Enter the connection details.
   d. For **Client Credentials** field, click **Select** to browse for the Client Credentials wallet file (for example, `wallet_ADWC1.zip`).

      The **Client Credentials** field displays the `cwallet.sso` file.
   e. Enter the **Username** and **Password**, and select a **Service Name** from the list.
   f. Save the details.

      You can now create data sets from the connection.

# Connect to Oracle Autonomous Transaction Processing

You can create a connection to Oracle Autonomous Transaction Processing and use the connection to access data.

1. Before you create connections to Oracle Autonomous Transaction Processing, you must obtain the client credentials zip file containing the trusted certificates that enable Oracle Analytics to connect to Oracle Autonomous Transaction Processing.

   See Download Client Credentials (Wallets) in *Using Oracle Autonomous Transaction Processing on Shared Exadata Infrastructure*.

   The credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Transaction Processing. The wallet file (for example, `wallet_SALESATP.zip`) that you upload must contain SSL certificates, to enable SSL on your Oracle Database Cloud connections.
2. To create a connection to Oracle Autonomous Transaction Processing:
   a. On the Home page, click **Create** then click **Connection**.
   b. Click **Oracle Autonomous Transaction Processing**.
   c. Enter the connection details.
   d. For **Client Credentials**, click **Select** to browse for the Client Credentials wallet file (for example, `wallet_SALESATP.zip`).

      The **Client Credentials** field displays the `cwallet.sso` file.
   e. Enter the **Username**, and **Password**, and select a **Service Name** from the list.
   f. Save the details.

      You can now create data sets from the connection.

# Connect to Oracle Talent Acquisition Cloud

You can create a connection to Oracle Talent Acquisition Cloud and use the connection to access data.

1. Click **Create** and then click **Connection**.
2. Click **Oracle Talent Acquisition** to display the fields for the connection.
3. Enter the connection details.
4. For **Host**, enter the URL for the Oracle Talent Acquisition data source.

   For example, if the Oracle Talent Acquisition URL is `https://example.taleo.net`, then the connection URL that you must enter is `https://example.taleo.net/smartorg/Bics.jss` .
5. Select an **Authentication** option.
   - Select **Always use these credentials**, so that the login name and password you provide for the connection are always used and users aren't prompted to log in.
   - Select **Require users to enter their own credentials** when you want to prompt users to enter their user name and password to use the data from the Oracle Talent Acquisition Cloud data source. Users are required to log in see only the data that they have the permissions, privileges, and role assignments to see.
6. Save the details.

   You can now create data sets from the connection.

# Connect to Snowflake Data Warehouse

You can create a connection to Snowflake Data Warehouse and use the connection to access data.

1. Click **Create**, and then click **Connection**.
2. Click **Snowflake Data Warehouse**.
3. Enter the connection details.
4. For **Hostname** enter the host account name for your data source.

   Use the format, for example:

   `<account>.snowflakecomputing.com`

   Where `account` is the Snowflake account name that you want to use to access the data.

   For example: `exampleaccountname.snowflakecomputing.com`.

   See format guidelines, `https://docs.snowflake.net/manuals/user-guide/connecting.html`.
5. For **Username** and **Password**, enter user credentials with access to the Snowflake data source.
6. For **Database Name**, enter the name of the database containing the schema tables and columns that you want to connect to.
7. For **Warehouse**, enter the name of the warehouse containing the database, schema tables and columns that you want to connect to. For example, `Example-WH`.
8. Save the details.

   You can now create data sets from the connection.

# Connecting to NetSuite

Connect to a NetSuite (NetSuite2.com) data source to visualize ERP and CRM data.

When you analyze data from a NetSuite2.com data source, you create a connection to store connection details. To obtain the connection details for your NetSuite application, go to the NetSuite Portal home page, click **Settings**, then click **Set up SuiteAnalytics Connect**, and copy the details displayed.

# 3
# Add Data Sets

Get data from data source connections or upload data from subject areas and spreadsheets and use it to build data sets for visualization and analysis.

**Topics**



## Create Data Sets from Databases

After you create database connections, you can use those connections to create data sets.

1. On the Home page, click **Create** and click **Data Set** to display the Create Data Set dialog.
2. If you haven't already selected a connection, click **Create Connection** and specify connection details for your data source.
3. In the Data Set editor, browse or search for and double-click a schema, and then choose the table that you want to use in the data set. When you double-click to select a table, a list of its columns is displayed.

   You can use breadcrumbs to quickly move back to the table or schema list.
4. In the column list, browse or search for the columns you want to include in the data set. You can use Shift-click or Ctrl-click to select multiple columns.

   Alternatively, you can select the **Enter SQL** option to view or modify the data source's SQL statement or to write a SQL statement.
5. Optionally use the **Data Access** option to specify how to refresh the data set.

   > **Note:**
   >
   > If you use the **Live** option (default), you'll see the data returned from the data source when you click **Refresh Data** on the Visualization canvas. If you use the **Automatic Caching** option, you'll see cached data. The **Automatic Caching** option is more suitable for small data sets.
6. Click **Add**. The View Data Source page is displayed.
7. In the View Data Source page you can optionally view the column properties and specify their formatting. The column type determines the available formatting options.

# Compose Data Sets from Subject Areas

You can build data sets from subject areas used in other Oracle products such as Oracle Fusion Applications with Oracle Transactional Business Intelligence.

You must create an Oracle Applications connection before you can create a subject area data set.

1. On the Home page, click **Create** and click **Data Set**. Click **Connection** and use the Create Connection dialog to specify the details for your data set.
2. In the Data Set editor, choose **Select Columns** to view, browse, and search the available subject areas and columns.
3. You can also optionally perform the following steps:
   - In the breadcrumbs click the **Add/Remove Related Subject Areas** option to include or exclude related subject areas. Subject areas are related when they use the same underlying business or logical model.
   - After you've selected columns, go to the Step editor at the top of the Data Set editor and click the **Filter** step to add filters to limit the data in the data set. After you've added filters, click **Get Preview Data** to see how the filters limit the data.
   - Click **Enter SQL** to display the logical SQL statement of the data source. View or modify the SQL statement in this field.

     If you edit the data source's logical SQL statement, then the SQL statement determines the data set and any of the column-based selection or specifications are disregarded.
   - Go to the Step editor at the top of the Data Set editor and click the last step in the Step editor to specify a description for the data set.
4. Before saving the data set, go to the **Name** field and confirm its name. Click **Add**.

   The Data Set page is displayed.
5. In the Data Set page you can optionally view the column properties and specify their formatting. The column type determines the available formatting options.

# Compose Data Sets from Analyses

You can build data sets from analyses created in other Oracle products such as Oracle Fusion Applications with Oracle Transactional Business Intelligence.

You must create an Oracle Applications connection before you can create an analysis data set.

1. On the Home page click **Create** and click **Data Set**. In the Create Data Set dialog, select **Create Connection** and use the Create Connection dialog to create the connection for your data set.
2. In the Data Set editor, select the **Select an Analysis** option to view, browse, and search the available analyses to use in your data set.
3. Double-click an analysis to use it for your data set.
4. You can also optionally perform the following steps:

- Click **Enter SQL** to display the SQL Statement of the data set. View or modify the SQL statement in this field.
- Click a column's gear icon to modify its attributes, like data type and whether to treat the data as a measure or attribute.
- Go to the Step editor at the top of the Data Set editor and click the last step in the Step editor to specify a description for the data set.

5. Before saving the data set, go to the **Name** field and confirm its name. Click **Add**.
6. In the Data Set page you can optionally view the column properties and specify their formatting. The column type determines the available formatting options.

# Create Data Sets from Essbase Cubes

After you create Essbase connections, you can use those connections to create data sets.

You must create the Essbase connection before you can create a data set for it.

1. On the Home page click **Create** and click **Data Set**.
2. In the Create Data Set dialog, select an existing Essbase connection.
3. In the Add Data Set page, double-click the Essbase cube that you want to use as a data set.

   Essbase cube details are displayed.
4. If required, you can edit the **Description**, and select an **Alias** value.

   If you select an alias value other than the default, then values from the selected alias table are displayed in visualizations that use this Essbase data set.
5. Click **Add** to save the Essbase cube.

# Add Spreadsheets as Data Sets

You can add a spreadsheet to use as a data set. You can browse for and upload spreadsheets from a variety of places, such as your computer, Google Drive, and Dropbox.

**Topics:**



## About Adding Spreadsheets or Other Data Files

You can create data sets from data stored in Microsoft Excel spreadsheets (XLSX and XLS ), CSV files, and TXT files. The maximum file size you can upload is 250 MB and the data column limit for a single file is 250 columns.

You must structure your Excel spreadsheets in a data-oriented way with no pivoted data. These rules apply for Excel tables:

- Tables must start in Row 1 and Column 1 of the Excel file.
- Tables must have a regular layout with no gaps or inline headings. An example of an inline heading is one that is repeated on every page of a printed report.
- Row 1 must contain the names of the columns in the table. For example, Customer Given Name, Customer Surname, Year, Product Name, Amount Purchased, and so on. In this example:
  - Column 1 has customer given names.
  - Column 2 has customer surnames.
  - Column 3 has year values.
  - Column 4 has product names.
  - Column 5 has the amount each customer purchased for the named product.
- The names in Row 1 must be unique. If two columns hold year values, then you must add a second word to one or both of the column names to make them unique. For example, if you have two columns named Year Lease, then you can rename the columns to Year Lease Starts and Year Lease Expires.
- Row 2 and greater must contain the data for the table, and those rows can't contain column names.
- Data in a column must be of the same type because it's often processed together. For example, the Amount Purchased column must contain only numbers (and possibly nulls), enabling it to be summed or averaged. The Given Name and Surname columns must contain text values because they might be concatenated, and you might need to split dates into months, quarters, or years.
- Data must be at the same granularity. A table can't contain both aggregations and details for those aggregations. For example, suppose that you have a sales table at the granularity of Customer, Product, and Year that contains the sum of Amount Purchased for each Product by each Customer by Year. In this case, you wouldn’t include invoice level details or daily summary values in the same table, because the sum of the Amount Purchased values wouldn't be calculated correctly. If you must analyze at the invoice level, the day level, and the month level, then you can do either of the following:
  - Have a table of invoice details: Invoice Number, Invoice Date, Customer, Product, and Amount Purchased. You can roll these up to the day, month, or quarter.
  - Have multiple tables, one at each granular level (invoice, day, month, quarter, and year).

## Add a Spreadsheet from Your Computer

You can create a data set from an Excel spreadsheet (XLSX or XLS), CSV file, or TXT file located on your computer.

You can't import an Excel spreadsheet that contains pivoted data. See About Adding Spreadsheets or Other Data Files.

1. On the Home page, click **Create**, and then click **Data Set**.

2. Click **File** and browse to select an XLSX or XLS (with unpivoted data), CSV, or TXT file.
3. Click **Open** to upload and open the selected spreadsheet.
4. Make any required name, description, or column attribute changes.
5. If you're uploading a CSV or TXT file, then in the **Separated By**, **Thousand Separator**, and **Decimal Separator** fields, confirm or change the default delimiters.

   If needed, choose Custom in the **Separated By** field and enter the character you want to use as the delimiter. In the CSV or TXT file, a custom delimiter must be one character. The following example uses a pipe (|) as a delimiter: Year|Product|Revenue|Quantity|Target Revenue| Target Quantity.
6. Click **Add** to create the data set.

## Add a Spreadsheet from Excel with the Smart View Plug-In

The Oracle Smart View Plug-In enables you to publish an XLSX or XLS spreadsheet, a CSV, or TXT file from Excel and use it as a data source.

When you import the spreadsheet and before you add it as a data source, you can modify column attributes, like data type and whether to treat the data as a measure or attribute.

Before you use the Smart View Plug-In, confirm you've done the following:

- Installed the latest version of Oracle Smart View for Office. To find the download, go to Oracle Smart View for Office. After you install Oracle Smart View for Office, be sure to restart all Microsoft Office applications.
- Confirmed that you've either an Excel spreadsheet in .XLSX or .XLS format, a .CSV file, or a .TXT file to use as the data source.
- Understand how the spreadsheet needs to be structured for successful import.

Follow these steps to publish an Excel spreadsheet, CSV, or TXT file to use it as a data source:

1. Open your Excel (.XLSX or XLS) spreadsheet, CSV, or TXT file in Microsoft Excel.

   If you're opening a .TXT file, follow the import steps for example, to specify the delimiter.
2. Click the **DV Desktop** tab.
3. If you're publishing a .XLSX or XLS file with pivot data, follow these steps:

   a. Select the upper-left numeric data cell, or select an area of data cells that you want to publish.

      Don't include grand totals when you select an area of data cells to publish.

   b. Click **Unpivot**.

   c. Click **OK**.
4. If required, format the new sheet content in Excel (for example, edit column heading names).
5. In the **DV Desktop** tab, click **Publish** to publish the new sheet.

If Oracle Analytics Desktop isn't running, it starts automatically. The spreadsheet data is displayed in the Data Set editor.

6. In the Data Set editor, make any required changes to **Name**, **Description**, or to column attributes.

   If you're uploading a CSV or TXT file, then in the **Separated By**, **Thousand Separator**, and **Decimal Separator** fields, confirm or change the default delimiters.

   If needed, choose Custom in the **Separated By** field and enter the character you want to use as the delimiter. In the CSV or TXT file, a custom delimiter must be one character. The following example uses a pipe (|) as a delimiter: Year|Product| Revenue|Quantity|Target Revenue| Target Quantity.

7. Click **Add**. If a data set exists with the same name, you're prompted to confirm that you want to overwrite it.

   You can update, re-pivot, or apply changes to the data set as needed.
   If you delete the Excel file created when un-pivoting, then the data set is no longer linked to the Excel file.

## Add a Spreadsheet from Windows Explorer

You can add a spreadsheet as a data source from within Windows Explorer.

Before you add a spreadsheet as a data source, do the following:

- Install the latest version of Oracle Smart View for Office. To find the download, go to Oracle Smart View for Office. After you install Oracle Smart View for Office, be sure to restart all Microsoft Office applications.
- Confirm that you're working with either an Excel spreadsheet in .XLSX or .XLS format or a .CSV file to use as the data source.
- For an Excel spreadsheet, ensure that it contains no pivoted data.
- Understand how the spreadsheet needs to be structured for successful import.

1. Open Windows Explorer and navigate to the spreadsheet file (.XLSX, .XLS, or .CSV) that you want to use as a data source.
2. Right-click the spreadsheet file icon.
3. Click **Open with** from the menu.
4. Select **Oracle Analytics Desktop**.
5. If a data set with the same name already exists, the Create or Reload Data Set window is displayed.
   - Click **Reload** and click **OK** to overwrite the existing data set with the same name.
     If you choose to reload, you don't need to follow the final step, and the new data set overwrites the existing data set.
   - Click **Create New**, and complete one of the following options:
     – Enter a new name, and click **OK**.
     – To save using an autogenerated data set name, click **OK**.
6. In the Data Set editor make any required changes to the Name, Description, or to column attributes.

If you're uploading a CSV file, then in the **Separated By**, **Thousand Separator**, and **Decimal Separator** fields, confirm or change the default delimiters.

If needed, choose Custom in the **Separated By** field and enter the character you want to use as the delimiter. For CSV files, a custom delimiter must be one character. The following example uses a pipe (|) as a delimiter: Year|Product|Revenue|Quantity|Target Revenue| Target Quantity.

7. Click **Add**.

## Add a Spreadsheet from Dropbox or Google Drive

You can upload Excel spreadsheets (XLSX or XLS), CSV files, and TXT files directly from Dropbox or Google Drive.

Before you start, you must set up a connection to Dropbox or Google Drive where your data files are stored. See Connect to Dropbox and Connect to Google Drive or Google Analytics.

- Confirm that the spreadsheet you want to use is either an Excel spreadsheet in .XLSX or .XLS format, a CSV file, or a TXT file.
- For an Excel spreadsheet, ensure that it contains no pivoted data.
- Understand how the spreadsheet needs to be structured for successful import.

1. In the Data page, click **Create** and click **Data Set**.
2. In the Create Data Set dialog, click the connection to Dropbox or Google Drive.
3. In the Data Set editor, search or browse the Dropbox or Google Drive directories and locate the spreadsheet that you want to use.

   You can use breadcrumbs to quickly move back through the directories.
4. Double-click a spreadsheet to select it.
5. Click **Add** to create the data set.

4

# Prepare Your Data Set for Analysis

Data preparation involves cleansing, standardizing, and enriching your data set before you analyze the data in a visualization canvas.

**Topics**



## Typical Workflow to Prepare Your Data Set for Analysis

Here are the common tasks for performing data preparation actions in the Prepare canvas.

| Task | Description | More Information |
|---|---|---|
| Apply enrichment recommendations | Enhance or add information to column data using the enrichment recommendations. | Accept Enrichment Recommendations |
| Apply transform recommendations | Modify column data using the transformation recommendations or available options. | Transform Data Using Column Menu Options |
| Change column properties | Change the column properties such as data type, number format. | Edit the Column Properties |
| Edit the data preparation script | Select and edit the changes applied to a column. | Edit the Data Preparation Script |

## About Data Preparation

You can transform and enrich the data that you're preparing for analysis.

Tutorial

When you create a project and add a data set to it, the data undergoes column level profiling that runs on a representative sample of the data. After profiling the data, you can implement transformation and enrichment recommendations provided for the recognizable columns in the data set. The following types of recommendations are provided to perform single-click transforms and enrichments on the data:

- Global positioning system enrichments such as latitude and longitude for cities or zip codes.
- Reference-based enrichments, for example, adding gender using on the person's first name as the attribute for make the gender decision.
- Column concatenations, for example, adding a column with the person's first and last name.
- Part extractions, for example, separating out the house number from the street name in an address.
- Semantic extractions, for example, separating out information from a recognized semantic type such as domain from an email address.
- Date part extractions, for example, separating out the day of week from a date that uses a month, day, year format to make the data more useful in the visualizations.
- Full and partial obfuscation or masking of detected sensitive fields.
- Recommendations to delete columns containing detected sensitive fields.

You can use and configure a wide range of data transformations from the column's **Options** menu. See Transform Data Using Column Menu Options.

When you transform data, a step is automatically added to the Preparation Script pane. A blue dot indicates that **Apply Script** hasn't been executed. After applying the script, you can make additional changes to the data set, or you can create a project, or click **Visualize** to begin your analysis.

As each transformation and enrichment change is applied to the data, you can review the changes. You can also compare the data changes with the original source data verify that the changes are correct.

The data transformation and enrichment changes that you apply to a data set affect all projects and data flows that use the same data set. When you open the project that shares the data set, a message appears indicating that the project uses updated data. You can create a data set from the original source that doesn't contain the data preparation changes. When you refresh the data in a data set, the preparation script changes are automatically applied to the refreshed data.

# Data Profiles and Semantic Recommendations

After creating a data set, the data set undergoes column-level profiling to produce a set of semantic recommendations to repair or enrich your data. These recommendations are based on the system automatically detecting a specific semantic type during the profile step.

There are various categories of semantic types such as geographic locations identified by city names, a specific pattern such as a credit card number or email address, a specific data type such as a date, or a recurring pattern in the data such as a hyphenated phrase.

**Topics**



## Semantic Type Categories

Profiling is applied to various semantic types.

Semantic type categories are profiled to identify:

- Geographic locations such as city names.
- Patterns such as those found with credit cards numbers or email addresses.
- Recurring patterns such as hyphenated phrase data.

## Semantic Type Recommendations

Recommendations to repair, enhance, or enrich the data set, are determined by the type of data.

Examples of semantic type recommendations:

- **Enrichments** - Adding a new column to your data that corresponds to a specific detected type, such as a geographic location. For example, adding population data for a city.
- **Column Concatenations** - When two columns are detected in the data set, one containing first names and the other containing last names, the system recommends concatenating the names into a single column. For example, a *first_name_last_name* column.
- **Semantic Extractions** - When a semantic type is composed of subtypes, for example a *us_phone* number that includes an area code, the system recommends extracting the subtype into its own column.
- **Part Extraction** - When a generic pattern separator is detected in the data, the system recommends extracting parts of that pattern. For example if the system detects a repeating hyphenation in the data, it recommends extracting the parts into separate columns to potentially make the data more useful for analysis.
- **Date Extractions** - When dates are detected, the system recommends extracting parts of the date that might augment the analysis of the data. For example, you might extract the day of week from an invoice or purchase date.
- **Full and Partial Obfuscation/Masking/Delete** - When sensitive fields are detected such as a credit card number, the system recommends a full or partial masking of the column, or even removal.

## Recognized Pattern-Based Semantic Types

Semantic types are identified based on patterns found in the data.

Recommendations are provided for these semantic types:

- Dates (in more than 30 formats)
- US Social Security Numbers (SSN)
- Credit Card Numbers
- Credit Card Attributes (CVV and Expiration Date)
- Email Addresses
- North American Plan Phone Numbers
- First Names (typical first names in the United States)
- Last Names (typical surnames in the United States)
- US Addresses

## Reference-Based Semantic Types

Recognition of semantic types is determined by loaded reference knowledge provided with the service.

Reference-based recommendations are provided for these semantic types:

- Country names
- Country codes
- State names (Provinces)
- State codes
- County names (Jurisdictions)
- City names (Localized Names)
- Zip codes

## Recommended Enrichments

Recommended enrichments are based on the semantic types.

Enrichments are determined based on the geographic location hierarchy:

- Country
- Province (State)
- Jurisdiction (County)
- Longitude
- Latitude
- Population
- Elevation (in Meters)

- Time zone
- ISO country codes
- Federal Information Processing Series (FIPS)
- Country name
- Capital
- Continent
- GeoNames ID
- Languages spoken
- Phone country code
- Postal code format
- Postal code pattern
- Phone country code
- Currency name
- Currency abbreviation
- Geographic top-level domain (GeoLTD)
- Square KM

## Required Thresholds

The profiling process uses specific thresholds to make decisions about specific semantic types.

As a general rule, 85% of the data values in the column must meet the criteria for a single semantic type in order for the system to make the classification determination. As a result, a column that might contain 70% first names and 30% “other”, doesn't meet the threshold requirements and therefore no recommendations are made.

# Accept Enrichment Recommendations

You can use the enrichment recommendations to enhance or add information to data.

You can upload or open an existing data set to modify the data using enrichment recommendations. After making the changes to the data set, you can create a project. You can add one or more data sets to the project and modify the data by using the enrichment recommendations.

If an enrichment recommendation adds information to data such as enhancing a zip code attribute column with the state name, a new column is added to the data set containing the name of the states associated with the zip codes. When you select a recommendation, the change is added to the Preparation Script. If you delete or undo the change, the recommendation is displayed once again as an available option in the Recommendation Panel.

If you don’t apply the Preparation Script and you close the project or the data set, you lose all the data changes you’ve performed.

1. Open a project or data set. If you're working with a project, go to the Prepare canvas. In the Metadata view data panel, select a column to enrich.

2. In the Recommendation Panel, select a recommendation to add the change to the Preparation Script.
3. Continue implementing enrichment recommendations on the data set.
4. In the Preparation Script Panel, click **Apply Script** to apply the data changes to the entire data set.

   If you're working with a project, click **Save** and click **Visualize** to review the enriched columns.

# Transform Data Using Column Menu Options

You can use column menu options to modify the data's format.

You can upload or open an existing data set to transform the data using column menu options. After making the changes to the data set, you can create a project or open an existing project and add the data set to the project.

The data transform changes update the column data using the selected option or add a new column to the data set. See Transform Recommendation Reference.

The list of available menu options for a column depends on the type of data in that column.

If you don't apply the transformation script and close the project or the data set, you lose all the data transform changes you've performed.

1. Open a project, and click **Prepare**. In the Preview data panel, select a column to transform.
2. Click **Options**, and select a transformation option.
3. In the step editor, update the fields to configure the changes. You can review the changes in the data preview table.
4. Click **Add Step** to apply the data changes, close the step editor, and add a step to the Preparation Script Panel.
5. Continue implementing data transform changes in the data set.
6. Click **Apply Script** in the Preparation Script Panel to apply the data transform changes to the entire data set.
7. (Optional) Click **Save**, and then click **Visualize** to see the transformed columns.

This example shows a Gender column with the data values F, f, M and m. You can use Group in the column menu options to change the column data to use Female and Male.

- Open the project or data set with the Gender column. If you're working on a project, go the Prepare canvas.
- Select the **Gender** column and click **Options**, then click **Group**.
- In the group editor, change the new column name to `Gender_Fix`.
- Change the name of Group 1 to `Female` and select **F** and **f**.
- Click **(+) Group** to add a new group and change the name to `Male`.
- Click **Add all** to select the remaining values in the Gender columns that should represent men.

- Click **Add Step** to include the new column Gender_Fix and standardized gender groups in the data set.
- In the Preparation Script pane, click **Apply Script** to apply the data changes.

# Convert Text Columns to Date or Time Columns

You can convert any text column to a date, time, or timestamp column.

For example, you can convert an attribute text column to a true date column.

1. Open the project or the data set that includes the column you want to convert. Confirm that you're working in the Prepare canvas.
2. Hover over the column that you want to convert.
3. Click **Options**, and select a conversion option (for example, **Convert to Number**, **Convert to Date**).

   You can also do this from the Data Sets page when you're editing a data set.
4. To further refine the format, select the column, and use the options on the properties pane.
5. If you want to change the Source Format's default value then click **Source Format** and select a format. For example, 2017.01.23, 01/23/2017, Mon, Jan 23, 2017, or Mon, Jan 23, 2017 20:00:00.

   The **Source Format** field automatically displays a suggested format based on the input column text. However, if the **Source Format** field doesn't display a suggested format (for example, for Sat 03/28 2017 20:10:30:222), then you can enter a custom format.
6. Click **Custom** if you need to enter your own format into the field at the bottom of the Convert to Date/Time dialog.

   The custom format you enter must be in a format recognized by Oracle Business Intelligence before conversion. If you enter a custom format that isn't recognized, an error message is displayed.
7. The **Hide Source Element** is selected by default and hides the original source column after conversion. If you deselect this option, the original column is displayed next to the converted column after conversion.
8. Click **Convert** to convert the text column into a date or time column.

   The changes you make apply to all projects using the data source with a modified date or time column.

# Adjust the Display Format of Date or Time Columns

You can adjust the display format of a date or a time column by specifying the format and the level of granularity.

For example, you might want to change the format of a transaction date column (which is set by default to show the long date format such as November 1, 2017) to display instead the International Standards Organization (ISO) date format (such as 2017-11-01). You might want to change the level of granularity (for example year, month, week, or day).

1. Open the project or the data set that includes the date and time column that you want to update. If you're working in a project, then confirm that you're working in the project's Prepare canvas.
2. Click the date or time column you want to edit.

   For example, click a date in the data elements area of the Data Panel, or click or hover over a date element on the main editing canvas.
3. If you're working in the main editing canvas, adjust the format by doing one of the following:
   - Click **Options**, then **Extract** to display a portion of the date or time (for example, the year or quarter only).
   - Click **Options**, then **Edit** to display an Expression Editor that enables you to create complex functions (for example, with operators, aggregates, or conversions).
   - In the properties pane, click the **Date/Time Format** tab, and use the options to adjust your dates or times (for example, click **Format**) to select from short, medium, or long date formats, or specify your own format by selecting **Custom** and editing the calendar string displayed.
4. If you're working in the data elements area of the Data Panel, adjust the format by doing one of the following:
   - If you want to display just a portion of a calendar column (for example, the year or quarter only), then select and expand a calendar column and select the part of the date that you want to display in your visualization. For example, to only visualize the year in which orders were taken, you might click Order Date and select Year.
   - In the properties pane, click the **Date/Time Format** tab, and use the options to adjust your dates or times.
5. If you're working in table view, select the column header and click **Options**, then in the properties pane click Date/Time Format to display or update the format for that column.

# General Custom Format Strings

You can use these strings to create custom time or date formats.

The table shows the general custom format strings and the results that they display. These allow the display of date and time fields in the user's locale.

| General Format String | Result |
|---|---|
| [FMT:dateShort] | Formats the date in the locale's short date format. You can also type `[FMT:date]`. |
| [FMT:dateLong] | Formats the date in the locale's long date format. |
| [FMT:dateInput] | Formats the date in a format acceptable for input back into the system. |
| [FMT:time] | Formats the time in the locale's time format. |
| [FMT:timeHourMin] | Formats the time in the locale's time format but omits the seconds. |
| [FMT:timeInput] | Formats the time in a format acceptable for input back into the system. |

| General Format String | Result |
|---|---|
| [FMT:timeInputHourMin] | Formats the time in a format acceptable for input back into the system, but omits the seconds. |
| [FMT:timeStampShort] | Equivalent to typing `[FMT:dateShort] [FMT:time]`. Formats the date in the locale's short date format and the time in the locale's time format. You can also type `[FMT:timeStamp]`. |
| [FMT:timeStampLong] | Equivalent to typing `[FMT:dateLong] [FMT:time]`. Formats the date in the locale's long date format and the time in the locale's time format. |
| [FMT:timeStampInput] | Equivalent to `[FMT:dateInput] [FMT:timeInput]`. Formats the date and the time in a format acceptable for input back into the system. |
| [FMT:timeHour] | Formats the hour field only in the locale's format, such as 8 PM. |
| YY or yy | Displays the last two digits of the year, for example `11` for 2011. |
| YYY or yyy | Displays the last three digits of the year, for example, `011` for 2011. |
| YYYY or yyyy | Displays the four-digit year, for example, `2011`. |
| M | Displays the numeric month, for example, 2 for February. |
| MM | Displays the numeric month, padded to the left with zero for single-digit months, for example, `02` for February. |
| MMM | Displays the abbreviated name of the month in the user's locale, for example, `Feb`. |
| MMMM | Displays the full name of the month in the user's locale, for example, `February`. |
| D or d | Displays the day of the month, for example, `1`. |
| DD or dd | Displays the day of the month, padded to the left with zero for single-digit days, for example, `01`. |
| DDD or ddd | Displays the abbreviated name of the day of the week in the user's locale, for example, `Thu` for Thursday. |
| DDDD or dddd | Displays the full name of the day of the week in the user's locale, for example, `Thursday`. |
| DDDDD or ddddd | Displays the first letter of the name of the day of the week in the user's locale, for example, T for Thursday. |
| r | Displays the day of year, for example, 1. |
| rr | Displays the day of year, padded to the left with zero for single-digit day of year, for example, 01. |
| rrr | Displays the day of year, padded to the left with zero for single-digit day of year, for example, 001. |
| w | Displays the week of year, for example, 1. |
| ww | Displays the week of year, padded to the left with zero for single-digit weeks, for example, 01. |
| q | Displays the quarter of year, for example, 4. |
| h | Displays the hour in 12-hour time, for example 2. |
| H | Displays the hour in 24-hour time, for example, `23`. |
| hh | Displays the hour in 12-hour time, padded to the left with zero for single-digit hours, for example, `01`. |

| General Format String | Result |
|---|---|
| HH | Displays the hour in 24-hour time, padded to the left with zero for single digit hours, for example, 23. |
| m | Displays the minute, for example, 7. |
| mm | Displays the minute, padded to the left with zero for single-digit minutes, for example, 07. |
| s | Displays the second, for example, 2.<br>You can also include decimals in the string, such as s.# or s.00 (where # means an optional digit, and 0 means a required digit). |
| ss | Displays the second, padded to the left with zero for single-digit seconds, for example, 02.<br>You can also include decimals in the string, such as ss.# or ss.00 (where # means an optional digit, and 0 means a required digit). |
| S | Displays the millisecond, for example, 2. |
| SS | Displays the millisecond, padded to the left with zero for single-digit milliseconds, for example, 02. |
| SSS | Displays the millisecond, padded to the left with zero for single-digit milliseconds, for example, 002. |
| tt | Displays the abbreviation for ante meridiem or post meridiem in the user's locale, for example, pm. |
| gg | Displays the era in the user's locale. |

# Create a Bin Column When You Prepare Data

Binning a measure creates a new column based on the value of the measure.

You can assign a value to the bin dynamically by creating the number of equal-sized bins or by explicitly specifying the range of values for each bin. You can create a bin column based on a data element.

1. Open a project, and click **Prepare**. In the Preview data panel, select a column that you want to modify using the bin option.
2. Click **Options** for the selected column, and select **Bin**.
3. In the Bin step editor, specify the options for the bin column.
   - Enter a number or use the arrows to increment or decrement the number of bins.
   - Based on your selection in the **Method** field, the range and count of the bins are updated.
     - In the **Manual** method, you select the boundary (that is, minimum and maximum) of each bin. You can also change the default name of each bin.
     - In the **Equal Width** method, the boundary of each bin is the same, but the count differs. Based on your selection in the **Bin Labels** field, the bin column labels are updated.
     - In the **Equal Height** method, the height of each bin is the same or very slightly different but the range is equal.

- If you select the **Equal Width** method, click to select a dimension (that is, an attribute data element) on which to apply the bin.

4. Click **Add Step** to apply the data changes, close the step editor, and add a step to the Preparation Script Panel.

# Edit the Column Properties

You can view and edit the properties of each column in the project's Prepare canvas.

Column property changes aren't affected by data transform changes. For example, if you've updated the name of a column after you use a data transform change on the same column, the name of the column is updated automatically.
You must select the **Result** step in the Preparation Script Panel to edit column properties in the properties pane or metadata view.

1. Confirm that you're working in the Prepare canvas of a project.

   If you've added more than one data set to the project, go to the tabs at the bottom of the window and select a data set in which you want to edit the column properties.

2. In the Preparation Script Panel, select the **Results** step and do one of the following:

   a. In the Metadata view data panel or Data Panel, select a column.

      - In the properties pane, use the General tab to change the selected column properties such as Treat As and Data Type.

   b. Click the view selector list on the Metadata view data panel toolbar and select metadata view, then click a row in the Metadata view data panel.

      You see the properties that you can edit (such as Data Element, Data Type, and Aggregation) in the Preview data panel. Each row represents a column in the data set.

      - Click a data element and click a property you want to edit, then select an option. For example, you might change the aggregation type from Sum to Average.
      - Toggle the checkbox in the Hidden column to hide and unhide a data column in the data set.

   For each property change, a step is added to the Preparation Script Panel, alongside any data transformation changes you have applied to the data set using the column's **Options** menu or the Recommendation Panel.

3. Click **Apply Script** in the Preparation Script Panel to apply the property changes to the data set.

The Hidden columns icon at the bottom of the window shows the number of hidden columns in the data set and is available for all views. You can click the Hidden columns icon to unhide one or all hidden columns.

You can also use the column menu options to hide a column. But you can only use the metadata view or Hidden columns icon to unhide a column.

# Edit the Data Preparation Script

You can edit the data transformation changes added to the Preparation Script.

Both before and after you've executed **Apply Script**, you can edit the data transformation steps. The edit option isn't available for all types of transform steps. If you're editing the steps after executing **Apply Script**, you must re-apply the script to the entire data set. If you don't save the updates to a step and navigate to another step, a warning message is displayed indicating that you haven't saved the changes.

The updates to the columns are applied only to the data set and not to the visualization. To ensure that you see the most up-to-date data, on the Visualize canvas, click **Menu**, then click **Refresh Data**.

1. Open a project or data set.

   If you're working with a project, go to the Prepare canvas and click the data set you want to edit.
2. Select a step in the Preparation Script pane and click **Edit Transform**.
3. Select a transform step in the Preparation Script pane and click the pencil icon or **Edit Transform**.

   If a step isn't editable, a message is displayed when you hover over the pencil icon that editing for the transform step has been disabled.
4. In the step editor, update the fields to edit the data transform changes that are applied to the columns.
5. Click **OK** to update the column and close the step editor.
6. Click **Apply Script** in the Preparation Script pane to apply the data transform changes to the entire data set.

# Add Columns in Data Preparation

You can create new columns to enhance your data.

1. Open a project or data set.

   If you're working with a project, go to the Prepare canvas and click the data set you want to edit.
2. In the Preparation Script pane, click **Add Preparation Step**.

   Alternatively, to create a column similar to an existing column, click **Options** on the column, and click **Create**.
3. Use the Create Column pane to specify the column details.

   Use the function picker to use operators and mathematical functions in your column. For example, you might specify `COLUMN1+COLUMN2` to calculate the sum of values in COLUMN1 and COLUMN2.
4. Click **Add Step**.

# Transform Data Using Replace

Transform data in a column using a simple pattern match or a regular (regex) expression. For example, in a State column, you might change CA to California.

You can use any Perl Compatible Regular Expression (PCRE), which are also referred to as regex expressions. Transforms are applied to every applicable row in a data set.

1. In your project (on the Prepare pane) or data set editor, select the text column that you want to transform.
2. Click **Options**, and then click **Replace**.
3. In the Replace step editor, specify how you'd like to match values.

   For example, click **Use regular expression** to match using a complex regular expression (regex), or click **Match partial values** to change part of a value.
4. In the **String to replace** field and **New string** field, specify the literal text or regular expressions to search and replace.

   For example, to change "2553 Bolman Court" to "2553 #Bolman# #Court#", select **Use regular expression**, enter `([A-Za-z]+)` in the **String to replace** field, and enter `#$1#` in the **New String** field.

   When you complete both fields, you'll see the transformed data in the preview pane.
5. Click **Add Step** to save the details and add the transform step to the Preparation Script pane.

   You'll apply the transformation when you click **Apply Script**.

## Example Replace Transforms Using Regular Expressions

Here're just a few examples of using regular (regex) expressions to transform data.

In these examples, the Search Expression column shows what you'd enter in the **String to replace** field, and the Replace Value shows what you'd enter in the **New string** field.

**Table 4-1 Example replace transforms**

| Search Expression | Replace Value | Original Text | Replaced Text | Notes |
|---|---|---|---|---|
| `@([a-z]+)(?=\.[a-z]{3})` | @example | MichelePFalk@yahoo.com | MichelePFalk@example.com | This example replaces domain details in email addresses. |

**Table 4-1 (Cont.) Example replace transforms**

| Search Expression | Replace Value | Original Text | Replaced Text | Notes |
|---|---|---|---|---|
| `^Gray\|Grey$` | Silver | Grey<br>Gray<br>Graystone | Silver<br>Silver<br>Graystone | The ^$ characters mean only look for entire string matches. The vertical bar is the regular expression for OR, so in this case the regular expression looks for either "Gray" or "Grey" and replace with "Silver".<br>The string Graystone isn't transformed because the regular expression is looking for entire value matches only. |
| `\d+` | 9999 | 8398 Park Street<br>123 Oracle Parkway | 9999 Park Street<br>9999 Oracle Parkway | This regular expression looks for one" \d" or more "+" digits and replaces them with "9999". The replace also works when the original text has only three digits. |
| `([A-z]+)(\d+)` | $2 | UA101654<br>US829383 | 101654<br>829383 | This regular expression is looking for one "[A-z]" or more "+" consecutive letters followed by one" \d" or more "+" digits - each of the two expressions is bounded by parentheses, which captures two groups - the first one ([A-z]) and the second one (\d+) - these groups are numbered automatically and can then be used in the replacement by using the dollar sign for the second group, for example, "$2". |
| `([A-z]+)(\d+)` | Postal Code: $2 | UA101654<br>US829383 | Postal Code: 101654<br>Postal Code: 829383 | This example uses the same match expression as the previous example, except that it shows how to insert your own replacement text in addition to a group. Text can be inserted before and after a recalled match group. |

5

# Manage Data and Data Sets

This topic describes the functions available to manage the data sets that you created from data sources.

**Topics:**



## Typical Workflow to Manage Added Data

Here are the common tasks for managing the data added from data sources.

| Task | Description | More Information |
|---|---|---|
| Refresh data | Refresh data in the data set when newer data is available. Or refresh the cache for Oracle Applications and databases if the data is stale. | Refresh Data in a Data Set |
| Update details of added data | Inspect and update the properties of the added data. | Update Details of Data that You Added |
| Manage data sets | See the available data sets and examine or update a data set's properties. | Manage Data Sets |
| Renaming a data set | Rename a data set listed on the data sets page. | Rename a Data Set |
| Duplicate data sets | Duplicate a data set listed on the data sets page. | Duplicate Data Sets |
| Blend data | Blend data from one data source with data from another data source. | Blend Data that You Added<br>About Mismatched Values in Blended Data |

# Manage Data Sets

You can update and delete the data that you added from various data sources.

You can use the Data Sets page to examine data set properties, and free up space by deleting data sets that you no longer need.

1. Go to the **Data** page, then **Data Sets** section.
2. Click the **Actions menu** of a data set or right-click the data set that you want to manage, and click **Inspect**.
3. Optionally, use the **Inspect** option to review data set columns and change the data set properties. For example, you can change the Name and Description.
4. Optionally, use the **Inspect** option to change whether to treat data set columns as measures or attributes.

   You can't change how a column is treated if it's already matched to a measure or attribute in the data model.
5. Optionally, use the **Inspect** option to change the Data Access for a database table. The default is **Live** because database tables are typically large and shouldn't be copied to cache. If your table is small, then select **Automatic Caching** and the data is copied into the cache if possible. If you select **Auto**, then you have to refresh the data when it's stale.
6. Optionally, download a data set created from a Microsoft Excel file by right-clicking the data set and selecting **Download File**. Note that the columns in the download match the columns in the file that you originally uploaded. Any derived columns that you added in the Visualize editor's Prepare canvas aren't included in the data set download.
7. Optionally, update data for a data set created from a Microsoft Excel file or Oracle Applications by right-clicking the data set and selecting **Reload Data**.

# Types of Data You Can Refresh

After you add data, the data might change, so you must refresh the data from its source.

Rather than refreshing a data set, you can replace it by loading a new data set with the same name as the existing one. However, replacing a data set can be destructive and is discouraged. Don't replace a data set unless you understand the consequences:

- Replacing a data set breaks projects that use the existing data set if the old column names and data types aren't all present in the new data set.
- Any data wrangling (modified and new columns added in the data stage) is lost and projects using the data set are likely to break.

### Databases

For databases, the SQL statement is rerun and the data is refreshed.

### CSV or TXT

To refresh a CSV or TXT file, you must ensure that it contains the same columns that are already matched with the date source. If the file that you reload is missing some

columns, then you'll see an error message that your data reload has failed due to one or more missing columns.

You can refresh a CSV or TXT file that contains new columns, but after refreshing, the new columns are marked as hidden and don't display in the Data Panel for existing projects using the data set.

**Excel**

To refresh a Microsoft Excel file, you must ensure that the newer spreadsheet file contains a sheet with the same name as the original one. In addition, the sheet must contain the same columns that are already matched with the data source. If the Excel file that you reload is missing some columns, then you'll see an error message that your data reload has failed due to one or more missing columns.

You can refresh an Excel file that contains new columns, but after refreshing, the new columns are marked as hidden and don't display in the Data Panel for existing projects using the data set. To resolve this issue, use the **Inspect** option of the data set to show the new columns and make them available to existing projects.

**Oracle Applications**

You can reload data and metadata for Oracle Applications data sources, but if the Oracle Applications data source uses logical SQL, reloading data only reruns the statement, and any new columns or refreshed data won't be pulled into the project. Any new columns come into projects as hidden so that existing projects that use the data set aren't affected. To be able to use the new columns in projects, you must unhide them in data sets after you refresh. This behavior is the same for file-based data sources.

# Refresh Data in a Data Set

You can refresh data in a data set from all source types such as databases, files, and Oracle Applications.

1. If you're in a project:
   - On the Visualize canvas, click **Refresh Data** to ensure that you see the most up-to-date data (by re-executing the visualization queries for all views in your project).
2. If you're on the **Data Sets** panel on the Data page, click **Actions menu** next to a data set and select **Reload Data**.

   After you reload a data set, in each project that uses that data set, on the Visualize canvas, click **Menu**, and select **Refresh Data**.

# Update Details of Data that You Added

After you add data, you can inspect its properties and update details such as the name and description.

1. Go to the Data page and select **Data Sets**.
2. Select the data set whose properties you want to update and click the **Actions menu** or right-click, then select **Inspect**.
3. View the properties in the following tabs and modify them as appropriate:

- General
- Data Elements

# Delete Data Sets

You can delete data sets when you need to free up space on your system.

Deleting a data set permanently removes it and breaks any projects that use the deleted data set. You can't delete subject areas that you've included in projects. Deleting data differs from removing a data set from a project.

1. Go to the Data page and select **Data Sets**.
2. Select the data set you want to delete and click the **Actions Menu** or right-click, then select **Delete**.

# Rename a Data Set

Renaming a data set helps you to quickly search and identify it in the data set library.

Even if you change the name of a data set, that change doesn't affect the reference for the project; that is, the project using the specific data set continues to work.

1. Go to the Data page and select **Data Sets**.
2. Select a data set and click the **Actions menu** or right-click, then select **Open**.
3. Click **Edit Data Set** on the Results toolbar.
4. Select the last step and go to the **Name** field, then change the value.
5. Click **Save**.

If a data set with the same name already exists in your system, an error message is displayed. Click **Yes** to overwrite the existing data set (with the data set whose name you're changing) or cancel the name change.

# Duplicate Data Sets

You can duplicate an uploaded data set that is listed in the Data Sets page to help you further curate (organize and integrate from various sources) data in projects.

For example, suppose an accounts team creates a specific preparation of a data set, and a marketing team wants to prepare the same data set but in a different way. The marketing team duplicates the data set for their own purposes.

