About This Guide

Purpose

The Schema Editor User Guide explains how to run the Schema Editor on all supported platforms, and how to use it to create and modify schema definitions for NoSQL data stores.

Audience

The guide is intended for end users of the Schema Editor who want to create or modify a schema definition for a connection to a NoSQL data store.

Knowledge Prerequisites

To use the Schema Editor, the following knowledge is helpful:

- An understanding of the data structures in the data store
- Familiarity with the data store for which you are modifying schemas
- Familiarity with the platform on which you are using the Schema Editor
- Exposure to SQL

Document Conventions

Italics are used when referring to book and document titles.

Bold is used in procedures for graphical user interface elements that a user clicks and text that a user types.

Monospace font indicates commands, source code or contents of text files.

Note:

A text box with a pencil icon indicates a short note appended to a paragraph.

Important:

A text box with an exclamation mark indicates an important comment related to the preceding paragraph.
Table of Contents

About the Schema Editor ................................................................. 5

Using the Schema Editor for ODBC Connections .................................. 6
  System Requirements ........................................................................ 6
  Starting the Schema Editor for ODBC Connections .............................. 7

Using the Schema Editor for JDBC Connections ...................................... 9
  System Requirements ........................................................................ 9
  Starting the Schema Editor for JDBC Connections .............................. 10

Sampling Data .................................................................................... 12
  Sampling Data for Couchbase ........................................................... 12
  Sampling Data for DocumentDB ......................................................... 14
  Sampling Data for DynamoDB ............................................................ 17
  Sampling Data for MongoDB ............................................................. 18

Modifying a Schema Definition .......................................................... 21
  Working in the Design View .............................................................. 21
  Opening a Schema Definition for Editing ............................................ 22
  Hiding and Unhiding Tables .............................................................. 23
  Renaming Tables .............................................................................. 23
  Editing Column Properties .................................................................. 24
  Column Properties ............................................................................ 25
  Adding and Deleting Columns ............................................................ 29
  Creating a Columnar View of Array Elements ..................................... 30
  Example of Virtual Tables vs. Columnar Views ................................ 30

Previewing Data .................................................................................. 33

Saving a Schema Definition ............................................................... 34

Connecting to a Data Store .................................................................. 35

Configuring Logging ........................................................................... 36

Third-Party Trademarks ....................................................................... 38

Third-Party Licenses .......................................................................... 39
About the Schema Editor

NoSQL data stores are able to store "denormalized" data, which does not follow the rules of data typing and structure that apply to traditional relational data. Denormalized data does not have a schema, and can include complex data types such as nested arrays or arrays of differently-typed elements. Because traditional ODBC and JDBC toolsets do not support these data structures, the data needs to be mapped to a relational form.

The Schema Editor is a Java application that enables you to create and modify schema definitions for NoSQL data stores. A schema definition serves as a database layer that maps NoSQL data to ODBC- and JDBC-compatible formats so that you can use standard BI tools to work with the data.

You can use the Schema Editor in conjunction with a Simba driver to create schema definitions. The driver generates the initial schema definition by sampling the data to detect its structure, and then assigning an appropriate data type to each column and generating virtual tables to represent complex data types. You can then use the options in the Schema Editor to manually modify the schema definition and make sure that your application can work with the data.

All Simba drivers that support schema definitions include the Schema Editor as part of the driver installation.

Note:
The AIX version of the Simba MongoDB ODBC Driver with SQL Connector does not include the Schema Editor, because the Schema Editor is not compatible with AIX at this time.
Using the Schema Editor for ODBC Connections

The Schema Editor application is packaged differently for ODBC and JDBC drivers. Depending on the platform that you are running and whether you are using a version of the application that is packaged with an ODBC driver or a JDBC driver, different system requirements apply, and different procedures and files are available for starting the application.

The following sections describe how to run the ODBC version of the Schema Editor. For information about the JDBC version, see Using the Schema Editor for JDBC Connections on page 9.

System Requirements

The ODBC version of the Schema Editor supports ODBC connections only. However, the schema definitions themselves are compatible with both the ODBC and JDBC versions of the Schema Editor, and can be used by both ODBC and JDBC drivers regardless of the Schema Editor that was used to create the definition.

Important:
The Schema Editor does not support Couchbase schema definitions that were created using version 1.1.2 or earlier of the ODBC driver.