1. Go to the Data page and select **Data Sets**.
2. Select a data set that you want to duplicate and click the **Actions menu** or right-click, then select **Duplicate**.
   - The duplication happens immediately.
   - The default name of the duplicated data set is `<Data set>Copy`.
   - If the data set name already exists, the new name is set to `<Data set>Copy#` in sequential order based on available names.
   - You can rename the duplicate data set by editing it in the Inspector dialog.
   - The user that duplicates the data set becomes the owner of the new data set.

- Any user who can view a data set can also duplicate the data set.
- All properties on the new data set, unless specifically stated, are reset (as if it's a new data set). For example, ACL, certified, indexed, custom-attributes.
- Data preparation changes made on the source are retained in the new data set.
- Conformance rules on the source are retained in the new data set.

# Blend Data that You Added

You might have a project where you added multiple data sets. You can blend data from one data set with data from another data set.

Video

For example, Data Set A might contain new dimensions that extend the attributes of Data Set B. Or Data Set B might contain new facts that you can use alongside the measures that already exist in Data Set A.
When you add more than one data set to a project, the system tries to find matches for the data that's added. It automatically matches external dimensions where they share a common name and have a compatible data type with attributes in the existing data set.

Data sets that aren't joined are divided by a line in the Data Panel of the project.
If the project includes multiple data sets and if any aren't joined, then you'll see restrictions between data elements and visualizations. For example, you can't use the data elements of a data set in the filters, visualizations, or calculations of another data set if they're not joined. If you try to do so, you see an error message. You can match data elements of data sets that aren't joined in the Data Diagram of a project, or you can create individual filters, visualizations, or calculations for each data set.

You can specify how you want the system to blend your data.

1. Add one or multiple data sets to your project. Confirm that you're working in the Prepare canvas.
2. Go to the tabs at the bottom of the Prepare canvas and click **Data Diagram**. Alternatively, in the Data Panel, right-click and select **Data Diagram**.
3. Click the number along the line that connects the external source to the newly loaded source to display the Connect Sources dialog.
4. In the Connect Sources dialog, make changes as necessary.
    a. To change the match for a column, click the name of each column to select a different column from the data sets.

        If columns have the same name and same data type, then they're recognized as a possible match. You can customize this and specify that one column matches another by explicitly selecting it even if its name isn't the same. You can select only those columns with a matching data type.

    b. Click **Add Another Match**, and then select a column from the data sets to match.
    c. For a measure that you're uploading for the first time, specify the aggregation type such as *Sum* or *Average*.
    d. Click the **X** to delete a match.

5. Click **OK** to save the matches.

# About Mismatched Values in Blended Data

In some cases when the rows of data that you expect to see in a data set are missing, then you must specify which data set to use for data blending.

Sometimes rows of data are missing when your project includes data from two data sets that contain a mixture of attributes and values, and there are match values in one source that don't exist in the other.

Suppose we have two data sets (Source A and Source B) with slightly different rows, as shown in the following image. Note that Source A doesn't include IN-8 and Source B doesn't include IN-7.

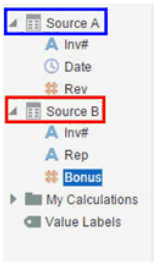
**T1 - Source A**

| Inv# | Date | Rev |
|---|---|---|
| IN-1 | 1/1/2015 | 100.00 |
| IN-2 | 1/1/2015 | 200.00 |
| IN-3 | 1/1/2015 | 300.00 |
| IN-4 | 1/2/2015 | 400.00 |
| IN-5 | 1/2/2015 | 500.00 |
| IN-6 | 1/2/2015 | 600.00 |
| IN-7 | 1/3/2016 | 800.00 |

**T2 - Source B**

| Inv# | Rep | Bonus |
|---|---|---|
| IN-1 | Billie | 1.00 |
| IN-2 | Joe | 2.00 |
| IN-3 | Kim | 3.00 |
| IN-4 | Billie | 4.00 |
| IN-5 | Joe | 5.00 |
| IN-6 | Kim | 6.00 |
| IN-8 | Mika | 8.00 |

The following results are displayed if you select the **All Rows** data blending option for Source A and select the **Matching Rows** data blending option for Source B. Because IN-7 doesn't exist in Source B, the results contain null Rep and null Bonus.




**T4 - SA All SB Match -- no Mika**

| Date | Rep | Rev | Bonus |
|---|---|---|---|
| 1/1/2015 | Billie | 100.00 | 1.00 |
| | Joe | 200.00 | 2.00 |
| | Kim | 300.00 | 3.00 |
| 1/2/2015 | Billie | 400.00 | 4.00 |
| | Joe | 500.00 | 5.00 |
| | Kim | 600.00 | 6.00 |
| 1/3/2016 | (null) | 800.00 | (null) |

**T5 - SA All SB Match -- no Mika**

| Date | Rev | Bonus |
|---|---|---|
| 1/1/2015 | 600.00 | 6.00 |
| 1/2/2015 | 1,500.00 | 15.00 |
| 1/3/2016 | 800.00 | (null) |

**T6 - SA All SB Match -- no Mika**

| Rep | Rev | Bonus |
|---|---|---|
| Billie | 500.00 | 5.00 |
| Joe | 700.00 | 7.00 |
| Kim | 900.00 | 9.00 |
| (null) | 800.00 | (null) |

The following results are displayed if you select the **Matching Rows** data blending option for Source A and select the **All Rows** data blending option for Source B. Because IN-8 doesn't exist in Source A, the results contain null Date and null Revenue.


The visualization for Source A includes Date as an attribute, and Source B includes Rep as an attribute, and the match column is Inv#. Under dimensional rules, you can't use these attributes with a measure from the opposite table unless you also use the match column.

There are two settings for blending tables that contain both attributes and measures. These are set independently in each visualization based on what columns are used in the visualization. The settings are **All Rows** and **Matching Rows** and they describe which source rows the system uses when returning data to be visualized.

The system automatically assigns data blending according to the following rules:

- If the visualization contains a match column, then the system sets sources with the match column to **All Rows**.
- If the visualization contains an attribute, then the system sets its source to **All Rows** and sets the other sources to **Matching Rows**.
- If attributes in the visualization come from the same source, then the system sets the source to **All Rows**, and sets the other sources to **Matching Rows**.
- If attributes come from multiple sources, then the system sets the source listed first in the project's elements panel to **All Rows** and sets the other sources to **Matching Rows**.

# Change Data Blending in a Project

You can change data blending in a project with multiple data sets. Data blending specifies which data set takes precedence over the other.

1. Select a visualization on the canvas that uses more than one data set and in the properties pane click **Data Sets**.
2. To change the default blending, click **Data Blending**, and select either **Auto** or **Custom**.

   If you choose **Custom**, you can set the blending to either **All Rows** or **Matching Rows**.

   - You must assign at least one source to **All Rows**.
   - If both sources are **All Rows**, then the system assumes that the tables are purely dimensional.
   - You can't assign both sources to **Matching Rows**.

# View and Edit Object Properties

You can view and edit the properties of standalone objects such as projects, data sets, connections, and data flows.

For example, you can check and modify the access permissions to specify the users who can change or read a data set, or see the source and target data sets for the data flow.

You can use the object inspector in the Home, Data, Catalog, and other top-level pages to view and edit the properties of an object. Based on the object's level, the properties also provide references to other objects, such as lower level objects that are part of the object that you're inspecting and other standalone objects that are referenced or used by that object. For example, a project property provides a list of data sets that are included in the project. The properties of lower level objects aren't part of the top-level object's inspector (such as data set properties), so they're not displayed as part of a project's properties. You can inspect the properties of the following objects:

- Projects
- Data Sets
- Connections
- Data Flows
- Sequences
- Schedules
- Folders

For example, perform the following steps to view or edit data flow properties in the inspector.

1. Go to the Data page and select **Data Flows**, then locate a data flow whose properties you want to view or edit.
2. Click the data flow's **Actions menu** or right-click and select **Inspect**.
3. In the Inspector dialog, check and modify the object properties (such as Name and Description).
4. In the Inspector dialog, modify the object properties (such as Name and Description). Common and type-specific properties are organized in tabs in the Inspector dialog, and the following tabs are displayed:
    - **General** - Lists standard life-cycle properties (such as Name, Description, Created By, and Modified By) that are common to all types of object. This tab also lists high-level properties (such as Type, File Name, File Size, and Location), depending on the type of object that you're inspecting.
    - **Sources/Targets** - Lists the source and target data sets for the data flow. The Parameter Name column is displayed if you've applied parameters to the data flow.
    - **Schedules** - Lists schedules for the object (such as Name, Frequency, and Next Start Time of the schedule).
    - **History** - Lists the recent activity for the object.

- **Permissions** - Lists each user's levels and level of permission.
- **Related** - Lists objects that are related, referenced, or used by the object that you're inspecting. The objects listed depend on the type of object that you're inspecting.

The Inspector dialog also displays other specific tabs (such as Data Elements, Parameters, and Data Flows), depending on the type of object that you're inspecting.

5. Click **Save**.

6

# Explore, Visualize, and Analyze Data

This topic describes the many ways that you can explore and analyze your data.

Video

**Topics:**



## Typical Workflow to Visualize Data

Here are the common tasks for visualizing your data.

| Task | Description | More Information |
|---|---|---|
| Create a project and add data sets to it | Begin a new project and select one or more data sets to include in the project. | Create a Project and Add Data Sets |
| Add data elements | Add data elements (for example, data columns or calculations) from the selected data set to the visualizations on the Visualize canvas. | Build a Visualization by Adding Data from Data Panel |
| Adjust the canvas layout | Add, remove, and rearrange visualizations. | Adjust the Visualize Canvas Layout and Properties |
| Filter content | Specify how many results and which items to include in the visualizations. | Create and Apply Filters |
| Deploy machine learning and Explain | Use Diagnostic Analytics (Explain) to show patterns and uncover insights in your data set, and add the visualizations that Explain provides to your projects. | Analyze Data with Explain |

# Create a Project and Add Data Sets

Projects contain visualizations that help you to analyze your data in a productive and meaningful ways.

When you create a project, you add one or more data sets containing the data that you want to visualize and explore. Data sets contain data from Oracle Applications, databases, or uploaded data files such as spreadsheets. You can also add multiple data sets to your existing projects.

 Tutorial

You can use the Data Set page to familiarize yourself with all available data sets. Data sets have distinct icons to help you quickly identify them by type.

1. To create a new project, go to the Home page, click **Create**, then click **Project**.
2. You can add data to a project using one of the following options:
   - If you're working with a new project, then in the Add Data Set dialog browse and select the data sources that you want to analyze, then click **Add to Project**.
   - If you're working with an existing project, then in the Data Panel click **Add** (+), then **Add Data Set** to display the Add Data Set dialog and add a data source.
   - You can also create a new data source based on a file or connection using the Create Data Set dialog, then add it to your projects.
3. Drag the data elements that you want to visualize from the Data Panel onto the visualization canvas, and start building your project. .
   - You can transform your data set to improve the quality of your analysis and visualization using data preparation script in the Prepare canvas.

# Build a Visualization by Adding Data from Data Panel

You can build visualizations by adding data elements such as measure, text, or attribute from the Data Panel to a canvas.

As you build the visualization, you add as many data elements as required or move them to specific areas on the canvas to explore and analyze your data.

**Topics:**



## Different Methods to Add Data

You can use different methods to add data from the Data Panel to create or update visualizations on a canvas.

Use one of the following methods to add data from the Data Panel:

- Drag data elements from the Data Panel and drop them onto the canvas.
- Select a data element or use Shift-click or Ctrl-click to select multiple data elements in the Data Panel, then right-click to select either a particular visualization type or the option to create a visualization automatically.
- Double-click a data element or use Shift-click or Ctrl-click to select multiple data elements in the Data Panel, then right-click to add them to the canvas.

When you update an existing visualization, you can add data from the Data Panel onto the Grammar Panel or Assignments Pane on the canvas. You can move the data elements from one area to another in the Grammar Panel or Assignments Pane. The visualization is updated based on your selection.

You can create a visualization by selecting a visualization type from the Visualization tab of the Data Panel and adding data elements to the canvas.

## Automatically Create Best Visualization

You can automatically create the best visualization on the canvas based on a set of data elements.

1. Confirm that you're working in the Visualize canvas.
2. To automatically create a visualization on the canvas, do one of the following:
   - Drag data elements from the Data Panel and drop them directly onto the canvas.
   - Right-click data elements on the Data Panel and click **Create Best Visualization**.
   - Double-click data elements to add it to the canvas.

A visualization is automatically created on the canvas, and the best visualization type is selected based on the preconfigured logic. The selected data element is also positioned on a specific area of the Grammar Panel. For example, if you add a revenue measure to the canvas, the data element is placed in the Values area of the Grammar Panel, and **Tile** is selected as the visualization type.

3. Continue adding data elements directly to the canvas to build your visualization.

The visualization type and the position of the data elements on the Grammar Panel might change when you add more data elements to the canvas.

You turn off the mode to automatically create visualizations when you perform actions such as:

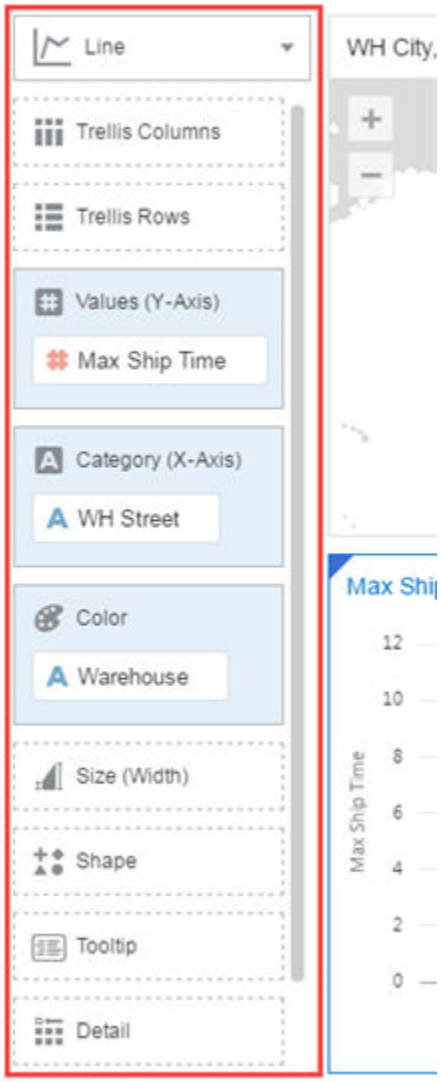
- Changing the visualization properties such as type.
- Adding or removing analyses from the Analytics tab in the Data Panel.
- Dropping a data element onto a specific area of the Grammar Panel or Assignments Pane.
- Changing the filter setting.

When the automatically create visualization mode is turned off, you can select the wizard icon or **Auto Visualization** in the visualization type list to turn on the automatically create mode.

## Add Data to the Visualization Using Grammar Panel

After you've selected the data sets for your project, you can begin to add data elements such as measures and attributes to visualizations.

You can select compatible data elements from the data sets and drop them onto the Grammar Panel in the Visualize canvas. Based on your selections, visualizations are created on the canvas. The Grammar Panel contains sections such as Columns, Rows, Values, and Category.




1. Open or create a project.
2. If you created a project, then add a data set to it.
3. Confirm that you're working in the Visualize canvas.

   Use one of the following methods to add data elements to the Grammar Panel:

   You can only drop data elements based on attribute and type onto a specific Grammar Panel section.

   - Drag and drop one or more data elements from the Data Panel to the Grammar Panel in the Visualize canvas.
     The data elements are automatically positioned, and if necessary the visualization changes to optimize its layout.
   - Replace a data element by dragging it from the Data Panel and dropping it over an existing data element.
   - Swap data elements by dragging a data element already inside the Visualize canvas and dropping it over another data element.

- Reorder data elements in the Grammar Panel section (for example, Columns, Rows, Values) to optimize the visualization, if you've multiple data elements in the Grammar Panel section.
- Remove a data element by selecting a data element in the Grammar Panel, and click **X**.

## Add Data to the Visualization Using Assignments Panel

You can use the Assignment Panel to help you position data elements in the optimal locations for exploring content.

You must create a project or open an existing project and add one or more data sets to the project before you can add data elements to the Assignment Panel. The sections in the Assignment Panel are the same as in the Grammar Panel.

1. Confirm that you're working on the Visualize canvas.
2. Select a visualization on the canvas.

   Alternatively, drag and drop a data element to the blank canvas or between visualizations on the canvas to create a new visualization.
3. Drag a data element to the visualization (but not to a specific drop target); you'll see a blue outline around the recommended assignments in the visualization.

   Hover the data element on the Assignment Panel to identify other valid assignments.
4. Drop the data element on the selected assignment.

To display the Assignment Panel, click **Show Assignments** on the visualization toolbar.

## Modify a Visualization's Tooltips

When you hover over a data point in a visualization, a tooltip displays and provides specific information about the data point. You can choose to see all tooltips or only the measures included in the Tooltip section of the Grammar Panel.

For example, if you create a simple bar chart visualization that shows revenue for countries in the Americas region, the tooltip displays the region's name, the country's name, and exact revenue amount. If you add Target Revenue to the Tooltip section of the Grammar Panel, then the target revenue amount is displayed in the tooltip and the user can easily compare the actual revenues with the target revenues. Set the **Tooltip** field in the General Properties Pane to **Tooltip Grammar Only** if you want the tool tip to contain only the target revenue amount.

Note the following restrictions:

- You can drag and drop only measure columns to the Tooltip section in the Grammar Panel.
- The Tooltip section in the Grammar Panel doesn't display for all visualization types.

1. Confirm that you're working on the Visualize canvas and select a visualization.
2. Drag and drop one or more measure columns from the Data Panel to the Tooltip section in the Grammar Panel.

Hover the mouse pointer over a data point on the visualization to display the tooltip. Because the **Tooltip** field is set to **All Data** by default, the tooltip contains the data point's values for all columns included in the visualization. The data values for the columns that you added to the Tooltip section are displayed at the bottom of the tooltip.

3. (Optional) Use the **Tooltip** field to display only the data values that you want or to turn off the tooltip.
   - If you want the tooltip to display data values for only the columns you dragged to the Tooltip section, then set the **Tooltip** field to **Tooltip Grammar Only**.
   - If you don't want the tooltip to be displayed, then confirm that there are no columns in the Tooltip section and set the **Tooltip** field to **Tool Tip Grammar Only**.

# Use Advanced Analytics Functions

Advanced analytics are statistical functions that you apply to enhance the data displayed in visualizations.

The Analytics area in the Data Panel contains standard analytics functions (for example, Clusters and Trend Line). You can use analytics functions as they are, or use them to create your own calculated columns that reference statistical scripts.

**Topics:**



## Add Advanced Analytics Functions to Visualizations

You can apply advanced analytics functions to your project's visualizations.

Before you can use analytic functions in visualizations, you must do the following:

- Install DVML.

  On Windows go to **Start**, browse to and expand your system's Oracle folder, and click **Install DVML**.

  On Mac, go to **Applications** and click **Oracle Analytics Desktop Configure Python**.
- Create a project or visualization that you can apply one or more analytic functions to.

1. Confirm that you're working in the Visualize canvas.
2. To display the available advanced analytic functions, click the **Analytics** icon in the Data Panel.
3. To edit the applied advanced analytics in a visualization, highlight the visualization, and in the properties pane click the **Analytics** icon.
4. To add advanced analytic functions to a visualization, do one of the following:
   - Drag and drop an advanced analytic function (such as Clusters, Outliers, Reference Line) from the **Analytics** pane to a visualization.
   - Right-click a visualization, and select an advanced analytic function.

- In the properties pane select the **Analytics** icon and click **Add (+)**, then select a function such as **Add Clusters** or **Add Outliers**.

## Add Reference Lines to Visualizations

You can use advanced analytics reference lines to identify the range of data element values in a visualization.

1. Confirm that you're working on the Visualize canvas.
2. In the Data Panel, click the **Analytics** icon.
3. Drag and drop **Reference Line** into a visualization. Alternatively, you can double-click **Reference Line** to add it to the selected visualization.
4. In the properties pane select the **Analytics** tab and do the following:
   a. Click the current **Method** and select **Line** or **Band**.
   b. Click the current reference **Function** (for example, **Average** and **Custom**) and select the reference function that you want to use.
   c. If you choose the **Line** method, you can select reference functions such as **Median**, **Percentile**, **Top N**, and **Constant**.
      - **Percentile** - Percentile rank number ranks the percentile of the data element added to the visualization.
      - **Top N** - N value marks the highest values (ranked from highest to lowest) of the data element added to the visualization.
      - **Bottom N** - N value marks the lowest values (ranked from highest to lowest) of the data element added to the visualization.
   d. If you choose the **Band** method, you can select either **Custom** or **Standard Deviation** as the reference function.
      - **Custom** - Select the *to and from* range of the data element values (such as **Median** to **Average**).
      - **Standard Deviation** - Select a value from 1 to 3 to show the standard deviation for the selected value of the data element.
5. Click **Save**.

Based on the selected **Method** or reference **Function**, a line is displayed in the visualization to highlight the value.

# Use Spark Charts to Examine Trends

You can add a spark chart to a tile visualization to view aggregate data trends over time.

1. Open or create a project and click **Visualize**.
2. In the Data pane, drag an aggregate data element to the canvas to create a tile visualization. For example, select **Revenue**.
3. In the Data pane drag a time-based data element to **Category (Chart)** to add a spark chart to the tile visualization. For example, select and drag **Month**.

   The red dot on the spark chart line shows the lowest value, and the green dot shows the highest value. The tooltip shows the first, last, lowest, highest and

average aggregate values for the selected category. Hover your cursor anywhere over the spark chart to display the tooltip.

4. If you want to add a filter, drag a data element from the Data pane to **Click here or drag data to add a filter**.

   For example, **Year**.
   The filter changes the spark line to show the trend for the filtered data.

5. If you want to change spark chart display settings, click General in the Properties pane.

| Property | Description |
|---|---|
| **Chart** | Click to display one of the following spark charts:<br>• **Line**<br>• **Bar**<br>• **Area** |
| **Reference Line** | Click to display or remove a reference line:<br>• **Average** - Displays a reference line showing the average trend.<br>• **None** - Removes the reference line. |
| **High/Low Marks** | Click to hide or show the high and low marks, displayed as a green and a red dot:<br>• **Hide**<br>• **Show** |

6. Click **Save** to save your changes.

# Create Calculated Data Elements in a Data Set

You can create a new data element (typically a measure) to add to your visualization. For example, you can create a new measure called Profit that uses the Revenue and Discount Amount measures.

The calculated data elements are stored in the data set's My Calculations folder and not in the project. In a project with a single data set only one My Calculations folder is available and the new calculated data elements are added to it. In a project with multiple data sets My Calculations folder is available for each set of joined and not-joined data sets. Ensure that you're creating the calculated data elements for the required data set or joined data set. The new calculated data elements are added to the My Calculations folder of the data sets (joined and non-joined) that you create the calculation for.

1. In the Visualize canvas navigate to the bottom of the Data Panel, right-click My Calculations, and click **Add Calculation** to open the New Calculation dialog.
2. Enter a name.
3. In the expression builder pane, compose and edit an expression. See About Composing Expressions.

   You can drag and drop a column into the expression builder pane only if the column is joined to the data set.
4. Click **Validate**.
5. Click **Save**.

# Undo and Redo Edits

You can quickly undo your last action and then redo it if you change your mind. For example, you can try a different visualization type when you don’t like the one you’ve just selected, or you can go back to where you were before you drilled into the data.

The undo and redo options are useful as you experiment with different visualizations. You can undo all the edits you've made since you last saved a project. However, in some cases, you can't undo and then redo an edit. For example, in the Create Data Set page, you've selected an analysis from an Oracle Application data source to use as a data set in the project. In the next step, if you use the undo option to remove the data set, you can't redo this change.

- To undo or redo an edit, go to the toolbar for the project or the data set and click **Undo Last Edit** or **Redo Last Edit**. You can use these options only if you haven't saved the project since making the changes.

- When you’re working on a project, click **Menu** on the project toolbar and select **Revert to Saved** to undo all the edits you've made since you last saved your project. **Revert to Saved** is enabled after you've saved the project for the first time. This option is automatically disabled if you select the **Auto Save** option.

# Refresh Data in a Project

Refresh ensures that you see the most up-to-date data in your projects.

- On the Visualize canvas, click **Refresh Data**. This option ensures that you see the most up-to-date data by re-executing the visualization queries for all views in your project.

# Pause Data Queries in a Project

You can disable the Auto Apply Data option to pause issuing new queries as you change visualization content in a project.

You can quickly configure a visualization without having to wait for data updates after each change.

1. Open or create a project.
2. If you've created a project, then add a data set to it.
3. Confirm that you’re working in the Visualize canvas.
4. Click the **Auto Apply Data** button to pause data queries.

   Data queries are temporarily disabled.
5. Make changes to a visualization in the project.

   Your changes are displayed but data isn't updated, and a bubble is displayed indicating the number of data requests that have been skipped.
6. Click the bubble if you want to refresh the data now.

Data queries are still disabled.

7. Click the **Auto Apply Data** button to re-enable data queries.

# Adjust the Visualize Canvas Layout and Properties

You can use the Visualize canvas' options to perform many different tasks. For example, copy and paste a visualization or the whole canvas, copy a visualization's data, or adjust the canvas' properties.

Note the following options.

| Option | Location | Description |
|---|---|---|
| Canvas Properties | Right-click a canvas tab | Change the name, layout, width, and height of the canvas in the Canvas Properties dialog.<br>Use the **Synchronize Visualizations** setting to specify how the visualizations on your canvas interact. |

| Option | Location | Description |
|---|---|---|
| Canvas Layout | Right-click a canvas tab | Select **Canvas Properties**, **Layout**, and one of the following:<br>• **Freeform** – If you select Freeform, you can perform the following functions:<br>– **Select All Visualizations** selects all the visualizations on a canvas for you to copy.<br>– **Order Visualization** and options: Bring to Front, Bring Forward, Send Backward, or Send to Back moves a visualization on a canvas with multiple visualizations.<br>– Rearrange a visualization on the canvas using drag and drop to a space between visualizations where you want to place it. The target drop area is displayed with a blue outline.<br>– Resize a visualization by dragging its edges to the appropriate dimensions.<br>• **Auto Fit** – Auto-arrange or correctly align the visualizations on a canvas with multiple visualizations.<br>Resize a visualization by dragging its edges to the appropriate dimensions:<br>– Resize one visualization by using **Ctrl** and dragging a visualization edge to an appropriate dimension. Other visualizations resize to fit the remaining space.<br>– Resize all visualizations to the same size by dragging a visualization edge to an appropriate dimension.<br>Use **Height**, and **Width** settings in conjunction with **Layout** settings to specify the layout size of visualizations in a canvas:<br>• **Screen** - Layout fits in the available screen space.<br>This option is removed when you select **Freeform**.<br>• **Grow** - Layout automatically grows in height or width to accommodate a visualization at its optimal size.<br>For example, if you add multiple visualizations, or rows of visualizations to a canvas, the layout grows so that the added visualizations display at their optimum size. |

| Option | Location | Description |
|---|---|---|
| | | • **Fixed** - Layout uses the specified size. |
| Add Canvas | Canvas tabs bar | Add a new canvas to the project.<br>You can drag a canvas to a different position on the canvas tabs bar. |
| Rename | Right-click a canvas tab | Rename a selected canvas. |
| Duplicate Canvas | Right-click a canvas tab | Add a copy of a selected canvas to the project's row of canvas tabs. |
| Copy Canvas | Right-click a canvas tab | Copies the whole canvas. You can use the **Paste Canvas** option to paste the canvas to the current project or to another project. |
| Clear Canvas | Right-click a canvas tab | Remove all the visualizations on the canvas. |
| Delete Canvas | Right-click a canvas tab | Delete a specific canvas of a project. |
| Duplicate Visualization | Visualization **Menu**, **Edit** or right-click a visualization | Add a copy of a selected visualization to the canvas. |
| Copy Visualization | Visualization **Menu**, **Edit** or right-click a visualization | Copy a visualization on the canvas. |
| Paste Visualization | Visualization **Menu**, **Edit**, or right-click a visualization or blank canvas | Paste a copied visualization into the current canvas, another canvas, or another project. |
| Copy All Data | Visualization **Menu**, **Edit** | Copy all of the visualization's data to the clipboard. You can then paste it to another application, like Word or Excel. Optionally, you can copy specific data in the visualization. Highlight the data you want to copy, right-click, and select **Copy Data**. |
| Delete Visualization | Visualization **Menu** or right-click a visualization | Delete a visualization from the canvas. |
| Brushing | Right-corner of canvas tabs bar | Turn on to keep multiple charts in sync when you make changes. |
| Toggle Grammar Panel | Right-corner of canvas tabs bar | Turn on to display the options for the currently displayed visualization, such as columns, rows, and colors. |
| Toggle Data Panel | Right-corner of canvas tabs bar | Turn on to display the left-hand panel to add data, visualizations, or analytics. |

# Copy and Paste a Visualization or Canvas

You can copy and paste a visualization or canvas within the same project, to another open project, or to another project open in a different browser tab.

When you copy a visualization or canvas from one project to another, Oracle Analytics does the following:

- Data - The data set for the pasted visualization or canvas is added to the target project. When you open or create the target project that you're pasting to, it doesn't need to include the data set used by the visualization or canvas that you'll copy and paste.

- Filters - The filters in the target project and in the pasted visualization or canvas are maintained. You don't need to add the visualization or canvas filters to the target project. If there's a conflict between the target project and the pasted visualization or canvas filters, then the pasted filters won't overwrite the target's filters.
- Color assignments - The color scheme of the target project is applied to the pasted visualization or canvas.
- Calculations - If the same calculation name exists in the target project, then the pasted calculation is added and renamed.

Use the following steps to copy and paste a visualization or canvas:

1. In the Home page, search for and open a project to copy a visualization or canvas from.
2. Copy a visualization or canvas.
   - To copy a visualization, click its **Menu**, hover over **Edit**, and then click **Copy Visualization**.
   - To copy a canvas, right-click it and click **Copy Canvas**.
3. Navigate to a visualization or canvas and paste the object.
   - To paste a visualization into a canvas that contains visualizations, click an existing visualization's **Menu**, hover over **Edit**, and then click **Paste Visualization**.
   - To paste a visualization into a blank canvas, right-click the canvas bar and select **Add Canvas**. Right-click the new canvas, hover over **Edit**, and then click **Paste Visualization**.
   - To paste a canvas, right-click the canvas bar and then click **Paste Canvas**.

# Change Visualization Types

You can change visualization types to best suit the data you're exploring.

When you create a project and add data elements to the canvas, Auto Visualization mode chooses the most appropriate visualization type based on the selected data element. The Auto Visualization mode is on (selected) by default. If you add more data elements, the visualization type is automatically updated, and the best type is selected based on the data elements.

If you want to use a different visualization type, then you need to select it from the visualization type list. When you change the visualization type, Auto Visualization mode is turned off. When the Auto Visualization mode is off (deselected), adding more data elements to the canvas won't change the visualization type automatically.

1. Confirm that you're working in the Visualize canvas. Select a visualization on the canvas, and on the visualization toolbar, click **Change Visualization Type**.
2. Select a visualization type. For example, select Treemap to change the visualization type from Pivot to Treemap.

   When you change the visualization type, the data elements are moved to matching drop target names. If an equivalent drop target doesn't exist for the new visualization type, then the data elements are moved to a Grammar Panel section labeled **Unused**. You can then move them to the Grammar Panel section you prefer.

# Adjust Visualization Properties

You can customize how the visualizations in your project look and function. For example, the legend's location, number format, and font used in titles and labels.

The tabs and fields displayed depend on the type of visualization that you're working with.

1. In the Visualize canvas, select a visualization.

   The visualization's properties display in the Properties pane.

2. Use the tabs in the Properties pane to adjust the visualization's properties as needed. The most common tabs and options are described in this table:

| Properties Pane Tab | Description |
|---|---|
| General | Change and format the visualization's title and legend. Change the title's font, size, and color.<br>Add a custom background image or fill color and adjust its transparency.<br>Turn the tooltip on or off. |
| Axis | Turn grid lines on or off. Set the axis labels for horizontal and vertical values and the axis values to start and end.<br>Display or hide the axis labels.<br>Change the axis label's text. Change the label text's font, size, and color. |
| Edge Labels | Show or hide a table visualization's headers. Change the table's header, data, and total text's font, size, and color.<br>Add or remove column totals.<br>Wrap column labels. Change a column text's font, size, and color. |
| Action | Add URLs to Tile, Image, and Text Box visualizations. |
| Values | Change the display and placement of the data labels. Change the labels' font, size, and color.<br>Change numbers to display as currency or percentage.<br>Change the aggregation method.<br>Display a trellis row (Y2 axis). |
| Analytics | Add clusters or outliers to the visualization.<br>Add reference lines, trend lines, and bands to display at the minimum or maximum values of a measure included in the visualization. |
| Filters | Change and format the visualization's filter title and selection names. Change the font, size, and color. |

# Apply Color to Visualizations

Use color to enhance visualizations.

**Topics:**

- About Color Assignments in Visualizations
- Access Color Options
- Change the Color Palette

- Assign Colors to Columns

# About Color Assignments in Visualizations

You can work with color to make visualizations more attractive, dynamic, and informative. You can color a series of measure values (for example, Sales or Forecasted Sales) or a series of attribute values (for example, Product and Brand).

Your color choices are shared across all visualizations on the canvas, so if you change the series or data point color in one visualization, then it appears on the other visualizations.

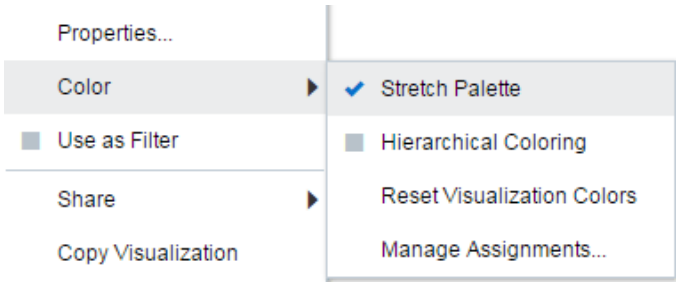
The **Visualize** canvas has a Color section in the Grammar Panel where you can put a measure column, attribute column, or set of attributes columns. Note how the canvas assigns color to the columns that are included in the Color section:

- When a measure is in the Color section, then you can select different measure range types (for example, single color, two color, and three color) and specify advanced measure range options (for example, reverse, number of steps, and midpoint).
- When you've one attribute in the Color section, then the stretch palette is used by default. Color palettes contain a set number of colors (for example, 12 colors), and those colors repeat in the visualization. The stretch palette extends the colors in the palette so that each value has a unique color shade.
- If you've multiple attributes in the Color section, then the hierarchical palette is used by default, but you can choose to use the stretch palette, instead. The hierarchical palette assigns colors to groups of related values. For example, if the attributes in the Color section are Product and Brand and you've selected Hierarchical Palette, then in your visualization, each brand has its own color, and within that color, each product has its own shade.

# Access Color Options

You can set color options for your project and for individual visualizations within your project.

1. If you want to edit color options for the whole project:
   a. Click **Menu** on the project toolbar and select **Project Properties**.
   b. Use the **General** tab to edit the color series or continuous coloring.
2. If you want to edit color options for a visualization.
   a. Select the visualization and click **Menu** or right-click.
   b. Select **Color**. The available color options depend on how the measures and attributes are set up in your visualization.




c. You can experiment with visualization colors and select **Reset Visualization Colors** to revert to the original colors.

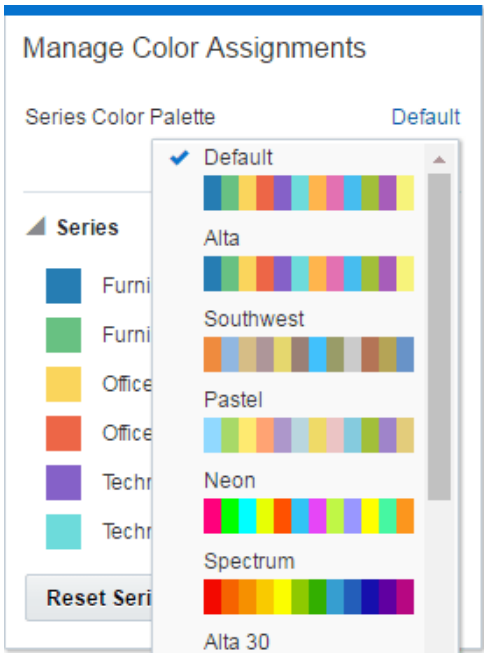
d. Select **Stretch Palette** to turn this option on or off.
Color palettes have a set number of colors, and if your visualization contains more values than the number of color values, then the palette colors are repeated. Use the Stretch Palette option to expand the number of colors in the palette. Stretch coloring adds light and dark shades of the palette colors to give each value a unique color. For some visualizations, stretch coloring is used by default.

## Change the Color Palette

You can switch between the various color palettes until you find the one you want.

Each color palette contains 12 colors that you can apply to a visualization.

1. Select the visualization that you want to change the color palette for.
2. Click **Menu** or right-click and select **Color**, then select **Manage Assignments**.
3. Locate the **Series Color Palette** and click the color palette that's currently used in the visualization (for example, Default or Alta).

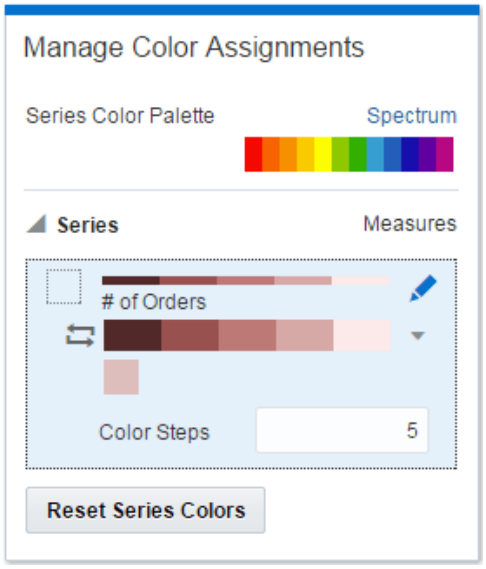


4. From the list, select the color palette that you want to apply to the visualization.

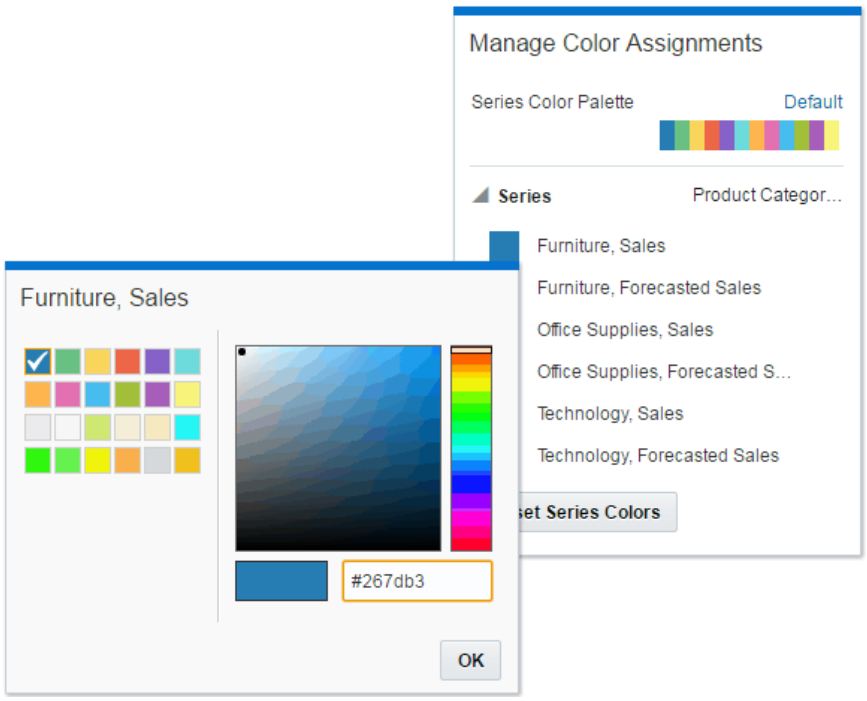
## Assign Colors to Columns

Instead of using the palette's default colors, you can choose specific colors to fine-tune the look of your visualizations.

1. Select the visualization that you want to manage the colors for.
2. Click **Menu** on the visualization toolbar or right-click and select **Color**, then select **Manage Assignments**.
3. If you're working with a measure column, you can do the following:
   - Click the box containing the color assigned to the measure. From the color picker dialog, select the color that you want to assign to the measure. Click **OK**.
   - Specify how you want the color range to be displayed for the measure (for example, reverse the color range, pick a different color range, and specify how many shades you want in the color range).




4. If you're working with an attribute column, then click the box containing the color assignment that you want to change. From the color picker dialog, select the color that you want to assign to the value. Click **OK**.




# Format Numeric Values of Columns

You can format numeric values of a column in your visualizations using a wide range of ready-to-use formats. For example, you might change the aggregation type from Sum to Average.

1. Create or open the project that contains the numeric column whose properties you want to change.
2. In the Data Panel, select the column.
3. In the properties pane for the selected column, use the **General** or **Number Format** tabs to change the numeric properties.
   - **General** - Change the column name, data type, treat as (measure or attribute), and aggregation type.
     For example, to change how a number is aggregated, use the **Aggregation** option.
   - **Number Format** - Change the default format of a number column.
4. Click **Save**.

# Set Currency Symbols for Visualizations

You can set measure values in a visualization to display an appropriate currency symbol.

You can configure a measure to use a custom currency to display the symbol associated with a currency. For example, if you set a canvas filter to display a European Ledger, then the Euro symbol is displayed for each measure value that's associated with the custom currency property. The project data must contain a currency code column, for example, a Ledger Currency column. The currency code

column enables the display of an appropriate currency symbol for the measure column, for example, a Profit column.

1. Create or open the project.
2. In the Properties panel for a selected measure, click the **Values** tab, click **Number Format**, and select **Currency**.
3. In the **Currency** field, click the currently displayed value and select **Custom**.
4. In the **Custom** field, add the column that determines the currency code.
5. Click **Save**.

# Format Numeric Values of Visualizations

You can format numeric properties of a visualization using a wide range of ready-to-use formats. For example, you might change the aggregation type from Sum to Average.

1. Create or open the project that contains the visualization whose properties you want to change.
2. In the Visualize canvas, select the visualization.
3. In the properties pane for the selected visualization, use the **Values** tab to change the numeric properties.

   For example, to change how a number is aggregated, use the **Aggregation Method** option.
4. Click **Save**.

# Apply Map Backgrounds and Map Layers to Enhance Visualizations

You can use geographical information to enhance the analysis of your data.

**Topics:**

- Create Cluster Layers on a Map Visualization
- Represent Point Data With Custom Icons on a Map
- Select Points or Area on a Map
- Represent Line Data Using Size and Color on a Map
- Make Maps Available to Users
- Make Map Backgrounds Available to Users

## About Map Backgrounds

You can enhance map visualizations in projects by adding and maintaining map backgrounds.

You can apply the ready-to-use map backgrounds to a project. You can also add backgrounds from the available list of Web Map Service (WMS) providers such as Google Maps and Baidu Maps. Background maps from these providers offer details and language support (such as city or region name) that certain geographic regions (such as Asian countries) require. You can enhance backgrounds in these ways:

- Modify the background parameters such as map type, format, language and API keys. The parameters are different for each WMS provider.
- Assign or change the default background in a project.
- Reverse the inherited default background settings in a project.

You can add a WMS provider and perform the following types of functions:

- Add the WMS map servers, and make them available as additional map background options.
- Select one or more map backgrounds available from the WMS provider.
- Assign an added WMS provider's map as the default map background.

## Enhance Visualizations with Map Backgrounds

You can use map backgrounds to enhance visualizations in a project.

Tutorial

Based on the column values, a specific set of dimensions and metrics is displayed on the map visualization. You'll see either the default map background or an existing Oracle map background if no default is set.

1. Create or open a project and confirm that you're working in the Visualize canvas.
2. To select a column and render it in a map view, do one of the following:
   - Right-click a map-related column in the Data Element pane and click **Pick Visualization**, then select **Map**.
   - Drag and drop a map-related column from the Data Element pane to the blank canvas, or between visualizations on the canvas. On the visualization toolbar, click **Change Visualization Type** and select **Map**.

   The selected column or attribute is displayed as a data layer in the Category (Geography) section of the Grammar Panel and in the Data Layers tab of the properties pane.

3. In the properties pane, click **Map** and specify the visualization properties.
4. If you want to use a different map background, click the **Background Map** value in the properties pane and select a background. For example, select Google Maps, and the visualization displays Google Maps as the map background.
   - If you want to see the list of available map backgrounds or change the backgrounds that you can use, do one of the following:
     - Click the **Background Map** value and select **Manage Map Backgrounds** to display the Map Background tab.
     - Open the Console page, click **Maps** and select the Backgrounds tab.
   - Optionally, select another map background such as Satellite, Road, Hybrid, or Terrain.
5. Click **Save**.