Windows

The Schema Editor supports the following versions of Windows (32- and 64-bit editions are supported):

- Windows Vista, 7, 8, or 10
- Windows Server 2008 or later

Linux

The Schema Editor supports the following Linux distributions (32- and 64-bit editions are supported):

- Red Hat® Enterprise Linux® (RHEL) 5, 6, or 7
- CentOS 5, 6, or 7
- SUSE Linux Enterprise Server (SLES) 11 or 12
- Debian 7 or 8
- Ubuntu 12.04, 14.04, or 16.04
To use the Schema Editor on Linux, you must have the following software installed:

- GTK+ 2.18, 2.20, or 2.24
- One of the following ODBC driver managers:
  - iODBC 3.52.7 or later
  - unixODBC 2.3.0 or later

**Note:**
If the LD_LIBRARY_PATH environment variable on your machine includes paths for both iODBC and unixODBC, unixODBC takes precedence.

**macOS**

The Schema Editor supports macOS version 10.9, 10.10, and 10.11.

To use the Schema Editor on macOS, you must have one of the following ODBC driver managers installed:

- iODBC 3.52.7 or later
- unixODBC 2.3.0 or later

**Note:**
If the DYLD_LIBRARY_PATH environment variable on your machine includes paths for both iODBC and unixODBC, iODBC takes precedence.

**Starting the Schema Editor for ODBC Connections**

Depending on which platform you are running, different procedures and files are available for starting the application.

**To start the Schema Editor on Windows:**

- Choose one:
  - From the program group of your Simba driver, click the **SchemaEditor** application shortcut.
  - Or, open the ODBC Data Source Administrator where you created the DSN for your connection, select the DSN, click **Configure**, and then click **Schema Editor**.
Note:
When you start the Schema Editor from the DSN configuration of a driver, the application automatically uses the connection information specified in your DSN.

To start the Schema Editor on Linux:

1. Navigate to the [DRIVER_DIR]/Tools/[Bitness]/SchemaEditor folder, where [DRIVER_DIR] is the root directory of the driver and [Bitness] is the bitness of your driver installation.
2. Run the SchemaEditor application.

To start the Schema Editor on macOS:

1. Navigate to the [DRIVER_DIR]/Tools/SchemaEditor folder, where [DRIVER_DIR] is the root directory of the driver.
2. Run the SchemaEditor application.

The Schema Editor opens on the start page. From there, you can choose to create a new schema definition by sampling data, or modify an existing schema definition stored in a JSON file or a NoSQL data store.
Using the Schema Editor for JDBC Connections

The Schema Editor application is packaged differently for ODBC and JDBC drivers. Depending on the platform that you are running and whether you are using a version of the application that is packaged with an ODBC driver or a JDBC driver, different system requirements apply, and different procedures and files are available for starting the application.

The following sections describe how to run the JDBC version of the Schema Editor. For information about the ODBC version, see Using the Schema Editor for ODBC Connections on page 6.

System Requirements

The Schema Editor requires Java Runtime Environment version 8 update 66 (JRE 8u66) or later.

The JDBC version of the Schema Editor supports JDBC connections only. However, the schema definitions themselves are compatible with both the ODBC and JDBC versions of the Schema Editor, and can be used by both ODBC and JDBC drivers regardless of the Schema Editor that was used to create the definition.

Windows

The Schema Editor supports the following versions of Windows (32- and 64-bit editions are supported):

- Windows Vista, 7, 8, or 10
- Windows Server 2008 or later

Linux

The Schema Editor supports the following Linux distributions (32- and 64-bit editions are supported):

- Red Hat® Enterprise Linux® (RHEL) 5, 6, or 7
- CentOS 5, 6, or 7
- SUSE Linux Enterprise Server (SLES) 11 or 12
- Debian 7 or 8
- Ubuntu 12.04, 14.04, or 16.04

To use the Schema Editor on Linux, you must have GTK+ 2.18, 2.20, or 2.24 installed.
macOS

The Schema Editor supports macOS version 10.9, 10.10, and 10.11.

Starting the Schema Editor for JDBC Connections

Use the SchemaEditor.jar file included in your JDBC driver package to start the Schema Editor. The application must be run using JRE 8u66 or later.

To verify the JRE version:

- Open a Command Prompt or Terminal window and then run the following command:

  \[ \text{java } -version \]

This command returns information about the JRE version that your machine runs by default. For example, the following message indicates that your machine runs JRE 8u66:

  \[ \text{java version "1.8.0_66"} \]
  \[ \text{Java(TM) SE Runtime Environment (build 1.8.0_66-b18)} \]

If the command fails to run or an earlier version of Java is returned, then you must specify the full path to the JRE. For example:

  \[ "C:\Program Files\Java\jre1.8.0_66\bin\java.exe" -version \]

To start the Schema Editor:

1. Open a Command Prompt or Terminal window and then navigate to the root directory of the driver.
2. Run one of the following commands, where \[JRE\_Path]\ is the full path to the bin subfolder in the JRE directory:
   - On Windows:
     \[JRE\_Path]\java -jar SchemaEditor.jar
   - Or, on Linux or macOS:
     \[JRE\_Path]/java -jar SchemaEditor.jar

\[
\text{Note:} \\
\text{If your machine uses JRE 8u66 or later by default, then it is not necessary to include } \[JRE\_Path]\ \text{in the command.}
\]
The Schema Editor opens on the start page. From there, you can choose to create a new schema definition by sampling data, or modify an existing schema definition stored in a JSON file or a NoSQL data store.
Sampling Data

You can use the options in the Sample View to specify how the driver samples data, and then generate a schema definition. The driver samples the data in the NoSQL data store in order to detect its structure and determine the data mappings that best support the data.

Depending on the data store that you are working with, different sampling options and workflows may be available. For detailed information about how to sample the data from your data store, see the following:

- Sampling Data for Couchbase on page 12
- Sampling Data for DocumentDB on page 14
- Sampling Data for DynamoDB on page 17
- Sampling Data for MongoDB on page 18

Sampling Data for Couchbase

You can use the options in the Sample View to specify how the driver samples data from Couchbase to generate a schema definition. The driver samples the data in order to detect its structure and determine the data mappings that best support the data.

To sample data for Couchbase:

1. Choose one:
   - From the start page, click **Create New**, provide your connection information, and then click **Connect**. For detailed information about how to specify connection information, **Connecting to a Data Store** on page 35.
   
   The Sample dialog box opens.
   - Or, from the Design View, click the **Sample View** tab. If you are not already connected to a data store, the Schema Editor prompts you to provide your connection information. For detailed information about how to specify connection information, **Connecting to a Data Store** on page 35.
   
   The Schema Editor displays the Sample View.

2. In the **Sampling Count** field, type the maximum number of records that the driver can sample to generate the schema definition. To sample every record in the database, set this option to 0.
Typically, sampling a large number of records results in a schema definition that is more accurate and better able to represent all the data in the database. However, the sampling process might take longer than expected when many records are sampled, especially if the database contains complex, nested data structures.

3. In the bottom pane, specify the collections that the driver samples records from by selecting the corresponding check boxes in the Selected column. You can select every collection in the database by selecting the check box in the Selected column header.

You can group and sort collections by clicking a column header. For example, to group collections based on the catalogs they belong to and then sort those groupings by catalog name in ascending order, click the Catalog column header. To sort the list in descending order, click the header again. To disable sorting, click the header a third time.

4. In each field in the Type Name Filter column, type the name of the attribute that the bucket (or collection) uses to specify document types.

When generating a schema definition, the driver creates a base table for each different document type. For example, if you specify type as the type name filter for a bucket, and that bucket contains documents that have the type values retail and internal, then the driver creates two tables named "retail" and "internal" in the schema definition.

Although specifying a type name filter is optional, it is highly recommended that you do so. If you do not provide a type name filter, then the driver uses the bucket name as the type for all the documents in the bucket, and the resulting schema definition is likely to contain errors.

In addition, make sure that you specify the type attribute for every data bucket in the Couchbase Server instance or cluster to which you are connecting. Otherwise, you might encounter issues when working with document types that are not part of the schema definition.

5. To generate the schema, click Sample.

The driver samples the data as specified and generates a schema definition, which opens in the Design View in the Schema Editor. If you return to the Sample View, you
will see that the check boxes in the Sampled column are selected for all the columns that were included in the sampling process.

**Sampling Data for DocumentDB**

You can use the options in the Sample View to specify how the driver samples DocumentDB data in order to generate a schema definition. The driver samples the data in order to detect its structure and determine the data mappings that best support the data.

When sampling your DocumentDB data to create a schema definition, the Schema Editor automatically identifies columns, assigns data types, and creates virtual tables for complex data types. To further refine your schema, you can use the following methods:

- **Collection Mapping (default):** The driver samples the data source and creates a single table schema.
- **Table Delimiters:** You can specify one or more attributes from the data, and the driver creates separate tables for each unique value of those selected attributes. Optionally, you can choose specific values for each attribute, and tables are only created for that subset of values. With this option, any unspecified values are excluded from the schema.

**Note:**

Collection Mapping and Table Delimiters are mutually exclusive. You can only sample a collection using one of these methods.

- **Views:** In addition to the two above methods, you can specify one or more views and the driver represents each as a separate table in the schema. A view is defined using the query language of the data source. For example:

```sql
SELECT c.Name
FROM AddressBook a
JOIN c IN a.children
WHERE a.id = '123'
ORDER BY a.address.city ASC
```
Important:

If you are using the 32-bit Schema Editor to work with deeply-nested structures or thousands of attributes, you may encounter a Java heap space error. In this case you may need to specify larger initial and maximum values for the memory allocation pool used by the Schema Editor. To do this, launch the Schema Editor from the command line using the following command:

```
java -jar SchemaEditor.jar -Xms[initial]m -Xmxmaximumm
```

where:

- `[initial]` is the initial size, in megabytes, of the memory pool
- `[maximum]` is the maximum size, in megabytes, of the memory pool

For example, to specify an initial memory pool of 512 megabytes and a maximum memory pool of 2048 megabytes, use the following command:

```
java -jar SchemaEditor.jar -Xms512m -Xmx2048m
```

To sample data using Collection Mapping:

1. Launch the Schema Editor.
2. Select your DSN from the list, or enter an appropriate connection string.
3. Click Connect.
4. Under Create A New Schema Definition, click Create New. Schema Editor shows a list of databases and collections from the data source.
5. Select the collection or collections you wish to sample from the list.
6. Click Sample.

To sample data using Table Delimiters:

1. Launch the Schema Editor.
2. Select your DSN from the list, or enter an appropriate connection string.
3. Click Connect.
4. Under Create A New Schema Definition, click Create New. Schema Editor shows a list of databases and collections from the data source.
5. Next to the collection you wish to sample, click Edit under the Mapping Method column.
6. Set the Mapping Method to Table Delimiters.
7. In the Attributes field, enter the Attribute you wish to use as a delimiter and press Enter. Repeat this process to add all relevant attributes.
You can delete attributes by selecting them and clicking **Delete**, or **Delete All** to remove all attributes.

**Note:**

If you do not specify values for the selected attribute or attributes, the Schema Editor creates tables for all unique values of the specified attributes, requiring it to survey the entire data source. For larger data sources this can take a significant amount of time.

8. In the **Values** field, enter the unique value you want used in your schema and press **Enter**. Repeat this process to add all relevant values. Any values that are not specified, and data related to them, will be excluded from the schema.

You can delete values by selecting them and clicking **Delete**, or **Delete All** to remove all values.

9. Click **OK** to return to the Sample dialog box.

10. To save the information entered here so it can be reused with another data source with the same structure, click **Save**.

11. Click **Sample**.

**To sample data using View definitions:**

1. Launch the Schema Editor.
2. Select your DSN from the list, or enter an appropriate connection string.
3. Click **Connect**.
4. Under Create A New Schema Definition, click **Create New**.
   Schema Editor shows a list of databases and collections from the data source.
5. Select the collection or collections you wish to sample from the list. Set up any Table Delimiters you want to use according to the previous procedure.
6. To add a new view definition, do the following:
   a. Under the **View Definitions** column for the appropriate collection, click **Add**.
   b. Click **New**.
   c. Type a unique name for the View Definition, then click **OK**.
   d. In the **Edit View** field, type a query that retrieves the information you want to view.
   e. To add an additional view, click **Add**.
      To remove views from the list, use **Delete**.
   f. To return to the Sample dialog box, click **OK**.
7. Click **Sample**.
The driver samples the data as specified and generates a schema definition, which opens in the Design View in the Schema Editor.

**Sampling Data for DynamoDB**

You can use the options in the Sample View to specify how the driver samples data from DynamoDB to generate a schema definition. The driver samples the data in order to detect its structure and determine the data mappings that best support the data.

**To sample data for DynamoDB:**

1. Choose one:
   - From the start page, click **Create New**, provide your connection information, and then click **Connect**. For detailed information about how to specify connection information, Connecting to a Data Store on page 35.

   The Sample dialog box opens.
   - Or, from the Design View, click the **Sample View** tab. If you are not already connected to a data store, the Schema Editor prompts you to provide your connection information. For detailed information about how to specify connection information, Connecting to a Data Store on page 35.

   The Schema Editor displays the Sample View.

2. From the **Sampling Method** drop-down list, select the direction in which the driver reads data during sampling. For example, if you select **Forward**, the driver samples data starting with the first record in the data store, then samples the next record, and so on.

3. In the **Sampling Count** field, type the maximum number of records that the driver can sample to generate the schema definition. To sample every record in the database, set this option to **0**.

   **Note:**
   Typically, sampling a large number of records results in a schema definition that is more accurate and better able to represent all the data in the database. However, the sampling process might take longer than expected when many records are sampled, especially if the database contains complex, nested data structures.

4. In the **Sampling Interval** field, type the interval at which the driver samples a record when scanning through the data store. For example, if you set this option to **2**, then the driver samples every second record in the data store.