## Use Different Map Backgrounds in a Project

As an author you can use different map backgrounds in map visualizations.

Here is an example of how you might use a map background in a project.

1. On the Home page click **Create**, then click **Project**.
2. Select a data set in the Add Data Set dialog.
3. Click **Add to Project**.

   The Project pane and list of Data Elements is displayed.
4. Select a map-related data element (for example, click **City**), and click **Pick Visualization**.
5. Select **Map** from the list of available visualizations.

   Either the default map background or an existing Oracle map background if no default is set is displayed.
6. In the visualization properties pane, select the **Map** tab.
7. Click the **Background Map** value and select a map from the drop-down list.

   For example, select Google Maps and Google Maps as the map background is displayed.
8. (Optional) Click another value to change the type of map (such as Satellite, Road, Hybrid, or Terrain).
9. (Optional) Click **Manage Map Backgrounds** from the **Background Map** options to display the Map Backgrounds pane.

   Use this option to maintain the map backgrounds that you want to use.

## Interpret Data Values with Color and Size in Map Visualizations

You can use the color and size of a shape such as a polygon or a bubble, to interpret values in a map visualization.

1. Create or open a project and confirm that you're working in the Visualize canvas.
2. Select a column and render it in a map view, doing one of the following:

- Right-click a map-related column in the Data Element pane and click **Pick Visualization**, then select **Map**.
- Drag and drop a map-related column from the Data Element pane to the blank canvas, or between visualizations on the canvas. On the visualization toolbar, click **Change Visualization Type** and select **Map**.

3. Drag and drop columns to the following sections on the Visualization Grammar Pane:

| Section | Description |
|---|---|
| Color | Change the color for geometries displayed in the corresponding map layer (for example, polygon fill color, bubble color) based on the values. |
| Size (Bubble) | Change color bubble size based on the measure column values.<br>To change the size of the color bubble you've to drag and drop measure columns only. The size shows the aggregated measure for a specific geographic location in a map visualization. |
| Trellis Columns / Rows | Compare multiple map visualizations based on the column values using filters. |

In the map visualization, you can also use the following to interpret measure columns and attribute values:

- **Legend** - If a measure column or an attribute has multiple values, then the legend is displayed that shows values by size or color.
- **Tooltip** - If you hover the mouse pointer over a color bubble or data point, then the values are displayed in a tooltip.

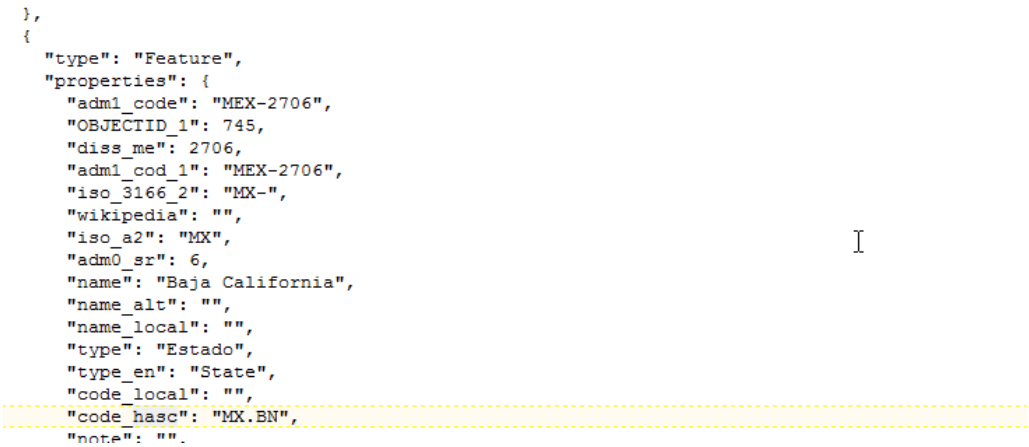
## Add Custom Map Layers

You can add custom map layers to use in map visualizations.

Video

You add a custom map layer using a geometric data file with the .json extension that conforms to GeoJSON schema https://en.wikipedia.org/wiki/GeoJSON (the maximum file size allowed is 20MB). You then use the custom map layer to view geometric map data in a project. For example, you might add a Mexico_States.json file to enable you to visualize geometric data in a map of Mexico States.

When creating a custom map layer, you must select layer keys that correspond with data columns that you want to analyze in a map visualization. For example, if you want to analyze Mexican States data on a map visualization, you might start by adding a custom map layer for Mexican States, and select HASC code layer key from the Mexican_States.json file. Here is an extract from the Mexican_States.json file that shows some of the geometric data for the Baja California state.




If you wanted to use the Mexican_States.json file, the layer keys that you select must match columns that you want to analyze from the Mexican States Data tables. For example, if you know there is a data cell for the Mexican state Baja California then select the corresponding name field in the JSON file to display state names in the Map visualization. When you create a project and select column (such as State, and HASC), then Mexican states are displayed on the map. When you hover the mouse pointer over a state, the HASC code (such as MX BN) for each state is displayed on the map.

1. Open the Console page and click **Maps** to display the Map Layers page.

   You can perform the following actions when managing System Map Layers and Custom Map Layers.

| What action can I perform? | System Map Layer | Custom Map Layer |
|---|---|---|
| Enable | Yes | Yes |
| Disable | Yes | Yes |
| Create | No | Yes |
| Delete | No | Yes |

2. To add a custom map layer, click **Add Custom Layer** or drag and drop a JSON file from File Explorer to the Custom Maps area.
3. Browse the Open dialog, and select a JSON file (for example, Mexico_States.json).

   The JSON file must be a GeoJSON file that conforms to the standard specified in https://en.wikipedia.org/wiki/GeoJSON (the maximum file size allowed is 20MB).

   Custom layers that use the Line String geometry type aren't fully supported. The Color and Size section on Visualization Grammar pane doesn't apply to line geometries.
4. Click **Open** to display the Map Layer dialog.
5. Enter a **Name** and an optional **Description**.
6. Select the layer keys that you want to use from the Layer Keys list.

   The layer keys are a set of property attributes for each map feature, such as different codes for each state in Mexico. The layer keys originate from the JSON file. Where possible, select only the layer keys that correspond with your data.

7. Click **Add**. A success message is displayed when the process is complete and the layer is added.

## Update Custom Map Layers

You can maintain custom map layers.

1. Open the Console page and click **Maps** to display the Map Layers page.
2. In the Custom Map Layers section, right-click the map layer and click **Options**, then do the following:
   - To view or make changes to the map layer settings, select **Inspect**. The Map Layer dialog is displayed where you can update the **Name**, **Description**, or the **Layer Keys** used in this layer.
   - To upload the JSON file again, select **Reload**.
   - To save the JSON file locally, select **Download**.
   - To delete the custom map layer, select **Delete**. You can disable or enable a System Map Layer and a Custom Map Layer, but you can't add or delete a System Map Layer.
3. Click the map layer to enable or disable it. For example, if you want to exclude us_states_hexagon_geo on the map, click the layer to disable it and remove it from searches.
4. To switch from using one map layer to another, do the following:
   a. In the properties pane, select the **Map** tab to display the map properties.
   b. Click the current **Map Layer** for example Mexican States. This displays a list of available custom map layers that you can choose from.
   c. Click the map layer that you want to use to match your data points.

## Apply Multiple Data Layers to a Single Map Visualization

You can use the data layer feature to display multiple data series (different sets of dimensions and metrics) on a single map visualization. The data layers are overlaid on one another in a single map visualization.

1. Create or open the project where you want to display multiple data layer overlays on a single map visualization. Confirm that you're working in the Visualize canvas.
2. Drag and drop a measure or attribute columns containing map-related data from the Data Panel to the Category (Geography) section on the Grammar Panel.

   If you're creating a map visualization, in the Data Panel, right-click an attribute column and click **Pick Visualization** then select **Map**.
3. Click **Layer options** in the Category (Geography) section of the Grammar Panel and click **Add Layer** to add a new data layer (for example, Layer 2).

   Alternatively in the Data Layers tab click **Add Layer (+)**.
4. Drag and drop a column to the Category (Geography) section. Based on the column values the map visualization automatically updates with a different set of dimensions, and it overlays on the previous layer.
5. Repeat step 3 and 4 to add multiple data layers on the map visualization.

6. Click **Layer options** to use other options such as Order Layer, Hide Layer, and Manage Layers.
7. In the Data Layers tab of the properties pane, you can specify the options for a layer such as Layer Type, Map Layer, Transparency, and Show Layer.
8. To refine the data shown for the measure and attribute columns in all the data layers, you can apply a filter such as a Range Filter or List Filter, to the map visualization. For example, you can select a measure or attribute for a layer, then apply filter to reduce the amount of data shown, and add the same measure or attribute to the Color section on the Grammar Panel.

# Use an Image as a Map Background and Draw Map Layer Shapes on the Image

You can upload an image, prepare the image as a map background, draw map layer shapes onto the image, and associate data with the map background layer.

**Topics:**



## Upload an Image as a Map Background

You can upload an image as a map background and then draw layers on top of the uploaded image.

1. On the Home page, click **Navigator**, and then click **Console**.
2. On the Console page, click **Maps**.
3. In Maps, click the **Backgrounds** tab, and expand **Image Backgrounds**.
4. Click **Add Image**, select your image, and click **Open**.
5. Enter a name and description for the uploaded image, and click **Save**.

## Draw Custom Map Layer Shapes on an Uploaded Image

You can draw and edit custom map layer shapes on an uploaded image and associate the shapes with data in map visualizations.

1. In Image Backgrounds, select the image, click **Options**, and then select **Create Map Layer**.
2. Select **Polygon**, **Line**, or **Point**, and draw a shape onto the image.

| Shape | Actions |
|---|---|
| **Polygon** | Click the image, drag and click to draw each edge of the polygon until the shape is complete, then click to finish. |

| Shape | Actions |
|---|---|
| **Circle** | Click the image, drag to increase the circle size until the circle reaches the required size, then click to finish. |
| **Line** | Click the image, then drag and click to draw each line edge until the line is complete. |
| **Point** | Click the image in the location where you want to draw a data point. |

If you uploaded a motorcycle image, you could draw a shape outline over each visible part. For example, you might draw a polygon to represent an irregular shape like the fuel tank, or a line to represent a fork, or a circle to represent a tire, and so on.

Each new shape that you create is given a default name and is listed under Features.

3. Enter a name for each shape that corresponds to a key column value in the data set.

   For example, if you drew a petrol tank polygon shape and the key column PartID in the data set has the value PT for petrol tank, then you must enter `PT` as the shape name.

   You can also edit a shape name by clicking the corresponding object in the Features list.

4. Optional: If you want to reposition a shape, click the shape, then drag it to move it to a new position.

5. Optional: If you want to resize a shape, click and hold the shape, or edge, and drag it until it reaches the required size, then click again to finish.

6. Click **Save**.

## Associate a Data Set with Map Layer Shapes Drawn on an Uploaded Image

You can associate a data set with the map background layer shapes that you drew on an uploaded image, and use it in a project.

1. In the Home page, click **Create** to bring in a data set file that you want to associate with your map background.

   For example, you might select `motorbike.xls` to create a motorcycle data set with a PartID key column containing values that match the part names of the shapes that you drew.

2. In the key column of the data set, click **Options**, select **Location Details**, choose the custom map layer, and click **OK** to assign the key column to the selected map layer.

3. Click **Apply Script** to associate the key column with the map layer.

4. Create a project.

5. Drag and drop the key column into the visualization. This automatically places the column into Category (Location).

   A map visualization is suggested based on the key column, and the associated map background is displayed.

6. Continue to add columns, and create visualizations as required.
7. Click **Save**.

## Assign a Map Layer to a Data Column

Assign a map layer to a data column to use it consistently in any project.

You can assign a map layer to a column that contains text or numeric attributes, for example, columns such as Airport Name, Latitude, and Longitude. When you select a data column with a map layer assignment for a visualization, Oracle Analytics automatically creates a map visualization.

1. On the Home page, select the project with the map layer.
2. In the Prepare tab, click **Options** for any numeric or text attribute column, and then click **Location Details**.
3. In Location Details, review the associated map layer, change the map layer if you want to, and then click **OK**.

   The property change is listed as a Change Property step in the Preparation Script pane.
4. In the Preparation Script pane, click **Apply Script**.

   The updated column displays the location icon indicating that the location preference has been set.
5. In the Properties dialog for the updated column, click the **Location** icon to verify the map layer associated with the column.
6. Create a visualization with the data columns for which you have set the location details.

   The visualization type is now automatically set as Map and the map layer is available for the specific columns. You no longer need to set the location details for each visualization.

## Auto Focus on Data for a Map Visualization

You can override the current viewport when you focus on new data in a map visualization.

The ability to auto focus on data enables you to view the visualization for the specific data that you displayed in a map. For example, if your map visualization displays sales by countries and you have first viewed sales for Australia, the map zooms to Australia. However, if you focus on sales for Italy, the map zooms to Italy.

1. Create or open the project with map visualizations and confirm that you're working on the Visualize canvas.
2. In Visualization Properties, click **Map**, and then set **Auto Focus on Data** to **On**.

   The default value of this property is **Off**.

# Review Location Matches for a Map Visualization

You can review mismatch issues between data and match results in map layers, such as when there are ambiguous or partial matches between words.

If ambiguous results are present, consider adding more columns to the map visualization to get the specific match. For example, your map might have layers for City, Country, and Continent. You can also exclude rows of data.

1. Create or open a project that contains map data with one or more data layers in a map visualization.
2. Click the **Visualization** tab.
3. Right-click the map visualization and select **Location Matches** from the menu to display the Location Matches dialog.
4. Select a tab representing a map layer in the current visualization to inspect how well your data matches the map layer.

   For example, select the **Country** tab to see how well your data matches with the Country map layer.
5. Optionally click **Map Layer** to select a different layer or click **Manage Map Layers** to display the Console page.
6. To review and resolve data mismatches use the columns:

   The summary section above the table displays the number of locations and any issues.

| Column Name | Column Description |
|---|---|
| Your Data | Displays comma-separated geographical data. |
| Match | Indicates what matches with the map layer information. A match can be anything from No Match, Partial Match, to a 100% match. Matches are displayed initially sorted top down from the worst to the best matches.<br>• No Match - Displays a red warning triangle indicator.<br>• A Match with an issue - Displays a yellow warning triangle indicator. The warning doesn't indicate a poor match but an imperfect match for which you might want to review the use case.<br>• A Perfect Match - Displays no warning triangle indicator.<br>If you are matching Latitude and Longitude, the match values are Valid or Invalid. |
| Match Quality | Quantifies how similar your mismatched strings are:<br>• No Match<br>• Number of Matches<br>• Percentage Confidence |
| Exclude | Enables you to exclude entire rows of data. |

7. Click the filter icon in the **Match** column title and select a filter option.

The selected filter option determines what is displayed in the matches table.

| Filter Option | Description |
|---|---|
| **All Data** | Displays all types of matches. |
| **Good Matches** | Displays only 100% perfect matches. |
| **All Issues** | Displays partial matches, multiple matches and no match. |
| **Partial Matches** | Indicates the percentage difference between the strings being matched. For example:<br>• Part of a string is exactly right, such as `Paulo` versus `Sao Paulo`.<br>• Most of a word is exactly right, such as `Caiyro` versus `Cairo`. |
| **Multiple Matches** | Indicates how many matches exist for ambiguous cases.<br>For example, you may see `Barcelona, Spain` matching with `Barcelona Argentina`. In this case you might want to revisit the data to add more detail to your GEO columns to ensure that you only match the correct columns. |
| **No Match** | Indicates that there's no match. |

8. Click in the **Exclude** column for each row of data that you want to exclude.
9. Click the **Exclude** menu:
   - Click **Select All** or **Deselect All**.
   - Click one of **Project Scope**, **Canvas Scope**, or **Visual Scope**.
10. Optionally add in more columns to the Category (Location) edge on the visualization to make your match more specific. For example, add Country data to remove a mismatch like Barcelona, Spain versus Barcelona, Argentina.
11. Display the Location Matches dialog to check the summary for any remaining mismatches, and click **OK** when you are satisfied, or repeat previous steps as required.

## Create Heatmap Layers on a Map Visualization

You can use a heatmap as a data layer type on a map visualization to identify the density or high concentration of point values or metric values associated with the points. For example, you can use a heatmap to identify the high profit stores in a geographic region or country.

You can create two types of heatmap layers:

- **Density heatmap** - Uses only map-related column data (such as latitude and longitude columns). Density heatmap layers show the cumulative sum of a point, where each point carries a specific weight. A point has a radius of influence around it, such that other points that fall in the same area also contribute to the total cumulative result of a point.
- **Metric heatmap** - Uses measure column data in the same layer. For example, if you add a measure column to the Color section on the Grammar Panel the heatmap is updated to show interpolated metric values.

1. Create or open the project where you want to use a heatmap layer on a map visualization. Confirm that you're working in the Visualize canvas.
2. Create an empty map visualization.
3. Drag and drop attribute columns containing map-related data from the Data Panel to the Category (Geography) section on the Grammar Panel.
   - If you're creating a project with a map visualization, in the Data Panel, right-click an attribute column and click **Pick Visualization** then select **Map**.
4. Go to the Data Layers tab of the properties pane.
   - Alternatively, click **Layer options** in the Category (Geography) section and click **Manage Layers**.
5. To create a density heatmap, click **Layer Type** value and select **Heatmap**.
   - Alternatively, you can add a new map layer, change the layer type to **Heatmap** and then add attribute columns to the Category (Geography) section.
6. To create a metric heatmap, drag and drop a metric column from the Data Panel to the Color section. The heatmap visualization changes from density to metric.
7. In the Data Layers tab of the properties pane, specify the options for the heatmap layer such as Radius, Interpolation, Transparency, and Color.
   - The default interpolation method is automatically selected based on the aggregation rule of the metric column or value that you've selected for the layer.
   - You can select the radius value in pixels (px). The radius value is the extent of influence of a measure around a point value on a map.

The heatmap is automatically updated based on the options selected in the Data Layers tab.

## Create Cluster Layers on a Map Visualization

You can use a cluster layer as a type of data layer on a map visualization. In a cluster layer, points that are positioned near each other are grouped together into a common bubble.

The number of points clustered in the group is indicated in the bubble's label. If selected points are grouped with unselected points, the circle is dotted to indicate a partial selection. Individual points are displayed as pin icons to emphasize the distinction between grouped and ungrouped points. Points are grouped based on their proximity in pixels and on the map's zoom factor.

1. Create or open the project where you want to use a point cluster layer on a map visualization. Confirm that you're working in the Visualize canvas.
2. Create an empty map visualization by dragging the Map visualization from the Data Panel to the canvas.
3. Drag and drop attribute columns containing map-related data from the Data Panel to the Category (Geography) section on the Grammar Panel.

   If you're creating a project with a map visualization, in the Data Panel, right-click an attribute column and click **Pick Visualization** then select **Map**.
4. Click the Data Layers tab of the properties pane.

Alternatively, click **Layer options** in the Category (Geography) section and click **Manage Layers**.

5. To create a point cluster, click **Layer Type** value and select **Cluster**.

   Alternatively, you can add a new map layer, change the layer type to **Cluster** and then add attribute columns to the Category (Geography) section.

The point cluster is automatically updated based on the zoom level.

## Represent Point Data With Custom Icons on a Map

You can use the Shapes edge to represent point data with custom icons in a map visualization.

You can associate a column with the Shape edge to display a custom shape for point data. For example, you can distinguish between cities by displaying them using custom shapes (for example, a square, a triangle, or a currency symbol). You can also change which custom shape you want to apply to one or more data points.

1. Create or open the project with a map visualization that contains point data and confirm that you're working on the Visualize canvas.
2. Drag and drop an attribute column containing point data (for example, city) from the Data Panel to the Category (Geography) edge on the Grammar Panel.
3. Drag and drop a column from the Data Panel to the **Shapes** edge and optionally to the **Color** edge on the Grammar Panel.

   The map visualization automatically updates based on your selection and overlays the previous layer.
4. (Optional) You can change how you assign custom shapes to data points and to the map legend.
   a. Highlight one or more data points on the map using one of the selection tools, or use Ctrl-click to select one or more data points.
   b. Right-click one of the multiple-selected data points, select **Custom Shapes** and then select **Series** or **Data Point**.
   c. Select a custom shape and click **Done**.

      Custom shapes are applied as follows:

      - **Series dialog** - custom data point shape not previously set
        Replaces highlighted data points and series items with the custom shape.
      - **Series dialog** - custom data point shape previously set
        Replaces only corresponding series items with the custom shape.
      - **Data Point dialog**
        Replaces only highlighted data points with the custom shape.
5. If you want to reassign the custom shape for a data point:
   a. Right-click any data point, select **Shape**, and click **Custom Shapes**.
   b. To change the custom shape assigned to a data point, click the shape corresponding to the data point that you want to change.
   c. Select a new custom shape and click **Done**, then click **Done** again.

6. If you want to reset all of the custom shapes currently applied to data points on a map visualization, right-click any data point, select **Shape**, and click **Reset Custom Shapes**.

   This resets all of the shapes applied to data points on the map to the default setting.

## Select Points or Area on a Map

You can select multiple points on the map in a specific area that you define using the selection tools.

1. Create or open the project with map visualizations and confirm that you're working on the Visualize canvas
2. Select the map visualization.
3. On the visualization toolbar, do one of the following:
   - Click the **Rectangle Selection** tool and drag a rectangle on the map to select the points or area you want.
   - Click the **Radial Selection** tool and select a point on the map, then drag outward to create a circle. The unit shows the total distance covered on the map.
   - Click the **Polygon Selection** tool and drag a freehand border around the points or area you want to select on the map.

   The selected points or area is highlighted on the map.

## Represent Line Data Using Size and Color on a Map

You can represent the weight of line data through thickness and color in a map visualization.

You can associate a measure with the Size edge to indicate the relative weight of a line. For example, to compare delays in airline routes, you can display flight routes with varying line thickness, where a thicker line and a darker color correspond to a higher number of delays.

1. Create or open the project with a map visualization that contains line data and confirm that you're working on the Visualize canvas.
2. Select the map visualization that displays the line data.
3. Drag and drop a column containing line data (for example airline routes) from the Data Panel to the Category (Geography) edge on the Grammar Panel.
4. Drag and drop a line measure column from the Data Panel to the **Size** edge and optionally to the **Color** edge on the Grammar Panel.

   The map visualization automatically updates based on your selection and overlays the previous layer.

## Make Maps Available to Users

For visualization projects, administrators make maps available to end users or hide them from end users.

You can include or exclude a map from users.

1. On the Home page, click **Console**.
2. Click **Maps**.
3. Use the **Include** option to make a map layer available to end users or hide it from end users.

   You can hide or display custom map and system map layers.

## Make Map Backgrounds Available to Users

Oracle provides two preconfigured map backgrounds. As an administrator, you can add map backgrounds for use in map visualizations.

1. On the Home page, click **Console**, select **Maps**, and then click **Backgrounds**.
   - To set a background as the default map background, click the **Default** column.
   - To include or exclude a map background as an available option to users, click the **Include** column.
2. To add a map background, click **Add Background** and select a map background from the list.

   Oracle Maps are preconfigured and shipped with the product. Other background maps that you can add are Google Maps. For a map provider other than Oracle (for example, Google), you must obtain a Map API access key for which you may be charged based on your usage.
3. Copy and paste in the appropriate Maps API access key.

   You must sign up with the provider to be able to add and use any of these map types.
   - To use the Google Maps tiles, you must obtain a Google Maps API access key from Google. Google prompts you to enter your Maps API access key and, when applicable, your Google “Client ID”. Usage of the tiles must meet the terms of service specified by Google in the Google Developers Site Terms of Service.
4. Select a default map type if applicable and enter a helpful description if needed.
5. Click **Add** to include the map in the list of available map backgrounds.

# Sort and Select Data in Visualization Canvases

While adding filters to visualizations helps you narrow your focus on certain aspects of your data, you can take a variety of other analytic actions to explore your data (for example, drilling, sorting, and selecting). When you take any of these analytic actions, the filters are automatically applied for you.

Select a visualization and click **Menu** or right-click, then select one of the following analytics actions:

- Use **Sort** to sort attributes in a visualization, such as product names from A to Z. If you’re working with a table view, then the system always sorts the left column first. In some cases where specific values display in the left column, you can’t sort the center column. For example, if the left column is Product and the center column is Product Type, then you can’t sort the Product Type column. To work around this issue, swap the positions of the columns and try to sort again.

- Use **Drill** to drill to a data element and drill through hierarchies in data elements, such as drilling to weeks within a quarter. You can also drill asymmetrically using multiple data elements. For example, you can select two separate year members that are columns in a pivot table, and drill into those members to see the details.
- Use **Drill to [*Attribute Name*]** to directly drill to a specific attribute within a visualization.
- Use **Keep Selected** to keep only the selected members and remove all others from the visualization and its linked visualizations. For example, you can keep only the sales that are generated by a specific sales associate.
- Use **Remove Selected** to remove selected members from the visualization and its linked visualizations. For example, you can remove the Eastern and Western regions from the selection.
- Use **Add Reference Line** to add a reference line to highlight an important fact depicted in the visualization, such as a minimum or maximum value. For example, you can add a reference line across the visualization at the height of the maximum revenue amount.

# Replace a Data Set in a Project

You can replace a data set by re-mapping columns used in a project to columns from a different data set.

As part of replacing a data set, you can review and re-map only those columns that are used in the project and replace them with columns of the same data type in the replacement data set. For example, you can replace a test data set with a production data set, or use a project as a template in which you can replace the data but maintain the added structures, visualizations, and calculations.

The **Replace Data Set** option is available for projects using multiple data sets that aren't joined.

1. Create or open the project where you want to replace the data set.

   Confirm that you're working in the **Visualize** canvas.
2. In the Data Panel pane, right-click the data set and select **Replace Data Set**.
3. In the Replace Data Set dialog, perform the following tasks:
   - Select the data set that replaces the existing data set in the project and click **Select**.
   - Review the mapping of the data between the existing and the new data sets in the data-mapping table. The data-mapping table includes all the data elements used in the project's visualizations, calculations, and filters. The data elements with similar type and names in the two data sets are automatically mapped. In the table, based on data types, the data elements are grouped and sorted alphabetically.
   - In the new data set column, click the drop-down arrow in a cell and select a specific data element to adjust the mapping of the data.
     - Only data elements of the same type are displayed in the data element selection dialog.
     - You can navigate back to select a different data set.
4. Click **Replace**.

The new data set replaces the existing data set in the project. You see a notification if you've selected a data set that is joined to other data sets in the project. Review and adjust the joins in the project's **Data Diagram**.

In the data-mapping table based on the selection, the data is updated throughout the project. For example, if you map a data element to **None**, the specific data is removed from the visualizations, calculations, and filters.

# Remove a Data Set from a Project

You can remove a data set from a project.

Removing data from a project, which only impacts the data for that project, differs from permanently deleting the data set from your system.

1. Open your project and in the Data Panel, select the data set that you want to remove.
2. Right-click and select **Remove from Project**.

# Analyze Data with Explain

Explain uses machine learning to find useful insights about your data.

**Topics:**



## What is Explain?

Explain analyzes the selected column within the context of its data set and generates text descriptions about the insights it finds. Explain creates corresponding visualizations that you can add to your project's canvas.

Video

Explain uses Oracle's machine learning to generate accurate, fast, and powerful information about your data. To use Explain, choose a column in your data set and select the Explain option. Explain automatically applies machine learning's statistical analysis to find the most significant patterns, correlations (drivers), classifications, and anomalies in your data. Explain then returns visualizations displaying the insights it found. Users can select a visualization to open the project editor and customize the visualizations and drill further into the data.

For example, suppose you want to look for information about your company's employee attrition. You create a project using a data set that contains attrition information and various profile attributes about employees who have left the organization compared to employees who are still in the organization. Select Explain for the Attrition column, and Explain reveals that one of the key drivers of employee attrition is marital status.

Explain is for data analysts who might not know what data trends they're looking for, and don't want to spend time experimenting by either dragging and dropping columns onto the canvas, or using data flows to train and apply predictive models.

Explain is also a useful starting point for data analysts to confirm a trend that they're looking for in their data, and then use that information to create and tune predictive models to apply to other data sets.

## What Are Insights?

Insights are categories that describe the selected column within the context of its data set.

The insights that Explain delivers are based on the column type or aggregation that you chose and will vary according to the aggregation rule set for the chosen metric. Explain generates only the insights that makes sense for the column type that you chose.

| Insight Type | Description |
|---|---|
| Basic Facts | Displays the basic distribution of the column's values. Column data is broken down against each of the data set's measures.<br>This insight is available for all column types.<br>• For a selected metric, this insight shows the distribution of the aggregated metric value for each member of each attribute column.<br>• For a selected attribute, this insight shows the value of each metric in the data set across the member values of the attribute. |
| Key Drivers | Shows the columns in the data set that have the highest degree of correlation with the selected column outcome. Charts display the distribution of the selected value across each correlated attributes value.<br>This tab displays only when explaining attribute columns, or when explaining a metric column that has an average aggregation rule. |
| Segments | Displays the key segments (or groups) from the column values. Explain runs a classification algorithm on the data to determine data value intersections and identifies ranges of values across all dimensions that generate the highest probability for a given outcome of the attribute. For example, a group of individuals of a certain age range, from a certain set of locations, with a certain range of years of education form a segment that has a very high probability of purchasing a given product.<br>This tab displays only when explaining attribute columns. |
| Anomalies | Identifies a series of values where one of the (aggregated) values deviates substantially from what the regression algorithms expect. |

## Use Explain to Discover Data Insights

When you select a column and choose the Explain feature, Oracle Analytics uses machine learning to analyze the column in the context of the data set. For example, Explain searches the selected data for key drivers and anomalies.

Explain displays its findings to you as text descriptions and visualizations. You can select key visualizations and add them to your project's canvas.

Tutorial

Explain is available after you install machine learning into your local Oracle Analytics Desktop directory. See How do I install Machine Learning for Desktop?

1. In the Home page, click **Create** and then **Project** to create a new project.
2. Click **Visualize** to open the Visualize canvas.
3. In the Data Panel, right-click a column and select **Explain <Data Element>**.

   For Explain to successfully analyze an attribute, the attribute must have three to 99 distinct values.

   The Explain dialog displays basic facts, anomalies, and other information about the selected column.
4. (Optional) In the Segments view, select the segments (or groups) that predict outcomes for the column you selected.
   - Click one or more columns to see how they impacts the column's outcome.
   - Sort how the information is displayed in the Segments. For example, confidence high to low or low to high.
5. (Optional) If your results contain too many correlated and highly ranked columns (for example, ZIP code with city and state), then excluding some columns from the data set so that Explain can identify more meaningful drivers.

   To do this, exit Explain, go to the Prepare canvas and either hide or delete columns, return to the Visualize canvas, locate and right-click the column, and select **Explain <Data Element>**.
6. For each visualization that you want to include in your project's canvas, hover over it and click its checkmark.
7. After you've selected all the visualizations that you want to include in the canvas, click **Add Selected**. You can manage the Explain (data insight) visualizations like any other visualizations you've manually created on the canvas.

# About Warnings for Data Issues in Visualizations

You see a data warning icon when the full set of data associated with a visualization isn't rendered or retrieved properly. If the full set of data can't be rendered or retrieved properly, then the visualization displays as much data as it can as per the fixed limit, and the remaining data or values are truncated or not displayed.

7

# Create and Apply Filters

This topic describes how you can use filters to find and focus on the data you want to explore.

**Topics:**



## Typical Workflow to Create and Apply Filters

Here are the common tasks for creating and applying filters to projects, visualizations, and canvases.

| Task | Description | More Information |
|---|---|---|
| Choose the appropriate filter type | Filter types (Range, Top / Bottom N filter, List, Date, and Expression) are specific to either a project, visualization, or canvas. | Apply Range Filters<br>Apply Top Bottom N Filters<br>Apply List Filters<br>Apply Date Range Filters |
| Create filters on projects and visualizations | Create filters on a project or visualization to limit the data displayed and focus on a specific section or category. | Create Filters on a Project<br>Create Filters on a Visualization |
| Build and use expression filters | You can build and use expression filters to define more complex filters using SQL expressions. | Build Expression Filters |

| Task | Description | More Information |
|---|---|---|
| Set visualization interaction properties | Define how you want visualizations to affect each other. | How Data Sets Interact with Filters |

# About Filters and Filter Types

Filters reduce the amount of data shown in visualizations, canvases, and projects.

The Range, List, Date, and Expression filter types are specific to either a visualization, canvas, or project. Filter types are automatically determined based on the data elements you choose as filters, but you can also switch to select another compatible filter type.

- **Date filters** - Use calendar controls to adjust time or date selections. You can either select a single contiguous range of dates, or you can use a date range filter to exclude dates within the specified range. See Apply Date Range Filters.
- **Expression filters** - Let you define more complex filters using SQL expressions. See Build Expression Filters.
- **List filters** - Applied to data elements that are text data types, number data types that aren't aggregatable, and date data types. List filters are applied to the whole canvas or to a specific visualization on the canvas. See Apply List Filters.
- **On-canvas filters** - Applied to attribute data elements. This filter type is added to and displays on the canvas and allows users to choose how to filter visualizations on the canvas. See Add an On-Canvas Filter.
- **Range filters** - Generated for data elements that are number data types and that have an aggregation rule set to something other than none. Range filters are applied to data elements that are measures, and that limit data to a range of contiguous values, such as revenue of $100,000 to $500,000. Or you can create a range filter that excludes (as opposed to includes) a contiguous range of values. Such exclusive filters limit data to noncontiguous ranges (for example, revenue less than $100,000 or greater than $500,000). See Apply Range Filters.
- **Top and bottom filters** - Applied to measure and attribute data elements. You can specify whether to filter by top or bottom, specify the number of items to display, and which measure or attribute to limit by. See Apply Top Bottom N Filters.

# How Data Sets Interact with Filters

There are several ways that data sets can interact with filters in a project. For example, filters might interact differently with visualizations depending on the number of data sets, whether the data sets are joined, and what the filters are applied to.

Various factors affect the interaction of data sets and filters in projects:

- The number of data sets within a project.
- The data sets that are joined (connected) or not-joined (for a project with multiple data sets).
- The data elements (columns) that are matched between joined data sets.

You can use the Data Diagram in the Prepare canvas of a project to:

- See joined and not-joined data sets.
- Join or connect multiple data sets by matching the data elements in the data sets.
- Disconnect the data sets by removing matched data elements.

# How the Number of Data Sets Affects Filters

Filters can interact differently with visualizations depending on the number of data sets, whether the data sets are joined, and what the filters are applied to.

You can add filters to the filter bar or to individual visualizations in a project.

| Single Data Set | Filter Interaction |
|---|---|
| Add a filter to the filter bar | It applies to all visualizations in the project. |
| Add a filter to a visualization | It is applied after filters on the filter bar are applied. |
| Add multiple filters | By default filters restrict each other based on the values that you select. |

| Multiple Data Sets | Filter Interaction |
|---|---|
| If you add filters to the filter bar. | • The filters apply to all the visualizations using the joined data sets. For visualizations using the not-joined data sets, you must add a separate filter to each data set.<br>• You can't specify data elements of a data set as a filter of other data sets, if the two data sets aren't joined.<br>• If a data element of a data set is specified as a filter, but doesn’t match the joined data sets, then the filter applies only to the visualization of that particular data set, and does not apply to other visualizations of joined or not-joined data sets.<br>• You can select **Pin to All Canvases** of a filter, to apply a filter to all canvases in the project. |
| If you hover over a filter name to see the visualization to which the filter is applied. | Any visualizations that don't use the data element of the filter are grayed out. |
| If you add filters to visualizations | • If you specify a filter on an individual visualization, that filter applies to that visualization after the filters on the filter bar are applied.<br>• If you select the **Use as Filter** option and select the data points that are used as a filter in the visualization, then filters are generated in the other visualizations of joined data sets and matched data elements. |

You can use the **Limit Values By** options to remove or limit how the filters in the filter bar restrict each other.

# Synchronize Visualizations in a Project

You can specify whether or not to synchronize Visualizations in a canvas.

You use the **Synchronize Visualizations** setting to specify how the visualizations on your canvas interact. By default, visualizations are linked for automatic synchronization. You can deselect **Synchronize Visualizations** to unlink your visualizations and turn automatic synchronization off.

When **Synchronize Visualizations** is on (selected), then all filters on the filter bar and actions that create filters (such as Drill) apply to:

- All the visualizations in a project with a single data set.
- All the visualizations of joined data sets with multiple data sets.

  If a data element from a data set is specified as a filter but isn't matched with the joined data sets, then the filter only applies to the visualization of the data set that it was specified for.

When you hover the mouse pointer over a visualization to see the filters applied to the visualization, any filter that isn't applied to the visualization is grayed out.

Any visualization-level filters are applied only to the visualization.

When **Synchronize Visualizations** is off (deselected), then analytic actions such as Drill affect the visualization to which you applied the action.

# About Automatically Applied Filters

By default, the filters in the filter bar and filter drop target are automatically applied. However, you can turn this behavior off if you want to manually apply the filters.

When the **Auto-Apply Filters** is selected in the filter bar menu, the selections you make in the filter bar or filter drop target are immediately applied to the visualizations. When **Auto-Apply Filters** is off or deselected, the selections you make in the filter bar or filter drop target aren’t applied to the canvas until you click the **Apply** button in the list filter panel.

# Create Filters on a Project

You can add filters to limit the data that’s displayed in the visualizations on the canvases in your project.

If your project contains multiple data sets and some aren’t joined, then there are restrictions for how you can use filters. Any visualization that doesn't use the data element of the filter is grayed out.

Instead of or in addition to adding filters to the project or to an individual canvas, you can add filters to an individual visualization.

1. Click **+ Add Filter**, and select a data element. Alternatively, drag and drop a data element from the Data Panel to the filter bar.

   You can't specify data elements of a data set as a filter of other data sets, if the two data sets aren’t joined.
2. Optionally, click the filter **Menu** and hover the mouse pointer over the **Filter Type** option to select an alternative compatible filter type.
3. Set the filter values. How you set the values depends upon the data type that you’re filtering.
   - Apply a range filter to filter on columns such as Cost or Quantity Ordered.
   - Apply a list filter to filter on columns such as Product Category, Product Name, or Date.
   - Apply a date filter to filters on columns such as Ship Date or Order Date.
4. Optionally, click the filter bar menu or right-click, then select **Add Expression Filter**.

5. Optionally, click the filter **Menu** and hover the mouse pointer over the **Limit Values By** option to specify how the filter interacts with the other filters in the filter bar. Note the following:
   - By default, the **Auto** option causes the filter to limit other related filters in the filter bar.
     For example, if you've filters for Product Category and Product Name, and if you set the Product Category filter to Furniture and Office Supplies, then the options in the Product Name filter value pick list is limited to the product names of furniture and office supplies. You can select **None** to turn this limit functionality off.
   - You can specify any individual filter in the filter bar that you don't want to limit. For example, if you have filters for Product Category, Product Sub Category, and Product Name, and in the **Limit Value By** option for the Product Category filter you click Product Sub Category, then the product subcategory filter shows all values and not a list of values limited by what you select for Product Category. However, the values shown for Product Name is limited to what you select for Product Category.
6. Optionally, click the filter bar menu or right-click and select **Auto-Apply Filters**, then click **Off** to turn off the automatic apply. When you turn off the automatic apply, then each filter's selection displays an **Apply** button that you must click to apply the filter to the visualizations on the canvas.
7. Click the filter bar menu or right-click and select **Pin to All canvases** of a filter to apply a filter to all canvases in the project.

You can also go to the filter bar and perform the following steps:

- Select a filter and right-click, then select **Delete** to remove it from the project.
- Right-click and select **Clear All Filter Selections** to clear the selection list of all the filters in the filter bar.
- Right-click and select **Remove All Filters** to remove all the filters in the filter bar.

# Create Filters on a Visualization

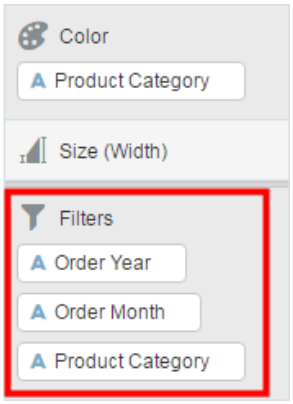
You can add filters to limit the data that's displayed in a specific visualization on the canvas.

If a project contains multiple data sets and some aren't joined, then there are restrictions for how you can use filters. Any visualization that doesn't use the data element of the filter is grayed out.

Visualization filters can be automatically created by selecting **Drill** on the visualization's **Menu** when the **Synchronize Visualizations** option is turned off in the **Canvas Properties** dialog.

Instead of or in addition to adding filters to an individual visualization, you can add filters to the project or to an individual canvas. Any filters included on the canvas are applied before the filters that you add to an individual visualization.

1. Confirm that you're working in the Visualize canvas.
2. Select the visualization that you want to add a filter to.
3. Drag and drop one or more data element from the Data Panel to the Filter drop target in the Grammar Panel.




To use data elements of a data set as a filter in the visualization of another data set, you've to join both the data set, before using the data elements as filters.

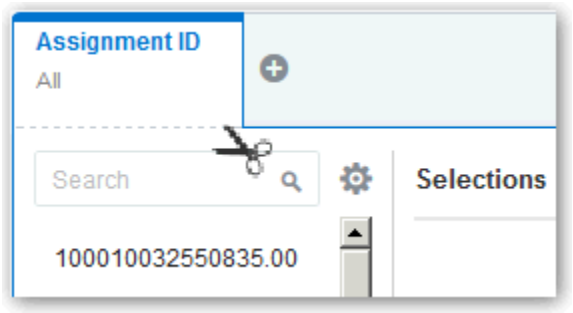
4. Set the filter values. How you set the values depends upon the data type that you're filtering.
   - To set filters on columns such as Cost or Quantity Ordered, see Apply Range Filters.
   - To set filters on columns such as Product Category or Product Name, see Apply List Filters.
   - To set filters on columns such as Ship Date or Order Date, see Apply Date Range Filters.
5. (Optional) Click the filter bar menu or right-click and click **Auto-Apply Filters**, then select **Off** to turn off automatic apply for all filters on the canvas and within the visualization. When you turn off automatic apply, then each filter's selection displays an **Apply** button that you must click to apply the filter to the visualization.

# Move Filter Panels

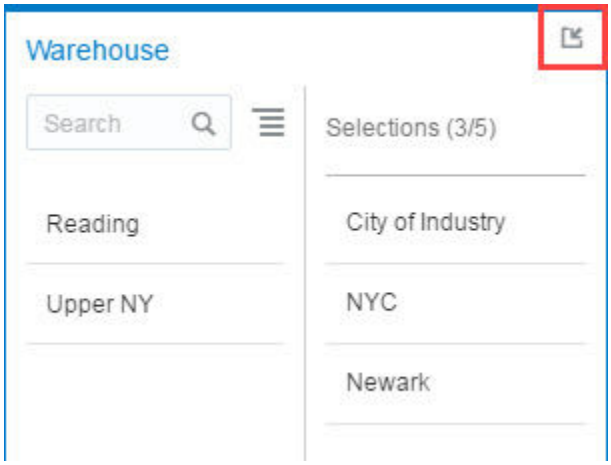
You can move filter panels from the filter bar to a different spot on the canvas.

When you expand filters in the filter bar, it can block your view of the visualization that you're filtering. Moving the panels makes it easy to specify filter values without having to collapse and reopen the filter selector.

1. To detach a filter panel from the filter bar, place the cursor at the top of the filter panel until it changes to a scissors icon, then click it to detach the panel and drag it to another location on the canvas.



2. To reattach the panel to the filter bar, click the reattach panel icon.




# Apply Range Filters

You use Range filters for data elements that are numeric data types and have an aggregation rule set to something other than none.

Range filters are applied only to measure columns and limits data to a range of contiguous values, such as revenue of $100,000 to $500,000. Alternatively, you can create a range filter that excludes (as opposed to includes) a contiguous range of values. Such exclusive filters limit data to two noncontiguous ranges (for example, revenue of less than $100,000 or greater than $500,000).

1. In the Visualize canvas, go to the filter bar and click the filter to view the Range list.
2. In the Range list, click **By** to view the selected list of Attributes. You can optionally perform any of the following steps:
   - Click a member to remove or add it to the selected list.
   - Click the **Plus (+)** icon to add a new member to the selected list.
   - Set the range that you want to filter on by moving the sliders in the histogram.
3. Click outside of the filter to close the filter panel.

# Apply Top Bottom N Filters

Use the Top Bottom N filter to filter a measure or attribute and display its highest or lowest values.

1. To apply the Top Bottom N filter to the canvas and all visualizations in the project:
   a. In the Data Panel, locate the attribute or measure that you want to filter on and drag and drop it to the filter bar.
   b. Click the filter menu and select **Filter Type**, then click **Top Bottom N**. You can only convert a range filter to Top Bottom N filter.
2. To apply the Top Bottom N filter to a specific visualization in the project:
   a. In the canvas, select the visualization that you want to filter.

b. In the Data Panel, locate the attribute or measure that you want to filter on and drag and drop it to the Filter drop target in the Grammar Panel.

c. In the Filter drop target, click the attribute or measure's name, hover over the **Filter Type** option, then click **Top Bottom N**.

3. To change the filter method (top or bottom), go to the **Method** value and click Top or Bottom to choose the method you want.
4. To specify the number of rows that are displayed, click the **Count** field and enter the number of rows.
5. To change which attribute or measure column to limit by, click the **By** field and select an attribute or measure or custom calculation included on the canvas. Or click **Plus (+)** to locate and select the attribute or measure or metric that you want to limit by.
6. Click outside of the filter to close the filter panel.

# Apply List Filters

List filters are applied to text, non-aggregatable numbers, and dates. After you add a list filter, you can change the selected members that it includes and excludes.

1. In the Visualize canvas, go to the filter bar and select a filter to view the Selections list.
2. Locate the member you want to include and click it to add it to the Selections list. Alternatively, use the **Search** field to find a member you want to add to the filter. Use the wildcards * and ? for searching.
3. Optionally, you can also perform the following steps:
   - In the Selections list click a member to remove it from the list of selections.
   - In the Selections list, you can click the eye icon next to a member to cause it to be filtered out but not removed from the selections list.
   - In the Selections list, you can click the actions icon at the top, and select **Exclude Selections** to exclude the members in the Selections list.
   - Click **Add All** or **Remove All** at the bottom of the filter panel to add or remove all members to or from the Selections list at one time.
4. Click outside of the filter to close the filter panel.

# Apply Date Range Filters

Date range filters use calendar controls to adjust time or date selections. You can select a single contiguous range of dates, or use a date range filter to exclude dates within the specified range.

1. In the Visualize canvas, go to the filter bar and click the filter to view the Calendar Date list.
2. In **Start**, select the date that begins the range that you want to filter.

   Use the **Previous** arrow and **Next** arrow to move backward or forward in time, or use the drop-down lists to change the month or year.
3. In **End**, select the date that ends the range that you want to filter.

4. Optionally, to start over and select different dates, right-click the filter in the filter bar and select **Clear Filter Selections**.
5. Click outside of the filter to close the filter panel.

# Apply Relative Time Filters

Use the relative time filter on a Date or Date/Time column to display data for a specified time period based upon the current date and time.

You can specify a relative time period as either an explicit number of past or future time units (for example 2 years), or you can specify a previous period. For example, Year To Date which includes data from 1-January this year to the current date, and Month To Date which includes data from the beginning of the month to the current date.

You can apply a Relative Time filter only to date columns that already exist in the data source, and not to derived columns such as Year, or Quarter. The Relative Time filter type supports Date (with no time of day portion) and DateTime (that is, TIMESTAMP with both date and time of day) column types.

The current date and time used in queries is the Oracle Analytics server host's date and time in the server's timezone (not the browser host's time or timezone). The starting day of the week (Sunday versus Monday) is based on the locale of the Oracle Analytics server which is set using the server configuration setting NLS_TERRITORY.

1. In the Visualize canvas, go to the filter bar and add a date range filter.
2. Click the filter context menu, select **Filter Type**, and **Relative Time** to switch to a relative time filter.
3. In the **Relative Time** filter, select a **Type** that defines the range that you want to filter.
    - **Last** - You specify a **Period** to apply to the selected **Time Level** (Years, Quarters, Months, Weeks, Days, and includes Hours, Minutes, and Seconds if the column time is TIMESTAMP) relative to today's date, to display records for the date values for that period.
      Last filters that are based on a DateTime column and which have a grain of Day or longer (for example, Year, Quarter, Month, Week, Day), retrieve data from the same time of day on the starting day. For example, if the server date/time is currently Thursday 3:15pm , a Last 2 Days filter on a DateTime column retrieves data with timestamps between Tuesday 3:15pm and Thursday 3:15pm in the server's timezone. Filter queries that are based on a DATE column type (which by definition have no time of day associated) only depend on the server host's date, not the time of day.
    - **Next** - You specify a future **Period** number to apply to the selected **Time Level** (Years, Quarters, Months, Weeks, Days, also Hours, Minutes, and Seconds if the column time is TIMESTAMP) relative to today's date, to display records for the date values for that period.
    - **To Date** - You specify a past **Time Level** (Year, Quarter, Month, Week, Day, also includes Hour, and Minute if the column time is TIMESTAMP) relative to today's date, that you want to apply to the date values in the visualization.
      A To Date filter retrieves data from the beginning of the chosen period, for example, Month to Date retrieves data from midnight of the first day of this month up until the current date and time (that is, Today or Now).

4. Click outside of the filter to close the filter panel.

# Build Expression Filters

Using expression filters, you can define more complex filters using SQL expressions. Expression filters can reference zero or more data elements.

For example, you can create the expression filter "Sample Sales"."Base Facts"."Revenue" < "Sample Sales"."Base Facts"."Target Revenue". After applying the filter, you see the items that didn't achieve their target revenue.

You build expressions using the Expression Builder. You can drag and drop data elements to the Expression Builder and then choose operators to apply. Expressions are validated for you before you apply them.

1. In the Visualize canvas, hover over the filter bar at the top of the pane and click **Menu**, then select **Add Expression Filter**.
2. In the Expression Filter panel, compose an expression.
3. In the **Label** field, give the expression a name.
4. Click **Validate** to check if the syntax is correct.
5. When the expression filter is valid, then click **Apply**. The expression is applied to the visualizations on the canvas.

# Add an On-Canvas Filter

You can add a filter directly to the canvas so that users can choose and display the results for some or all of the visualizations on the canvas. This filter is available for attribute columns.

The on-canvas filters that you add are displayed in the Visualize canvas, Narrate canvas, and in presentation mode.

By default the on-canvas filter's values are limited by other filters (canvas, on-canvas, and visualization filters). You can use the Properties pane to modify how the filters interact.

1. Confirm that you're working on the Visualize canvas.
2. Go to the Data Panel, locate the attribute that you want to filter on, and drag and drop it to the canvas to create a new visualization.
3. On the visualization's toolbar, click **Change Visualization Type** and select **List Box**.
4. (Optional) Go to the filter's list box Properties pane, click the Filters tab, and specify how you want the on-canvas filter to interact with the canvas and visualization filters and which items on the canvas to apply the filter to.
5. (Optional) Go to the list box's General pane and adjust how you want the filter to display (for example, **Title**, **Label** and **Align**) and how you want it to function (for example, **Multi Select** and **Default Value**).

8

# Use Other Functions to Visualize Data

This topic describes other functions that you can use to visualize your data.

**Topics:**



## Typical Workflow to Prepare, Connect, and Search Artifacts

Here are the common tasks for using available functions to prepare, connect, and search artifacts.

| Task | Description | More Information |
|---|---|---|
| Build stories | Capture the insights that you discover in your visualizations into a story that you can revisit later, include in a presentation, or share with team members. | Build Stories |
| Manage custom plug-ins | Upload, download, search for, and delete custom plug-ins that you can use to customize various objects such as visualizations types or projects. | Manage Custom Plug-ins |
| Compose expressions | Compose expressions to use in filters or in calculations. | Compose Expressions |
| Create and apply data actions | Create data action links to pass context values from canvases to URLs or project filters. | Use Data Actions to Connect to Canvases and External URLs and Use in External Containers |

# Build Stories

This topic covers how you capture insights and group them into stories.

**Topics:**



## Capture Insights

As you explore data in visualizations, you can capture memorable information in one or more insights, which build your story. For example, you might notice before and after trends in your data that you'd like to add to a story to present to colleagues.

Tutorial

Using insights, you can take a snapshot of any information that you see in a visualization and keep track of any moments of sudden realization while you work with the data. You can share insights in the form of a story, but you don't have to. Your insights can remain a list of personal moments of realization that you can go back to, and perhaps explore more. You can combine multiple insights in a story. You can also link insights to visualizations using the Interaction property.

Insights don't take a snapshot of data. They take a snapshot of the project definition at a certain point in time.

1. Display the Narrate pane, and build your story:
   - Use the **Search** option in the Canvases pane to locate visualizations to include in your story. Right-click each canvas to include and click **Add To Story**.
   - Click **Add Note** to annotate your canvases with insights, such as notes or web links.
   - Use the tabs on the properties pane to further refine your story. For example, click **Presentation** to change the presentation style from compact to film strip.
   - To synchronize your story canvases with your visualizations, display the Visualize pane, click **Canvas Settings**, then select **Synchronize Visualizations**. Alternatively, click **Canvas Properties** and select this option.
   - To synchronize your story canvases with your visualizations, display the Visualize pane and right-click the canvas tabs to select **Canvas Properties**, then change the **Synchronize Visualizations** setting to on (selected).
2. Continue adding insights to build a story about your data exploration.

   The story builds in the Narrate canvas. Each insight has a tab.

## Create Stories

After you begin creating insights within a story, you can cultivate the look and feel of that story. For example, you can rearrange insights, include another insight, or hide an insight title. Each project can have one story comprising multiple pages (canvas).

1. In your project, click **Narrate**.
2. Create the story in the following ways:
   - Add one or more canvases to the story and select a canvas to annotate.
   - To annotate a story with insights, click **Add Note**. You can add text and web links.
   - To change the default configuration settings for a story, use the properties pane on the Canvases panel.
   - To edit an insight, click or hover the mouse pointer over the insight, click the menu icon, and select from the editing options.
   - To include or exclude an insight, right-click the insight and use the **Display** or **Hide** options. To display insights, on the canvas property pane, click **Notes**, then **Show All Notes**.
   - To show or hide insight titles or descriptions, on the canvas property pane, click **General**, and use the **Hide Page** and **Description** options.
   - To rearrange insights, drag and drop them into position on the same canvas.
   - To limit the data displayed in a story, on the canvas property pane, click **Filters**. If no filters are displayed, go back to the Visualize pane and add one or more filters first, then click **Save**.
   - To update filters for a story, on the canvas property pane, click **Filters**, and use the options to hide, reset, or selectively display filters.
   - To rename a story, click the story title and update.
   - To add the same canvas multiple times to a story, right-click a canvas and click **Add to Story**. You can also right-click the canvases at the bottom of the Narrate pane and click **Duplicate**.
   - To display the story at any time click **Present**.
   - To close present mode and return to the **Narrate** pane click **X**.
   - To toggle insights use the **Show Notes** option.

   You can modify the content on a canvas for an insight. For example, you can add a trend line, change the chart type, or add a text visualization. After changing an insight, you'll notice that its corresponding wedge (in the Insight pane) or dot (in the Story Navigator) changes from solid blue to hollow. When you select **Update** to apply the changes to the insight, you'll see the wedge or dot return to solid blue.

## View Streamlined Content

You can use the presentation mode to view a project and its visualizations without the visual clutter of the canvas toolbar and authoring options.

1. On the Narrate toolbar, click **Present**.

2. To return to the interaction mode, click **X**.

# Add Notes

You can add notes to a canvas on the Narrate pane to refer to a specific spot or data point such as a column in a table, a particular horizontal bar, or a cluster in a Scatter plot.

1. In your project, click **Narrate** and select the canvas where you want to add notes.

   Alternatively, in the Canvases Panel, select a canvas and click **Canvases List** or right-click, then click **Add to Story**. You can also drag and drop the canvas from the Canvases Panel to the Story page.

2. Select a data point or spot on the visualization where you want to add a data reference annotation and click **Add Note**.

   Alternatively, click **Add Note** to annotate your canvas with insights.

3. Enter the text you want to show in the note.
4. In the format text dialog, do the following:
   - Define the format of the note text.
   - Click **Link** if you want to insert a web address in the note. In the Hyperlink dialog, enter the web address and click **OK**.
5. Continue adding notes to your canvas.

A connector or line connects the note to the data point or spot on the visualization. If you select more than ten data points, the connectors won't be displayed. To show the connectors select the note body and click the drop-down arrow, then select **Show Connector**. If you change or remove a data point on the Visualize canvas, the note attached to that data point is automatically hidden.
You can’t connect a note to a data point or spot on the following visualization types:

- List
- Tile
- Correlation Matrix
- Map
- Parallel Coordinates
- Chord Diagram

## Edit a Note

You can edit the text or web addresses inserted in notes.

1. Select the canvas where you added the note you want to edit.
2. Click the note on the visualization.

   Alternatively, select the note body and click the drop-down arrow, then click **Edit**.

3. Edit the text or web address in the note.
4. Click outside of the note body.

## Adjust a Note

You can adjust a note after you add it to a canvas on the Narrate pane. For example, you can move the note around on the canvas, connect or detach it from a data point, resize it, or hide it.

1. In the Narrate pane, select the canvas where you added the note that you want to adjust.
2. Select the note body and click the drop-down arrow, then select the appropriate option such as **Duplicate**, **Show Connector**, or **Hide Note**.
3. In the properties pane, click the **Note** tab, and use the options to hide or delete the note.
4. To connect the note to a data point, hover the mouse pointer over the note to see the connection dots be displayed on the note body.

   Click a connection dot and drag the cursor to a data point on the visualization.
5. To alter the data points attached to the note, do the following:
   a. Select the note body and click the drop-down arrow, then select **Detach from Data**.
   b. Select a connection dot and drag the cursor to the new data points.
6. Select a note body and perform the following actions:
   - Drag the selected note to a new position.
   - Drag the note body sizing handle left and right or up and down.

# Identify Content with Thumbnails

You can quickly visually identify content on the Home page and within projects by looking at thumbnail representations.

Project thumbnails on the Home page show a miniature visualization of what projects look like when opened. Project thumbnails are regenerated and refreshed when projects are saved. If a project uses a Subject Area data set, then the project is represented with a generic icon instead of a thumbnail.

# Manage Custom Plug-ins

You can upload, download, search for, and delete custom plug-ins. Plug-ins are custom visualization types or custom data actions that you create externally and then import into your system.

For example, you can upload a custom plug-in that provides a visualization type that you can use in projects.

Tutorial

1. On the Home page, click the **Navigator**, and then click **Console**.
2. Click **Extensions**.

   You use this page to upload, search for, delete, or download a custom plug-in.

3. To upload a custom plug-in, click **Extension** and perform one of the following actions.
   - Browse to the required plug-in file in your file system, and click **Open** to select the plug-in.
   - Drag the required plug-in file to the **Upload Custom Plugin** object.

   If the uploaded custom plug-in file name is the same as an existing custom plug-in, then the uploaded file replaces the existing one and is displayed in visualizations.
4. Perform any of the following tasks.
   - If the plug-in provides a visualization type, you can select that type from the list of available types when you create or switch the type of a visualization.
   - To search for a custom plug-in, enter your search criteria in the **Search** field and click **Return** to display search results.
   - To delete a custom plug-in, click **Options** on the custom plug-in and select **Delete**, and click **Yes** to delete the custom plug-in.
     If you delete a custom visualization type that's used in a project, then that project displays an error message in place of the visualization. Either click **Delete** to remove the visualization, or upload the same custom plug-in so that the visualization renders correctly.
   - To download a custom plug-in, click **Options** on the custom plug-in and select **Download**.

# Compose Expressions

You can use the Expression window to compose expressions to use in expression filters or in calculations. Expressions that you create for expression filters must be Boolean (that is, they must evaluate to true or false).

While you compose expressions for both expression filters and calculations, the end result is different. A calculation becomes a new data element that you can add to your visualization. An expression filter, on the other hand, appears only in the filter bar and can't be added as a data element to a visualization. You can create an expression filter from a calculation, but you can't create a calculation from an expression filter. See Create Calculated Data Elements and Build Expression Filters.

You can compose an expression in various ways:

- Directly enter text and functions in the Expression window.
- Add data elements from the Data Elements pane (drag and drop, or double-click).
- Add functions from the function panel (drag and drop, or double-click).

See Expression Editor Reference.

# Use Data Actions to Connect to Canvases and External URLs and Use in External Containers

A Data Action link can pass context values as parameters to external URLs, filters to other projects or to visualizations embedded in external containers.

When a link navigates to a project, the data context is displayed in the form of canvas scope filters in the filter bar. The links data context may include attributes associated with the selections or cell from which the link was initiated.

**Topics:**



# Create Data Actions to Connect Visualization Canvases

You can create data actions to navigate to a canvas in the current project or to a canvas in another project.

Video

You can also use data actions to transfer context-related information (for example, an order number) where the link displays details about an order number in another visualization or project.

1. Create or open a project and confirm that you're working in the Visualize canvas.
2. Click **Menu** on the project toolbar and click **Data Actions**.
3. Click **Add Action** and enter a name for the new navigation link.
   - You can use only letters and numbers in the navigation link's name.
   - You can add multiple navigation links.
4. Click the **Type** field and select **Analytics Link**.
5. Click the **Anchor To** field and select the columns from the current visualization to associate with this data action. Don't select measure columns or hidden columns. If you don't specify a value for the **Anchor To** field, then the data action applies to all data elements in the visualizations.
6. Click the **Project** field and select the project you want to use for the anchor:
   - **Use This Project** - Select if you want to link to a canvas in the active project. Columns that you select must be in the current visualization.
   - **Select from Catalog** - Select to browse for and select the project that you want to use.
7. Click the **Canvas Link** field and select the canvas that you want to use.
8. Click the **Pass Values** field and select which values you want the data action to pass.

   For example, if in the **Anchor To** field, you specified order number column, then in the **Pass Values** field, select **Anchor Data** to pass the specified column values.
   - **All** - Dynamically determines the intersection of the cell that you click and passes those values to the target.

- **Anchor Data** - Ensures that the data action is displayed at runtime, but only if the required columns specified in the **Anchor To** field are available in the view context.
- **None** - Opens the page (URL or canvas) but doesn't pass any data.
- **Custom** - Enables you to specify a custom set of columns to pass.

9. Click **OK** to save.

## Create Data Actions to Connect to External URLs from Visualization Canvases

You can use data actions to navigate to an external URL from a canvas so that when you select a column such as the supplier ID, it displays a specific external website.

1. Create or open a project and confirm that you're working in the Visualize canvas.
2. Click **Menu** and select **Data Actions**.
3. Click **Add Action** and enter a name for the new navigation link.

   You can add multiple navigation links.
4. Click the **Type** field and select **URL Navigation**.
5. Click the **Anchor To** field and select the columns that you want the URL to apply to. If you don't specify a value for the **Anchor To** field, then the data action applies to all data elements in the visualizations.
6. Enter a URL address and optionally include notation and parameters.

   For example, where `http://www.example.com?q=${keyValuesForColumn:"COLUMN"}` is displayed like `www.oracle.com?q=${keyValuesForColumn:"Sales"."Products"."Brand"}` The column names that you select here are replaced with values when you invoke the data action.
7. Click **OK** to save.
8. In the **Canvas**, click a cell, or use Ctrl-click to select multiple cells.
9. Right-click and select from the menu the navigation name that you created earlier.

   Selecting the cells determines the values to pass to the parameters (that is, the URL tokens).

## Create Data Actions to Connect to REST APIs from Visualization Canvases

You can use HTTP API data actions in a canvas so that when you select a column, it sends the value to a REST API which returns a response from the external website.

You must configure the domain that you're trying to connect to as a safe domain before you create an HTTP API data action. See Register Safe Domains.

1. Create or open a project and confirm that you're working in the Visualize area.
2. Click **Menu** and then select the **Data Actions** tab.
3. Click **Add Action** and enter a name for the new HTTP API data action. For example, enter `HTTP API Example`.

You can add multiple HTTP API data actions.

4. Click the **Type** field and select **HTTP API**.
5. Click the **Anchor To** field and select the columns that you want the HTTP API data action to apply to. Don't select measure columns or hidden columns. If you don't specify a value for the **Anchor To** field, then the data action applies to all data elements in the visualizations.
6. Click the **HTTP Method** field and select an appropriate value (that is, GET, POST, PUT, DELETE) to send to the REST API.
7. Enter the URL for the REST API that starts with http or https and optionally includes replacement tokens.

   Replacement tokens in the URL pass contextual values to a data action (for example, data values, user name, project path, canvas name).

   For example, you can pass a column value (for example, a value for the Category column) to a REST API (for example, the Google books API) using: `http://www.googleapis.com/books/v1/volumes?q=${valuesForColumn:"Category"}` The value that you select from a cell in the Category column (for example, "Books") passes to the REST API.
8. If you selected the POST method, a field is displayed where you enter each parameter on a separate line.

   You enter each parameter as a name-value pair with the name and value separated by "**=**". You can use the same URL token syntax in the name-value pairs as required by the API that you're calling.

   For example:

   - `paramName1=paramValue1`
   - `paramName2=${valuesForColumn:"Product"}`
9. Click **OK** to save.
10. Click a data point in the visualization.

    For example, you might select "Books" from the Category column.
11. Right-click and select **HTTP API Example** to display the result.

    The selected data-points determine which values to pass to the URL tokens.

    A success or failure message is displayed confirming the REST API has been successfully invoked using the selected value.

## Create Data Actions to Connect to Oracle Business Intelligence Publisher Reports

You can create an analytics data action link to transfer selected data points from an Oracle Analytics project to an Oracle Business Intelligence Publisher report.

The Oracle Analytics project, BI Publisher report, and analysis can be in different folders.

1. Open an Oracle Analytics project that uses the data model used in the BI Publisher report.
2. Click **Menu** on the project toolbar and click **Data Actions**.

3. Click **Add Action** and enter a name for the new navigation link.
4. Click the **Type** field and select **Analytics Link**.
5. Enter a name for the data action in the **Name** field.
6. Click the **Target** field, select **Select from Catalog**, then browse to select the BI Publisher report that you want the data action to pass data to, and click **OK**.
7. Verify that **Parameter Mapping** is set to `Default`.
8. Verify the **Pass Values** field value is set to `All`.
9. Click **OK**.
10. Select data points in the visualization and choose the data action to test that the values are passed to the BI Publisher report.

## Invoke Data Actions from Visualization Canvases

You can invoke data actions from a canvas to other canvases and URLs.

1. Create or open a project. Confirm that you're working in the Visualize canvas.
2. On the canvas that contains a Data Action link leading to another canvas or URL, perform the following steps:
   a. Right-click a data element, or select multiple elements (using Ctrl-click).
   b. Select **Data Actions** from the context menu.
   c. Complete the Project Properties dialog.

   The name of the data actions that apply in the current view context are displayed in the context menu.

   All the values defined in the **Anchor To** field must be available in the view context in order for a data action to be displayed in the context menu.

   The following rules apply to matching data elements passed as values with data elements on the target canvas:

   - If the same data element is matched in the target project's canvas, and if the target canvas doesn't have an existing canvas filter for the data element, a new canvas filter is added. If there is an existing canvas filter, it's replaced by the value from the source project's canvas.
   - If the expected data set is unavailable but a different data set is available, the match is made by using the column name and data type in the different data set, and the filter is added to that.
   - If there are multiple column matches by name and data type, then the filter is added to all those columns in the target project or canvas.

   The data action navigates to the target cell or URL that is mapped and filters the data displayed based on the values specified in the Data Actions dialog.
   The Pass Values context, set in the **Pass Values** field, consists of data elements used in the visualization from which the data action is invoked. The Pass Values context doesn't include data elements in the project, canvas, or visualization level filters.

# Visualize Data from the Home Page

Use the search bar on the Home page to find your data and quickly visualize it. You can use this functionality to perform impromptu visualizations without having to first build a project.

1. On the Home page, click the search bar.
2. Enter a search term (for example, "Office") and press SHIFT + ENTER to select the default result or select a specific result displayed in the drop-down list. You'll see your data visualized.
   - What you select determines the data set for the visualization, and all other criteria that you enter is limited to columns or values in that data set.


3. Build your visualization by searching for and selecting other items.
4. When you're happy with your visualization and you'd like to save it or further enhance it, hover over it and click **Explore in Data Visualization**.

# Find Data, Projects, and Visualizations

This topic describes how you can search for items such as objects, projects, and columns.

**Topics:**



## How Is Data Indexed?

When you search or visualize data from the Home page, your results are determined by what information has been indexed.

Every two minutes, the system runs a process to index your saved objects, project content, and data set column information. The indexing process also updates the index file to reflect any objects, projects, or data sets that you deleted from your system so that these items are no longer displayed in your search results.

For all data sets, the column metadata is indexed. For example, column name, the data type used in the column, aggregation type, and so on. Column data is indexed for Excel spreadsheet, CSV, and TXT data set columns with 1,000 or fewer distinct rows. Note that no database column data is indexed and therefore that data isn't available in your search results.

## Search for Content

Use the search bar on the Home page to find items such as data sets, projects, and machine learning scripts.

1. On the Home page, locate the search bar.
2. Enter your search criteria. Note the following options:
   - Click in the search bar for a drop down list of all content types, such as project, report, and data set. Click a content type to add it to the search bar. Or below the search bar, click one of the search tags to add it to the search bar.
   - Build or modify a search tag by adding or removing other items.
   - Specify the full or partial name of what you're looking for. The search is case-insensitive.
   - To clear your search terms, in the search bar click X or select search tags and delete.
3. In the search results, click an object to display it.

## Search Tips

Use these tips to help you find your content.

- **Searching in Non-English Locales** - When you enter criteria in the search field, what displays in the drop-down list of suggestions can differ depending upon your locale setting. For example, if you're using an English locale and enter *sales*, then the drop-down list of suggestions contains items named *sale* and *sales*. However, if you're using a non-English locale such as Korean and type sales, then the drop-down list of suggestions contains only items that are named *sales* and items such as *sale* aren't included in the drop-down list of suggestions.
- **Searching for New Objects and Data** - If you create or save a project or create a data set and then immediately try to search for it, then it's likely that your search results won't contain matches. If this happens, refresh your browser. If you still can't find the new object or data, then wait a few minutes for the indexing process to run, and retry your search. The system automatically runs the indexing process every two minutes. Users can access only the data they've been given permission to access.

# Save Your Changes Automatically

You can use the auto save option to save your updates to a visualization project in real-time automatically.

If you've already saved your project in a specific location, the Save Project dialog isn't displayed after you click **Auto Save**.

1. Create or open a project.

2. From the Save menu, select **Auto Save**.
3. In the Save Project dialog, enter the **Name** and optional **Description** to identify your project.
4. Select the folder where you want to save your project.
5. Click **Save**. Any project updates are saved in real-time.

Suppose that two users are updating the same project and **Auto Save** is enabled. The **Auto Save** option is automatically disabled when different types of updates are made to the project. A message is displayed that states that another user has updated the project.

# Sort the Items in a Page

You can quickly reorganize items in the Catalog, Data, and Machine Learning pages by sorting the items based on their attributes.

As you add more data or projects, sorting items helps you to control the order in which they're listed. For example, you can reorganize data sets on the Data page based on their modification dates and times.

1. Go to the page with the items you want to sort.
2. Click the **Sort By** menu on the page toolbar and select a sort option such as **Modified** or **Reverse Order**.
3. Click the **List View** icon to switch to list view. In the table, click a column header to sort the items in that column in ascending or descending order.

The selected sort and column options are saved as user preferences.

9

# Create Custom Data Action Plug-ins

You can create custom data action plug-ins to use in Oracle Analytics.

Data action plug-ins extend Oracle Analytics and enable users to select data-points in visualizations and to invoke specific actions. Oracle Analytics provides a core set of data actions that cover many common use cases, but by writing your own data action plug-in, you can extend this functionality even further.

The tutorial uses an example to help you understand how to create a custom data action plug-in.

 Tutorial

You must have a basic understanding of the following to create custom data action plug-ins:

- JavaScript
- RequireJS
- JQuery
- KnockoutJS

**Topics:**



## About Data Action Plug-ins and the Data Actions Framework

Data action plug-ins leverage the data actions framework to provide custom, data-driven actions that are tightly integrated into the Oracle Analytics user interface.

When a user invokes a data action, the Data Action Manager passes the request context (for example, qualified data reference, measure values, filters and metadata) to the data action plug-in which is responsible for handling the request. Oracle provides four types of data action plug-ins: `CanvasDataAction`, `URLNavigationDataAction`, `HTTPAPIDataAction` and `EventDataAction`. You can

extend these data action plug-in types along with their abstract base classes to provide your own data actions.

**Topics:**



# Data Action Categories

The data action categories include Navigate to URL, HTTP API, Navigate to Canvas, and Event actions:

- **Navigate to URL**: Opens the specified URL in a new browser tab.
- **HTTP API**: Uses the `GET/POST/PUT/DELETE/TRACE` commands to target an HTTP API and doesn't result in a new tab. Instead the HTTP status code is examined and a transient success or failure message is displayed.
- **Navigate to Canvas**: Enables the user to navigate from a source canvas to a target canvas in either the same or a different visualization. Any filters that are in effect in the source canvas are passed to the target canvas as external filters. When the target canvas opens, it attempts to apply the external filters to the visualization. The mechanism by which external filters are applied isn't described here.
- **Event Actions**: Publishes an event using the Oracle Analytics event router. Any JavaScript code (for example, a third-party plug-in) can subscribe to these events and handle their custom response accordingly. This provides the maximum flexibility because the plug-in developer can choose how the data action responds. For example, they can choose to display a user interface or pass data to multiple services at once.

Both the **Navigate to URL** and **HTTP API** data action category types can use a token syntax to inject data or metadata from the visualization into the `URL` and `POST` parameters.

**URL Token Replacement**

HTTP data actions can replace tokens in URLs with values from the context passed to the data action. For example, qualified data reference values, filter values, username, project path, and canvas name.

| Token | Notes | Replace With | Example | Result |
|---|---|---|---|---|
| `${valuesForColumn:COLUMN}` | NA | Column display values from the qualified data reference. | `${valuesForColumn:"Sales"."Products"."Brand"}` | `BizTech,FunPod` |
| `${valuesForColumn:COLUMN, separator:"/"}` | Any token that can potentially be replaced with multiple values supports the optional separator option. The `separator` defaults to a comma (,) but you can set it to any string. You can escape double quotes inside this string by using a backslash (\). | Column display values from the qualified data reference. | `${valuesForColumn:"Sales"."Products"."Brand"}` | `BizTech,FunPod` |
| `${valuesForColumn:COLUMN, separationStyle:individual}` | Any `separationStyle` defaults to `delimited` but you can set it to `individual` if the user needs to generate separate URL parameters for each value. | Column display values from the qualified data reference. | `&myParam=${valuesForColumn:"Sales"."Products"."Brand"}` | `&myParam=BizTech &myParam=FunPod` |
| `${keyValuesForColumn:COLUMN}` | NA | Column key values from the qualified data reference. | `${keyValuesForColumn:COLUMN}` | `10001,10002` |
| `${env:ENV_VAR}` | Supported environment variables are: `sProjectPath`, `sProjectName`, `sCanvasName`, `sUserID`, and `sUserName`. | An environment variable. | `${env:'sUserID'}` | `myUserName` |

# Data Action Context

You can define a context that is passed when the user invokes a data action.

You define how much of the context is passed to the data action when you create the data action.

### Qualified Data Reference

When the data action is invoked a qualified data reference is generated for each marked data point using an array of `LogicalFilterTree` objects. A `LogicalFilterTree` consists of multiple `LogicalFilterNode` objects arranged in a tree structure. This object includes:

- The attributes on the row or column edges of the data layout.
- The specific measure on the measure edge that addresses each marked cell.
- The specific measure value for each marked cell.
- Key values and display values.

**Environment Variables**

In addition to the data and metadata describing each marked data point, certain data actions may need further context describing the environment from where the data action is invoked. Such environment variables include:

- Project Path
- Project Name
- Canvas Name
- User ID
- User Name

## Data Action Code Design

You create data actions using API classes.

- There are four concrete classes of data action that inherit from the `AbstractDataAction` class:
  - `CanvasDataAction`
  - `URLNavigationDataAction`
  - `HTTPAPIDataAction`
  - `EventDataAction`
- You can create new types of data actions using the data action plug-in API. See Data Visualizer SDK Reference.
- The registry of data action types is managed by the `DataActionPluginHandler`.
- Code that creates, reads, edits, deletes, or invokes instances of data actions does so by publishing events.
- Events are handled by the `DataActionManager`.

## Data Action Model Classes

There are several different types of data action model classes.

**AbstractDataAction**

This class is responsible for:

- Storing the Knockout Model (subclasses are free to extend this with their own properties).
- Defining the abstract methods that subclasses must implement:
  - `+ invoke(oActionContext: ActionContext, oDataActionContext:DataActionContext) <<abstract>>`

Invokes the data action with the passed context - should only be called by the `DataActionManager`.

  - `+ getGadgetInfos(oReport): AbstractGadgetInfo[] <<abstract>>`
    Constructs and returns the `GadgetInfos` responsible for rendering the user interface fields for editing this type of data action.
  - `+ validate() : DataActionError`
    Validates the data action and returns null if valid or a `DataActionError` if it's invalid.

- Providing the default implementation for the following methods used to render generic parts of the data action user interface fields:
  - `+ getSettings():JSON`
    Serializes the data action's Knockout Model to JSON ready to be included in the report (uses komapping.toJS(_koModel)).
  - `+ createNameGadgetInfo(oReport) : AbstractGadgetInfo`
    Constructs and returns the `GadgetInfo` that can render the data action's **Name** field.
  - `+ createAnchorToGadgetInfo(oReport) : AbstractGadgetInfo`
    Constructs and returns the `GadgetInfo` that can render the data action's **Anchor To** field.
  - `+ createPassValuesGadgetInfo(oReport) : AbstractGadgetInfo`
    Constructs and returns the `GadgetInfo` that can render the data action's **Pass Values** field.

Subclasses may not need all of the `GadgetInfos` that the base class provides so they may not need to call all of these methods. By separating out the rendering of each field in this way, subclasses are free to pick and choose the gadgets they need. Some subclasses may even choose to provide a different implementation of these common data action gadgets.

#### CanvasDataAction, URLNavigationDataAction, HTTPAPIDataAction, EventDataAction

These are the concrete classes for the basic types of data actions. These classes work by themselves to provide the generic user interface for these types of data action. They can also act as convenient base classes for custom data action plug-ins to extend.

- **CanvasDataAction**: Used to navigate to a canvas.
- **URLNavigationDataAction**: Used to open a web page in a new browser window.
- **HTTPAPIDataAction**: Used to make a `GET/POST/PUT/DELETE/TRACE` request to an HTTP API and handle the `HTTP Response` programatically.
- **EventDataAction**: Used to publish JavaScript events through the Event Router.

Each class is responsible for:

- Implementing the abstract methods from the base class.
  - `invoke(oActionContext: ActionContext, oDataActionContext:DataActionContext)`
    This method should invoke the data action by combining the properties defined in the KOModel with the specified `DataActionContext` object.

  - `getGadgetInfos(oReport): AbstractGadgetInfo[]`
    This method should:
    * Create an array containing `AbstractGadgetInfos`.
    * Call individual `createXXXGadgetInfo()` methods pushing each `AbstractGadgetInfo` into the array.
    * Return the array.

- Providing the additional methods for creating the individual gadgets that are specific to the particular subclass of data action.

Subclasses of these concrete classes may not need to use all of the gadgets provided by their superclasses in their custom user interfaces. By separating out the construction of each gadget in this way, subclasses are free to pick and choose the gadgets they need.

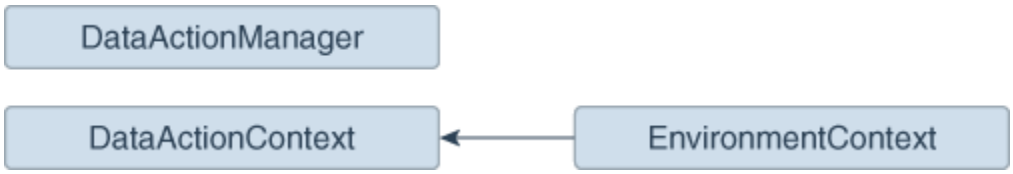
#### DataActionKOModel, ValuePassingMode

The `DataActionKOModel` class provides the base KOModel shared by the different subclasses of `AbstractDataAction`. See DataActionKOModel Class.

## Data Action Service Classes

There are several different data action service classes.

#### DataActionManager




All communication with `DataActionManager` uses `ClientEvents.DataActionManager` which implements event handlers for:

- Managing the set of data actions defined in the current project.
- Invoking a data action.
- Retrieving all the data actions defined in the current project.
- Retrieving all the data actions that are applicable to the current marked data points.

#### DataActionContext, EnvironmentContext

When a data action is invoked, the `DataActionContext` class contains the context that's passed to the target.

- `getColumnValueMap()`
  Returns a map of attribute column values keyed by attribute column names. These define the qualified data reference for the data points that the data action is invoked from.
- `getLogicalFilterTrees()`

Returns a `LogicalFilterTrees` object describing the qualified data references for the specific data points that the data action is invoked from (see the `InteractionService` for details).

- `getEnvironmentContext()`
  An instance of the `EnvironmentContext` class describing the source environment such as:
  - `getProjectPath()`
  - `getCanvasName()`
  - `getUserID()`
  - `getUserName()`
- `getReport()`
  Returns the report that the data action is invoked from.

**DataActionHandler**

The `DataActionHandler` class registers the various data action plug-ins. Its API is broadly consistent with the other plug-in handlers (for example, `VisualizationHandler`).

The `DataActionHandler` class provides the following public methods:

- `getClassName(sPluginType:String) : String`
  Returns the fully qualified class name for the specified data action type.
- `getDisplayName(sPluginType:String) : String`
  Returns the translated display name for the specified data action type.
- `getOrder(sPluginType:String) : Number`
  Returns a number used to sort lists of the types of data action into the preferred order.

The `DataActionHandler` class provides the following static methods:

- `getDependencies(oPluginRegistry:Object) : Object.<String, Array>`
  Returns a dependency map covering all the registered data action types.
- `getHandler(oPluginRegistry:Object, sExtensionPointName:String, oConfig:Object) : DataActionPluginHandler`
  Constructs and returns a new instance of the `DataActionHandler` class.

**DataActionUpgradeHandler**

The `DataActionUpgradeHandler` class is called by the `UpgradeService` when a report is opened.

The `DataActionHandler` class provides two main methods:

- `deferredNeedsUpgrade(sCurrentVersion, sUpgradeTopic, oDataActionJS, oActionContext) : Promise`
  Returns a `Promise` that resolves to a Boolean indicating whether the specified data action must be upgraded (`true`) or not (`false`). The method decides whether the data action must be upgraded by comparing the version number of the data action instance with the version number obtained from the data action's constructor. If the instance's version number is less than the constructor's version number, the data action instance must be upgraded.

- `performUpgrade(sCurrentVersion, sUpgradeTopic, oDataActionJS, oActionContext, oUpgradeContext) : Promise`
  Carries out the upgrade on the specified data action and resolves the `Promise`. The upgrade itself is carried out by calling the `upgrade()` method on the data action (only the specific subclass of data action being upgraded is qualified to upgrade itself).
- `getOrder(sPluginType:String) : Number`
  Returns a number used to sort lists of the types of data action into the preferred order.

# Data Action Code Interactions

A data action interacts with Oracle Analytics code when it creates a user interface field, and when a user invokes a data action.

### Create the Field for a New Data Action Instance

This interaction starts when Oracle Analytics wants to render a data action user interface field. To do so, it:

1. Creates a `PanelGadgetInfo` that acts as the parent `GadgetInfo` for the `GadgetInfos` that the data action returns.
2. Calls `getGadgetInfos()` on the data action.
3. Adds the data action's `GadgetInfos` as children of the `PanelGadgetInfo` created in the first step.
4. Creates the `PanelGadgetView` that renders the `PanelGadgetInfo`.
5. Sets the `HTMLElement` that's the container of the `PanelGadgetView`.
6. Registers the `PanelGadgetView` as a child `HostedComponent` of a `HostedComponent` that's already attached to the `HostedComponent` tree.
   This renders the data action's gadgets inside the Panel gadget in the order they appear in the array returned by `getGadgetInfos()`.

### Invoke a Data Action

This interaction starts when the user invokes a data action through the Oracle Analytics user interface (for example, from the context menu on a data point in a visualization).

In response to the user interaction, the code:

1. Publishes an `INVOKE_DATA_ACTION` event containing the data action's ID, the `DataVisualization` that the data action is invoked from, and a `TransientVizContext` object.
2. The `DataActionManager` handles this event by:
   a. Obtaining the data action instance from its ID.
   b. Obtaining the `LogicalFilterTrees` for the marked data points in the specified `DataVisualization`.
   c. Constructing a `DataActionContext` that contains all the information to pass to the data action's target.
   d. Calling `invoke(oDataActionContext)` on the data action.

# Example Data Action plugin.xml File

This topic shows an example plugin.xml file for a `CanvasDataAction` data action.

**Example plugin.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:obiplugin xmlns:tns="http://plugin.frameworks.tech.bi.oracle"
               xmlns:viz="http://plugin.frameworks.tech.bi.oracle/
extension-points/visualization"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               id="obitech-currencyconversion"
               name="Oracle BI Currency Conversion"
               version="0.1.0.@qualifier@"
               optimizable="true"
               optimized="false">


   <tns:resources>
      <tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
      <tns:resource-folder id="nls" path="resources/nls"
optimizable="true">
         <tns:extensions>
            <tns:extension name="js" resource-type="script"/>
         </tns:extensions>
      </tns:resource-folder>
   </tns:resources>


   <tns:extensions>
      <tns:extension id="oracle.bi.tech.currencyconversiondataaction"
point-id="oracle.bi.tech.plugin.dataaction" version="1.0.0">
         <tns:configuration>
         {
            "resourceBundle": "obitech-currencyconversion/nls/messages",
            "properties":
            {
               "className": "obitech-currencyconversion/
currencyconversion.CurrencyConversionDataAction",
               "displayName": { "key" : "CURRENCY_CONVERSION",
"default" : "Currency Conversion" },
               "order": 100
            }
         }
         </tns:configuration>
      </tns:extension>
   </tns:extensions>


</tns:obiplugin>
```

## Data Action Plug-in Files and Folders

The following files and folders are used to implement data action plug-ins.

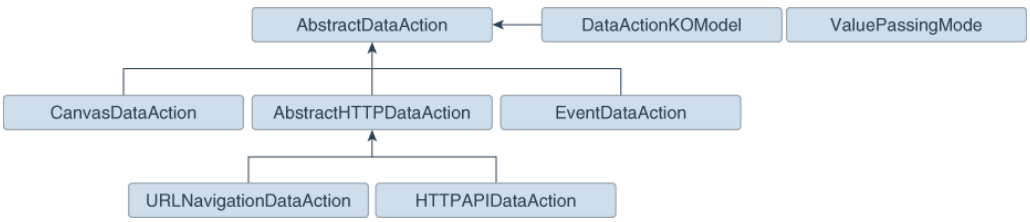
`bitech/client/plugins/src/`

- `report`
  - `obitech-report`
    - `scripts`
      - `dataaction`
        - `dataaction.js`
        - `dataactiongadgets.js`
        - `dataactionpanel.js`
        - `dataactionupgradehandler.js`
- `obitech-reportservice`
  - `scripts`
    - `dataaction`
      - `dataactionmanager.js`
      - `dataactionhandler.js`

# Choose the Best Data Action Class to Extend

Before you start writing your custom data action plug-in, decide which of the existing data action classes you want to extend. Choose the data action class that provides functionality that most closely matches what you want your data action to do.

Each data action inherits from the `AbstractDataAction` class as shown in the class diagram. The class diagram shows the two abstract data action classes (`AbstractDataAction` and `AbstractHTTPDataAction`) and the four concrete data action classes (`CanvasDataAction`, `URLNavigationDataAction`, `HTTPAPIDataAction`, and `EventDataAction`) that you can extend. Each data action that you provide must extend one of these classes. Which class you extend depends on the behavior you want to implement when you invoke your data action. Most third-party data actions are likely to extend either `URLNavigationDataAction`, `HTTPAPIDataAction` or `EventDataAction`.



Regardless of which class you extend, when your data action is invoked, you're provided with metadata describing the full context of the data-point from which the data action is invoked. See Data Action Context.

## AbstractDataAction Class

`AbstractDataAction` is the abstract base class from which all types of data action inherit. It's responsible for providing common functionality and default behavior that the subclasses can use.

### AbstractDataAction

All types of data action are subclasses of the `AbstractDataAction` base class. It provides the core set of functionality common to all data actions. Unless you're creating a complex data action that carries out multiple types of action when invoked, or you need to do something not supported by the concrete classes, you shouldn't extend this class directly. If you need to create a complex data action then consider extending the concrete class that most closely provides the functionality you require.

### AbstractDataAction Syntax

```
+ AbstractDataAction(oKOModel)



+ getKOViewModel():DataActionKOModel



+ createFromJS(fDataActionConstructor, sClassName,
oDataActionKOModelUS) : AbstractDataAction


+ invoke(oActionContext, oDataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
+ validate() : DataActionError


+ getSettings() : Object
+ requiresActionContextToInvoke() : Boolean
+ isAllowedHere() : Boolean


# createNameGadgetInfo(oReport) : AbstractGadgetInfo
# createAnchorToGadgetInfo(oReport) : AbstractGadgetInfo
# createPassValuesGadgetInfo(oReport) : AbstractGadgetInfo
```

# DataActionKOModel Class

Each subclass of `AbstractDataAction` is likely to create its own subclass of `DataActionKOModel`. The `DataActionKOModel` base class provides the following properties:

**DataActionKOModel, ValuePassingMode**

- `sID:String`
  The unique ID given to the data action instance.
- `sClass:String`
  The class name of this specific type of data action.
- `sName:String`
  The display name given to the data action instance.
- `sVersion`
- `sScopeID`
- `eValuePassingMode:ValuePassingMode`
  The mode used when passing context values. The mode can be one of the `ValuePassingMode` values (`ALL, ANCHOR_DATA, NONE, CUSTOM`).
- `aAnchorToColumns: ColumnKOViewModel[]`
  The columns that this data action is anchored to. This is optional. If not supplied, then the data action is available on all columns.
- `aContextColumns : ColumnKOViewModel[]`
  The columns that this data action includes in the context passed to the data action target when the data action is invoked. If not supplied, all marked columns are included in the context.

# CanvasDataAction Class

`CanvasDataAction` is a subclass of the `AbstractDataAction` base class. You can extend this concrete class to provide the functionality you require.

**CanvasDataAction**

Use the `CanvasDataAction` class to navigate from a data point in a visualization to a different canvas. The canvas you're navigating to can be in the same project or a different one. All the active filters for the source visualization are passed to the target canvas along with new filters that describe the Qualified Data Reference of the data

point itself. If your data action needs to navigate to a different canvas then this is the class your data action should extend.