5. In the bottom pane, specify the collections that the driver samples records from by...
selecting the corresponding check boxes in the **Selected** column. You can select every collection in the database by selecting the check box in the **Selected** column header.

**Note:**

You can group and sort collections by clicking a column header. For example, to group collections based on the catalogs they belong to and then sort those groupings by catalog name in ascending order, click the **Catalog** column header. To sort the list in descending order, click the header again. To disable sorting, click the header a third time.

6. To generate the schema, click **Sample**.

The driver samples the data as specified and generates a schema definition, which opens in the Design View in the Schema Editor. If you return to the Sample View, you will see that the check boxes in the Sampled column are selected for all the columns that were included in the sampling process.

**Sampling Data for MongoDB**

You can use the options in the Sample View to specify how the driver samples data from MongoDB to generate a schema definition. The driver samples the data in order to detect its structure and determine the data mappings that best support the data.

**To sample data for MongoDB:**

1. Choose one:
   - From the start page, click **Create New**, provide your connection information, and then click **Connect**. For detailed information about how to specify connection information, **Connecting to a Data Store** on page 35.

      The Sample dialog box opens.
   - Or, from the Design View, click the **Sample View** tab. If you are not already connected to a data store, the Schema Editor prompts you to provide your connection information. For detailed information about how to specify connection information, **Connecting to a Data Store** on page 35.

      The Schema Editor displays the Sample View.

2. From the **Sampling Method** drop-down list, select the sampling method to use. You can sample record sequentially by selecting the **Forward** (starting from the first record), or **Backward** (starting from the last record) options, or at random by selecting the **Random** option.
3. In the **Sampling Count** field, type the maximum number of records that the driver can sample to generate the schema definition. To sample every record in the database, set this option to **0**.

**Note:**

The random sampling method is only supported by MongoDB Server 3.2 or higher.

4. In the **Sampling Interval** field, type the interval at which the driver samples a record when scanning through the data store. For example, if you set this option to **2**, then the driver samples every second record in the data store. This option is ignored if the Sampling Method is set to Random.

5. In the bottom pane, specify the collections that the driver samples records from by selecting the corresponding check boxes in the **Selected** column. You can select every collection in the database by selecting the check box in the **Selected** column header.

**Note:**

Typically, sampling a large number of records results in a schema definition that is more accurate and better able to represent all the data in the database. However, the sampling process might take longer than expected when many records are sampled, especially if the database contains complex, nested data structures.

6. To filter the data from a collection so that only certain documents are included in the sampling process, type a JSON filter in the corresponding field in the **JSON Filter** column. The driver samples data only from documents in the collection that meet the filter conditions.

**Note:**

You can group and sort collections by clicking a column header. For example, to group collections based on the catalogs they belong to and then sort those groupings by catalog name in ascending order, click the **Catalog** column header. To sort the list in descending order, click the header again. To disable sorting, click the header a third time.
7. To generate the schema, click **Sample**.

The driver samples the data as specified and generates a schema definition, which opens in the Design View in the Schema Editor. If you return to the Sample View, you will see that the check boxes in the Sampled column are selected for all the columns that were included in the sampling process.
Modifying a Schema Definition

In some cases, you might need to manually modify a schema definition to make sure that your data displays as desired in ODBC and JDBC applications. Use the options in the Design View to modify your schema definitions as needed.

📝 Note:
Catalog and schema properties cannot be modified.

For detailed information about how you can modify your schema definition, see the following sections:

- Opening a Schema Definition for Editing on page 22
- Hiding and Unhiding Tables on page 23
- Renaming Tables on page 23
- Editing Column Properties on page 24
- Adding and Deleting Columns on page 29
- Creating a Columnar View of Array Elements on page 30

Working in the Design View

The Design View lets you inspect and modify the schema definition that is currently open in the application. The following image shows the Design View, with the left pane outlined in red and the main pane outlined in green:
Left Pane

The left pane contains a tree structure representing the relational database structure that the NoSQL data is mapped to. You can browse through the tree structure to inspect the database structure created by the schema definition, or select an object to view its properties in the main pane.

Each node in the tree is a database object, such as a catalog, table, or virtual table. The icons beside the object names represent the object type and indicate whether the object is hidden from ODBC and JDBC applications:

- 💼 represents catalogs.
- 📖 represents schemas (Couchbase only).
- 📊 represents tables.
- 📊 represents virtual tables, which are used to renormalize complex data.
- ⭕️ indicates that the table or virtual table is hidden.

To locate a specific object, you can type the object name in the Filter field at the top. The left pane excludes any objects with names that do not contain the value in the Filter field.

You can also exclude hidden tables and virtual tables from the left pane by clicking Exclude Hidden Tables at the top. However, if a hidden object has an unhidden object nested under it, then the hidden object continues to appear in the pane.