```
+ CanvasDataAction(oKOModel)



+ create(s)ID_sName) : CanvasDataAction
+ upgrade(oOldDataActionJS) : Object



+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
+ validate() : DataActionError


# createProjectGadgetInfo(oReport) : AbstractGadgetInfo
# createCanvasGadgetInfo(oReport) : AbstractGadgetInfo
```

## EventDataAction Class

`EventDataAction` is a subclass of the `AbstractDataAction` base class. You can extend this concrete class to provide the functionality you require.

### EventDataAction

Use the `EventDataAction` class to publish a client-side event. You can then register one or more subscribers that listen for that event and perform their own actions. Use this type of data action in more complex use cases where you've a large amount of code and can benefit from keeping your data action code loosely coupled to the code that performs the necessary actions when the data action is invoked.

```
+ EventDataAction(oKOModel)



+ create(sID_sName) : EventDataAction
+ upgrade(oOldDataActionJS) : Object


+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
+ validate() : DataActionError


# createEventGadgetInfo(oReport) : AbstractGadgetInfo
```

# AbstractHTTPDataAction Class

`AbstractHTTPDataAction` is the abstract base class that the `URLNavigationDataAction` and `HTTPAPIDataAction` subclasses inherit common functionality and default behavior from.

### AbstractHTTPDataAction

The `AbstractHTTPDataAction` abstract base class is shared by both the `URLNavigationDataAction` and `HTTPAPIDataAction` classes. If your data action needs to open a web page in a new browser tab you must extend `URLNavigationDataAction`. If your data action needs to invoke an HTTP API then you should extend `HTTPAPIDataAction`. You may decide it's better to extend `AbstractHTTPDataAction` directly.

```
+ HTTPDataAction(oKOModel)


+ validate() : DataActionError


# createURLGadgetInfo(oReport) : AbstractGadgetInfo
```

# URLNavigationDataAction Class

`URLNavigationDataAction` is a subclass or the `AbstractHTTPDataAction` base class.

### URLNavigationDataAction

Use the `URLNavigationDataAction` class to open a specific URL in a new browser tab. You compose the URL using tokens that are replaced with values derived from data points that the user selects when they invoke the data action. The data point values are passed as part of the data action context to the external web page. For example, create a data action invoked using a `CustomerID` column that opens a customer's web page in your Customer Relations Management application such as Oracle Sales Cloud.

```
+ URLNavigationDataAction(oKOModel)


+ create(sID_sName) : URLNavigationDataAction
+ upgrade(oOldDataActionJS) : Object


+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
```

## HTTPAPIDataAction Class

`HTTPAPIDataAction` is a subclass or the `AbstractHTTPDataAction` base class. You can extend this concrete class to provide the functionality you require.

**HTTPAPIDataAction**

Use the `HTTPAPIDataAction` class to invoke HTTP APIs by creating an asyncronous `XMLHTTPRequest` (XHR) and submitting it to the specified URL. The HTTP response code enables a message to be displayed briefly on the canvas. For example, you can customize the request to send JSON or XML payloads to a REST or SOAP server and you can customize the response handler to show a custom user interface.

For the `HTTPAPIDataAction` data action to work, you must add the URL of the HTTP API you want to access to your list of Safe Domains and grant it **Connect** access. See Register Safe Domains.

```
+ HTTPAPIDataAction(oKOModel)


+ create(sID_sName) : HTTPAPIDataAction
+ upgrade(oOldDataActionJS) : Object


+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]


# createHTTPMethodGadgetInfo(oReport) : AbstractGadgetInfo
# createPostParamGadgetInfo(oReport) : AbstractGadgetInfo
```

# Generate Data Action Plug-ins from a Template

You use a series of commands to generate a development environment and populate it with a HTTP API Data Action along with the necessary folders and files that you need to create a custom data action plug-in.

All plug-in files follow the same basic structure. You can manually create the files and folders or you can generate them from a template. The tools to do this are part of the Oracle Analytics Desktop software development kit (SDK) which is included with Oracle Analytics Desktop. See Oracle Analytics Desktop SDK Reference.

Use Oracle Analytics Desktop version 5.4 or later to access the classes required to create a custom data action plug-in.

Use these commands to generate your development environment and populate it with a HTTP API data action.

1. At a command prompt, specify the root folder of your Oracle Analytics Desktop installation:

   ```
   set DVDESKTOP_SDK_HOME=C:\Program Files\Oracle Analytics Desktop
   ```

2. Specify the location to store your custom plug-ins:

```
set PLUGIN_DEV_DIR=C:\temp\dv-custom-plugins
```

3. Add the SDK command line tools to your path using:

   ```
   set PATH=%DVDESKTOP_SDK_HOME%\tools\bin;%PATH%
   ```

4. Create a folder for the directory used to store the custom plug-ins using:

   ```
   mkdir %PLUGIN_DEV_DIR%
   ```

5. Change the directory to the folder for storing custom plug-ins:

   ```
   cd %PLUGIN_DEV_DIR%
   ```

6. Create the environment variables:

   ```
   bicreateenv
   ```

7. Create the template files needed to start developing a custom HTTP API data action, for example:

   ```
   bicreateplugin -pluginxml dataaction -id company.mydataaction -subType
   httpapi
   ```

   Use the `-subType` option to specify the data action type that you want to create from: `httpapi`, `urlNavigation`, `canvasNavigation`, `event`, or `advanced`. The `advanced` option extends from the `AbstractDataAction` base class.

# Generated Folders and Files

Your newly generated data action development environment contains these folders and files:

```
1    %PLUGIN_DEV_DIR%\src\customdataaction
2       company-mydataaction\
3          extensions\
4             oracle.bi.tech.plugin.dataaction\
5                company.mydataaction.json
6          nls\
7             root\
8                messages.js
9             messages.js
10          mydataaction.js
11          mydataactionstyles.css
12          plugin.xml
```

- **Line 2**: The `company-mydataaction` folder is the ID that you specify.
- **Line 6**: The `nls` folder contains the files for externalizing strings that enable your plug-in to provide Native Language Support.
- **Line 7**: The strings in the files under the `nls\root` folder are the default strings used when translations for a requested language aren't available.
- **Line 8**: The `messages.js` file contains externalized strings for your plug-in that you can add.
- **Line 9**: The `messages.js` file must contain an entry that you add for each additional language that you want to provide localized strings for. You must add a corresponding folder under the `nls` folder for each locale that you want to add

translations for. Each folder must contain the same set of files, with the same file names as those added under the `nls\root` folder.

- **Line 10**: The `mydataaction.js` file is the newly generated JavaScript module template that provides a starting point to develop your custom data action.
- **Line 11**: The `mydataactionstyles.css` file can contain any CSS styles that you want to add, and which your data action's user interface can use.
- **Line 12**: The `plugin.xml` file registers your plug-in and its files with Oracle Analytics.

# Extend a Data Action Base Class

Once you've chosen the subclass of data action that you want to extend and have generated the necessary folders and files, you're ready to start writing the code specific to your new data action.

You can find your newly generated data action code under `%PLUGIN_DEV_DIR%\src\dataaction`. See Generated Folders and Files for an explanation of the files and folder structure. The main file you must edit is the JavaScript file. For example, if your custom data action ID is `company.MyDataaction`, then the file you're looking for is `%PLUGIN_DEV_DIR%\src\dataaction\company-mydataaction\mydataaction.js`.

**Extending Your Data Action's Knockout Model**
If your data action has additional properties that need to be stored, then you must add them as observable properties to the Knockout Model. If your data action is given the ID `company.MyDataaction`, then the Knockout Model is called `mydataaction.MyDataActionKOModel` which is located near the top of `mydataaction.js`. By default, this Knockout Model is configured to extend the Knockout Model used by your data action's superclass so you only need to add additional properties to the model.

For a data action that's extending the `HTTPAPIDataAction` base class, use code similar to the following:

```
1 - mydataaction.MydataactionKOModel = function (sClass, sID, sName,
sVersion, sScopeID, aAnchorToColumns, eValuePassingMode, sURL,
        eHTTPMethod, sPOSTParams)
2 - {
3 - mydataaction.MydataactionKOModel.baseConstructor.call(this, sClass,
sID, sName, sVersion, sScopeID, aAnchorToColumns, eValuePassingMode,
sURL, eHTTPMethod, sPOSTParams);
4 - };
5 - jsx.extend(mydataaction.MydataactionKOModel,
dataaction.HTTPAPIDataActionKOModel);
```

- **Line 1**: This is the constructor for your Knockout Model. It accepts the properties that the model needs to store.
- **Line 3**: This is the superclass's constructor, otherwise known as the `baseConstructor` to which you pass the values for all of the properties that are handled by one of the Knockout Model's superclasses.
- **Line 5**: This sets the superclass for this Knockout Model class.

Use code similar to the following to add a string and an array to set properties that are persisted by the data action.

```
1   mydataaction.MydataactionKOModel = function (sClass, sID,
sName, sVersion, sScopeID, aAnchorToColumns, eValuePassingMode, sURL,
eHTTPMethod, sPOSTParams)
2   {
3   mydataaction.MydataactionKOModel.baseConstructor.call(this, sClass,
sID, sName, sVersion, sScopeID, aAnchorToColumns, eValuePassingMode,
sURL, eHTTPMethod, sPOSTParams);
4
5
6   // Set Defaults
7   sMyString = sMyString || "My default string value";
8   aMyArray = aMyArray || [];
9
10
11  // Asserts
12  jsx.assertString(sMyString, "sMyString");
13  jsx.assertArray(aMyArray, "aMyArray");
14
15
16  // Add observable properties
17  this.sMyString = ko.observable(sMyString);
18  this.aMyArray = ko.observableArray(aMyArray);
19  };
20  jsx.extend(mydataaction.MydataactionKOModel,
dataaction.HTTPAPIDataActionKOModel);
```

# Choose Which Data Action Inherited Methods to Override

Each data action must implement various methods in order to function properly, so you only need to override those methods that implement behavior that you want to change.

If you're extending one of the concrete data actions classes, for example `HTTPAPIDataAction`, then most of the required methods are already implemented and you only need to override the methods that implement the behavior you want to change.

### Generic Methods

This section describes the various methods and what's expected of them.

All types of data action must implement the methods that are described here.

**create(sID, sName)**

The `create()` static method is called when you're creating a new data action and select a **Data Action Type** from the drop-down menu. This method is responsible for:

- Constructing the Knockout Model class that your data action uses.
  The Knockout Model class must have the ID and name that's passed to the `create()` method along with sensible defaults for all other properties. For example, for a currency conversion data action you might want to set the default currency to convert into Dollars. The Knockout Model is the correct place to provide your default values.

- Constructing an instance of your data action from the Knockout Model.
- Returning the instance of your data action.

**invoke(oActionContext, oDataActionContext)**

The `invoke()` method is called when the user invokes your data action from the context menu for a data point in a visualization. The method passes the `DataActionContext` argument which contains metadata describing the selected data points, visualization, filters, project, and session. See Data Action Service Classes.

**validate()**

The `validate()` method is called on each data action when the user clicks **OK** in the Data Actions dialog. The `validate()` method returns a `null` to indicate that everything is valid or a `DataActionError` if something is invalid. If there's an error in one of the data actions in the dialog, the error prevents the dialog from closing and an error message is displayed to the user. This method validates the name of the data action using the `this.validateName()` method.

**getGadgetInfos(oReport)**

The `getGadgetInfos()` method is called to enable the user interface to display data action property fields. The method returns an array of `GadgetInfos` in the order you want them to appear in the user interface. Gadgets are provided for all of the most common types of fields (for example, text, drop-down, password, multi-select, radio button, check box) but you can create custom gadgets if you want more complicated fields (for example, where multiple gadgets are grouped together, or where different gadget fields display depending on which option you select). It's a best practice to create a method that constructs each `GadgetInfo` you want in your array, as it makes it easier for potential subclasses to pick and choose from the `GadgetInfos` you've provided. If you follow this best practice there are already various methods implemented by the different data action base classes that can return a `GadgetInfo` for each of the fields that they use in their user interfaces. If you also need one of these `GadgetInfos` then you call the corresponding `create****GadgetInfo()` method and push its return value into your array of gadgets.

**isAllowedHere(oReport)**

The `isAllowedHere()` method is called when the user right-clicks on a data-point in a visualization and the user interface starts to generate the context menu. If a data action exists that's relevant to the selected data-points, then the method returns `true` and the data action appears in the context menu. If the method returns `false`, then the data action doesn't appear in the context menu. Consider accepting the default behavior inherited from the superclass.

**upgrade(oOldDataActionJS)**

If you're creating your first data action then don't use the `upgrade(oOldDataActionJS)` method. Only use this method after you've created your first Knockout Model and are making significant changes to properties for a second version of your Knockout Model. For example, if the first version of your data action stores a URL in its Knockout Model, but you decide that the next version will store URL component parts in separate properties (for example, `protocol`, `hostname`, `port`, `path`, `queryString` and `bookmark`).

The second version of your Knockout Model code would request to open a data action that had been saved with the first version of your Knockout Model code which can cause problems. To resolve this issue, the system identifies that your current data action code version is newer than that of the data action being opened and it calls the

`upgrade()` method on your new data action class and passes in the old data action Knockout Model (serialized to a JSON object). You can then use the old JSON object to populate your new Knockout Model and return an upgraded version of the JSON object. This ensures that old data action metadata continues to work as you improve your data action code.

### HTTPAPIDataAction Methods

If you're extending the `HTTPAPIDataAction` class, then it provides the following additional method that you may choose to override:

#### getAJAXOptions(oDataActionContext)

The `getAJAXOptions()` method is called by the data action's `invoke()` method. The `getAJAXOptions()` method creates the `AJAX Options` object that describes the HTTP request that you want your data action to make. The `getAJAXOptions()` method is passed the `oDataActionContext` object that contains the metadata describing the selected data-points, visualization, filters, project, and session. Set the `AJAX Options` as required by the HTTP API you're trying to integrate with and specify the functions you want to be called when the `HTTPRequest` is successful or results in an error. See the JQuery website for an explanation of the `jQuery.ajax` object and its properties.

The following implementation is inherited from the `HTTPAPIDataAction` class. You need to rewrite the inherited method to specify requirements. For example, forming the HTTP request, and the code that handles the HTTP response. This implementation is useful as it shows the parameters passed to the `getAJAXOptions()` function, the object that it's expected to return, and gives a clear example of how to structure the code inside the method.

```
1 /**
2  * This method returns an object containing the AJAX settings used
when the data action is invoked.
3  * Subclasses may wish to override this method to provide their own
behavior.
4  * @param {module:obitech-reportservices/
dataactionmanager.DataActionContext} oDataActionContext The context
metadata describing where the data action was invoked from.
5  * @returns {?object} A JQuery AJAX settings object (see http://
api.jquery.com/jQuery.ajax/ for details) - returns null if there is a
problem.
6  */
7 dataaction.HTTPAPIDataAction.prototype.getAJAXOptions = function
(oDataActionContext)
8 {
9    jsx.assertInstanceOfModule(oDataActionContext,
"oDataActionContext", "obitech-reportservices/dataactionmanager",
"DataActionContext");
10
11   var oAJAXOptions = null;
12   var oKOViewModel = this.getKOViewModel();
13   var sURL = oKOViewModel.sURL();
14   if (sURL)
15   {
16      // Parse the URL
17      var sResultURL = this._parseURL(sURL, oDataActionContext);
18      if (sResultURL)
```

```
19      {
20         // Parse the POST parameters (if required)
21         var eHTTPMethod = oKOViewModel.eHTTPMethod()[0];
22         var sData = null;
23         if (eHTTPMethod ===
dataaction.HTTPDataActionKOModel.HTTPMethod.POST)
24         {
25            var sPOSTParams = oKOViewModel.sPOSTParams();
26            sData =
sPOSTParams.replace(dataaction.AbstractHTTPDataAction.RegularExpressions
.LINE_END, "&");
27            sData = this._parseURL(sData, oDataActionContext, false);
28         }
29         oAJAXOptions = {
30            type: eHTTPMethod,
31            url: sResultURL,
32            async: true,
33            cache: false,
34            success: function (/*oData, sTextStatus, oJQXHR*/)
35            {
36
oDataActionContext.getReport().displaySuccessMessage(messages.HTTP_API_D
ATA_ACTION_INVOCATION_SUCCESSFUL.format(oKOViewModel.sName()));
37            },
38            error: function (oJQXHR/*, sTextStatus, sError*/)
39            {
40
oDataActionContext.getReport().displayErrorMessage(messages.HTTP_API_DAT
A_ACTION_INVOCATION_FAILED.format(oKOViewModel.sName(),
oJQXHR.statusText, oJQXHR.status));
41            }
42         };
43         if (sData)
44         {
45            oAJAXOptions.data = sData;
46         }
47      }
48   }
49   return oAJAXOptions;
50 };
```

# Test, Package, and Install Your Data Action

You use Oracle Analytics Desktop to test your data action from its source location before you install it.

1. If Oracle Analytics Desktop is currently running, close it.
2. If you're working behind a proxy, set the proxy settings in `%PLUGIN_DEV_DIR%\gradle.properties`. For information about accessing the web through HTTP proxy, see Gradle User Manual.
3. Run Oracle Analytics Desktop in SDK mode by using the command prompt you started in Choose Which Data Action Inherited Methods to Override and enter the following commands:

```
cd %PLUGIN_DEV_DIR%

.\gradlew run
```

Oracle Analytics Desktop starts in SDK mode. Your data action plug-in appears in the Console | Extensions page.

Create a project and test your data action. If you find any issues, you can debug your code using your browser's built-in developer tools.

4. If you created an HTTP API data action:
   a. Go to the Console and display the Safe Domains page.
   b. Add each domain that you want to access.

      For example, if you need access to the apilayer.com APIs, add `apilayer.net` to the list of safe domains.
   c. Click the **Connect** column checkbox for the selected domain.
   d. Reload the Safe Domains page in your browser for the changes to take effect.
5. If you want to prepare your data action plug-in to distribute to other people or to install in Oracle Analytics:
   - Package all of the files into a single ZIP file containing the `%PLUGIN_DEV_DIR%\src\customdataaction` folder and its contents.
   - Name the zip using the same ID you gave to your data action plug-in when you created it.
6. Install your data action plug-in. See Manage Custom Plug-ins.

# Use an Upgrade Handler for Knockout Model Changes

For some Knockout Model changes you need to upgrade your data action plug-in using an upgrade handler.

When you're making improvements to your data action plug-in without making changes to the Knockout Model you normally edit your JavaScript or CSS files, create a new ZIP file, and replace the existing data action plug-in with the new ZIP file. However, if you've made changes to your data action's Knockout Model then you might need to change the data action `VERSION` property and provide an upgrade handler.

Decide whether you need to use an upgrade handler:

**Upgrade Handler Required**

- If you rename a property in your Knockout Model.
- If you combine multiple properties into a single property in your Knockout Model.
- If you split a single property into multiple properties in your Knockout Model.
- If you add a new property to the Knockout Model and the correct default value for it depends on other values in the Knockout Model.

**Upgrade Handler Not Required**

- If you add a new property to the Knockout Model and can provide a default value that's correct for all existing usages of your data action.
- If you remove a property from the Knockout Model because it's no longer used by your data action code.

# Upgrade Data Action Plug-ins

Upgrade your data action plug-ins to improve the data action code or upgrade the metadata to enable existing data actions to work with new data action code.

Use an upgrade handler to upgrade a data action plug-in.

1. Increase the version number of your data action.

   For example, if your data action is called company.MyDataAction, then search `mydataaction.js` for the `mydataaction.MyDataAction.VERSION` property. If it's currently set to `1.0.0` then change it to `1.0.1`.

2. Add a `static upgrade(oOldDataActionJS)` method to your data action's class.

   If the `VERSION` property differs from the `sVersion` value stored in the data action metadata then the Data Action Manager calls the `static upgrade()` method on your data action's class.

3. Implement your `upgrade()` method by calling the `upgrade()` method on the superclass and capture its response.

4. Continue to implement your `upgrade()` method by making further edits to the partially upgraded data action JSON returned by the superclass, until the object matches the correct set of properties required by your latest Knockout Model.

5. To finish call `var oUpgradedDataAction = dataaction.AbstractDataAction.createFromJS(fDataActionClass, sFullyQualifiedDataActionClassName, oUpgradedDataActionJS)`.

   This command constructs a new instance of your data action from the upgraded data action JSON and returns `oUpgradedDataAction.getSettings()`.

# Data Action Plug-in File Reference

Each data action plug-in requires a plugin.xml file and each plugin.xml file can contain any number of data actions.

**Topics:**

# Data Action plugin.xml File Example

The plugin.xml file has three main sections, `tns:obiplugin`, `tns:resources`, and `tns:extension`.

**Example plugin.xml**

This example shows a typical plugin.xml file for one data action.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tns:obiplugin xmlns:tns="http://plugin.frameworks.tech.bi.oracle"
3                id="obitech-currencyconversion"
4                name="Oracle BI Currency Conversion"
5                version="0.1.0.@qualifier@"
6                optimizable="true"
7                optimized="false">
8
9
10   <tns:resources>
11      <tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
12      <tns:resource-folder id="nls" path="resources/nls"
optimizable="true">
13         <tns:extensions>
14            <tns:extension name="js" resource-type="script"/>
15         </tns:extensions>
16      </tns:resource-folder>
17   </tns:resources>
18
19
20   <tns:extensions>
21      <tns:extension id="oracle.bi.tech.currencyconversiondataaction"
point-id="oracle.bi.tech.plugin.dataaction" version="1.0.0">
22         <tns:configuration>
23         {
24            "host": { "module": "obitech-currencyconversion/
currencyconversion" },
25            "resourceBundle": "obitech-currencyconversion/nls/
messages",
26            "properties":
27            {
28               "className": "obitech-currencyconversion/
currencyconversion.CurrencyConversionDataAction",
29               "displayName": { "key" : "CURRENCY_CONVERSION",
"default" : "Currency Conversion" },
30               "order": 100
31            }
32         }
33         </tns:configuration>
34      </tns:extension>
35   </tns:extensions>
36
37 </tns:obiplugin>
```

# Data Action plugin.xml File Properties Section - tns:obiplugin

The `tns:obiplugin` section defines properties common to all types of plug-ins.

#### Plug-in Properties

The `tns:obiplugin` section defines properties common to all types of plug-ins.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tns:obiplugin xmlns:tns="http://plugin.frameworks.tech.bi.oracle"
3                id="obitech-currencyconversion"
4                name="Oracle BI Currency Conversion"
5                version="0.1.0.@qualifier@"
6                optimizable="true"
7                optimized="false">
```

- **Line 1**: The XML declaration.
- **Line 2**: The opening tag for the plug-in's root XMLElement and the declaration for the `tns` namespace that's used throughout plugin.xml files.
- **Line 3**: The plug-in's unique ID.
- **Line 4**: The plug-in's default display name (used when a localized version isn't available).
- **Line 5**: The plug-in's version number.
- **Line 6**: A boolean indicating whether or not the JS/CSS can be optimized (compressed).
- **Line 7**: A boolean indicating whether or not the JS/CSS has been optimized (compressed).

# Data Action plugin.xml File Resources Section - tns:resources

The `tns:resources` section registers all of the files that contribute to your plug-in.

#### Resources

```
1 <tns:resources>
2    <tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
3    <tns:resource-folder id="nls" path="resources/nls"
optimizable="true">
4       <tns:extensions>
5          <tns:extension name="js" resource-type="script"/>
6       </tns:extensions>
7    </tns:resource-folder>
8 </tns:resources>
```

You need to register each JavaScript, CSS, Image, and Translation Resource File here. The section is contained within the `<tns:resources>` element and contains any number of the following elements:

- `<tns:resource>`
  These elements are used to register a single file (for example, a JavaScript or CSS file).
- `<tns:resource-folder>`
  These elements are used to register all the files under a specified folder at the same time. For example, an image folder or the folder containing the resource files for Native Language Support.

More information on how to register each type of file is provided in the following sections.

**JavaScript Files**

Each JavaScript file in your plug-in must be registered with a line similar to the one shown below.

```
<tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
```

Where:

- **id** is the ID given to the file.
  Set the ID to match the JavaScript filename without the .js extension.
- **path** is the relative path to the JavaScript file from the plugin.xml file. JavaScript files should be stored under your plug-in's `scripts` directory.
  Use all lowercase for your JavaScript files with no special characters (for example, underscore, hyphen).
- **type** is the type of file being registered. It must be set to `script` for JavaScript files.
- **optimizedGroup** groups multiple JavaScript files into a single compressed file. Third-party plug-ins must leave this set to `base`.

**CSS Files**

Each CSS file in your plug-in must be registered with a line similar to the one shown below.

```
<tns:resource id="currencyconversionstyles" path="resources/
currencyconversion.css" type="css"/>
```

Where:

- **id** is the ID given to the file.
  Set the ID to match the CSS filename without the .css extension.
- **path** is the relative path to the CSS file from the plugin.xml file. CSS files should be stored under your plug-in's `resources` directory.
  Use all lowercase for your CSS files with no special characters (for example, underscore, hyphen).

- **type** is the type of file being registered. It should always be set to `css` for CSS files.

**Image Folders**

If your plug-in has images that you need to refer to from within your JavaScript code, then put them in a `resources/images` directory within your plug-in's directory structure and add a `<tns:resource-folder>` element to your `plugin.xml` as follows:

```
<tns:resource-folder id="images" path="resources/images"
optimizable="false"/>
```

If your images are only referenced by your CSS files, then you don't need to add this `<tns:resource-folder>` element to your `plugin.xml` file. In this case, you must still add them to the `resources/images` directory so that you can then refer to them using a relative path from your CSS file.

**Native Language Support Resource Folders**

Oracle Analytics implements Native Language Support. This requires developers to externalize the strings they display in their user interface into separate JSON resource files. You can then provide different localized versions of those files in a prescribed directory structure and Oracle Analytics automatically uses the correct file for the user's chosen language. You can provide as many translated versions of the resource files as needed. A Native Language Support resource folder points Oracle Analytics to the root of the prescribed Native Language Support directory structure used by your plug-in. All plug-ins that use Native Language Support resource files must have a `<tns:resource-folder>` entry that looks exactly like the example below.

```
1 <tns:resource-folder id="nls" path="resources/nls" optimizable="true">
2    <tns:extensions>
3       <tns:extension name="js" resource-type="script"/>
4    </tns:extensions>
5 </tns:resource-folder>
```

See Generated Folders and Files for details about the contents of the files and the prescribed directory structure that you should follow.

## Data Action plugin.xml File Extensions Section - tns:extension

For each data action you want your plug-in to provide, you must register a data action extension using a `<tns:extension>` element similar to this:

```
<tns:extension id="oracle.bi.tech.currencyconversiondataaction" point-
id="oracle.bi.tech.plugin.dataaction" version="1.0.0">
   <tns:configuration>
   {
      "host": { "module": "obitech-currencyconversion/
currencyconversion" },
      "resourceBundle": "obitech-currencyconversion/nls/messages",
      "properties":
      {
         "className": "obitech-currencyconversion/
```