Main Pane

The main pane shows the properties of the selected object and its contents. For example, when you select a table, the top area of the pane shows table properties while the middle area shows column properties.

You can use the main pane to edit column properties and preview the results of your schema definition. For more information, see Editing Column Properties on page 24 and Previewing Data on page 33.

Opening a Schema Definition for Editing

📝 Note:

Opening a schema definition from the data store is not supported in DocumentDB.

You can modify schema definitions that are saved in a local JSON file or in a data store. Open a schema definition in the Schema Editor and then use the options in the Design View to modify it as needed.
To open a schema definition from a JSON file:

1. Choose one:
   - In the start page, click **Modify Local Schema Definition**.
   - Or, in the Design View or Sample View, from the menu bar at the top, select **File > Open**.
2. Browse to select your JSON file and then click **Open**. A connection dialog for the data source will be displayed.
3. Enter any required data for connecting to your data source and click **Connect**. For more information see **Connecting to a Data Store** on page 35.

The schema definition opens in the Design View.

To open a schema definition from a data store:

- In the start page, click **Modify Database Schema Definition**, provide your connection information, and then click **Connect**. For detailed information about how to specify connection information, see **Connecting to a Data Store** on page 35.

The schema definition opens in the Design View.

### Hiding and Unhiding Tables

You can hide tables so that they cannot be viewed or queried in ODBC and JDBC applications.

**To hide a table:**

- In the Design View, in the left pane, right-click the table that you want to hide and then click **Hide Table**.

In the left pane, the icon displays beside the table name to indicate that it is hidden.

**To unhide a table:**

- In the Design View, in the left pane, right-click the table that you want to unhide and then click **Unhide Table**.

The icon indicating a hidden table no longer displays beside the table name.

### Renaming Tables

You can change the table name that displays in ODBC and JDBC applications (the **SQL Name** table property).
To rename a table:

1. In the Design View, in the left pane, right-click the table that you want to rename and then click **Rename Table**.
2. In the Rename Table dialog box, type a new name in the field and then click **OK**.

The table is shown with the specified name in ODBC and JDBC applications.

**Editing Column Properties**

When you select a table or virtual table from the left pane of the Design View, the main pane displays its properties as well as the properties of the columns that it contains. You can edit column properties to change the way that the columns are handled in ODBC and JDBC applications.

⚠️ **Note:**

Some properties are read-only and cannot be edited. This includes properties such as **Column #**, **Primary Key Column**, and **Foreign Key Column**. For more information, see **Column Properties** on page 25.

The center area of the main pane displays column properties. The header contains the property names, and each row contains the property values for one column.

To locate the row for a specific column, in the **Search Column Names** field, type the column name, and then click ⬆️ or ⬇️ to browse through the search results. The Schema Editor highlights any rows where the column name contains the search term.

The exact column properties that are available in the main pane differ depending on the specific NoSQL data store that you are working with, but typically include the following:

- **SQL Name**: The column name that is used in SQL operations.
- **Source Name**: The column name that is used in the NoSQL data store.
- **SQL Type**: The SQL data type that the column is mapped to for SQL operations.
- **Source Type**: The NoSQL data type that the column is mapped to in the NoSQL data store.

For more information about specific column properties, see **Column Properties** on page 25.
To edit column properties:

1. In the Design View, in the left pane, select the table containing the properties that you want to edit.
2. In the main pane, click the cell containing the value that you want to edit and then do one of the following:
   - If the cell is an editable field, then type your new value and press ENTER.
   - If the cell is a drop-down list, then select a value from the list.

Column Properties

General

The following table lists and describes the column properties that are available in the Design View for all NoSQL data stores. The properties are listed in alphabetical order. Only writable properties can be edited.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Writable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column #</td>
<td>A unique identifier for the column in the schema definition.</td>
<td>No</td>
</tr>
<tr>
<td>Foreign Key Column</td>
<td>A Boolean indicating whether the column contains foreign key values.</td>
<td>No</td>
</tr>
<tr>
<td>Hide Column</td>
<td>A Boolean indicating whether the column displays in ODBC and JDBC applications.</td>
<td>Yes</td>
</tr>
<tr>
<td>Primary Key Column</td>
<td>A Boolean indicating whether the column contains primary key values.</td>
<td>No</td>
</tr>
<tr>
<td>Source Name</td>
<td>The column name that is used in the NoSQL data store.</td>
<td>Yes</td>
</tr>
<tr>
<td>Source Type</td>
<td>The NoSQL data type that the column is mapped to in the NoSQL data store.</td>
<td>Yes</td>
</tr>
<tr>
<td>SQL Name</td>
<td>The column name that is used in SQL operations.</td>
<td>Yes</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Writable</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>SQL Type</td>
<td>The SQL data type that the column is mapped to for SQL operations.</td>
<td>Yes</td>
</tr>
<tr>
<td>User Generated Column</td>
<td>A Boolean indicating whether the column was manually created through the Add Column or Move to Parent options in the Schema Editor. User-generated columns can be deleted.</td>
<td>No</td>
</tr>
</tbody>
</table>

**Note:**
The Move to Parent option is not supported for Couchbase and DocumentDB.

**Couchbase**

The following is an additional column property that is available when you work with Couchbase data stores. Writable properties can be edited.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Writable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>The precision for fractional seconds in TIME and TIMESTAMP data. This value indicates the number of digits that exist to the right of the decimal in the seconds component of TIME and TIMESTAMP data. This property does not affect data that is not of type TIME or TIMESTAMP.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**DocumentDB**

The following table lists and describes additional column properties that are available when you work with DocumentDB data stores. The properties are listed in alphabetical order. Only writable properties can be edited.
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Writable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Length</td>
<td>The maximum number of characters that the column can contain.</td>
<td>Yes</td>
</tr>
<tr>
<td>Precision</td>
<td>The number of digits in numeric data. This property does not affect non-numeric data directly, but it is used if the data is converted to a numeric type.</td>
<td>Yes</td>
</tr>
<tr>
<td>Scale</td>
<td>The number of digits to the right of the decimal in numeric data. This property does not affect non-numeric data directly, but it is used if the data is converted to a numeric type.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**DynamoDB**

The following table lists and describes additional column properties that are available when you work with DynamoDB data stores. The properties are listed in alphabetical order. Only writable properties can be edited.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Writable</th>
</tr>
</thead>
</table>
| Column Type   | An identifier for the type of data that the column contains:  
  - Data indicates that the column contains raw data values.  
  - Index indicates that the column contains index values for lists.  
  - Set indicates that the column contains data values from a set. | Yes |
| Length        | The maximum number of characters that the column can contain. | Yes |
| Precision     | The number of digits in numeric data. This property does not affect non-numeric data directly, but it is used if the data is converted to a numeric type. | Yes |
# Property Name | Description | Writable
---|---|---
Scale | The number of digits to the right of the decimal in numeric data. This property does not affect non-numeric data directly, but it is used if the data is converted to a numeric type. | Yes
Write Permission | A value specifying whether the data in the column is writable:  
- **Read Only** indicates that the data in the column cannot be changed.  
- **Write** indicates that the data in the column can be changed using DML statements. | Yes

## MongoDB

The following table lists and describes additional column properties that are available when you work with MongoDB data stores. The properties are listed in alphabetical order. Only writable properties can be edited.

| Property Name | Description | Writable |
---|---|---|
Column Size | The maximum number of characters that the column can contain. | Yes
Key Type | An identifier for the type of data that the column contains:  
- **DATA_COLUMN** indicates that the column contains raw data values.  
- **ID_COLUMN** indicates that the column contains _id values, which are primary key values used specifically in MongoDB.  
- **INDEX_COLUMN** indicates that the column contains index values for arrays. | No
Nullable | A Boolean indicating whether the column can contain NULL values. | Yes
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Writable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>The number of digits in numeric data. This property does not affect non-numeric data directly, but it is used if the data is converted to a numeric type.</td>
<td>Yes</td>
</tr>
<tr>
<td>Scale</td>
<td>The number of digits to the right of the decimal in numeric data. This property does not affect non-numeric data directly, but it is used if the data is converted to a numeric type.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Adding and Deleting Columns

**Note:**
This feature is not currently available for DocumentDB.

In some cases, due to the configuration settings used during the sampling process, the driver might generate a schema definition that does not contain all the desired table columns. To correct the schema definition, you can use the Schema Editor to add or delete columns. However, you can only delete columns that have the User Generated Column property set to true.

**To add a column:**

1. In the Design View, in the left pane, select the table that you want to add a column to.
2. In the main pane, click **Add Column**. A row representing the new column is added to the main pane.
3. Define properties for the new column. For detailed instructions, see **Editing Column Properties** on page 24.

**To delete a column:**

1. In the Design View, in the left pane, select the table that you want to delete a column from.
2. In the main pane, specify the column that you want to delete by selecting the corresponding check box in the **Selected** column.
3. Click **Delete Column**.

The selected column is deleted from the schema definition.
Creating a Columnar View of Array Elements

Note:
- This feature is not supported for Couchbase and DocumentDB.
- The Schema Editor supports columnar views for all forms of array data structures. For example, in addition to creating columnar views of arrays from MongoDB, you can also create columnar views of lists, maps, and sets from DynamoDB.