```
currencyconversion.CurrencyConversionDataAction",
         "displayName": { "key" : "CURRENCY_CONVERSION", "default" :
"Currency Conversion" },
         "order": 100
      }
   }
   </tns:configuration>
</tns:extension>
```

Where:

- **id** is the unique ID you give to your data action.
- **point-id** is the type of extension you want to register. For data action extensions, this must be set to `oracle.bi.tech.plugin.dataaction`.
- **version** is the extension API version that your extension definition uses (leave this set to **1.0.0**).

The `<tns:configuration>` element contains a JSON string that defines:

- **host.module** - This is the fully qualified name of the module containing your data action. This fully qualified module name is formulated as `%PluginID%/%ModuleName%`, where:
  - `%PluginID%` must be replaced with the plug-in ID you specified in the id attribute of the `<tns:obiplugin>` element.
  - `%ModuleName%` must be replaced with the resource ID you specified in the id attribute of the `<tns:resource>` element for the JavaScript file containing your data action.
- **resourceBundle** - This is the Native Language Support path to the resource file that contains this data action's localized resources. If your resource files are named messages.js and stored correctly in the prescribed `nls` directory structure, then set this property to `%PluginID%/nls/messages` (where `%PluginID%` must be replaced with the plug-in ID you specified in the id attribute of the `<tns:obiplugin>` element at the top of the `plugin.xml` file).
- **properties.className** - This is the fully qualified class name given to the data action you're registering. This fully qualified class name is formulated as `%PluginID%/%ModuleName%.%ClassName%`, where:
  - `%PluginID%` must be replaced with the plug-in ID you specified in the id attribute of the `<tns:obiplugin>` element.
  - `%ModuleName%` must be replaced with the resource ID you specified in the id attribute of the `<tns:resource>` element for the JavaScript file containing your data action.
  - `%ClassName%` must be replaced with the name you gave to the data action class in your JavaScript file.
- **properties.displayName** - This property contains an object and two further properties:
  - **key** is the Native Language Support message key that can be used to lookup the data action's localized display name from within the specified `resourceBundle`.

  - **default** is the default display name to use if for some reason the localized version of the display name can't be found.

- **properties.order** - This property enables you to provide a hint that's used to determine the position that this data action should appear when shown in a list of data actions. Data actions with lower numbers in their order property appear before data actions with higher numbers. When there's a tie, the data actions are displayed in the order they're loaded by the system.

10

# Use Oracle Analytics Predictive Models and Oracle Machine Learning Models

Predict data with Oracle Analytics predictive models or machine learning models from Oracle Database or Oracle Autonomous Data Warehouse.

**Topics:**



## Create and Use Oracle Analytics Predictive Models

Oracle Analytics predictive models use several embedded machine learning algorithms to mine your data sets, predict a target value, or identify classes of records. Use the data flow editor to create, train, and apply predictive models to your data.

**Topics:**



### What Are Oracle Analytics Predictive Models?

An Oracle Analytics predictive model applies a specific algorithm to a data set to predict values, predict classes, or to identify groups in the data.

You can also use Oracle machine learning models to predict data. See

How Can I Use Oracle Machine Learning Models in Oracle Analytics?

Oracle Analytics includes algorithms to help you train predictive models for various purposes. Examples of algorithms are classification and regression trees (CART), logistic regression, and k-means.

You use the data flow editor to first train a model on a training data set. After the predictive model has been trained, you apply it to the data sets that you want to predict.

You can make a trained model available to other users who can apply it against their data to predict values. In some cases, certain users train models, and other users apply the models.

> **Note:**
>
> If you're not sure what to look for in your data, you can start by using Explain, which uses machine learning to identify trends and patterns. Then you can use the data flow editor to create and train predictive models to drill into the trends and patterns that Explain found. See What is Explain?

You use the data flow editor to train a model:

- First, you create a data flow and add the data set that you want to use to train the model. This training data set contains the data that you want to predict (for example, a value like sales or age, or a variable like credit risk bucket).
- If needed, you can use the data flow editor to edit the data set by adding columns, selecting columns, joining, and so on.
- After you've confirmed that the data is what you want to train the model on, you add a training step to the data flow and choose a classification (binary or multi), regression, or cluster algorithm to train a model. Then name the resulting model, save the data flow, and run it to train and create the model.
- Examine the properties in the machine learning objects to determine the quality of the model. If needed, you can iterate the training process until the model reaches the quality you want.

Use the finished model to score unknown, or unlabeled, data to generate a data set within a data flow or to add a prediction visualization to a project.

**Example**

Suppose you want to create and train a multi-classification model to predict which patients have a high risk of developing heart disease.

1. Supply a training data set containing attributes on individual patients like age, gender, and if they've ever experienced chest pain, and metrics like blood pressure, fasting blood sugar, cholesterol, and maximum heart rate. The training data set also contains a column named "Likelihood" that is assigned one of the following values: absent, less likely, likely, highly likely, or present.
2. Choose the CART (Decision Tree) algorithm because it ignores redundant columns that don't add value for prediction, and identifies and uses only the columns that are helpful to predict the target. When you add the algorithm to the data flow, you choose the Likelihood column to train the model. The algorithm uses machine learning to choose the driver columns that it needs to perform and output predictions and related data sets.
3. Inspect the results and fine tune the training model, and then apply the model to a larger data set to predict which patients have a high probability of having or developing heart disease.

# How Do I Choose a Training Model Algorithm?

Oracle Analytics provides algorithms for any of your machine learning modeling needs: numeric prediction, multi-classifier, binary classifier, and clustering.

Oracle's machine learning functionality is for advanced data analysts who have an idea of what they're looking for in their data, are familiar with the practice of predictive analytics, and understand the differences between algorithms.

Normally users want to create multiple prediction models, compare them, and choose the one that's most likely to give results that satisfy their criteria and requirements. These criteria can vary. For example, sometimes users choose models that have better overall accuracy, sometimes users choose models that have the least type I (false positive) and type II (false negative) errors, and sometimes users choose models that return results faster and with an acceptable level of accuracy even if the results aren't ideal.

Oracle Analytics contains multiple machine learning algorithms for each kind of prediction or classification. With these algorithms, users can create more than one model, or use different fine-tuned parameters, or use different input training datasets and then choose the best model. The user can choose the best model by comparing and weighing models against their own criteria. To determine the best model, users can apply the model and visualize results of the calculations to determine accuracy, or they can open and explore the related data sets that Oracle Analytics used the model to output. See What Are Related Data Sets?

Consult this table to learn about the provided algorithms:

| Name | Type | Category | Function | Description |
|---|---|---|---|---|
| CART | Classification<br>Regression | Binary Classifier<br>Multi-Classifier<br>Numerical | - | Uses decision trees to predict both discrete and continuous values. Use with large data sets. |
| Elastic Net Linear Regression | Regression | Numerical | ElasticNet | Advanced regression model. Provides additional information (regularization), performs variable selection, and performs linear combinations. Penalties of Lasso and Ridge regression methods. Use with a large number of attributes to avoid collinearity (where multiple attributes are perfectly correlated) and overfitting. |

| Name | Type | Category | Function | Description |
|---|---|---|---|---|
| Hierarchical | Clustering | Clustering | AgglomerativeClustering | Builds a hierarchy of clustering using either bottom-up (each observation is its own cluster and then merged) or top down (all observations start as one cluster) and distance metrics. Use when the data set isn't large and the number of clusters isn't known beforehand. |
| K-Means | Clustering | Clustering | k-means | Iteratively partitions records into k clusters where each observation belongs to the cluster with the nearest mean. Use for clustering metric columns and with a set expectation of number of clusters needed. Works well with large datasets. Result are different with each run. |
| Linear Regression | Regression | Numerical | Ordinary Least Squares<br>Ridge<br>Lasso | Linear approach for a modeling relationship between target variable and other attributes in the data set. Use to predict numeric values when the attributes aren't perfectly correlated. |
| Logistic Regression | Regression | Binary Classifier | LogisticRegressionCV | Use to predict the value of a categorically dependent variable. The dependent variable is a binary variable that contains data coded to 1 or 0. |
| Naive Bayes | Classification | Binary Classifier<br>Multi-Classifier | GaussianNB | Probabilistic classification based on Bayes' theorem that assumes no dependence between features. Use when there are a high number of input dimensions. |

| Name | Type | Category | Function | Description |
|---|---|---|---|---|
| Neural Network | Classification | Binary Classifier<br>Multi-Classifier | MLPClassifier | Iterative classification algorithm that learns by comparing its classification result with the actual value and returns it to the network to modify the algorithm for further iterations.<br>Use for text analysis. |
| Random Forest | Classification | Binary Classifier<br>Multi-Classifier<br>Numerical | - | An ensemble learning method that constructs multiple decision trees and outputs the value that collectively represents all the decision trees.<br>Use to predict numeric and categorical variables. |
| SVM | Classification | Binary Classifier<br>Multi-Classifier | LinearSVC, SVC | Classifies records by mapping them in space and constructing hyperplanes that can be used for classification. New records (scoring data) are mapped into the space and are predicted to belong to a category, which is based on the side of the hyperplane where they fall. |

# Typical Workflow to Create and Use Oracle Analytics Predictive Models

Here are the common tasks for creating predictive models, and how to apply the models to data sets and use them in projects.

| Task | Description | More Information |
|---|---|---|
| Train a model using sample data | Use one of the supplied algorithms to train a model to predict trends and patterns in your sample data. | Create and Train a Predictive Model |
| Evaluate a model | Use related data sets to evaluate the effectiveness of your model, and iteratively refine the model until you're satisfied with it. | Inspect a Training Model |

| Task | Description | More Information |
|---|---|---|
| Apply a model to your data using a data flow | Apply a training predictive model to your data to generate a data set that includes the predicted trends and patterns. | Apply a Predictive or Oracle Machine Learning Model to a Data Set |
| Apply a predictive model to your project data | Use a scenario to add a predictive model to your project. | Add a Predictive Model to a Project |

# Create and Train a Predictive Model

Based on the problem that needs to be solved, an advanced data analyst chooses an appropriate algorithm to train a predictive model and then evaluates the model's results.

Arriving at an accurate model is an iterative process and an advanced data analyst can try different models, compare their results, and fine tune parameters based on trial and error. A data analyst can use the finalized, accurate predictive model to predict trends in other data sets, or add the model to projects.

Oracle Analytics provides algorithms for numeric prediction, multi-classification, binary-classification and clustering. For information about how to choose an algorithm, see How Do I Choose a Training Model Algorithm?

The algorithms aren't available until you install Oracle machine learning into your local Oracle Analytics Desktop directory. See How do I install Machine Learning for Desktop?

1. In the Home page, click **Create** and select **Data Flow**.
2. Select the data set that you want to use to train the model. Click **Add**.

   Typically you'll select a data set that was prepared specifically for training the model and contains a sample of the data that you want to predict. The accuracy of a model depends on how representative the training data is.
3. In the data flow editor, click **Add a step (+)**.

   After adding a data set, you can either use all columns in the data set to build the model or select only the relevant columns. Choosing the relevant columns requires an understanding of the data set. Ignore columns that you know won't influence the outcome behavior or that contain redundant information. You can choose only relevant columns by adding the **Select Columns** step. If you're not sure about the relevant columns, then use all columns.
4. Navigate to the bottom of the list and click the train model type that you want to apply to the data set.
5. Select an algorithm and click **OK**.
6. If you're working with a supervised model like prediction or classification, then click **Target** and select the column that you're trying to predict. For example, if you're creating a model to predict a person's income, then select the Income column.

   If you're working with an unsupervised model like clustering, then no target column is required.
7. Change the default settings for your model to fine tune and improve the accuracy of the predicted outcome. The model you're working with determines these settings.

8. Click the **Save Model** step and provide a name and description. This will be the name of the generated predictive model.
9. Click **Save**, enter a name and description of the data flow, and click **OK** to save the data flow.
10. Click **Run Data Flow** to create the predictive model based on the input data set and model settings that you provided.

## Inspect a Training Model

After you create the training model and run the data flow, you can review information about the model to determine its accuracy. Use this information to iteratively adjust the model settings to improve accuracy and predict better results.

1. Click the Navigator icon and select **Machine Learning**.
2. Click the **Models** tab.
3. Click the menu icon for a model and select **Inspect**.

   The Inspect dialog is displayed.
4. Browse the dialog's tabs for information about the model and to view the model's accuracy to determine if you need to adjust the model's parameters or select a more suitable training algorithm. Note the following information:
   - Quality tab - This tab contains model quality details that include accuracy metrics like model accuracy, precision, recall, F1 value, false positive rate, and so on. Oracle Analytics provides similar metrics irrespective of the algorithm used to create the model thereby making comparison between different models easy.
     During the model creation process, the input data set is split into two parts to train and test the model based on the Train Partition Percent parameter. The model uses the test portion of the data set to test the accuracy of the model that is built.
   - Related tab - Use to navigate to the data sets generated when you train a model. Depending on the algorithm, these data sets contain details about the model like: prediction rules, accuracy metrics, confusion matrix, key drivers for prediction, and so on.
     These parameters help you understand the rules the model used to determine the predictions and classifications. You can double-click a related data set to view it or to use it in a project.
5. If based on your findings in the Quality and Related tabs you need to adjust the model parameters and retrain it, then close the information dialog, click the Navigator icon, select **Data**, click the **Data Flows** tab, locate the data flow, and click **Open**.

## What Are Related Data Sets?

When you run the data flow to create the training model, Oracle Analytics creates a set of related data sets. You can open and create projects on these data sets to learn about the accuracy of the model.

Depending on the algorithm you chose for your model, related data sets contain details about the model such as prediction rules, accuracy metrics, confusion matrix, and key drivers for prediction. You can use this information to fine tune the model to

get better results, and you can use related data sets to compare models and decide which model is more accurate.

For example, you can open a Drivers data set to discover which columns have a strong positive or negative influence on the model. By examining those columns, you find that some columns aren't treated as model variables because they aren't realistic inputs or that they're too granular for the forecast. You use the data flow editor to open the model and based on the information you discovered, you remove the irrelevant or too-granular columns, and regenerate the model. You check the Quality and Results tab and verify if the model accuracy is improved. You continue this process until you're satisfied with the model's accuracy and it's ready to score a new data set.

To find and open a model, see Inspect a Training Model.

Different algorithms generate similar related data sets. Individual parameters and column names may change in the data set depending on the type of algorithm, but the functionality of the data set stays the same. For example, the column names in a statistics data set may change from Linear Regression to Logistic Regression, but the statistics data set contains accuracy metrics of the model.

These are the related data sets:

### CARTree

This data set is a tabular representation of CART (Decision Tree), computed to predict the target column values. It contains columns that represent the conditions and the conditions' criteria in the decision tree, a prediction for each group, and prediction confidence. The Inbuilt Tree Diagram visualization can be used to visualize this decision tree.

The CARTree data set is outputted when you select these model and algorithm combinations.

| Model | Algorithm |
|---|---|
| Numeric | CART for Numeric Prediction |
| Binary Classification | CART (Decision Tree) |
| Multi Classification | CART (Decision Tree) |

### Classification Report

This data set is a tabular representation of the accuracy metrics for each distinct value of the target column. For example, if the target column can have the two distinct values Yes and No, this data set shows accuracy metrics like F1, Precision, Recall, and Support (the number of rows in the training data set with this value) for every distinct value of the target column.

The Classification data set is outputted when you select these model and algorithm combinations.

| Model | Algorithms |
|---|---|
| Binary Classification | Naive Bayes<br>Neural Network<br>Support Vector Machine |

| Model | Algorithms |
|---|---|
| Multi Classification | Naive Bayes<br>Neural Network<br>Support Vector Machine |

### Confusion Matrix

This data set, which is also called an error matrix, is a pivot table layout. Each row represents an instance of a predicted class, and each column represents an instance in an actual class. This table reports the number of false positives, false negatives, true positives, and true negatives, which are used to compute precision, recall, and F1 accuracy metrics.

The Confusion Matrix data set is outputted when you select these model and algorithm combinations.

| Model | Algorithms |
|---|---|
| Binary Classification | Logistics Regression<br>CART (Decision Tree)<br>Naive Bayes<br>Neural Network<br>Random Forest<br>Support Vector Machine |
| Multi Classification | CART (Decision Tree)<br>Naive Bayes<br>Neural Network<br>Random Forest<br>Support Vector Machine |

### Drivers

This data set provides information about the columns that determine the target column values. Linear regressions are used to identify these columns. Each column is assigned coefficient and correlation values. The coefficient value describes the column's weight-age used to determine the target column's value. The correlation value indicates the relationship direction between the target column and dependent column. For example, if the target column's value increases or decreases based on the dependent column.

The Drivers data set is outputted when you select these model and algorithm combinations.

| Model | Algorithms |
|---|---|
| Numeric | Linear Regression<br>Elastic Net Linear Regression |
| Binary Classification | Logistics Regression<br>Support Vector Machine |
| Multi Classification | Support Vector Machine |

### Hitmap

This data set contains information about the decision tree's leaf nodes. Each row in the table represents a leaf node and contains information describing what that leaf node represents, such as segment size, confidence, and expected number of rows. For example, expected number of correct predictions = Segment Size * Confidence.

The Hitmap data set is outputted when you select these model and algorithm combinations.

| Model | Algorithm |
|---|---|
| Numeric | CART for Numeric Prediction |

### Residuals

This data set provides information on the quality of the residual predictions. A residual is the difference between the measured value and the predicted value of a regression model. This data set contains an aggregated sum value of absolute difference between the actual and predicted values for all columns in the data set.

The Residuals data set is outputted when you select these model and algorithm combinations.

| Model | Algorithms |
|---|---|
| Numerics | Linear Regression<br>Elastic Net Linear Regression<br>CART for Numeric Prediction |
| Binary Classification | CART (Decision Tree) |
| Multi Classificatin | CART (Decision Tree) |

### Statistics

This data set's metrics depend upon the algorithm used to generate it. Note this list of metrics based on algorithm:

- Linear Regression, CART for Numeric Prediction, Elastic Net Linear Regression - These algorithms contain R-Square, R-Square Adjusted, Mean Absolute Error(MAE), Mean Squared Error(MSE), Relative Absolute Error(RAE), Related Squared Error(RSE), Root Mean Squared Error(RMSE).
- CART(Classification And Regression Trees), Naive Bayes Classification, Neural Network, Support Vector Machine(SVM), Random Forest, Logistic Regression - These algorithms contain Accuracy, Total F1.

This data set is outputted when you select these model and algorithm combinations.

| Model | Algorithm |
|---|---|
| Numeric | Linear Regression<br>Elastic Net Linear Regression<br>CART for Numeric Prediction |

| Model | Algorithm |
|---|---|
| Binary Classification | Logistics Regression<br>CART (Decision Tree)<br>Naive Bayes<br>Neural Network<br>Random Forest<br>Support Vector Machine |
| Multi Classification | Naive Bayes<br>Neural Network<br>Random Forest<br>Support Vector Machine |

**Summary**

This data set contains information such as Target name and Model name.

The Summary data set is outputted when you select these model and algorithm combinations.

| Model | Algorithms |
|---|---|
| Binary Classification | Naive Bayes<br>Neural Network<br>Support Vector Machine |
| Multi Classification | Naive Bayes<br>Neural Network<br>Support Vector Machine |

# Register and Use Oracle Machine Learning Models

You can register and use Oracle machine learning models from Oracle Database or Oracle Autonomous Data Warehouse to score data in Oracle Analytics. Use the data flow editor to apply the machine learning models to your data.

**Topics:**



## How Can I Use Oracle Machine Learning Models in Oracle Analytics?

Oracle Analytics allows you to register and use Oracle machine leaning models that reside in database data sources.

Using Oracle machine learning models with Oracle Analytics greatly increases the level of predictive analytics that you can perform on data sets because the data and the model reside in the database, the data scoring is performed in the database, and

the resulting data set is stored in the database. This allows you to use the Oracle machine learning execution engine to score large data sets.

You can register and use Oracle machine learning models from these database data sources:

- Oracle Autonomous Data Warehouse
- Oracle Database

All of the database's Oracle machine learning models that were created using the Oracle Machine Learning for SQL API (OML 4SQL) are retrieved for registration in Oracle Analytics. Your database permissions determine the Oracle machine learning models that are available for you to register and use.

You can also create predictive models in Oracle Analytics. See What Are Oracle Analytics Predictive Models?

## Typical Workflow to Register and Use Oracle Machine Learning Models

Here are the common tasks for registering Oracle machine learning models, and how to apply the models to data sets and use them in projects.

| Task | Description | More Information |
|---|---|---|
| Register the Oracle machine learning model | Use an Oracle Database or Oracle Autonomous Data Warehouse connection to find Oracle machine learning models and register them for use with Oracle Analytics. | Register Oracle Machine Learning Models in Oracle Analytics |
| Apply a model to your data using a data flow | Create a data flow to apply an Oracle machine learning model to your data and generate a data set that includes the predicted trends and patterns. | Apply a Predictive or Oracle Machine Learning Model to a Data Set |

## Register Oracle Machine Learning Models in Oracle Analytics

The Oracle machine learning models must be registered in Oracle Analytics before you can use them to predict data. You can register and use models that reside in your Oracle Database or Oracle Autonomous Data Warehouse connections.

1. In the Home page, click Page Menu and select **Register ML Model**.
2. In the Register an ML Model dialog, select a connection. If needed, click **Create Connection** to create a connection to the Oracle Database or Oracle Autonomous Data Warehouse data source containing the Oracle machine learning model that you want to use.

   The Select a Model to Register dialog is displayed and contains a list of the Oracle machine learning models that you have permission to access.
3. In the Select a Model to Register dialog, click the model that you want to register and review the model's information. For example, the mining class and algorithm

used to build the model, the target the model predicts, the columns the model is trained on, model predictions, and parameters.

4. Click **Register**.
5. From the Home page, click **Navigator**, and then click **Machine Learning** to confirm that the model was imported successfully.

# Apply a Predictive or Oracle Machine Learning Model to a Data Set

Use the data flow editor to score a predictive model on any data set, or score an Oracle machine learning model on a data set in its corresponding database.

Running the model outputs a new data set with columns containing predicted values that can be used for analysis and visualization.

When you run a predictive model, the data is moved into and processed by Oracle Analytics. When you run an Oracle machine learning model, data isn't moved from the database into Oracle Analytics. Instead the model resides, is processed, and the output data set is stored in the database.

1. In the Home page, click **Create** and select **Data Flow**.

   The Add Data Set pane is displayed.
2. Select the data set that you want to apply the model to. If you're applying an Oracle machine learning model, then you must pick a data set from the same Oracle Database or Oracle Autonomous Data Warehouse where the model exists. Click **Add**.
3. In the data flow editor, click **Add a step (+)**.
4. Navigate to the bottom of the list and click **Apply Model**.
5. In the Select Model dialog, select the model. If you're applying an Oracle machine learning model, then only the models registered for the corresponding Oracle Database or Oracle Autonomous Data Warehouse data set are displayed. Click **OK**.
6. Go to the Outputs section and inspect the columns returned by the model. Select the columns that you want outputted with the data set, and update the **Column Name** fields as needed.

   Output columns vary depending on the model type. For example, for numeric prediction, output columns include PredictedValue and PredictedConfidence. And for clustering, output columns include the clusterId.
7. Go to the Inputs section and inspect and adjust how the columns in the scoring data set are mapped to the columns in the model. The parameters section displays parameters specific to the model type. For example, if you use a clustering model for scoring, maximum null values present is a parameter that you can provide for the scoring process. This parameter is used in the missing value imputation.

   If you're working with an Oracle machine learning model, then the model and the mapped input data types must match. To view the model's data types:

   a. On the Home Page, click **Console**.
   b. Click **Machine Learning** and go to the Models tab.

c. Locate the model, click its **Actions Menu**, and click **Inspect**.

d. Go to the Details tab and expand the Input Columns section to view the data types.

8. In the data flow editor, click **Add a step (+)** and select **Save Data**.
9. Enter a name in the **Name** field, and in the **Save data to** field confirm or select where to save the output data. If you're working with an Oracle machine learning model, then the data set's connection information defaults to the input data set's connection.
10. Set data preferences as needed in the **Treat As** and **Default Aggregation** fields.

    When you save data, the apply model appends the model's output columns that you selected to the data set.
11. Click **Save**, enter a name and description for the data flow, and click **OK** to save the data flow.
12. Click **Run Data Flow** to create the data set.

# Add a Predictive Model to a Project

When you create a scenario in a project, you apply a predictive model to the project's data set to reveal the trends and patterns that the model was designed to find.

> **Note:**
>
> You can't apply an Oracle machine learning model to a project's data.

After you add the model to the project and map the model's inputs to the data set's columns, the Data Panel contains the model's objects, which you can drag and drop onto the canvas. Machine learning generates the model's values based on the visualization's corresponding data columns.

1. In the Home page, click **Create** and select **Project**.

   The Add Data Set pane is displayed.
2. Select the data set that you want to use to create the project and click **Add to Project**.
3. In the Data pane, click **Add**, and select **Create Scenario**.
4. In the Create Scenario - Select Model dialog, select a model and click **OK**.

   You can only apply a predictive model. You can't apply an Oracle machine learning model.

   If each model input can't be matched to a data element, then the Map Your Data to the Model dialog is displayed.
5. If the Map Your Data to the Model dialog is displayed, then in the **Data Set** field, select the data set to use with the model.
6. Match the model input and data elements as needed. Click **Done**.

   The scenario is displayed as a data set in the Data Elements pane.

7. Drag and drop elements from the data set and the data model onto the Visualize canvas.
8. To adjust the scenario, right-click the scenario in the Data Elements pane and select **Edit Scenario**.
9. Change the data set and update the model input and data elements mapping as needed.
10. Click **Save** to save the project.

11

# Curate Your Data Using Data Flows

You can use data flows to produce curated (combined, organized, and integrated) data sets.

Video

**Topics:**



## Typical Workflow to Curate Data with Data Flows

Here are the common tasks for creating curated data sets with data flows.

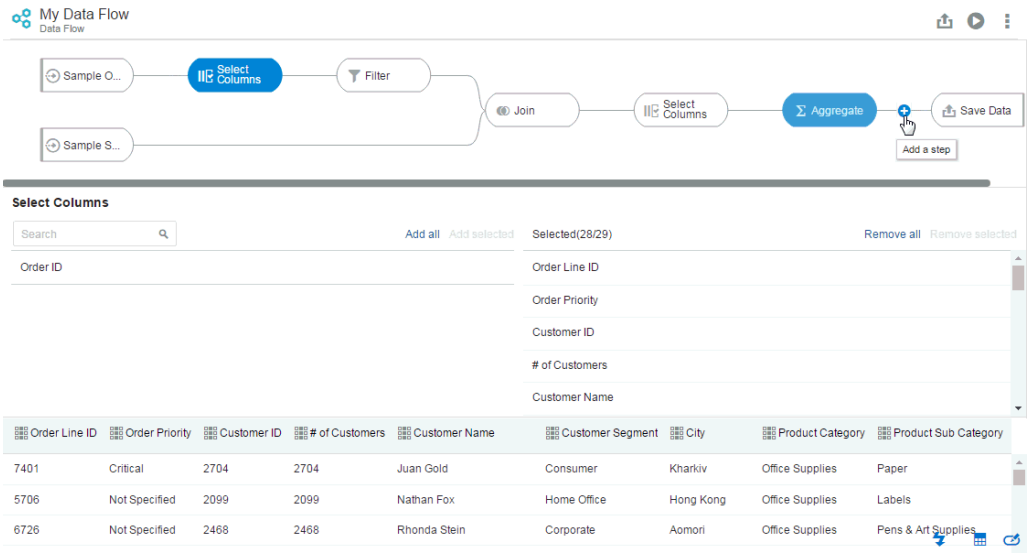
| Task | Description | More Information |
|---|---|---|
| Create a data flow | Create data flows from one or more data sets. | Create a Data Flow |
| Run a data flow | Run (that is, execute) data flows to create data sets. | Run a Data Flow |
| Run a saved data flow | Run a saved data flow to create data sets or to refresh the data in a data set. | Run a Saved Data Flow |
| Create a sequence of data flows | Create and save a sequential list of data flows. | Create a Sequence of Data Flows |
| Use Machine Learning to analyze data | Create, interpret, and score models using data flows. | Typical Workflow to Create and Use Oracle Analytics Predictive Models |
| Manage data flows | Monitor, run, schedule, import and export flows. | Manage Your Data Flows |

# About Data Flows

Data flows enable you to organize and integrate your data to produce a curated data set that your users can analyze.

For example, you might merge two data sets containing order data, strip out columns you don't need, aggregate the value of orders, and save the results in a new data set named `Order Summary`.

**Figure 11-1 A data flow in the data flow editor**




To build a data flow, you add steps. Each step performs a specific function, for example, add data, join tables, merge columns, transform data, save your data. Use the data flow editor to add and configure your steps. Each step is validated when you add or change it. When you've configured your data flow, you execute it to produce a data set.

To add steps, either right-click an existing step and click **Add step** or drag a step from the left-hand **Data Flow Steps** pane and drop it into the data flow editor.

When you add your own columns or transform data, you can use a wide range of SQL operators (for example, BETWEEN, LIKE, IN), conditional expressions (for example, CASE), and functions (for example, Avg, Median, Percentile).

You can save the output data from a data flow in either a data set or in one of the supported database types. If you save data to a database, you can transform the data source by overwriting it with data from the data flow. The data source and data flow tables must be in the same database and have the same name. Before you start, create a connection to one of the supported database types.

To find out which databases you can write to from a data flow, refer to the More Information column in Supported Data Sources.

# Create a Data Flow

You can create a data flow from one or more data sets. With a data flow, you produce a curated data set that you can use to create meaningful visualizations.

1. On the Home or Data page, click **Create** and select **Data Flow**.
2. In the Add Data Set dialog, select a data set or data source and click **Add**.

   You can select an existing data set or click **Create Data Set** to create a new one based on a file, local subject area, or database connection.

   In the data flow editor, you can add more data sources at any time by clicking **Add Step (+)**, then clicking **Add Data**.
3. In the Add Data pane, configure your data. For example, you can:
   - Replace the selected data set (click the data source name next to **Add Data -**).
   - Include or exclude columns.
   - Rename columns.
   - Prompt for a data source when the data flow is executed (using the **When Run Prompt to select Data Set** option).
4. Build your data flow:
   - For each function that you want to perform, click **Add a step (+)**, click the step type you want, then specify the properties in the Step editor pane.
     **Tip**: Hover over the last step to display the **Add a step (+)** option. You can also edit your flow and add steps using **Options** in the Column header. For example, you can rename, reformat, merge, or transform columns.
   - To remove a step, hover over the step and click **X** or right-click the step and click **Delete**. If you've invalidated part of the data flow by deleting a step you'll see a red X icon over the offending step.
   - To undo or redo an edit in a data flow that you haven't saved, go to the workflow diagram panel toolbar and click **Undo Last Edit** or **Redo Last Edit**.
   - At the end of your data flow, add a **Save Data** step and specify a meaningful name.
5. Save your data flow.

   You can start the data flow now using the **Run Data Flow** option or later using the **Data Flows** panel on the Data page. If you run it now you can access the generated data set on the **Data Sets** panel on the Data page).

## Zoom Controls in Data Flow Editor

The zoom controls help you increase or decrease the data flow diagram view and inspect the flow.

Use the zoom in button **(+)** or the zoom out button **(-)** in the data flow editor toolbar to reset the current zoom level. You can set the zoom level from 30 to 100. In the zoom controls, you can also insert a zoom level number or select a predefined level from the menu.

You can't see the **Remove step** and **Add a step (+)** icons when zoom level is below 100. Go to a step and right-click to select the **Delete** or **Add step** options.

# Run a Data Flow

Run (that is, execute) a data flow to produce a data set that you can use to create visualizations.

To successfully run a data flow, it must be free of validation errors.

1. Create or open the data flow that you want to execute and produce a data set from.
2. Click **Run Data Flow** to execute the data flow. If there is no validation error, a completion message is displayed.

   When you execute a data flow without saving it, the data flow isn't saved and isn't displayed in the Data Flows list. Therefore, the data flow isn't available for you to modify or run.

   Go to the Data page and select Data Sets to see your resulting data set in the list.
3. Click **Save** or **Save As**. In the Save Data Flow As dialog enter a **Name** and **Description** to identify your data flow.

   Go to the Data page and select Data Flows to see your resulting data flow in the list.

# Run a Saved Data Flow

You can run a saved data flow to create a new data set or to refresh existing data.

1. In the Data page, go to the Data Flows section, and locate the data flow that you want to run.
2. Right-click the data flow and select **Run** now (or click **New schedule** to run later).
   - Complex data flows take longer to run. While the data flow is running, you can go to and use other parts of the application, and then come back to the Data Flows pane to check the status of the data flow.
   - You can cancel a long-running data flow. To do so, go to the Data Flows section, click the data flow's **Action menu** and select **Cancel**.
   - If it's the first time you're running a data flow, then a new data set is created, and you can find it in the Data Sets section of the Data page. The data set contains the name that you specify on the data flow's **Save Data** step. If you've run the data flow before, then the resulting data source already exists, and its data is refreshed.
   - When creating a new database data source, set the database's query mode to **Live**. Setting the query mode to **Live** allows the data flow to access data from the database (versus the data cache) and pushes any expensive operations such as joins to the database.

# Reuse a Data Flow

In a data flow, you can add parameters so you can reuse the data flow with a different source data set or use different criteria to process and select data. Parameters help

you identify the type of data appropriate for the data flow and if you want to select an alternative data set when running or scheduling the data flow. You can also apply parameters to modify default values when creating an Essbase cube.

For example, using a parameter you can:

- Process a new data set that has the same format as the default input data set.
- Process and store different aspects of a large data set based on date range, individual departments, or regions into alternative target data sets.

In the Step editor pane, you can apply parameters for the following steps:

| Step Name | Parameter Field |
|---|---|
| Add Data | 1. Select the **When Run Prompt to select Data Set** option.<br>2. Provide the **Name** and **Prompt** values for the parameter. |
| Save Data | 1. Select the **When Run Prompt to specify Data Set** option.<br>2. Provide the **Name** and **Prompt** values for the parameter. |
| Create Essbase Cube | 1. Select the **When Run Prompt to specify Data Set** option.<br>2. Provide the **Cube** name, **Application** name, and **Prompt** value for the parameter. |

# Apply Incremental Processing to a Data Flow

Use incremental processing to determine the last data processed in the data flow and to process only the newly added data.

1. Select a data element column as an incremental identifier for the data set.

   You can select an incremental identifier only for those data sets that are sourced through database connections.

   a. Go to the Data page and select **Data Sets**.

   b. Select a data set and click the **Actions menu** or right-click, then select **Open**.

   c. Click **Edit Data Set** on the Results toolbar.

   d. Select the data set node in the diagram. From the **New Data Indicator** list, select a column, then click **Save**.

2. Apply incremental processing to the data flow using the data sets for which you've selected the incremental identifier.

   a. Create or open the data flow in which you want to apply incremental processing.

   b. In the Data Flow editor select the data set.

   c. In the Step editor pane, select **Add new data only** to mark the data set as incremental.

   d. Click **Save**.

In a data flow with multiple data sets, you can select only one data set as incremental. If you try to select a second data set as incremental, you see a warning message. Click **Yes** to enable incremental processing for the second data set for which you've

selected **Add new data only**. Incremental processing is deselected for the first data set.

# Modify Parameter Prompts When You Run a Data Flow

Parameter prompts are displayed before the job runs, when you run a data flow with parameter prompts. Prompts allow you to review the default values or settings and to select or define an alternate value or setting.

1. Go to the Data page and click Data Flows to select the data flow with parameter prompts that you want to run.
2. Click the data flow's **Actions menu** or right-click and select **Run**.
3. In the Data Flow Prompt dialog, either use the default values or define alternate values.
   - In the Sources section, click the default Target - existing data set name, then select a new source data set in the Add Data Set dialog. Click **Add**.
   - In the Targets section, do one of the following:
     - Change the default Target - existing data set name.
     - For a data flow with **Create Essbase Cube** step, change the default Target - Application and Target - Cube names.
4. Click **OK**.

# Customize the Names and Descriptions of Data Flow Steps

You can change the names of steps to make flows easier to understand, and annotate data flows by adding your own descriptions.

1. Open a data flow.
2. Click the step you want to customize.

   Notice that the step name is displayed in the panel below the step diagram.
3. In the panel below the step diagram, click the step name (for example, **Merge Columns**).
4. Use the **Name** and **Description** fields to change the default values.

   The new name that you specify is appended to the default name.
5. To save your changes, press **Enter**.

# Create a Sequence of Data Flows

A sequence is a saved sequential list of specified data flows and is useful when you want to run multiple data flows as a single transaction. If any flow within a sequence fails, then all the changes done in the sequence are rolled back.

Video

1. On the Home page click **Create** and select **Sequence**.
2. Drag and drop the data flows and sequences to the Sequence pane.

3. Click the menu icon to move an item up or down in the list, and to remove an item.
4. Click **Save**. When you save a sequence, it's displayed in the Sequence area of the Data page.
5. Go to the Sequence area of the Data page, select the sequence, and click **Execute Sequence**.
6. Go to the Data page and click **Data Sets** to see the list of resulting data sets.

# Manage Your Data Flows

Manage your data flows on the Data Flows page.

From the Home page, click **Navigator**, then **Data**, then **Data Flows**. Hover over a data flow and use the **Actions menu** to access the options described in the table below.

| Option | Description |
|---|---|
| Run | Execute the data flow. |
| Open/Open in a New Tab | Open the data flow for editing. |
| New schedule | Execute the data flow regularly. |
| Inspect | View general information about the data flow, such as the source and target data, when the flow was last executed, scheduled executions, and execution history. |
| Export | Export a data flow project with its dependent data and credentials as a .DVA file to the download folder of your machine. Use the export and import options to migrate data flow projects from one system to another or back up your data flow projects.<br>To import a data flow project that you've downloaded, on the Data Flows page, click **Page Menu**, then click **Import Project/Flow**. Follow the on-screen instructions to select a local .DVA file to import. |
| Delete | Remove the data flow from your system. |

# Using Steps

You build data flows using steps to curate your data. Steps are functions that change your data in a specific way. For example, steps can aggregate values, perform time series analysis, or perform machine learning algorithms.

| Step | Use this step to: | More Information |
|---|---|---|
| Add Columns | Add a new output data column to your data flow using a wide range of functions, conditional expressions, and SQL operators. | Add Columns in a Data Flow |
| Add Data | Add a data source to your data flow. | Add Data in a Data Flow |
| Aggregate | Apply aggregate functions to group data in a data flow. | Add Aggregates to a Data Flow |
| Analyze Sentiment | Detect sentiment for a text column by applying a sentiment analysis to the data flow. | Add a Sentiment Analysis to a Data Flow |
| Apply Model | Apply a machine learning model to your data (also known as scoring a data model). | Apply a Predictive or Oracle Machine Learning Model to a Data Set |

| Step | Use this step to: | More Information |
|---|---|---|
| Bin | Assign your data values into categories, such as high, low, or medium. | Create a Bin Column in a Data Flow |
| Branch | Creates multiple outputs from a data flow using a branch. | Create Multiple Pipelines in a Data Flow Using a Branch |
| Create Essbase Cube | Create an Essbase cube from a data set. | Create and Customize an Essbase Cube in a Data Flow |
| Cumulative Value | Group data by applying cumulative aggregate functions in a data flow. | Add Cumulative Values to a Data Flow |
| Database Analytics | Use advanced analytic functions, such as anomaly detection, unpivot, sampling, and advanced clustering (Requires Oracle database or Oracle Autonomous Data Warehouse). | Add Database Analytics to a Data Flow |
| Filters | Use filters to limit the data in a data flow output. | Filter Your Data in a Data Flow |
| Group | Create a group column of attribute values in a data set. | Create a Group in a Data Flow |
| Join | Join multiple tables or data sets. | Add a Join in a Data Flow |
| Merge Columns | Combine two or more columns in your data flow. | Merge Columns in a Data Flow |
| Merge Rows | Combine two or more rows in your data flow. | Merge Rows in a Data Flow |
| Rename Columns | Change the name of data columns to something more meaningful. | Rename Columns in a Data Flow |
| Save Data | Before running a data flow, modify or select the database name, attribute or measure, and aggregation rules for each columns of the output data set. | Save Output Data from a Data Flow |
| Save Model | Change the default model name (untitled) and provide a description. | Save Model |
| Select Columns | Specify which data columns to include in your data flow. | Select Columns to Include in a Data Flow |
| Split Columns | Extract useful data from within data columns. | Split Columns in a Data Flow |
| Time Series Forecast | Apply a time series forecast calculation to a data set to create additional rows. | Add a Time Series Forecast to a Data Flow |
| Train Binary-Classifier | Train a machine learning model to classify your data into one of two predefined categories. | Train a Binary Classifier Model in a Data Flow |
| Train Clustering | Train a machine learning model to segregate groups with similar traits and assign them into clusters. | Train a Clustering Model in a Data Flow |
| Train Multi-Classifier | Train a machine learning model to classify your data into three or more predefined categories. | Train a Multi-Classifier Model in a Data Flow |
| Train Numeric Prediction | Train a machine learning model to predict a numeric value based on known data values. | Train a Numeric Prediction Model in a Data Flow |
| Transform Column | Modify data in a column using a wide range of functions, conditional expressions, and SQL operators. | Transform Data in a Data Flow |

# Add Columns in a Data Flow

You can add columns to your target data and customize the format. For example, you might calculate the value of your stock by multiplying the number of units in a `UNITS` column by the sale price in a `RETAIL_PRICE` column.

Use the **Add Columns** step in the data flow editor.

1. Click **Add a step (+)**, and select **Add Columns**.
2. In the Add Columns pane, use the expression builder to define your column. For example, to calculate the value of stock items you might specify `UNITS * RETAIL_PRICE`.

   Select SQL operators, functions, and conditional expressions from the expression pick list.

# Add Data in a Data Flow

When you create a new data flow and select a data set, you'll see a step with the name of your data set. You can add additional data from multiple data sources to your data flow.

Use the **Add Data** step in the data flow editor. If you've created a new data flow project, your data set will be selected.

1. Use the options on the Add Data pane to configure the data set. For example, change the default name, or include and exclude columns.
2. To add another data set to your flow, click **Add a step (+)**, and select **Add Data**.

   If matching columns are found in data sets, a Join step is automatically added to enable you to define the relationship between the data sets. For example, you might want to combine rows from two data sets where the `CustomerID` in the first data set matches the `CustomerID` in the second data set.
3. If you don't get a Join step automatically, click **Add a step (+)**, and select **Join**.

   To complete the join, on the data flow diagram click the circle on the dotted line between the data source step and the Join step. Then use the Join pane to configure the relationship between the data sets
4. Click your data set step again and use the options on the Add Data pane to configure the data set.

| Field | Description |
|---|---|
| Add Data - *<Data source name>* | Click this pane heading to edit the step name and description. |
| Select... | Use this option to change the data set or data source. Changing the data set or data source might break other steps in your flow. |
| When Run Prompt to select Data Set | Select this option to supply the name of the output data set when the data flow is executed. For example, you might want to specify a different name for the output data set each time the flow is executed. |

# Add Aggregates to a Data Flow

Create group totals by applying aggregate functions such as count, sum, and average.

Use the **Aggregate** step in the data flow editor.

1. Click **Add a step (+)** and select **Aggregate**.

   In the Aggregate pane you'll see a suggested aggregate column for each numeric column.

2. Use the options on the Aggregate pane to configure your aggregate:

| Field | Description |
|---|---|
| Aggregate | Select a column you want to add to the aggregate |
| Function | Select an aggregate function such as Sum, Average, Minimum, or Count to apply to the selected column. |
| New column name | Change the name of the aggregate column. |

3. Add or remove aggregates.
   - To remove an aggregate, select the aggregate and click **X**.
   - To see the **Add Aggregate** option, scroll to the bottom of the Aggregate pane.

# Create a Bin Column in a Data Flow

Use a bin to categorize your data by creating a new column based on the value of a measure. For example, you might categorize values for `RISK` into three bins for low, medium, and high.

Use the **Bin** step in the data flow editor.

1. Click **Add a step (+)**, and select **Bin**.

   You also create bins when you add columns using the **Add Column** step.

2. Select the column whose values you want to categorize.

3. Use the options on the Bin pane to configure your bin:

| Field | Description |
|---|---|
| Bin | You'll see the column that you selected in Step 2. To categorize values in a different column, click the column name and select a different column. |
| Method | Specify how the data boundaries are calculated.<br>• In the **Manual** method, the range is divided by the number of bins.<br>• In the **Equal Width** method, the histogram range is divided into intervals of the same size. For equal width binning, the column values are measured, and the range is divided into equal-sized intervals. The edge bins can accommodate very low or very high values in the column.<br>• In the **Equal Height** method, the height of each bin is same or very slightly different but the histogram range is equal. For equal height or frequency binning, the intervals of each bin is based on each interval containing approximately the equal number of elements (that is, records). **Equal Height** method is preferred specifically for the skewed data. |

| Field | Description |
|---|---|
| Histogram View | Based on the **Method** selected, the histogram range (width) and histogram count (height) of the bins are updated. |
| List View | If you select the **Manual** method, you can change the name of the bins, and you can define the range for each bin. |

Based on your changes, the data preview (for example, the bin column name) is updated.

## Create Multiple Pipelines in a Data Flow Using a Branch

Creates multiple outputs from a data flow using a branch. For example, if you have sales transactions data based on country, you might save data for United States in the first branch and data for Canada in the second branch.

Use the **Branch** step in the data flow editor.

1. Click **Add a step (+)** and select **Branch**.

   You'll see a **Branch** step and two **Save Data** steps added to the data flow. Select the **Branch** step and use the **Branch into** option to add or remove branches. The minimum number of branches is two, and the maximum is five.

2. To configure each branch, click connection line between the **Branch** step and the **Save Data** step, click **Add a step (+)** and select a step type that processes your branch.

   For example, you might add a **Filter** to the first branch that saves data from United States, and add a **Filter** to the second branch that saves data from Canada. Or, you might use the **Split Columns** step to save some columns in the first branch and other columns in the second branch.

3. Click each **Save Data** step and in the Save Data Set pane specify the properties for saving the output data sets.

## Create and Customize an Essbase Cube in a Data Flow

Create an Essbase cube from a spreadsheet or database.

You can create Essbase cubes only for Oracle Analytics Cloud – Essbase. When selecting an Essbase connection for creating a cube, you might see remote connections to on-premises Oracle Essbase instances. You can't create a cube from data in on-premises Oracle Essbase instances.

Use the **Create Essbase Cube** step in the data flow editor.

1. Click **Add a step (+)**, and select **Create Essbase Cube**.
2. In the Create Essbase Cube pane, specify the values for creating the cube such as connection and application name.
3. To configure the input columns, do the following:
   a. Move the slider to enable the **Customize Cube** option.
   b. Select the number of rows you want to analyze and click **Configure**.
   c. Perform the following actions for each column in the Dimensions, Measure, and Skip sections:
      - Cut

- Paste as Sibling
- Paste as Child
- Skip
- Delete

| Section and Designation Type | Cut | Paste as Sibling | Paste as Child | Skip |
|---|---|---|---|---|
| Dimension Header | No | No | Yes | No |
| Dimension | Yes | Yes | Yes | Yes |
| Generation | Yes | Yes | Yes | Yes |
| Alias, Attribute, UDA | Yes | Yes | No | Yes |
| Measure Header | No | No | Yes | No |
| Measure | Yes | Yes | Yes | Yes |
| Skip Header | No | No | Yes | No |
| Skip | Yes | Yes | No | No |

d. Change the following column values:
   - Column name in the Data Elements column.
   - Designation type in the Treat As column.

4. Select the **When Run Prompt to specify Data Set** option to apply parameters to change the default values when creating the Essbase cube.

## Cut, Paste, and Skip Rules

The cut, paste, and skip actions you perform for each column follow pre-configured rules.

- When you skip a column, it moves to the Skip section of the table. You can only paste a column as a sibling of the Skip header, or as a sibling of any skipped column.
- Any columns that are pasted as a Measure follow the rule of the paste command. Measure hierarchies are allowed, but the designation type doesn't change.
- **Paste as Child** action for Dimension columns:
  - When a column is pasted as a child of the Dimensions header, the cut column is pasted as a Dimension.
  - When a column is pasted as a child of the Dimension column:
    * The cut column is pasted as a Generation.
    * If the Dimension column already has a Generation child, the existing Generation (and its children) becomes the children of the new Generation column.
  - When a column is pasted as a child of the Generation column:
    * The cut column is pasted as a child of the Generation if the cut column is an Alias, Attribute, or UDA.

    - The cut column is pasted as a Generation if the cut column isn’t an Alias, Attribute, or UDA.
  - **Paste as Child** for any Dimension column isn't allowed if the target is an Alias, Attribute, or UDA.
- **Paste as Sibling** action for Dimension columns:
  - When a column is pasted as a sibling of a Dimension column, it's pasted as a Dimension.
  - When a column is pasted as a sibling of an Attribute, Alias, or UDA and it isn’t an Alias, Attribute, or UDA, the column is pasted as an Attribute.
  - **Paste as Sibling** for any Dimension column isn't allowed if the target is a Generation.

## Designation Change Rules for Generation Columns

The Generation columns follow specific pre-configured rules when you change their designation type.

- **Generation** to **Attribute**/**Alias**/**UDA** - If the Generation column has any children, they move up a level and become children of the Generation column’s parent.
- **Attribute**/**Alias**/**UDA** to **Generation** - If the new Generation column has a sibling Generation column, the existing Generation column (and its children) become children of the new Generation column.

# Add Cumulative Values to a Data Flow

You can calculate cumulative totals such as moving aggregate or running aggregate.

Use the **Cumulative Value** step in the data flow editor.

1. Click **Add a step (+)**, and select **Cumulative Value**.
2. Use the options on the Cumulative Value pane to configure your aggregate.

| Field | Description |
|---|---|
| Aggregate | Select the data column to calculate. |
| Function | Select the cumulative function to apply. |
| Rows | You can edit this field only for specific functions. |
| New column name | Change the aggregate column name. |
| (+) Aggregate | Add an aggregate column. |
| (+) Sort Column | Specify how you'd like to sort each new cumulative column. |

# Add Database Analytics to a Data Flow

Database analytics enable you to detect anomalies, cluster data, sample data, and unpivot data. Database analytics are executed in the database, not in Oracle

Analytics, therefore you must be connected to an Oracle database or Oracle Autonomous Data Warehouse.

Use the **Database Analytics** step in the data flow editor.

Before you start, create a connection to your Oracle database or Oracle Autonomous Data Warehouse and use it to create a data set.

1. In the data flow editor, click **Add a step (+)**, and select **Database Analytics**.

   If you aren't connected to an Oracle database or Oracle Autonomous Data Warehouse, you won't see the **Database Analytics** option.

2. At the Select Database Analytics page, select a function type then click OK.

| Function Types | Description |
|---|---|
| Dymanic Anomaly Detection | Detect anomalies in your input data without a pre-defined model. For example, you might want to highlight unusual financial transactions.<br>When you deploy this function with large data sets, configure the partition columns to maximise performance. |
| Dynamic Clustering | Cluster your input data without a pre-defined model. For example, you might want to characterize and discover customer segments for marketing purposes.<br>When you deploy this function with large data sets, configure the partition columns to maximise performance. |
| Un-pivoting Data | Transpose data that's stored in columns into row format. For example, you might want to transpose multiple columns showing a revenue metric value for each year to a single revenue column with multiple value rows for the year dimension. You simply select the metric columns to transpose and specify a name for the new column.You'll get a new dataset with fewer columns and more rows. |
| Sampling Data | Selects a random sample percentage of data from a table. You simply specify the percentage of data you want to sample. For example, you might want to randomly sample ten percent of your data. |

3. On the **Analytics Operation <type>** pane, configure the operation.
   - Use the **Outputs** area to specify the data columns to analyze.
   - Use the **Parameters** area to configure options for the operation.

## Filter Your Data in a Data Flow

You use filters to limit the amount of data included in the data flow output. For example, you might create a filter to limit sales revenue data to the years 2017 through 2019.

Use the **Filter** step in the data flow editor.

1. Click **Add a step (+)**, and select **Filter**.
2. In the Filter pane, select the data element you want to filter:

| Field | Description |
|---|---|
| Add Filter (+) | Select the data element you want to filter, in the Available Data dialog. Alternatively, click **Data Elements** in the Data Panel, and drag and drop a data element to the Filter pane. |

| Field | Description |
|---|---|
| Filter fields | Change the values, data or selection of the filter (for example, maximum and minimum range). Based on the data element, specific filter fields are displayed. You can apply multiple filters to a data element. |
| Filter menu icon | Select a function to clear the filter selection and disable or delete a filter. |
| Filter pane menu icon | Select a function to clear all filter selections, remove all filters, and auto-apply filters. You can select to add an expression filter. |
| Add Expression Filter | Select to add an Expression Filter. Click **f(x)**, select a function type, and then double-click to add a function in the **Expression** field.<br>Click **Apply**. |
| Auto-Apply Filters | Select an auto-apply option for the filters, such as Default (On). |

The data preview is updated using the applied filter.

## Create a Group in a Data Flow

You can categorize non-numeric data into groups that you define. For example, you might put orders for lines of business `Communication` and `Digital` into a group named `Technology`, and orders for `Games` and `Stream` into a group named `Entertainment`.

Use the **Group** step in the data flow editor.

1. Click **Add a step (+)**, and select **Group**.
2. For each group that you want to create, use the Group pane:
   a. Use the pop list of columns to select the column you'd like categorize. For example, to categorize orders by line of business, you might select LINE_OF_BUSINESS.
   b. (Optional) Click the group name to change the default name `Group 1`. For example, you might change `Group 1` to `Technology`.
   c. (Optional ) In the **Name** field, change the default name of the new column from `new_name1` to a more meaningful name.
   d. In the center box, select one of more categories to add to the group. For example, to analyze line of business you might put `Communication` and `Digital` in a group named Technology.

      In the Preview Data pane, you'll see a new column with the groups that you defined displayed as the value for each row. For example, values might be Technology or Entertainment.
3. To add more groups, click **Group (+)**.

## Add a Join in a Data Flow

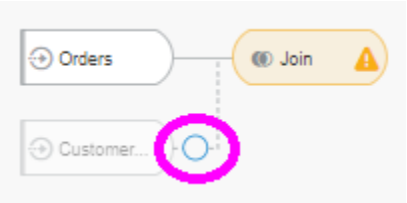
When you add data from multiple data sources to your data flow, you can join them on a common column. For example, you might join an `Orders` data set to a `Customer_orders` data set using a customer ID field.

When you use the **Add Data** step to add an extra data source, a **Join** step is automatically added to your data flow. But you can also manually add a **Join** step if you have more than one data source defined in your data flow.

Use the **Join** step in the data flow editor.

1. Add the data sources you'd like to join.
2. Select a data source, click **Add a step**, then click **Join**.

   You'll see a suggested connection with a node on the connection line.




3. Click the node on the connection line to complete the connection.
4. Use the options on the Join pane to configure your step.

| Field | Description |
|---|---|
| Keep rows | Use these options to specify how you want to join your data. Click an option to preview your merged data (if you're displaying the Data Preview pane). |
| Match columns | Specify the common field on which you'd like to join the data sources. |

## Merge Columns in a Data Flow

You can combine multiple columns into a single column. For example, you might merge the street address, street name, state, and ZIP code columns so that they display as one item in visualizations.

Use the **Merge Columns** step in the data flow editor.

1. Click **Add a step (+)**, and select **Merge Columns**.
2. Use the options on the Merge Columns pane to configure your merge:
   - **(+) Column** field - Select more columns you want to merge.
   - **Delimiter** field - Select a delimiter to separate column names (for example, Space, Comma, Dot, or Custom Delimiter).

## Rename Columns in a Data Flow

Rename columns to create more meaningful data column names in your generated data sets.

Use the **Rename Columns** step in the data flow editor.

1. Click **Add a step (+)**, and select **Rename Columns**.
2. Use the **Rename** fields to specify a more meaningful name for columns in your generated data set.

## Save Output Data from a Data Flow

For the data created by a data flow you can change the default name and description, specify where to save the data, and specify runtime parameters. If you're saving the

output from your data flow to a database, before you start, create a connection to one of the supported database types.

Use the **Save Data** step in the data flow editor.

1. Click **Add a step (+)** and select **Save Data**. Or, if you've already saved the data flow, then click the **Save Data** step.
2. In the Save Data Set pane, optionally change the default **Name** and add a **Description**.

   If you don't change the default **Name** value, you'll generate a data set named 'untitled'. After you run this data flow, you'll see the generated data set in the Data Sets page (click **Data** from the navigator on the Home page).
3. Click **Save data to** and select a location:
   - Choose **Data Set Storage** to save the output data in a data set in Oracle Analytics.
   - Choose **Database Connection** save the output data in one of the supported database types.
4. If you've selected **Database Connection**, specify details about the database connection.

   Before you start, create a connection to one of the supported database types.

   a. Click **Select connection** to display the Save Data to Database Connection dialog, and select a connection.

      You can save to a range of databases, including Oracle, Oracle Autonomous Data Warehouse, Apache Hive, Hortonworks Hive, and Map R Hive.

      To find out which databases you can write to, refer to the More Information column in Supported Data Sources.

   b. In the **Table** field, optionally change the default table name.

      The table name must conform to the naming conventions of the selected database. For example, the name of a table in an Oracle database can't begin with numeric characters.

   c. In the **When run** field, specify whether you'd like to replace existing data or add new data to existing data.
5. Select the **When Run Prompt to specify Data Set** option if you want to specify the name of the output data set or table at run time.
6. In the **Columns** table, change or select the database name, the attribute or measure, and the aggregation rules for each column in the output data set:

| Column name | Description |
|---|---|
| Treat As | Select how each output column is treated, as an attribute or measure. |
| Default Aggregation | Select the aggregation rules for each output column (such as Sum, Average, Minimum, Maximum, Count, or Count Distinct).<br>You can select the aggregation rules if a specific column is treated as a measure in the output data set. |
| Database Name | Change the database name of the output columns.<br>You can change the column name if you're saving the output data from a data flow to a database. |

When you run the data flow

- If you've selected data set storage, go to the Data page and select Data Sets to see your output data set in the list.
  - Click **Actions menu** or right-click and select **Inspect**, to open the data set dialog.
  - In the data set dialog, click **Data Elements** and check the Treat As and Aggregation rules that you've selected for each column in the **Save Data** step.
- If you're saving output data to a database, go to the table in that database and inspect the output data.

## Save Model

You can change the default name of your model and add a description.

Use the **Save Model** step in the data flow editor. You'll see this step added automatically in the data flow editor when you add one of the train model steps, for example, Train Numeric Prediction, or Train Binary Classifier.

1. Add one of the train model steps to your data flow. For example, Train Numeric Prediction, or Train Binary Classifier.
2. Click the **Save Model** step.
3. In the Save Model pane, optionally change the default **<Model name**, and specify a **Model description** to identify the model type and script used.

   If you don't change the default **Model name** value, you'll save a model named untitled. After you run this data flow, you'll see your new model in the Machine Learning page. Click **Machine Learning** from the navigator on the Home page to apply a saved model to your data.

## Select Columns to Include in a Data Flow

Select which columns to include in your data flow. By default, all data columns are included in your data flow.

Use the **Select Columns** step in the data flow editor.

1. Click **Add a step (+)**, and select **Select Columns**.
2. Use the on-screen options to select or remove columns.

## Add a Sentiment Analysis to a Data Flow

You can detect sentiment for a given text column by applying a sentiment analysis to your data flow.

Sentiment analysis evaluates text based on words and phrases that indicate a positive, neutral, or negative emotion. Based on the outcome of the analysis, a new column contains a Positive, Neutral, or Negative string type result.

Use the **Analyze Sentiment** step in the data flow editor.

1. Click **Add a step (+)**, and select **Analyze Sentiment**.
2. In the Analyze Sentiment pane and Output section, specify an output column for the emotion result value.

3. Optionally change the default column name 'emotion'.
4. In the Analyze Sentiment pane and Parameters section, specify the value for **Text to Analyze**.

   Select a text column with natural language content to analyze.

## Split Columns in a Data Flow

You can strip out useful data from columns of concatenated data. For example, if a column contains `001011Black`, you might split this data into two separate columns, `001011` and `Black`.

Use the **Split Columns** step in the data flow editor.

Before you start, turn on Data Preview so that you can see the new columns as you configure the split. If your data source has many columns, use a **Select Columns** step to remove extraneous columns first to improve the preview.

1. Click **Add a step (+)**, and select **Split Columns**.
2. Use the options on the Split Columns panel to configure the data flow.

| Field | Description |
|---|---|
| Split Column | Click **Select Column** to specify the data column you'd like to split. If a column is already chosen, click the column name to choose a different column. |
| On | Specify whether to split the column by delimiter or by position.<br>Select **Delimiter** if the column has separator characters, such as commas or spaces.<br>Select **Position** if the column doesn't have separator characters. If you split on position, you can only create two new columns. |
| Delimiter | (Displayed when **On** is set to **Delimiter**) Specify the separator used in your data column (for example, space, comma, custom). |
| Position | (Displayed when **On** is set to **Position**) Specify where the second column starts. For example, if your column contains `AABBBCCCDDD`, specify 6 to put `AABBB` in the first column and `CCCDDD` in the second column. |
| Number of parts to create | Specify the number of new columns to create when **On** is set to **Delimiter** (you can't change the default value 2 if **On** is set to **Position**). For example, if your source data column contains `AA BBBBB CCC DD`, you might select **4** to put each sub-string into a different column. |
| Occurrence | Specify how many of the sub-strings in the source column to include in each new column.<br>Examples based on data `AA BBBBB CCC DD` with **Delimiter** set to **Space**:<br>• If you set **Occurrence** to **1**, **Number of parts to create** to **1**, the new column contains AA. If you set the **Occurrence** to **2**, the new column contains `AA BBBBB`.<br>• If you set **Occurrence** to **1**, **Number of parts to create** to **2**, the first new column contains AA and the second new column contains `BBBBB CCC DD`.<br>• If you set **Occurrence** to **1**, **Number of parts to create** to **4**, the first new column contains AA, the second new column contains `BBBBB`, the third new column contains `CCC`, and the fourth new column contains `DD`. |
| New column *<number>* name | Change the default name (**New column <number>**) for new columns to a more meaningful name.<br>Use the adjacent check box to display or hide new columns. |

# Add a Time Series Forecast to a Data Flow

You can calculate forecasted values by applying a Time Series Forecast calculation.

A forecast takes a time column and a target column from a given data set and calculates forecasted values for the target column and puts the values in a new column. All additional columns are used to create groups. For example, if an additional column 'Department' with values 'Sales', 'Finance', and 'IT' is present, the forecasted values of the target column are based on the past values of the given group. Multiple columns with diverse values lead to a large number of groups that affect the precision of the forecast. Select only columns that are relevant to the grouping of the forecast.

Use a **Time Series Forecast** step in the data flow editor.

1. Click **Add a step (+)**, and select **Time Series Forecast**.
2. In the Time Series Forecast pane and Output section, specify an output column for the forecasted value.
3. In the Time Series Forecast pane, configure your forecast calculation:

| Field | Description |
|---|---|
| Target | Select a data column with historical values. |
| Time | Select a column with date information. Forecasted values use a daily grain. |
| Periods | Select the value that indicates how many periods (days) are forecasted per group. |

# Train a Binary Classifier Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Binary Classifier model to evaluate how accurately it classifies your data into one of two predefined categories. For example, you might predict whether a product instance will pass or fail a quality control test.

Use the **Train Binary Classifier** step in the data flow editor.

1. Click **Add a step (+)**, and select **Train Binary Classifier**.
2. At the Select Train Two-Classification Model Script dialog, select a script type, then click **OK**. For example, you might select Naive Bayes.
3. Click **Select a column** and select the data column to analyze.
4. Use the on-screen options to configure the script parameters.

Create and Train a Predictive Model

# Train a Clustering Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Clustering model to evaluate how accurately it segregates groups with similar traits and assigns them into clusters. For example, you might assign your costumers

into clusters (such as big-spenders, regular spenders and so on) based on their purchasing habits.

Use the **Train Clustering** step in the data flow editor.

1. Click **Add a step (+)**, and select **Train Clustering**.
2. At the Select Train Clustering Model Script dialog, select a script type, then click **OK**. For example, you might select Hierarchical Clustering for model training.
3. Use the on-screen options to configure the script parameters.

Create and Train a Predictive Model

## Train a Multi-Classifier Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Multi-Classifier model to evaluate how accurately it classifies your data into three or more predefined categories. For example, you might predict whether a piece of fruit is an orange, apple, or pear.

Use the **Train Multi-Classifier** step in the data flow editor.

1. Click **Add a step (+)**, and select **Train Multi-Classifier**.
2. At the Select Train Two-Classification Model Script dialog, select a script type, then click **OK**. For example, you might select Naive Bayes.
3. Click **Select a column** and select the data column to analyze.
4. Use the on-screen options to configure the script parameters.

Create and Train a Predictive Model

## Train a Numeric Prediction Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Numeric Prediction model to evaluate how accurately it predicts a numeric value based on known data values. For example, you might predict the value of a property based on square-footage, number of rooms, zip code, and so on.

Use the **Train Numeric Prediction** step in the data flow editor.

1. Click **Add a step (+)**, and select **Train Numeric Prediction**.
2. At the Select Train Numeric Prediction Model Script dialog, select a script type, then click **OK**. For example, you might select Random Forest for Numeric model training.
3. Click **Select a column** and select the data column to analyze.
4. Use the on-screen options to configure the train model.

Create and Train a Predictive Model

# Transform Data in a Data Flow

You can transform the column data of a data set in a data flow.

You can transform data in a column.

You can also quickly transform the data in a column by using the column menu option in Data Preview. The list of available menu options for a column depends on the type of data in that column. You can perform the following types of data transforms:

- Update or modify the data in a column.
- Group or merge multiple columns in a data set.
- Add a column to or remove a column from a data set.

Use the **Transform Column** step in the data flow editor.

1. To add a data transform step, do one of the following:
   - Click **Add a step (+)**, select **Transform Column**, then select a column
   - Drag and drop the **Transform Column** step from the Data Flow Steps panel to the workflow diagram panel and select a column.
   - Select a column in the Preview data panel and click **Options**, then select a transform option. See Column Menu Options for Quick Data Transformations.
2. In the Step editor pane, compose an expression or update the fields to configure the changes. You can review the changes in the Preview data panel.

   If you're composing an expression, do the following:
   - Click **Validate** to check if the syntax is correct.
   - If the expression is valid, click **Apply** to transform the column data.

# Merge Rows in a Data Flow

You can merge the rows of two data sources (known as a UNION command in SQL terminology).

Before you merge the rows, do the following:

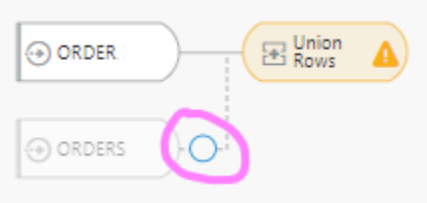
- Confirm that each data set has the same number of columns.
- Check that the data types of the corresponding columns of the data sets match. For example, column 1 of data set 1 must have the same data type as column 1 of data set 2.

Use the **Union Rows** step in the data flow editor.

1. In your data flow, add the data sources you want to merge.

   For example, you might add data sets named Order and Orders.
2. On one of the data sources, click **Add a step (+)** and select **Union Rows**.

   You'll see a suggested connection with a node on the connection line.




3. Click the node on the connection line to complete the connection.
4. Use the options on the Union Rows pane to configure your step.

| Field | Description |
|---|---|
| Keep | Use these options to specify how you want to join your data. Click an option to display an explanatory diagram and preview your merged data (if you're displaying the Data Preview pane). |

12

# Import and Share

You can import projects, or share projects with other users.

**Topics:**



## Import a Project File

You can import a project file created and exported by another user, or exported from another product such as Oracle Fusion Applications.

The import includes everything that you need to use the project such as associated data sets, connection string, connection credentials, and stored data.

1. On the Home page, click the **Page Menu** icon and then select **Import Project/ Flow**.
2. In the Import Project/Flow dialog, click **Select File** or drag a project file onto the dialog, then click **Import**.

## Export a Project or Folder as a File

You can export a project or folder as an archive file (.DVA) that another user can import.

The .DVA file includes items that you specify such as the associated data sets, connection strings, connection credentials, and stored data.

1. On the Home page, click **Navigator**, and then click **Catalog**.
2. In the Catalog page, highlight the project or folder that you want to share and click **Actions menu**, then select **Export** to open the Export dialog.
3. Click **File**.
4. For **Name**, keep the default name or enter a new name for the export file (.DVA file).
5. Enable the **Include Data** option to include the data when sharing a project or folder.

6. Enable the **Include Connection Credentials** option so that users can open the project without having to sign in. Use the following guidelines to set this field:
   - **Excel, CSV, or TXT data source** - These data sources don't use a data connection, therefore you can clear the **Include Connection Credentials** option.
   - **Database data source** - If you enable the **Include Connection Credentials** option, the user must provide a valid user name and password to load data into the imported project.
   - **Oracle Applications or Oracle Essbase data source** - Selecting the **Include Connection Credentials** option works if on the connection setup's Create Connection dialog you specified the **Always use these credentials** option in the **Authentication** field.

     If you clear the **Include Connection Credentials** option or specify the **Require users to enter their own credentials** option in the **Authentication** field, then the user must provide a valid user name and password to load data into the imported project.
7. If you enable **Include Data** or **Include Connection Credentials**, enter and confirm a password that the user must provide to import the project or folder and decrypt its connection credentials and data.
8. Click **Save**.

# Export a Visualization or Story as a File

You can share visualizations or stories as a file in a variety of formats, such as DVA (a visualization project), Powerpoint (PPTX), Adobe Acrobat (PDF), PNG, or CSV (data only).

1. Create or open a project.
2. On the Visualize or Narrate canvas, click the **Export** icon on the project toolbar, then click **File**.
3. Use the **Format** option to select the output format you want, and specify output options:
   - For **Powerpoint (pptx)**, **Acrobat (pdf)**, and **Image (png)** - Specify the file name, paper size, and orientation.
     When you share any of these visual formats, the visualization or pages are re-rendered based on the size and orientation you select. So if you're sharing a table, then your output file might not contain all of the table's rows and columns displayed in your visualization, canvas, or story.
   - For **Data (csv)** - Specify the output file name. This option only includes the data used in the project. The outputted file uses the data delimiter for your computer's locale. For example, if your locale is set to Brazil, then the delimiter for numeric decimals is a comma instead of a period, which is used when your locale is set to United States.
   - For **Package (dva)** - Specify whether to include project data and connection credentials. To enable users to open the project DVA file without having to enter a password, click **Include Connection Credentials** and specify the password.
4. Click **Save**.

# Export a Visualization's Data to a CSV File

You can export the data from a visualization to a CSV file. This enables you to open and update the visualization data in a compatible application such as Excel.

1. Open the visualization.
2. On the Visualize or Narrate canvas, click the **Export** icon on the project toolbar, then click **File** to open the File dialog.
3. In the **Format** option, select **Data (csv)**.
4. Click **Save**.

   The exported file uses the data delimiter for your computer's locale. For example, if your locale is set to Brazil, then the delimiter for numeric decimals is a comma instead of a period, which is used when your locale is set to United States.

# Email Projects and Folders

You can email the .DVA file of a project or folder to other users.

Selecting the option to email a project or folder initiates an export process that produces a .DVA file. The .DVA file includes everything needed to use the project or folder (such as associated data sets, the connection string and credentials, and stored data)

1. On the Home page, click **Navigator**, and then click **Catalog**.
2. On the Catalog page select the project or folder that you want to share and click **Actions menu**, then select **Export** to open the Export dialog.
3. Click **Email** to open the Email dialog.
4. Enable the **Include Data** option if you're sharing a project or folder that uses an Excel data source and you want to include the data with the export.
5. Enable the **Connection Credentials** option if retrieving the data requires connection credentials. Then enter and confirm the password.

   If your project or folder includes data from Oracle Applications or a database and you've selected the **Include Data** option, then enter a password to send to the database for authentication when the user opens the application to access the data. Disable the **Include Data** option if you want users to enter the password when they open the application to access the data.
6. Click **Email**.

   Your email client opens a new partially composed email with the .DVA file attached.

# Email a File of a Visualization, Canvas, or Story

You can email visualizations, canvases, or stories in formats like Powerpoint (PPTX), Acrobat (PDF), Image (PNG), CSV (data only), or Package (the whole project including connection credentials).

1. Open the project, and on the Visualize or Narrate canvas, click the **Export** icon on the project toolbar, then click **Email**.

2. Use the **Format** option to select the output format you want, and specify output options:
   - For **Powerpoint (pptx)**, **Acrobat (pdf)**, and **Image (png)** - Specify the file name, paper size, and orientation.
     When you email any of these visual formats, the visualization or pages are re-rendered based on the size and orientation you select. So if you're emailing a table, then your output file might not contain all of the table's rows and columns included in your visualization, canvas, or story.
   - For **Data (csv)** - Specify the output file name. This option only includes the data used in the project. The outputted file uses the data delimiter for your computer's locale. For example, if your locale is set to Brazil, then the delimiter for numeric decimals is a comma instead of a period, which is used when your locale is set to United States.
   - For **Package (dva)** - Specify whether to include project data, connection credentials,). To enable users to open the project DVA file without having to enter a password, click **Include Connection Credentials** and specify the password.
3. Click **Email**.

   Your email client opens a new partially composed email with the .DVA file attached.

# Print a Visualization, Canvas, or Story

You can print your project's visualizations, canvases, or stories.

1. Open the project, and on the Visualize or Narrate canvas, click the **Export** icon on the project toolbar.
2. Set the **Include**, **Size**, and **Orientation** fields as needed.
3. Click **Print**.

   When you print, the visualization or pages are re-rendered based on the size and orientation you selected. So if you're printing a table, then your printed copy might not contain all of the table's rows and columns included in your visualization, canvas, or story.

# A
# Frequently Asked Questions

This reference provides answers to frequently asked questions for Oracle Analytics Desktop.

**Topics:**



## Oracle Analytics Desktop Installation FAQs

This topic answers common installation questions.

**How do I install Machine Learning and Advanced Analytics?**

Machine learning and advanced analytics are optional components that aren't included in the Oracle Analytics Desktop installation. You must install machine learning to use Diagnostics Analytics (Explain), Machine Learning Studio, or advanced analytics.

Follow these steps to install Machine Learning Framework on Windows.

1. Go to the Windows Start menu, browse to Oracle, and click **Install DVML**.
2. The installer starts on completion of the download. Follow the displayed instructions to install machine learning to the selected install path.
3. Click **Finish** to close the installer.
4. When prompted, press any key to close the terminal window.
5. If Oracle Analytics Desktop was running during the installation, then restart it.

Follow these steps to install Machine Learning Framework on Apple Mac.

1. Double-click the application **Oracle Analytics Configure Python** in Finder under Applications or in Launchpad.
2. The installer starts on completion of the download. Follow the displayed instructions to install machine learning to the selected install path. Enter an administrator user name and password to run the installation.
3. Click **Close** after the installation is completed.
   The Machine Learning Framework was installed in `/Library/Frameworks/DVMLruntime.framework`
4. If Oracle Analytics Desktop was running during the installation, then restart it.

**Why can’t I install Oracle Analytics Desktop?**

To perform the installation, you must have administrator privileges. If you try to install without administrator privileges, the following error message is displayed: `Error in creating registry key. Permission denied`.

To check to see if you've the required administrator privileges, go to Windows Control Panel and check your user accounts. If you don't have administrator privileges, then see your administrator to help you set up the required privileges.

**Why can't I successfully upgrade?**

If you have issues upgrading, then delete the previous version and try to the installation again.

**How will I know when to upgrade?**

You'll see a message when a newer version is available. The message will guide you to Oracle Technology Network where you can download the latest installer. See Oracle Analytics Desktop Installation Download.

# Oracle Analytics Desktop Project and Data Source FAQs

This topic answers common questions about projects and data sources.

**What data sources are supported?**

You can use data only from specific types and versions of sources. See Supported Data Sources.

**What if I'm using an unsupported version of Teradata?**

If you're using an unsupported version of Teradata, then you must update the extdriver.paths configuration file before you can successfully build a connection to Teradata. This configuration file is located here: `C:\<your directory>\AppData\Local\OracleAnalyticsDesktop\extdrvier.paths`. For example, `C:\Users\jsmith\AppData\Local\OracleAnalyticsDesktop\extdriver.paths`.

When updating the extdriver.paths configuration file, remove the default Teradata version number and replace it with the Teradata version number that you're using. Make sure that you include `\bin` in the path. For example if you're using Teradata 14.10, change `C:\Program Files\Teradata\Client\15.10\bin` to `C:\Program Files\Teradata\Client\14.10\bin`.

# Oracle Analytics Desktop Printing and Exporting FAQs

This topic answers common questions about printing and exporting.

**Why don't I see images in projects or from background maps when I print pages or when I export images in formats such as PDF, PPT, and PNG?**

You or a visualization builder might have added an image to a project or background map by referencing that image with a URL. For the image to print or be exported in various formats, the external website hosting that image must have the Access-Control-Allow-Origin header from the host server, to ensure proper security. If a map background includes an image reference that's taken from an external website that doesn't have this header, you won't see the image.

For more information about this header, see `https://www.w3.org/wiki/CORS_Enabled`.

# B

# Troubleshoot Visualization Issues

This topic describes common problems that you might encounter when working with visualizations and explains how to solve them.

**When I import a project, I get an error stating that the project, data source, or connection already exists**

When you're trying to import a project, you might receive the following error message:

"There is already a project, data source or connection with the same name as something you're trying to import. Do you want to continue the import and replace the existing content?"

This error message is displayed because one or more of the components exported with the project is already on your system. When a project is exported, the outputted .DVA file includes the project's associated data sources and connection string. To resolve this error, you can either click **OK** to replace the components on your system, or you can click **Cancel** and go into your system and manually delete the components.

This error message is also displayed when the project you're trying to import contains no data. When you export a project without data, the project's and data sources' metadata are included in the .DVA. To resolve this issue, you can click **OK** to replace the components on your system, or you can click **Cancel** and go into your system and manually delete the data source or connection that's causing the error.

**When I try to build a connection to Teradata, I get an error and the connection is not saved**

When you're trying to create a connection to Teradata, you might receive the following error message:

"Failed to save the connection. Cannot create a connection since there are some errors. Please fix them and try again."

This error message is displayed because the version of Teradata that you're using is different from the version supported by Oracle Analytics Desktop. To resolve this issue, update the extdriver.paths configuration file. This configuration file is located here: `C:\<your directory>\AppData\Local\OracleAnalyticsDesktop\extdrvier.paths`. For example, `C:\Users\jsmith\AppData\Local\OracleAnalyticsDesktop\extdriver.paths`.

To update the extdriver.paths configuration file, remove the default Teradata version number and replace it with the Teradata version number that you're using. Make sure that you include \bin in the path. For example if you're using Teradata 14.10, then change `C:\Program Files\Teradata\Client\15.10\bin` to `C:\Program Files\Teradata\Client\14.10\bin`. See What if I'm using a Teradata version different that the one supported by Oracle Analytics Desktop?

### I have issues when I try to refresh data for file-based data sources

Keep in mind the following requirements when you refresh data for Microsoft Excel, CSV, or TXT data sources:

- To refresh an Excel file, ensure that the newer spreadsheet file contains a sheet with the same name as the original file you uploaded. If a sheet is missing, then you must fix the file to match the sheets in the original uploaded file.
- If the Excel, CSV, or TXT file that you reload is missing some columns, then you'll get an error stating that your data reload has failed. If this happens, then you must fix the file to match the columns in the original uploaded file.
- If the Excel, CSV, or TXT file you used to create the data source was moved or deleted, then the connection path is crossed out in the Data Source dialog. You can reconnect the data source to its original source file, or connect it to a replacement file, by right-clicking the data source in the Display pane and in the Options menu select **Reload Data**. You can then browse for and select the file to load.
- If you reloaded an Excel, CSV, or TXT file with new columns, then the new columns are marked as hidden and don't display in the Data Panel for existing projects using the data set. To unhide these columns, click the **Hidden** option.

Your Excel spreadsheets must have a specific structure. See About Adding Spreadsheets or Other Data Files.

### I can't refresh data from a MongoDB data source

The first time you connect to MongoDB, the MongoDB driver creates a cache file. If the MongoDB schema was renamed and you try to reload a MongoDB data source or use the data source in a project, then you might get an error or Oracle Analytics doesn't respond.

To correct this error, you need to clear the MongoDB cache. To clear the cache, delete the contents of the following directory: `C:\<your directory>\AppData\Local\Progress\DataDirect\MongoDB_Schema`. For example, `C:\Users\jsmith\AppData\Local\Progress\DataDirect\MongoDB_Schema`

### Oracle Support needs a file to help me diagnose a technical issue

If you're working with the Oracle Support team to resolve a specific issue, they may ask you to generate a diagnostic dump file. This file contains the following information:

- Version information
- Installer logs
- Application component logs containing status information for Oracle Business Intelligence Presentation Server, Oracle Business Intelligence Server, and other critical components
- Jetty logs
- Data Security Standard (DSS) logs
- Webcat metadata plugin logs
- Derby logs
- Server Administration Workbench (SAW) Server logs

- Oracle Business Intelligence Presentation Server logs
- Oracle Business Intelligence Server logs

1. Open the command prompt and change the directory to the Oracle Analytics Desktop installation directory (for example, C:\Program Files\Oracle Analytics Desktop).
2. Type `diagnostic_dump.cmd` and then provide a name for the .zip output file (for example, output.zip).
3. Press Enter to execute the command.
   You can find the diagnostic output .zip file in your installation directory.

**I need to find more information about a specific issue**

The community forum is another great resource that you can use to find out more information about the problem you're having.

You can find the forum here: Oracle Community Forum.

C

# Accessibility Features and Tips

This topic describes accessibility features and information for Oracle Analytics Desktop.

**Topics:**



## Start Oracle Analytics Desktop with Accessibility Features Enabled

You can enable features that improve navigation and make the interface accessible.

To enable the accessibility features, you must start Oracle Analytics Desktop from the command line. Open a command window and enter the following:

On Windows:

```
dvdesktop.exe - sdk
```

On Mac:

```
open /Applications/dvdesktop.app --args -sdk
```

When you run the command, Oracle Analytics Desktop opens in a web browser.

## Keyboard Shortcuts for Visualizations

You can use keyboard shortcuts to navigate and to perform actions in visualizations.

Use these keyboard shortcuts for working with a project in the Visualize Canvas.

| Task | Keyboard Shortcut |
|---|---|
| Add data columns to a project. | Shift + F10 |
| Add insights to a project. | Ctrl + I |
| Copy the selected items to the clipboard. | Ctrl + C |
| Export content to PDF. | Ctrl + Shift + E (Win)<br>Command + Shift + E (Mac) |
| Export content to PPT. | Ctrl + Shift + X (Win)<br>Command + Shift + X (Mac) |

| Task | Keyboard Shortcut |
|---|---|
| Print content. | Ctrl + Shift + P (Win)<br>Command + Shift + P (Mac) |
| Reverse the last undo. | Ctrl + Y |
| Save a newly created project with a specific name. | Ctrl + Shift + S |
| Save a project with the changes. | Ctrl + S |
| Undo the last change. | Ctrl + Z |

Use these keyboard shortcuts while working on a visualization in the Visualize canvas.

| Task | Keyboard Shortcut |
|---|---|
| Copy a visualization to paste it to another canvas in the project or to a canvas in another project. | Ctrl + C |
| Paste the visualization into a canvas in the project or into a canvas in another project. | Ctrl + V |
| Duplicate a visualization. | Ctrl + D |
| Delete a visualization. | Delete key |

Use these keyboard shortcuts while working with a filter in the filter panel on the filter bar.

| Task | Keyboard Shortcut |
|---|---|
| Search items in a filter. | Enter key |
| Add the search string to the selection list. | Ctrl + Enter |

Use these keyboard shortcuts when you want to open, create, or edit artifacts such as data sets, projects, data flows, and sequences in a new tab or window.

| Task | Keyboard Shortcut |
|---|---|
| Open an artifact in a new browser tab. | Ctrl + Click the artifact |
| Open an artifact in a new browser window. | Shift + Click the artifact |

# Keyboard Shortcuts for Data Flow

Use these keyboard shortcuts to perform actions in the data flow editor.

| Task | Keyboard Shortcut |
|---|---|
| Undo the last change. | Ctrl + Z / Command + Z |
| Reverse the last undo. | Ctrl + Y / Command + Y |

D

# Data Sources and Data Types Reference

Find out about supported data sources, databases, and data types.

**Topics**



## Supported Data Sources

With Oracle Analytics Desktop, you can connect to many different data sources. Data sources are sorted alphabetically by Oracle databases first, then other databases.

| Data Source | Version | Oracle Analytics Desktop for Windows | Oracle Analytics Desktop for Mac | More Information |
|---|---|---|---|---|
| Oracle Applications | 11.1.1.9+ or Fusion Applications Release 8 and later | Yes | Yes | Connector supports several Oracle SaaS Applications. See Oracle Applications Connector.<br>See also Create an Oracle Applications Connection. |
| Oracle Autonomous Data Warehouse | - | Yes | Yes | Connection to public IP address only.<br>You can connect to multiple Oracle Autonomous Data Warehouse data sources. Upload a wallet for each connection.<br>Supports saving output from data flows.<br>See Connect to Oracle Autonomous Data Warehouse . |

| Data Source | Version | Oracle Analytics Desktop for Windows | Oracle Analytics Desktop for Mac | More Information |
|---|---|---|---|---|
| Oracle Database | 11.2.0.4+<br>12.1+<br>12.2+<br>18+<br>19+ | Yes | Yes | Use the Oracle Database connection type to connect to Oracle Database Classic Cloud Service.<br>You can connect to multiple database services. Upload a wallet for each connection.<br>Supports saving output from data flows.<br>Ensure that the appropriate security access rules are in place to allow a network connection to the database service on the database listening port.<br>See Create a Database Connection. |
| Oracle Analytics Cloud – Essbase | Oracle Essbase 11.1.2.4.0 +<br>Oracle Analytics Cloud – Essbase | Yes | Yes | See Create a Connection to Oracle Essbase. |
| Oracle Netsuite | Netsuite Release 2019.2 (JDBC Driver 8.10.85.0) | Yes | Yes | See Connecting to NetSuite . |
| Oracle Service Cloud | 1.2 | Yes | No | - |
| Oracle Talent Acquisition | - | Yes | Yes | - |
| Actian Ingres | 5.0+ | Yes | No | - |
| Actian Matrix | 5.0+ | Yes | No | - |
| Actian Vector | 5.0+ | Yes | No | - |
| Amazon Aurora | - | Yes | No | - |

| Data Source | Version | Oracle Analytics Desktop for Windows | Oracle Analytics Desktop for Mac | More Information |
|---|---|---|---|---|
| Amazon EMR | Amazon EMR 4.7.2 running Amazon Hadoop 2.7.2 and Hive 1.0.0<br>Amazon EMR (MapR) - Amazon Machine Image (AMI) 3.3.2 running MapR Hadoop M3 and Hive 0.13.1 | Yes | No | Complex data types not supported. |
| Amazon Redshift | 1.0.1036 + | Yes | No | - |
| Apache Drill | 1.7+ | Yes | No | - |
| Apache Hive | 2.3.0+<br>3.0+ | Yes | No | Supports Kerberos.<br>Supports saving output from data flows. |
| Cassandra | 3.10 | Yes | No | - |
| CSV File | - | Yes | Yes | - |
| DB2 | 10.1+<br>10.5+ | Yes | No | - |
| DropBox | - | Yes | No | - |
| Elastic Search | 5.6.4+ | Yes | - | - |
| Google Analytics | - | Yes | No | - |
| Google Cloud | - | Yes | No | - |
| Google Drive | - | Yes | No | - |
| GreenPlum | 4.3.8+ | Yes | No | - |
| HortonWorks Hive | 1.2+ | Yes | No | Supports Kerberos.<br>Supports saving output from data flows. |
| HP Vertica | 7+ | Yes | No | - |
| IBM BigInsights Hive | 1.2+ | Yes | No | Supports Kerberos. |
| Impala | 2.7+ | Yes | No | |
| Informix | 12.1+ | Yes | No | - |
| JDBC | Generic JDBC driver support | Yes | Yes | See Create Generic JDBC Connections. |

| Data Source | Version | Oracle Analytics Desktop for Windows | Oracle Analytics Desktop for Mac | More Information |
|---|---|---|---|---|
| MapR Hive | 1.2+ | - | - | Supports Kerberos.<br>Supports saving output from data flows. |
| Microsoft Access | 2013<br>2016 | Yes | No | - |
| Microsoft Azure SQL Database | 5.6.0 | Yes | TBD | - |
| Microsoft Excel | - | Yes | Yes | Only XLSX files (and XLS with unpivoted data). |
| MonetDB | 5+ | Yes | No | - |
| MongoDB | 3.2.5 | Yes | No | - |
| MySQL | 5.6+<br>5.7+ | Yes | No | Connections to MySQL Community Edition aren't supported. |
| Netezza | 7 | Yes | No | - |
| OData | 4.0+ | Yes | No | - |
| ODBC | Generic ODBC driver support | - | - | - |
| Pivotal HD Hive | - | Yes | No | Supports Kerberos. |
| PostgreSQL | 9.0+ | Yes | No | - |
| Presto | - | Yes | No | - |
| Salesforce | - | Yes | No | - |
| Spark | 1.6+ | Yes | No | Supports saving output from data flows. |
| SQL Server | 2014<br>2016 | Yes | No | - |
| Sybase ASE | 15.7+ | Yes | No | - |
| Sybase IQ | 16+ | Yes | No | - |
| Teradata | 14<br>15<br>16<br>16.10 | Yes | No | - |
| Teradata Aster | 6.10+ | Yes | No | - |

# Oracle Applications Connector

Oracle Applications Connector supports Oracle Applications Cloud. You can also use Oracle Applications Connector to connect to your on-premises Oracle BI Enterprise Edition deployments (if patched to an appropriate level) and another Oracle Analytics service.

Oracle Applications Cloud applications you can connect to:

- Oracle Sales Cloud
- Oracle Financials Cloud
- Oracle Human Capital Management Cloud
- Oracle Supply Chain Cloud
- Oracle Procurement Cloud
- Oracle Project Cloud
- Oracle Loyalty Cloud

# Supported Data Types

Read about the data types that Oracle Analytics supports.

**Topics:**



## Supported Base Data Types

When reading from a data source, Oracle Analytics attempts to map incoming data types to the supported data types.

For example, a database column that contains only date values is formatted as a DATE, a spreadsheet column that contains a mix of numerical and string values is formatted as a VARCHAR, and a data column that contains numerical data with fractional values uses DOUBLE or FLOAT.

In some cases Oracle Analytics can’t convert a source data type. To work around this data type issue, you can manually convert a data column to a supported type by entering SQL commands. In other cases, Oracle Analytics can't represent binary and complex data types such as BLOB, JSON, and XML.

Note that some data types aren't supported. You'll see an error message if the data source contains unsupported data types.

Oracle Analytics supports the following base data types:

- **Number Types** — SMALLINT, SMALLUNIT, TINYINT, TINYUINT, UINT, BIT, FLOAT, INT, NUMERIC, DOUBLE
- **Date Types** — DATE, DATETIME, TIMESTAMP, TIME
- **String Types** — LONGVARCHAR, CHAR, VARCHAR

# Supported Data Types by Database

Oracle Analytics supports the following data types.

| Database Type | Supported Data Types |
|---|---|
| Oracle | BINARY DOUBLE, BINARY FLOAT<br>CHAR, NCHAR<br>CLOB, NCLOB<br>DATE<br>FLOAT<br>NUMBER, NUMBER (p,s),<br>NVARCHAR2, VARCHAR2<br>ROWID<br>TIMESTAMP, TIMESTAMP WITH LOCAL TIMEZONE, TIMESTAMP WITH TIMEZONE |
| DB2 | BIGINT<br>CHAR, CLOB<br>DATE, DECFLOAT, DECIMAL, DOUBLE<br>FLOAT<br>INTEGER<br>LONGVAR<br>NUMERIC<br>REAL<br>SMALLINT<br>TIME, TIMESTAMP<br>VARCHAR |
| SQL Server | BIGINT, BIT<br>CHAR<br>DATE, DATETIME, DATETIME2, DATETIMEOFFSET, DECIMAL<br>FLOAT<br>INT<br>MONEY<br>NCHAR, NTEXT, NUMERIC, NVARCHAR, NVARCHAR(MAX)<br>REAL<br>SMALLDATETIME, SMALLINT, SMALLMONEY<br>TEXT, TIME, TINYINT<br>VARCHAR, VARCHAR(MAX)<br>XML |

| Database Type | Supported Data Types |
|---|---|
| MySQL | BIGINT, BIGINT UNSIGNED<br>CHAR<br>DATE, DATETIME, DECIMAL, DECIMAL UNSIGNED, DOUBLE, DOUBLE UNSIGNED<br>FLOAT, FLOAT UNSIGNED<br>INTEGER, INTEGER UNSIGNED<br>LONGTEXT<br>MEDIUMINT, MEDIUMINT UNSIGNED, MEDIUMTEXT<br>SMALLINT, SMALLINT UNSIGNED<br>TEXT, TIME, TIMESTAMP, TINYINT, TINYINT UNSIGNED, TINYTEXT<br>VARCHAR<br>YEAR |
| Apache Spark | BIGINT, BOOLEAN<br>DATE, DECIMAL, DOUBLE<br>FLOAT<br>INT<br>SMALLINT, STRING<br>TIMESTAMP, TINYINT<br>VARCHAR |
| Teradata | BIGINT, BYTE, BYTEINT<br>CHAR, CLOB<br>DATE, DECIMAL, DOUBLE<br>FLOAT<br>INTEGER<br>NUMERIC<br>REAL<br>SMALLINT<br>TIME, TIMESTAMP<br>VARCHAR |

E

# Data Preparation Reference

This topic describes the set and types of recommendation and options you can use to perform data transform changes to a data set.

**Topics:**



## Transform Recommendation Reference

Find out about the data transform options in the project's Prepare canvas.

| Option | Description |
|---|---|
| **Edit** | Edits the column. For example, you can change the name, select another column, or update functions. |
| **Hide** | Hides the column in the Data Panel and in the visualizations. If you want to see the hidden columns, click **Hidden columns** (ghost icon) on the page footer. You can then unhide individual columns or unhide all the hidden columns at the same time. |
| **Group, Conditional Group** | Select **Group** to create your own custom groups. For example, you can group States together with custom regions, and you can categorize dollar amounts into groups indicating small, medium, and large. |
| **Split** | Splits a specific column value into parts. For example, you can split a column called, Name, into first and last name. |
| **Uppercase** | Updates the contents of a column with the values in all uppercase letters. |
| **Lowercase** | Updates the contents of a column with the values all in lowercase letters. |
| **Sentence Case** | Updates the contents of a column to make the first letter of the first word of a sentence uppercase. |
| **Rename** | Allows you to change the name of any column. |
| **Duplicate** | Creates a column with identical content of the selected column. |
| **Convert to Text** | Changes the data type of a column to text. |
| **Replace** | Changes specific text in the selected column to any value that you specify. For example, you can change all instances of *Mister* to *Mr.* in the column. |
| **Create** | Creates a column based on a function. |
| **Convert to Number** | Changes the data type of the column to number, which deletes any values that aren't numbers from the column. |
| **Convert to Date** | Changes the data type of the column to date and deletes any values that aren’t dates from the column. |
| **Bin** | Creates your own custom groups for number ranges. For example, you can create bins for an Age column with age ranges binned into Pre-Teen, Young Adult, Adult, or Senior based on custom requirements. |
| **Log** | Calculates the natural logarithm of an expression. |
| **Power** | Raises the values of a column to the power that you specify. The default power is 2. |

| Option | Description |
|---|---|
| **Square Root** | Creates a column populated with the square root of the value in the column selected. |

# Column Menu Options for Quick Data Transformations

You can use the following column menu options to transform data in a data flow.

| Options | Description |
|---|---|
| **Rename** | Change the column name. |
| **Duplicate** | Create a column with data that's identical to the selected column. |
| **Delete** | Select and remove a column from the data set. |
| **Convert to Number** | Change the data type of the column to number and delete any values that aren't numbers. |
| **Convert to Text** | Change the data type of a column to text. |
| **Uppercase** | Convert all the text in the column to uppercase. |
| **Lowercase** | Convert all the text in the column to lowercase. |
| **Sentence Case** | Convert the first letter of the first word to uppercase on each row in a column. |
| **Group** | Create a custom group to combine related values. For example, you can group states with custom regions and categorize dollar amounts into groups showing small, medium, and large. |
| **Merge Columns** | Combine two or more columns to display as one. |
| **Transform** | Modify the column data by using an expression. |
| **Bin** | Create your custom groups for number ranges. |

# Developer Reference

The Visualize canvas includes Developer options that enable you to embed analytics content in other applications. To display Developer options, open a project, display the Visualize pane, and click **Options**, then **Developer**.

**Table E-1 Developer tabs**

| Tab | Description |
|---|---|
| Embed | Embed analytics content in other applications. |
| JSON | Lookup a column reference for an embed that uses filters. |

**Table E-1 (Cont.) Developer tabs**

| Tab | Description |
| --- | --- |
| XML, Data Sets, Prep Data Sets, Prep UIOptions | Don't use these tabs. There're for internal diagnostics only. |

F

# Expression Editor Reference

This section describes the expression elements that you can use in the Expression Editor.

Topics:



## SQL Operators

SQL operators are used to specify comparisons between expressions.

You can use various types of SQL operators.

| Operator | Example | Description | Syntax |
|---|---|---|---|
| BETWEEN | `"COSTS"."UNIT_COST" BETWEEN 100.0 AND 5000.0` | Determines if a value is between two non-inclusive bounds.<br>`BETWEEN` can be preceded with `NOT` to negate the condition. | `BETWEEN [LowerBound] AND [UpperBound]` |
| IN | `"COSTS"."UNIT_COST" IN(200, 600, 'A')` | Determines if a value is present in a set of values. | `IN ([Comma Separated List])` |
| IS NULL | `"PRODUCTS"."PROD_NAME" IS NULL` | Determines if a value is null. | `IS NULL` |
| LIKE | `"PRODUCTS"."PROD_NAME" LIKE 'prod%'` | Determines if a value matches all or part of a string. Often used with wildcard characters to indicate any character string match of zero or more characters (%) or any single character match (_). | `LIKE` |
| + | `(FEDERAL_REVENUE + LOCAL_REVENUE) - TOTAL_EXPENDITURE` | Plus sign for addition. | `+` |

| Operator | Example | Description | Syntax |
|---|---|---|---|
| - | `(FEDERAL_REVENUE + LOCAL_REVENUE ) - TOTAL_EXPENDITURE` | Minus sign for subtraction. | - |
| * *or* X | `SUPPORT_SERVICES_EXPENDITURE * 1.5` | Multiply sign for multiplication. | *<br>X |
| / | `CAPITAL_OUTLAY_EXPENDITURE /1.05` | Divide by sign for division. | / |
| % | | Percentage | % |
| \|\| | `STATE\|\| CAST(YEAR AS CHAR(4))` | Character string concatenation. | \|\| |
| ( | `(FEDERAL_REVENUE + LOCAL_REVENUE ) - TOTAL_EXPENDITURE` | Open parenthesis. | ( |
| ) | `(FEDERAL_REVENUE + LOCAL_REVENUE ) - TOTAL_EXPENDITURE` | Close parenthesis. | ) |
| > | `YEAR > 2000 and YEAR < 2016 and YEAR <> 2013` | Greater than sign, indicating values higher than the comparison. | > |
| < | `YEAR > 2000 and YEAR < 2016 and YEAR <> 2013` | Less than sign, indicating values lower than the comparison. | < |
| = | | Equal sign, indicating the same value. | = |
| >= | | Greater than or equal to sign, indicating values the same or higher than the comparison. | >= |
| <= | | Less than or equal to sign, indicating values the same or lower than the comparison. | <= |
| <> | `YEAR > 2000 and YEAR < 2016 and YEAR <> 2013` | Not equal to, indicating values higher or lower, but different. | <> |

| Operator | Example | Description | Syntax |
|---|---|---|---|
| , | `STATE in ('ALABAMA','CALIFORNIA')` | Comma, used to separate elements in a list. | , |

# Conditional Expressions

You use conditional expressions to create expressions that convert values.

The conditional expressions described in this section are building blocks for creating expressions that convert a value from one form to another.

Follow these rules:

- In `CASE` statements, `AND` has precedence over `OR`
- Strings must be in single quotes

| Expression | Example | Description | Syntax |
|---|---|---|---|
| CASE (If) | `CASE`<br>`WHEN score-par < 0 THEN 'Under Par'`<br>`WHEN score-par = 0 THEN 'Par'`<br>`WHEN score-par = 1 THEN 'Bogey'`<br>`WHEN score-par = 2 THEN 'Double Bogey'`<br>`ELSE 'Triple Bogey or Worse'`<br>`END` | Evaluates each `WHEN` condition and if satisfied, assigns the value in the corresponding `THEN` expression.<br>If none of the `WHEN` conditions are satisfied, it assigns the default value specified in the `ELSE` expression. If no `ELSE` expression is specified, the system automatically adds an `ELSE NULL`. | `CASE WHEN request_condition1 THEN expr1 ELSE expr2 END` |
| CASE (Switch) | `CASE Score-par`<br>`WHEN -5 THEN 'Birdie on Par 6'`<br>`WHEN -4 THEN 'Must be Tiger'`<br>`WHEN -3 THEN 'Three under par'`<br>`WHEN -2 THEN 'Two under par'`<br>`WHEN -1 THEN 'Birdie'`<br>`WHEN 0 THEN 'Par'`<br>`WHEN 1 THEN 'Bogey'`<br>`WHEN 2 THEN 'Double Bogey'`<br>`ELSE 'Triple Bogey or Worse'`<br>`END` | Also referred to as `CASE (Lookup)`. The value of the first expression is examined, then the `WHEN` expressions. If the first expression matches any `WHEN` expression, it assigns the value in the corresponding `THEN` expression.<br>If none of the `WHEN` expressions match, it assigns the default value specified in the `ELSE` expression. If no `ELSE` expression is specified, the system automatically adds an `ELSE NULL`.<br>If the first expression matches an expression in multiple `WHEN` clauses, only the expression following the first match is assigned. | `CASE expr1 WHEN expr2 THEN expr3 ELSE expr4 END` |
| IfCase > ELSE | | | `ELSE [expr]` |

| Expression | Example | Description | Syntax |
|---|---|---|---|
| IfCase > IFNULL | | | `IFNULL([expr], [value])` |
| IfCase > NULLIF | | | `NULLIF([expr], [expr])` |
| IfCase > WHEN | | | `WHEN [Condition] THEN [expr]` |
| IfCase > CASE | | | `CASE WHEN [Condition] THEN [expr] END` |
| SwitchCase > ELSE | | | `ELSE [expr]` |
| SwitchCase >IFNULL | | | `IFNULL([expr], [value])` |
| SwitchCase > NULLIF | | | `NULLIF([expr], [expr])` |
| SwitchCase > WHEN | | | `WHEN [Condition] THEN [expr]` |

# Functions

There are various types of functions that you can use in expressions.

**Topics:**



## Aggregate Functions

Aggregate functions perform operations on multiple values to create summary results.

The following list describes the aggregation rules that are available for columns and measure columns. The list also includes functions that you can use when creating calculated items for analyses.

- **Default** — Applies the default aggregation rule as in the data model or by the original author of the analysis. Not available for calculated items in analyses.
- **Server Determined** — Applies the aggregation rule that's determined by the Oracle Analytics (such as the rule that is defined in the data model). The aggregation is performed within Oracle Analytics for simple rules such as Sum, Min, and Max. Not available for measure columns in the Layout pane or for calculated items in analyses.
- **Sum** — Calculates the sum obtained by adding up all values in the result set. Use this for items that have numeric values.
- **Min** — Calculates the minimum value (lowest numeric value) of the rows in the result set. Use this for items that have numeric values.
- **Max** — Calculates the maximum value (highest numeric value) of the rows in the result set. Use this for items that have numeric values.
- **Average** — Calculates the average (mean) value of an item in the result set. Use this for items that have numeric values. Averages on tables and pivot tables are rounded to the nearest whole number.
- **First** — In the result set, selects the first occurrence of the item for measures. For calculated items, selects the first member according to the display in the Selected list. Not available in the Edit Column Formula dialog.
- **Last** — In the result set, selects the last occurrence of the item. For calculated items, selects the last member according to the display in the Selected list. Not available in the Edit Column Formula dialog.
- **Count** — Calculates the number of rows in the result set that have a non-null value for the item. The item is typically a column name, in which case the number of rows with non-null values for that column are returned.
- **Count Distinct** — Adds distinct processing to the Count function, which means that each distinct occurrence of the item is counted only once.
- **None** — Applies no aggregation. Not available for calculated items in analyses.
- **Report-Based Total (when applicable)** — If not selected, specifies that the Oracle Analytics should calculate the total based on the entire result set, before applying any filters to the measures. Not available in the Edit Column Formula dialog or for calculated items in analyses. Only available for attribute columns.

| Function | Example | Description | Syntax |
|---|---|---|---|
| AGGREGATE AT | `AGGREGATE(sales AT month, region)` | Aggregates columns based on the level or levels you specify.<br>*measure* is the name of a measure column. level is the level at which you want to aggregate. You can optionally specify more than one level. You can't specify a level from a dimension that contains levels that are being used as the measure level for the measure you specified in the first argument. For example, you can't write the function as AGGREGATE(yearly_sales AT month) because *month* is from the same time dimension that's being used as the measure level for *yearly_sales*. | `AGGREGATE(measure AT level [, level1, levelN])` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| AVG | `Avg(Sales)` | Calculates the average (mean) of a numeric set of values. | `AVG(expr)` |
| AVGDISTINCT | | Calculates the average (mean) of all distinct values of an expression. | `AVG(DISTINCT expr)` |
| BIN | `BIN(revenue BY productid, year WHERE productid > 2 INTO 4 BINS RETURNING RANGE_LOW)` | Classifies a given numeric expression into a specified number of equal width buckets. The function can return either the bin number or one of the two end points of the bin interval. numeric_expr is the measure or numeric attribute to bin. BY grain_expr1,..., grain_exprN is a list of expressions that define the grain at which the numeric_expr is calculated. BY is required for measure expressions and is optional for attribute expressions. WHERE a filter to apply to the numeric_expr before the numeric values are assigned to bins INTO number_of_bins BINS is the number of bins to return BETWEEN min_value AND max_value is the min and max values used for the end points of the outermost bins RETURNING NUMBER indicates that the return value should be the bin number (1, 2, 3, 4, etc.). This is the default. RETURNING RANGE_LOW indicates the lower value of the bin interval RETURNING RANGE_HIGH indicates the higher value of the bin interval | `BIN(numeric_expr [BY grain_expr1, ..., grain_exprN] [WHERE condition] INTO number_of_bins BINS [BETWEEN min_value AND max_value] [RETURNING {NUMBER \| RANGE_LOW \| RANGE_HIGH}])` |
| BottomN | | Ranks the lowest n values of the expression argument from 1 to n, 1 corresponding to the lowest numerical value.<br>*expr* is any expression that evaluates to a numerical value. integer is any positive integer. Represents the bottom number of rankings displayed in the result set, 1 being the lowest rank. | `BottomN(expr, integer)` |
| COUNT | `COUNT(Products)` | Determines the number of items with a non-null value. | `COUNT(expr)` |
| COUNTDISTINCT | | Adds distinct processing to the COUNT function.<br>*expr* is any expression. | `COUNT(DISTINCT expr)` |
| COUNT* | `SELECT COUNT(*) FROM Facts` | Counts the number of rows. | `COUNT(*)` |
| First | `First(Sales)` | Selects the first non-null returned value of the expression argument. The `First` function operates at the most detailed level specified in your explicitly defined dimension. | `First([NumericExpression)]` |
| Last | `Last(Sales)` | Selects the last non-null returned value of the expression. | `Last([NumericExpression)]` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| MAVG | | Calculates a moving average (mean) for the last n rows of data in the result set, inclusive of the current row.<br>*expr* is any expression that evaluates to a numerical value. integer is any positive integer. Represents the average of the last n rows of data. | `MAVG(expr, integer)` |
| MAX | `MAX(Revenue)` | Calculates the maximum value (highest numeric value) of the rows satisfying the numeric expression argument. | `MAX(expr)` |
| MEDIAN | `MEDIAN(Sales)` | Calculates the median (middle) value of the rows satisfying the numeric expression argument. When there are an even number of rows, the median is the mean of the two middle rows. This function always returns a double. | `MEDIAN(expr)` |
| MIN | `MIN(Revenue)` | Calculates the minimum value (lowest numeric value) of the rows satisfying the numeric expression argument. | `MIN(expr)` |
| NTILE | | Determines the rank of a value in terms of a user-specified range. It returns integers to represent any range of ranks. NTILE with numTiles=100 returns what is commonly called the "percentile" (with numbers ranging from 1 to 100, with 100 representing the high end of the sort).<br>*expr* is any expression that evaluates to a numerical value. numTiles is a positive, nonnull integer that represents the number of tiles. | `NTILE(expr, numTiles)` |
| PERCENTILE | | Calculates a percentile rank for each value satisfying the numeric expression argument. The percentile rank ranges are between 0 (0th percentile) to 1 (100th percentile).<br>*expr* is any expression that evaluates to a numerical value. | `PERCENTILE(expr)` |
| RANK | `RANK(chronological_key, null, year_key_columns)` | Calculates the rank for each value satisfying the numeric expression argument. The highest number is assigned a rank of 1, and each successive rank is assigned the next consecutive integer (2, 3, 4,...). If certain values are equal, they'reare assigned the same rank (for example, 1, 1, 1, 4, 5, 5, 7...).<br>*expr* is any expression that evaluates to a numerical value. | `RANK(expr)` |
| STDDEV | `STDDEV(Sales)`<br>`STDDEV(DISTINCT Sales)` | Returns the standard deviation for a set of values. The return type is always a double. | `STDDEV(expr)` |
| STDDEV_POP | `STDDEV_POP(Sales)`<br>`STDDEV_POP(DISTINCT Sales)` | Returns the standard deviation for a set of values using the computational formula for population variance and standard deviation. | `STDDEV_POP([NumericExpression])` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| SUM | `SUM(Revenue)` | Calculates the sum obtained by adding up all values satisfying the numeric expression argument. | `SUM(expr)` |
| SUMDISTINCT | | Calculates the sum obtained by adding all of the distinct values satisfying the numeric expression argument.<br>*expr* is any expression that evaluates to a numerical value. | `SUM(DISTINCT expr)` |
| TOPN | | Ranks the highest n values of the expression argument from 1 to n, 1 corresponding to the highest numerical value.<br>*expr* is any expression that evaluates to a numerical value. integer is any positive integer. Represents the top number of rankings displayed in the result set, 1 being the highest rank. | `TOPN(expr, integer)` |

# Analytics Functions

Analytics functions allow you to explore data using models such as trendline and cluster.

| Function | Example | Description | Syntax |
|---|---|---|---|
| TRENDLINE | `TRENDLINE(revenue, (calendar_year, calendar_quarter, calendar_month) BY (product), 'LINEAR', 'VALUE')` | Fits a linear or exponential model and returns the fitted values or model. The *numeric_expr* represents the Y value for the trend and the *series* (time columns) represent the X value. | `TRENDLINE(numeric_expr, ([series]) BY ([partitionBy]), model_type, result_type)` |
| CLUSTER | `CLUSTER((product, company), (billed_quantity, revenue), 'clusterName', 'algorithm=k-means;numClusters=%1;maxIter=%2;useRandomSeed=FALSE;enablePartitioning=TRUE', 5, 10)` | Collects a set of records into groups based on one or more input expressions using K-Means or Hierarchical Clustering. | `CLUSTER((dimension_expr1, ... dimension_exprN), (expr1, ... exprN), output_column_name, options, [runtime_binded_options])` |
| OUTLIER | `OUTLIER((product, company), (billed_quantity, revenue), 'isOutlier', 'algorithm=kmeans')` | Classifies a record as Outlier based on one or more input expressions using K-Means or Hierarchical Clustering or Multi-Variate Outlier detection Algorithms. | `OUTLIER((dimension_expr1, ... dimension_exprN), (expr1, ... exprN), output_column_name, options, [runtime_binded_options])` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| REGR | `REGR(revenue, (discount_amount), (product_type, brand), 'fitted', '')` | Fits a linear model and returns the fitted values or model. This function can be used to fit a linear curve on two measures. | `REGR(y_axis_measure_expr, (x_axis_expr), (category_expr1, ..., category_exprN), output_column_name, options, [runtime_binded_options])` |
| EVALUATE_SCRIPT | `EVALUATE_SCRIPT('filerepo://obiee.Outliers.xml', 'isOutlier', 'algorithm=kmeans;id=%1;arg1=%2;arg2=%3;useRandomSeed=False;', customer_number, expected_revenue, customer_age)` | Executes a Python script as specified in the *script_file_path*, passing in one or more columns or literal expressions as input. The output of the function is determined by the *output_column_name*. | `EVALUATE_SCRIPT(script_file_path, output_column_name, options, [runtime_binded_options])` |

## Calendar Functions

Calendar functions manipulate data of the data types `DATE` and `DATETIME` based on a calendar year.

| Function | Example | Description | Syntax |
|---|---|---|---|
| CURRENT_Date | `CURRENT_DATE` | Returns the current date.<br>The date is determined by the system in which the Oracle BI is running. | `CURRENT_DATE` |
| CURRENT_TIME | `CURRENT_TIME(3)` | Returns the current time to the specified number of digits of precision, for example: HH:MM:SS.SSS<br>If no argument is specified, the function returns the default precision. | `CURRENT_TIME(expr)` |
| CURRENT_TIMESTAMP | `CURRENT_TIMESTAMP(3)` | Returns the current date/timestamp to the specified number of digits of precision. | `CURRENT_TIMESTAMP(expr)` |
| DAYNAME | `DAYNAME(Order_Date)` | Returns the name of the day of the week for a specified date expression. | `DAYNAME(expr)` |
| DAYOFMONTH | `DAYOFMONTH(Order_Date)` | Returns the number corresponding to the day of the month for a specified date expression. | `DAYOFMONTH(expr)` |
| DAYOFWEEK | `DAYOFWEEK(Order_Date)` | Returns a number between 1 and 7 corresponding to the day of the week for a specified date expression. For example, 1 always corresponds to Sunday, 2 corresponds to Monday, and so on through to Saturday which returns 7. | `DAYOFWEEK(expr)` |
| DAYOFYEAR | `DAYOFYEAR(Order_Date)` | Returns the number (between 1 and 366) corresponding to the day of the year for a specified date expression. | `DAYOFYEAR(expr)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| DAY_OF_QUARTER | `DAY_OF_QUARTER(Order_Date)` | Returns a number (between 1 and 92) corresponding to the day of the quarter for the specified date expression. | `DAY_OF_QUARTER(expr)` |
| HOUR | `HOUR(Order_Time)` | Returns a number (between 0 and 23) corresponding to the hour for a specified time expression. For example, 0 corresponds to 12 a.m. and 23 corresponds to 11 p.m. | `HOUR(expr)` |
| MINUTE | `MINUTE(Order_Time)` | Returns a number (between 0 and 59) corresponding to the minute for a specified time expression. | `MINUTE(expr)` |
| MONTH | `MONTH(Order_Time)` | Returns the number (between 1 and 12) corresponding to the month for a specified date expression. | `MONTH(expr)` |
| MONTHNAME | `MONTHNAME(Order_Time)` | Returns the name of the month for a specified date expression. | `MONTHNAME(expr)` |
| MONTH_OF_QUARTER | `MONTH_OF_QUARTER(Order_Date)` | Returns the number (between 1 and 3) corresponding to the month in the quarter for a specified date expression. | `MONTH_OF_QUARTER(expr)` |
| NOW | `NOW()` | Returns the current timestamp. The `NOW` function is equivalent to the `CURRENT_TIMESTAMP` function. | `NOW()` |
| QUARTER_OF_YEAR | `QUARTER_OF_YEAR(Order_Date)` | Returns the number (between 1 and 4) corresponding to the quarter of the year for a specified date expression. | `QUARTER_OF_YEAR(expr)` |
| SECOND | `SECOND(Order_Time)` | Returns the number (between 0 and 59) corresponding to the seconds for a specified time expression. | `SECOND(expr)` |
| TIMESTAMPADD | `TIMESTAMPADD(SQL_TSI_MONTH, 12,Time."Order Date")` | Adds a specified number of intervals to a timestamp, and returns a single timestamp.<br>Interval options are: *SQL_TSI_SECOND*, *SQL_TSI_MINUTE*, *SQL_TSI_HOUR*, *SQL_TSI_DAY*, *SQL_TSI_WEEK*, *SQL_TSI_MONTH*, *SQL_TSI_QUARTER*, *SQL_TSI_YEAR* | `TIMESTAMPADD(interval, expr, timestamp)` |
| TIMESTAMPDIFF | `TIMESTAMPDIFF(SQL_TSI_MONTH, Time."Order Date",CURRENT_DATE)` | Returns the total number of specified intervals between two timestamps.<br>Use the same intervals as TIMESTAMPADD. | `TIMESTAMPDIFF(interval, expr, timestamp2)` |
| WEEK_OF_QUARTER | `WEEK_OF_QUARTER(Order_Date)` | Returns a number (between 1 and 13) corresponding to the week of the quarter for the specified date expression. | `WEEK_OF_QUARTER(expr)` |
| WEEK_OF_YEAR | `WEEK_OF_YEAR(Order_Date)` | Returns a number (between 1 and 53) corresponding to the week of the year for the specified date expression. | `WEEK_OF_YEAR(expr)` |
| YEAR | `YEAR(Order_Date)` | Returns the year for the specified date expression. | `YEAR(expr)` |

# Conversion Functions

Conversion functions convert a value from one form to another.

| Function | Example | Description | Syntax |
|---|---|---|---|
| CAST | `CAST(hiredate AS CHAR(40)) FROM employee` | Changes the data type of an expression or a null literal to another data type. For example, you can cast a *customer_name* (a data type of CHAR or VARCHAR) or *birthdate* (a datetime literal). | `CAST(expr AS type)` |
| IFNULL | `IFNULL(Sales, 0)` | Tests if an expression evaluates to a null value, and if it does, assigns the specified value to the expression. | `IFNULL(expr, value)` |
| INDEXCOL | `SELECT INDEXCOL(VALUEOF (NQ_SESSION.GEOGRAPHY _LEVEL), Country, State, City), Revenue FROM Sales` | Uses external information to return the appropriate column for the signed-in user to see. | `INDEXCOL([integer literal], [expr1] [, [expr2], ?-])` |
| NULLIF | `SELECT e.last_name, NULLIF(e.job_id, j.job_id) "Old Job ID" FROM employees e, job_history j WHERE e.employee_id = j.employee_id ORDER BY last_name, "Old Job ID";` | Compares two expressions. If they're equal, then the function returns NULL. If they're not equal, then the function returns the first expression. You can't specify the literal NULL for the first expression. | `NULLIF([expression], [expression])` |
| To_DateTime | `SELECT To_DateTime ('2009-03-0301:01:00' , 'yyyy-mm-dd hh:mi:ss') FROM sales` | Converts string literals of *DateTime* format to a *DateTime* data type. | `To_DateTime([expressi on], [literal])` |
| VALUEOF | `SalesSubjectArea.Cust omer.Region = VALUEOF("Region Security"."REGION")` | References the value of an Oracle BI repository variable in a filter.<br>Use *expr* variables as arguments of the VALUEOF function. Refer to static repository variables by name. | `VALUEOF(expr)` |

# Display Functions

Display functions operate on the result set of a query.

| Function | Example | Description | Syntax |
|---|---|---|---|
| BottomN | `BottomN(Sales, 10)` | Returns the *n* lowest values of expression, ranked from lowest to highest. | `BottomN([NumericExpre ssion], [integer])` |
| FILTER | `FILTER(Sales USING Product = 'widget')` | Computes the expression using the given preaggregate filter. | `FILTER(measure USING filter_expr)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| MAVG | `MAVG(Sales, 10)` | Calculates a moving average (mean) for the last *n* rows of data in the result set, inclusive of the current row. | `MAVG([NumericExpression], [integer])` |
| MSUM | `SELECT Month, Revenue, MSUM(Revenue, 3) as 3_MO_SUM FROM Sales` | Calculates a moving sum for the last *n* rows of data, inclusive of the current row.<br>The sum for the first row is equal to the numeric expression for the first row. The sum for the second row is calculated by taking the sum of the first two rows of data, and so on. When the *n* th row is reached, the sum is calculated based on the last *n* rows of data. | `MSUM([NumericExpression], [integer])` |
| NTILE | `NTILE(Sales, 100)` | Determines the rank of a value in terms of a user-specified range. It returns integers to represent any range of ranks. The example shows a range from 1 to 100, with the lowest sale = 1 and the highest sale = 100. | `NTILE([NumericExpression], [integer])` |
| PERCENTILE | `PERCENTILE(Sales)` | Calculates a percent rank for each value satisfying the numeric expression argument. The percentile rank ranges are from 0 (1st percentile) to 1 (100th percentile), inclusive. | `PERCENTILE([NumericExpression])` |
| RANK | `RANK(Sales)` | Calculates the rank for each value satisfying the numeric expression argument. The highest number is assigned a rank of 1, and each successive rank is assigned the next consecutive integer (2, 3, 4,...). If certain values are equal, they're assigned the same rank (for example, 1, 1, 1, 4, 5, 5, 7...). | `RANK([NumericExpression])` |
| RCOUNT | `SELECT month, profit, RCOUNT(profit) FROM sales WHERE profit > 200` | Takes a set of records as input and counts the number of records encountered so far. | `RCOUNT([NumericExpression])` |
| RMAX | `SELECT month, profit, RMAX(profit) FROM sales` | Takes a set of records as input and shows the maximum value based on records encountered so far. The specified data type must be one that can be ordered. | `RMAX([NumericExpression])` |
| RMIN | `SELECT month, profit, RMIN(profit) FROM sales` | Takes a set of records as input and shows the minimum value based on records encountered so far. The specified data type must be one that can be ordered. | `RMIN([NumericExpression])` |
| RSUM | `SELECT month, revenue, RSUM(revenue) as RUNNING_SUM FROM sales` | Calculates a running sum based on records encountered so far.<br>The sum for the first row is equal to the numeric expression for the first row. The sum for the second row is calculated by taking the sum of the first two rows of data, and so on. | `RSUM([NumericExpression])` |
| TOPN | `TOPN(Sales, 10)` | Returns the *n* highest values of expression, ranked from highest to lowest. | `TOPN([NumericExpression], [integer])` |

# Evaluate Functions

Evaluate functions are database functions that can be used to pass through expressions to get advanced calculations.

Embedded database functions can require one or more columns. These columns are referenced by %1 ... %N within the function. The actual columns must be listed after the function.

| Function | Example | Description | Syntax |
|---|---|---|---|
| EVALUATE | `SELECT EVALUATE('instr( %1, %2)', address, 'Foster City') FROM employees` | Passes the specified database function with optional referenced columns as parameters to the database for evaluation. | `EVALUATE([string expression], [comma separated expressions])` |
| EVALUATE_AGGR | `EVALUATE_AGGR('REGR_SLOPE(%1, %2)', sales.quantity, market.marketkey )` | Passes the specified database function with optional referenced columns as parameters to the database for evaluation. This function is intended for aggregate functions with a `GROUP BY` clause. | `EVALUATE_AGGR('db_agg _function(%1...%N)' [AS datatype] [, column1, columnN])` |

# Mathematical Functions

The mathematical functions described in this section perform mathematical operations.

| Function | Example | Description | Syntax |
|---|---|---|---|
| ABS | `ABS(Profit)` | Calculates the absolute value of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `ABS(expr)` |
| ACOS | `ACOS(1)` | Calculates the arc cosine of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `ACOS(expr)` |
| ASIN | `ASIN(1)` | Calculates the arc sine of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `ASIN(expr)` |
| ATAN | `ATAN(1)` | Calculates the arc tangent of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `ATAN(expr)` |
| ATAN2 | `ATAN2(1, 2)` | Calculates the arc tangent of *y* /*x*, where *y* is the first numeric expression and *x* is the second numeric expression. | `ATAN2(expr1, expr2)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| CEILING | `CEILING(Profit)` | Rounds a non-integer numeric expression to the next highest integer. If the numeric expression evaluates to an integer, the `CEILING` function returns that integer. | `CEILING(expr)` |
| COS | `COS(1)` | Calculates the cosine of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `COS(expr)` |
| COT | `COT(1)` | Calculates the cotangent of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `COT(expr)` |
| DEGREES | `DEGREES(1)` | Converts an expression from radians to degrees.<br>*expr* is any expression that evaluates to a numerical value. | `DEGREES(expr)` |
| EXP | `EXP(4)` | Sends the value to the power specified. Calculates *e* raised to the n-th power, where *e* is the base of the natural logarithm. | `EXP(expr)` |
| ExtractBit | `Int ExtractBit(1, 5)` | Retrieves a bit at a particular position in an integer. It returns an integer of either 0 or 1 corresponding to the position of the bit. | `ExtractBit([Source Number], [Digits])` |
| FLOOR | `FLOOR(Profit)` | Rounds a non-integer numeric expression to the next lowest integer. If the numeric expression evaluates to an integer, the `FLOOR` function returns that integer. | `FLOOR(expr)` |
| LOG | `LOG(1)` | Calculates the natural logarithm of an expression.<br>*expr* is any expression that evaluates to a numerical value. | `LOG(expr)` |
| LOG10 | `LOG10(1)` | Calculates the base 10 logarithm of an expression.<br>*expr* is any expression that evaluates to a numerical value. | `LOG10(expr)` |
| MOD | `MOD(10, 3)` | Divides the first numeric expression by the second numeric expression and returns the remainder portion of the quotient. | `MOD(expr1, expr2)` |
| PI | `PI()` | Returns the constant value of pi. | `PI()` |
| POWER | `POWER(Profit, 2)` | Takes the first numeric expression and raises it to the power specified in the second numeric expression. | `POWER(expr1, expr2)` |
| RADIANS | `RADIANS(30)` | Converts an expression from degrees to radians.<br>*expr* is any expression that evaluates to a numerical value. | `RADIANS(expr)` |
| RAND | `RAND()` | Returns a pseudo-random number between 0 and 1. | `RAND()` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| RANDFromSeed | `RAND(2)` | Returns a pseudo-random number based on a seed value. For a given seed value, the same set of random numbers are generated. | `RAND(expr)` |
| ROUND | `ROUND(2.166000, 2)` | Rounds a numeric expression to *n* digits of precision.<br>*expr* is any expression that evaluates to a numerical value.<br>*integer* is any positive integer that represents the number of digits of precision. | `ROUND(expr, integer)` |
| SIGN | `SIGN(Profit)` | Returns the following:<br>• 1 if the numeric expression evaluates to a positive number<br>• -1 if the numeric expression evaluates to a negative number<br>• 0 if the numeric expression evaluates to zero | `SIGN(expr)` |
| SIN | `SIN(1)` | Calculates the sine of a numeric expression. | `SIN(expr)` |
| SQRT | `SQRT(7)` | Calculates the square root of the numeric expression argument. The numeric expression must evaluate to a nonnegative number. | `SQRT(expr)` |
| TAN | `TAN(1)` | Calculates the tangent of a numeric expression.<br>*expr* is any expression that evaluates to a numerical value. | `TAN(expr)` |
| TRUNCATE | `TRUNCATE(45.12345, 2)` | Truncates a decimal number to return a specified number of places from the decimal point.<br>*expr* is any expression that evaluates to a numerical value.<br>*integer* is any positive integer that represents the number of characters to the right of the decimal place to return. | `TRUNCATE(expr, integer)` |

## Running Aggregate Functions

Running aggregate functions perform operations on multiple values to create summary results.

| Function | Example | Description | Syntax |
|---|---|---|---|
| MAVG | | Calculates a moving average (mean) for the last *n* rows of data in the result set, inclusive of the current row.<br>*expr* is any expression that evaluates to a numerical value. *integer* is any positive integer. Represents the average of the last *n* rows of data. | `MAVG(expr, integer)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| MSUM | `select month, revenue, MSUM(revenue, 3) as 3_MO_SUM from sales_subject_area` | Calculates a moving sum for the last n rows of data, inclusive of the current row.<br>*expr* is any expression that evaluates to a numerical value. *integer* is any positive integer. Represents the sum of the last n rows of data. | `MSUM(expr, integer)` |
| RSUM | `SELECT month, revenue, RSUM(revenue) as RUNNING_SUM from sales_subject_area` | Calculates a running sum based on records encountered so far.<br>*expr* is any expression that evaluates to a numerical value. | `RSUM(expr)` |
| RCOUNT | `select month, profit, RCOUNT(profit) from sales_subject_area where profit > 200` | Takes a set of records as input and counts the number of records encountered so far.<br>*expr* is an expression of any datatype. | `RCOUNT(expr)` |
| RMAX | `SELECT month, profit,RMAX(profit) from sales_subject_area` | Takes a set of records as input and shows the maximum value based on records encountered so far.<br>*expr* is an expression of any datatype. | `RMAX(expr)` |
| RMIN | `select month, profit,RMIN(profit) from sales_subject_area` | Takes a set of records as input and shows the minimum value based on records encountered so far.<br>*expr* is an expression of any datatype. | `RMIN(expr)` |

## String Functions

String functions perform various character manipulations. They operate on character strings.

| Function | Example | Description | Syntax |
|---|---|---|---|
| ASCII | `ASCII('a')` | Converts a single character string to its corresponding ASCII code, between 0 and 255. If the character expression evaluates to multiple characters, the ASCII code corresponding to the first character in the expression is returned.<br>*expr* is any expression that evaluates to a character string. | `ASCII(expr)` |
| BIT_LENGTH | `BIT_LENGTH('abcdef')` | Returns the length, in bits, of a specified string. Each Unicode character is 2 bytes in length (equal to 16 bits).<br>*expr* is any expression that evaluates to a character string. | `BIT_LENGTH(expr)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| CHAR | `CHAR(35)` | Converts a numeric value between 0 and 255 to the character value corresponding to the ASCII code.<br>*expr* is any expression that evaluates to a numerical value between 0 and 255. | `CHAR(expr)` |
| CHAR_LENGTH | `CHAR_LENGTH(Customer_Name)` | Returns the length, in number of characters, of a specified string. Leading and trailing blanks aren't counted in the length of the string.<br>*expr* is any expression that evaluates to a character string. | `CHAR_LENGTH(expr)` |
| CONCAT | `SELECT DISTINCT CONCAT ('abc', 'def') FROM employee` | Concatenates two character strings.<br>*exprs* are expressions that evaluate to character strings, separated by commas.<br>You must use raw data, not formatted data, with `CONCAT`. | `CONCAT(expr1, expr2)` |
| INSERT | `SELECT INSERT('123456', 2, 3, 'abcd') FROM table` | Inserts a specified character string into a specified location in another character string.<br>*expr1* is any expression that evaluates to a character string. Identifies the target character string.<br>*integer1* is any positive integer that represents the number of characters from the beginning of the target string where the second string is to be inserted.<br>*integer2* is any positive integer that represents the number of characters in the target string to be replaced by the second string.<br>*expr2* is any expression that evaluates to a character string. Identifies the character string to be inserted into the target string. | `INSERT(expr1, integer1, integer2, expr2)` |
| LEFT | `SELECT LEFT('123456', 3) FROM table` | Returns a specified number of characters from the left of a string.<br>*expr* is any expression that evaluates to a character string<br>*integer* is any positive integer that represents the number of characters from the left of the string to return. | `LEFT(expr, integer)` |
| LENGTH | `LENGTH(Customer_Name)` | Returns the length, in number of characters, of a specified string. The length is returned excluding any trailing blank characters.<br>*expr* is any expression that evaluates to a character string. | `LENGTH(expr)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| LOCATE | `LOCATE('d' 'abcdef')` | Returns the numeric position of a character string in another character string. If the character string isn't found in the string being searched, the function returns a value of 0.<br>*expr1* is any expression that evaluates to a character string. Identifies the string for which to search.<br>*expr2* is any expression that evaluates to a character string.<br>Identifies the string to be searched. | `LOCATE(expr1, expr2)` |
| LOCATEN | `LOCATEN('d' 'abcdef', 3)` | Like LOCATE, returns the numeric position of a character string in another character string. LOCATEN includes an integer argument that enables you to specify a starting position to begin the search.<br>*expr1* is any expression that evaluates to a character string. Identifies the string for which to search.<br>*expr2* is any expression that evaluates to a character string. Identifies the string to be searched.<br>*integer* is any positive (nonzero) integer that represents the starting position to begin to look for the character string. | `LOCATEN(expr1, expr2, integer)` |
| LOWER | `LOWER(Customer_Name)` | Converts a character string to lowercase.<br>*expr* is any expression that evaluates to a character string. | `LOWER(expr)` |
| OCTET_LENGTH | `OCTET_LENGTH('abcdef')` | Returns the number of bytes of a specified string.<br>*expr* is any expression that evaluates to a character string. | `OCTET_LENGTH(expr)` |
| POSITION | `POSITION('d', 'abcdef')` | Returns the numeric position of *strExpr1* in a character expression. If *strExpr1* isn't found, the function returns 0.<br>*expr1* is any expression that evaluates to a character string. Identifies the string to search for in the target string.<br>*expr2* is any expression that evaluates to a character string. Identifies the target string to be searched. | `POSITION(expr1 IN expr2)` |
| REPEAT | `REPEAT('abc', 4)` | Repeats a specified expression *n* times.<br>*expr* is any expression that evaluates to a character string<br>*integer* is any positive integer that represents the number of times to repeat the character string. | `REPEAT(expr, integer)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| REPLACE | `REPLACE('abcd1234', '123', 'zz')` | Replaces one or more characters from a specified character expression with one or more other characters.<br>*expr1* is any expression that evaluates to a character string. This is the string in which characters are to be replaced.<br>*expr2* is any expression that evaluates to a character string. This second string identifies the characters from the first string that are to be replaced.<br>*expr3* is any expression that evaluates to a character string. This third string specifies the characters to substitute into the first string. | `REPLACE(expr1, expr2, expr3)` |
| RIGHT | `SELECT RIGHT('123456', 3) FROM table` | Returns a specified number of characters from the right of a string.<br>*expr* is any expression that evaluates to a character string.<br>*integer* is any positive integer that represents the number of characters from the right of the string to return. | `RIGHT(expr, integer)` |
| SPACE | `SPACE(2)` | Inserts blank spaces.<br>*integer* is any positive integer that indicates the number of spaces to insert. | `SPACE(expr)` |
| SUBSTRING | `SUBSTRING('abcdef' FROM 2)` | Creates a new string starting from a fixed number of characters into the original string.<br>*expr* is any expression that evaluates to a character string.<br>*startPos* is any positive integer that represents the number of characters from the start of the left side of the string where the result is to begin. | `SUBSTRING([SourceString] FROM [StartPostition])` |
| SUBSTRINGN | `SUBSTRING('abcdef' FROM 2 FOR 3)` | Like SUBSTRING, creates a new string starting from a fixed number of characters into the original string.<br>*SUBSTRINGN* includes an integer argument that enables you to specify the length of the new string, in number of characters.<br>*expr* is any expression that evaluates to a character string.<br>*startPos* is any positive integer that represents the number of characters from the start of the left side of the string where the result is to begin. | `SUBSTRING(expr FROM startPos FOR length)` |

| Function | Example | Description | Syntax |
|---|---|---|---|
| TrimBoth | `Trim(BOTH '_' FROM '_abcdef_')` | Strips specified leading and trailing characters from a character string.<br>*char* is any single character. If you omit this specification (and the required single quotes), a blank character is used as the default.<br>*expr* is any expression that evaluates to a character string. | `TRIM(BOTH char FROM expr)` |
| TRIMLEADING | `TRIM(LEADING '_' FROM '_abcdef')` | Strips specified leading characters from a character string.<br>*char* is any single character. If you omit this specification (and the required single quotes), a blank character is used as the default.<br>*expr* is any expression that evaluates to a character string. | `TRIM(LEADING char FROM expr)` |
| TRIMTRAILING | `TRIM(TRAILING '_' FROM 'abcdef_')` | Strips specified trailing characters from a character string.<br>*char* is any single character. If you omit this specification (and the required single quotes), a blank character is used as the default.<br>*expr* is any expression that evaluates to a character string. | `TRIM(TRAILING char FROM expr)` |
| UPPER | `UPPER(Customer_Name)` | Converts a character string to uppercase.<br>*expr* is any expression that evaluates to a character string. | `UPPER(expr)` |

## System Functions

The `USER` system function returns values relating to the session.

It returns the user name you signed in with.

| Function | Example | Description | Syntax |
|---|---|---|---|
| DATABASE | | Returns the name of the Oracle BI subject area to which you're logged on. | `DATABASE()` |
| USER | | Returns the user name for the Oracle BI Repository to which you're logged on. | `USER()` |

## Time Series Functions

Time series functions are aggregate functions that operate on time dimensions.

Time dimension members must be at or below the level of the function. Because of this, one or more columns that uniquely identify members at or below the given level must be projected in the query.

| Function | Example | Description | Syntax |
|---|---|---|---|
| AGO | `SELECT Year_ID, AGO(sales, year, 1)` | Calculates the aggregated value of a measure from the current time to a specified time period in the past. For example, AGO can produce sales for every month of the current quarter and the corresponding quarter-ago sales. | `AGO(expr, time_level, offset)` |
| PERIODROLLING | `SELECT Month_ID, PERIODROLLING (monthly_sales, -1, 1)` | Computes the aggregate of a measure over the period starting *x* units of time and ending *y* units of time from the current time. For example, PERIODROLLING can compute sales for a period that starts at a quarter before and ends at a quarter after the current quarter.<br>*measure* is the name of a measure column.<br>*x* is an integer that specifies the offset from the current time.<br>*y* specifies the number of time units over which the function computes.<br>*hierarchy* is an optional argument that specifies the name of a hierarchy in a time dimension, such as *yr*, *mon*, *day*, that you want to use to compute the time window. | `PERIODROLLING(measure , x [,y])` |
| TODATE | `SELECT Year_ID, Month_ID, TODATE (sales, year)` | Aggregates a measure from the beginning of a specified time period to the currently displayed time. For example, this function can calculate Year to Date sales.<br>*expr* is an expression that references at least one measure column.<br>*time_level* is the type of time period, such as quarter, month, or year. | `TODATE(expr, time_level)` |

### FORECAST Function

Creates a time-series model of the specified measure over the series using Exponential Smoothing (ETS) or Seasonal ARIMA or ARIMA, and outputs a forecast for a set of periods as specified by *numPeriods*.

**Syntax** `FORECAST(numeric_expr, ([series]), output_column_name, options, [runtime_binded_options])])`

Where:

- *numeric_expr* indicates the measure to forecast, for example, revenue data to forecast.
- *series* indicates the time grain at which the forecast model is built. This is a list of one or more time dimension columns. If you omit series, then the time grain is determined from the query.
- *output_column_name* indicates the output column. The valid values are *forecast*, *low*, *high*, and *predictionInterval*.
- *options* indicates a string list of name/value pairs separated by a semi-colon (;). The value can include *%1 ... %N* specified in `runtime_binded_options`.

- *runtime_binded_options* indicates a comma separated list of runtime-binded columns and options.

**FORECAST Function Options** The following table list available options to use with the `FORECAST` function.

| Option Name | Values | Description |
|---|---|---|
| numPeriods | Integer | The number of periods to forecast |
| predictionInterval | 0 to 100, where higher values specify higher confidence | The confidence level for the prediction. |
| modelType | ETS<br>SeasonalArima<br>ARIMA | The model to use for forecasting. |
| useBoxCox | TRUE<br>FALSE | If *TRUE*, then use Box-Cox transformation. |
| lambdaValue | Not applicable | The Box-Cox transformation parameter.<br>Ignore if NULL or when `useBoxCox` is *FALSE*.<br>Otherwise the data is transformed before the model is estimated. |
| trendDamp | TRUE<br>FALSE | This is a parameter for ETS model.<br>If *TRUE*, then use damped trend. If FALSE or NULL, then use non-damped trend. |
| errorType | Not applicable | This is a parameter for ETS model. |
| trendType | none("N")<br>additive("A")<br>multiplicative("M")<br>automatically selected("Z") | This is a parameter for ETS model. |
| seasonType | none("N")<br>additive("A")<br>multiplicative("M")<br>automatically selected("Z") | This is a parameter for ETS model. |
| modelParamIC | ic_auto<br>ic_aicc<br>ic_bic<br>ic_auto (this is the default) | The information criterion (IC) used in the model selection. |

### Revenue Forecast by Day Example

This example selects revenue forecast by day.

```
FORECAST("A - Sample Sales"."Base Facts"."1- Revenue" Target,
("A - Sample Sales"."Time"."T00 Calendar Date"),'forecast',
'numPeriods=30;predictionInterval=70;') ForecastedRevenue
```

### Revenue Forecast by Year and Quarter Example

This example selects revenue forecast by year and quarter.

```
FORECAST("A - Sample Sales"."Base Facts"."1- Revenue",
("A - Sample Sales"."Time"."T01 Year" timeYear, "A - Sample Sales"."Time"."T02
Quarter" TimeQuarter),'forecast', 'numPeriods=30;predictionInterval=70;')
ForecastedRevenue
```

# Constants

You can use constants to include specific dates and times in expressions.

Available constants include Date, Time, and Timestamp.

| Constant | Example | Description | Syntax |
|---|---|---|---|
| DATE | `DATE [2014-04-09]` | Inserts a specific date. | `DATE [yyyy-mm-dd]` |
| TIME | `TIME [12:00:00]` | Inserts a specific time. | `TIME [hh:mi:ss]` |
| TIMESTAMP | `TIMESTAMP [2014-04-09 12:00:00]` | Inserts a specific timestamp. | `TIMESTAMP [yyyy-mm-dd hh:mi:ss]` |

# Types

You can use data types, such as `CHAR`, `INT`, and `NUMERIC` in expressions.

For example, you use types when creating `CAST` expressions that change the data type of an expression or a null literal to another data type.

G

# Oracle Analytics Desktop SDK Reference

This topic describes the software development kit (SDK) that you can use to develop and deploy visualization plug-ins to your Oracle Analytics Desktop installation.

**Topics:**



## Oracle Analytics Desktop SDK

The SDK provides a development environment where you can create and develop custom visualization plug-ins and deploy them to your Oracle Analytics Desktop installation.

**Topics:**



### Scripts

Your installation includes the scripts that you use to create a development environment and create skeleton visualization plug-ins.

The scripts are located in this directory: `<your_installation_directory>\Oracle Analytics Desktop\tools\bin`.

For example, `C:\Program Files\Oracle Analytics Desktop\tools\bin`
Note the following script names and descriptions:

- **bicreateenv** - Run this script to create the development environment where you develop your plug-ins.
- **bicreateplugin** - Run this script to create a skeleton visualization to quickly get started on developing your custom plug-in.

- **bideleteplugin** - Run this script to delete a plug-in from your development environment.
- **bivalidate** - Run this script with the `gradlew validate` command to call the bivalidate script. The bivalidate script validates whether the JSON configuration files are properly formatted and contain appropriate visualization configuration.

## Other Resources

You can use resources other than scripts to develop your custom visualization plug-ins.

Note the following available resources:

- **circlePack sample** - The circlePack sample is included in your development environment. You can deploy and use this sample immediately. However, the sample is designed for you to use with the provided tutorial to learn how to develop a visualization plug-in. You can also copy the sample and use it as a template for the visualization plug-ins that you want to create.

  The circlePack sample is located in `<your_development_directory>\src\sampleviz\sample-circlepack`

  For example, `C:\OracleDVDev\src\sampleviz\sample-circlepack`
- **Other visualization plug-in samples** - You can download plug-in examples from the Oracle Analytics Library.
- **Tutorial** - The tutorial contains information and instructions to help you understand how to create a robust visualization plug-in. This tutorial provides step-by-step instructions for modifying the circlePack sample included in your plug-in development environment.

  Tutorial
- **JS API documentation** - This documentation contains JavaScript reference information that you need to develop a visualization plug-in. See Oracle Analytics Desktop SDK JavaScript Reference.

# Create the Visualization Plug-in Development Environment

You need to set the PATH environment variable and create the development environment before you can create visualization plug-ins.

1. Using the command prompt, create an empty development directory. For example, `C:\OracleAnalyticsDev`.
2. Set the PATH environment variable. For example,

```
set ANALYTICSDESKTOP_SDK_HOME="C:\Program Files\Oracle Analytics
Desktop"
set PLUGIN_DEV_DIR=C:\OracleAnalyticsDev
REM add tools\bin to path:
set PATH=%ANALYTICSDESKTOP_SDK_HOME%\tools\bin;%PATH%
```

3. Run the **bicreateenv** script included in your installation to create the development environment in the empty directory. For example,

   ```
   cd C:\OracleAnalyticsDev
   bicreateenv
   ```

   For information about the options available for running this script, see the script's command-line help. For example,

   ```
   C:\OracleAnalyticsDev>bicreateenv -help
   ```

   The complete development environment, including build.gradle and gradlew, is created in the directory that you specified.

4. (Optional) If you're working behind a web proxy, then you need to set gradle.properties to point to your proxy. The gradle.properties are located in your development environment, for example `C:\OracleAnalyticsDev\gradle.properties`.

   Use the following example to set your gradle.properties:

   ```
   systemProp.https.proxyHost=www-proxy.somecompany.com
   systemProp.https.proxyPort=80
   systemProp.https.nonProxyHosts=*.somecompany.com|
   *.companyaltname.com
   ```

# Create a Skeleton Visualization Plug-in

After you create a skeleton visualization plug-in in your development environment, you then develop it into a robust visualization plug-in and deploy it to your Oracle Analytics Desktop environment.

1. Run the **bicreateplugin** script included in your installation to create a skeleton visualization. Use the following syntax:

   ```
   bicreateplugin viz -<subType> -<id> -<name>
   ```

   - `<subType>` is the type of visualization that you want to create. Your choices are:
     - **basic** - Use this option to create a visualization that doesn't use any data from Oracle Analytics Desktop or use any data model mapping. This is like the Image and Text visualization types delivered with Oracle Analytics Desktop. For example, you can use this visualization type to show an image or some text that's coded into the plug-in or from a configuration. You can use this type of visualization to improve formatting.
     - **dataviz** -This type renders data from data sources registered with Oracle Analytics Desktop into a chart or table or some other representation on the screen. It also respond to marking events from other visualizations on the same canvas and publish interaction events to affect other visualizations on the same canvas.
     - **embeddableDataviz** - This type renders data from data sources registered with Oracle Analytics Desktop into the cells of a trellis visualization. It also responds to marking events from other visualizations on the same canvas and publish interaction events to affect other visualizations on the same canvas.

- <id> is your domain and the name that you want to give the visualization directory and components in your development environment. For example, com-company.basicviz.
- <name> is the name of the visualization plug-in that you test, deploy, and use in projects.

For example to create a basic visualization, name its development directory com-company-basicviz, and name the visualization plug-in helloViz, enter and run the following command:

```
C:\OracleDevDir>bicreateplugin viz –subType basic –id
com.company.basicviz –name helloViz
```

2. (Optional) Open the script's command-line help for information about the options available for running this script. For example, `C:\OracleDVDev> bicreateplugin -help`

When you run the `bicreateplugin -viz` command for the first time, the system creates the customviz directory in the following location.
`<your_development_environment>\src\customviz`
All custom visualization development directories that you create are added to this directory.

For example, `C:\OracleDVDev\src\customviz\com-company-basicviz`

# Create a Skeleton Skin or Unclassified Plug-in

The `bicreateplugin -unclassified` command creates an empty plug-in with plugin.xml, localization bundles. The `bicreateplugin -skin` command creates a skeleton skin plug-in.

- Run the **createplugin** script included in your installation to create a skeleton plug-in. Use one of the following syntaxes:

  ```
  bicreateplugin -skin -<id>
  ```

  ```
  bicreateplugin -unclassified -<id>
  ```

  - <id> is your domain and the name that you want to give the visualization. For example, com-company.newskin

  For example, to create a skin plug-in, enter and run the following command:

  ```
  C:\OracleDevDir>bicreateplugin skin –id com.company.newskin
  ```

# Develop a Visualization Plug-in

After you create the skeleton visualization plug-in, you can use resources provided by Oracle to help you develop your plug-in.

The directories for dataviz and embeddableDataviz types include the datamodelhandler.js file, which contains the physical-to-logical data mapping format. This file also determines how Oracle Analytics Desktop renders and passes user interactions to the server.

- Use the tutorial to learn how to perform development tasks such as implement data mapping.

  Tutorial

- Use the .JS API documentation to learn how to add dependencies. See Oracle Analytics Desktop SDK JavaScript Reference.

# Run in SDK Mode and Test the Plug-in

You can run Oracle Analytics Desktop in SDK mode from your browser when you're developing your visualization plug-in or when you want to test your visualization plug-in.

1. Execute the `gradlew run` command. For example, `C:\OracleDevDir>gradlew run`

   After you run the command, note the following results:

   - Oracle Analytics Desktop opens in SDK mode in your default browser. Use the browser's JavaScript debugger to test and debug the application.
   - The visualization that you created is available in the Visualizations pane of Oracle Analytics Desktop.
   - A system tray is displayed in the operating system's toolbar and includes three links: Launch Browser, which you use to launch or relaunch your default browser to display Oracle Analytics Desktop; Copy URL to Clipboard, which you can use to copy the URL and paste it into a different browser; and Shutdown, which you use to shut down the development browser.
2. Test your visualization by dragging and dropping it to a project's canvas and adding data elements.
3. If necessary, continue developing the visualization plug-in. When working in SDK mode in the browser, you can update the .JS definition and refresh the browser to see your changes.

# Validate the Visualization Plug-in

After you've tested your visualization plug-in and before you can package and deploy it, you must validate it.

1. Run the `gradlew validate` command. For example,

   ```
   cd C:\OracleDVDev
   .\gradlew validate
   ```

   This step validates whether the JSON configuration files are properly formatted and contain appropriate visualization configuration. If the validation discovers any errors, then the system displays error messages.
2. To check for errors in the JavaScript source files, use your browser's development tools.

# Build, Package, and Deploy the Visualization Plug-in

After you validate the visualization plug-in, you've to build and package it, and then copy the resulting distributions into your installation directory.

The build and package process runs for all of the visualizations in your development directory, and each plug-in is contained in its own zip file. There's no way to build and

package specific visualizations. If you want to exclude visualizations from the build and package process, then you've to move the visualizations that you want to exclude out of your development directory, or delete them from the directory before you perform the build. See Delete Plug-ins from the Development Environment.

1. Run the `gradlew build` command. For example,

   ```
   cd C:\OracleDVDev
   .\gradlew clean build
   ```

   A build directory is added to your development environment. For example, `C:\OracleDVDev\build\distributions`. This directory contains a zip file for each visualization. The zip file's name is the one that you gave the visualization when you created its skeleton. For example, basicviz.zip.

2. Copy the zip files to your installation directory. For example, `%localappdata%\OracleAnalyticsDesktop\plugins`.

# Delete Plug-ins from the Development Environment

You can use the `bideleteplugin` script to delete the unneeded plug-ins from your development environment.

The build and package process includes all of the visualizations contained in your development directory. There is no way to build and package specific visualizations. To exclude any unwanted visualizations from the build, you can delete them before you perform the build and package process.

1. If you want to delete a visualization plug-in, then run the `bideleteplugin` command, using the following syntax:

   ```
   cd C:\<your_development_directory>
   bideleteplugin viz -id <name_of_your_domain>.<name_of_viz_plugin>
   ```

2. If you want to delete an unclassified plug-in, then run the `bideleteplugin` command, using the following syntax:

   ```
   cd C:\<your_development_directory>
   bideleteplugin unclassified -id
   <name_of_your_domain>.<name_of_unclassified_plugin>
   ```

3. If you want to delete a skin plug-in, then run the `bideleteplugin` command, using the following syntax:

   ```
   cd C:\<your_development_directory>
   bideleteplugin skin -id <name_of_your_domain>.<name_of_skin_plugin>
   ```