To support arrays, the Schema Editor creates virtual tables. Each virtual table is a child object of the table that contains the array, and each row in a virtual table contains an array element. As an alternative, you can create a columnar view of the array elements in the parent table. For an example of how a virtual table representation of an array compares to a columnar view, see Example of Virtual Tables vs. Columnar Views on page 30.

To create a columnar view of array elements:

1. In the Design View, in the left pane, select the virtual table containing the array elements for which you want to create a columnar view.
2. In the main pane, select the virtual table column that contains the array elements.
3. Click Move to Parent.
4. When prompted, in the Number of Array Elements to Show in Parent field, type the number of array elements for which you want to create a columnar view, and then click OK.

In the parent table, the Schema Editor creates columns that contain the array elements.

Example of Virtual Tables vs. Columnar Views

Note:
- This feature is not supported for Couchbase and DocumentDB.
- The following example uses arrays from MongoDB. Columnar views are supported for all forms of array data structures, so the concepts in this example are also applicable to other array data structures such as lists, maps, and sets from DynamoDB.

Because arrays are not natively supported in standard ODBC and JDBC applications, the Schema Editor renormalizes arrays by mapping them to a supported format.

Array data can be renormalized into virtual tables or columnar views. In virtual tables, each row contains one array element. In columnar views, each column contains one...
array element. These columns are not created in the database. Rather, they are generated at query time based on the data structure specified in the schema definition.

The following example shows a comparison between a virtual table of arrays and a columnar view of the same arrays.

Consider the following array data in a collection named testscores:

```json
{
    "grades": [
        { "course": "BIOL100", "grade": "A", "score": 96 },
        { "course": "CHEM100", "grade": "A", "score": 88 },
        { "course": "PHYS100", "grade": "B", "score": 75 },
        { "course": "MATH100", "grade": "B", "score": 72 },
        { "course": "ENGL100", "grade": "C", "score": 60 }
    ],
    "name": "John Doe",
    "student_id": "11875445"
}
```

```json
{
    "grades": [
        { "course": "BIOL100", "grade": "A", "score": 86},
        { "course": "CHEM100", "grade": "B", "score": 78},
        { "course": "PHYS100", "grade": "A", "score": 85},
        { "course": "MATH100", "grade": "A", "score": 90 }
    ],
    "name": "Jane Doe",
    "student_id": "11812340"
}
```

By default, your Schema Editor driver renormalizes arrays into virtual tables. The following two tables show the base table and virtual table that would be created in the schema definition.

<table>
<thead>
<tr>
<th>_id</th>
<th>name</th>
<th>student_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>John Doe</td>
<td>11875445</td>
</tr>
<tr>
<td>2222</td>
<td>Jane Doe</td>
<td>11812340</td>
</tr>
</tbody>
</table>
To create a columnar view of the score array, you would select the score column, click **Move to Parent**, and then specify 5 as the number of elements to move. The Schema Editor creates columns in the base table containing up to 5 elements from each score array. The following table shows what the base table looks like after the columnar view is created. The virtual table remains unchanged.

<table>
<thead>
<tr>
<th>_id</th>
<th>name</th>
<th>student_id</th>
<th>score_0</th>
<th>score_1</th>
<th>score_2</th>
<th>score_3</th>
<th>score_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>John Doe</td>
<td>11875445</td>
<td>96</td>
<td>88</td>
<td>75</td>
<td>72</td>
<td>60</td>
</tr>
<tr>
<td>2222</td>
<td>Jane Doe</td>
<td>11812340</td>
<td>86</td>
<td>78</td>
<td>85</td>
<td>90</td>
<td></td>
</tr>
</tbody>
</table>

www.simba.com
The preview feature shows you the relational tables that are created as a result of the data mapping from the schema definition. Preview your data to see how a table or virtual table would appear in ODBC and JDBC applications.

**Note:**

When working with Couchbase data, make sure that the driver is configured to execute queries in SQL instead of N1QL. The Schema Editor uses a SQL query to determine which data to preview.

For information about configuring the driver to use SQL or N1QL, see the descriptions of "Query Mode" or "QueryMode" in the Installation and Configuration Guide for your Couchbase driver.

**To preview data:**

1. In the Design View, in the left pane, select the table that you want to preview.
2. In the main pane, click **Preview**.

The bottom area of the main pane displays up to 10 rows showing how the table appears in ODBC and JDBC applications. The configuration of the database and the driver determine which rows are selected to be included in the preview.
Saving a Schema Definition

After creating or modifying a schema definition, you can preserve your changes by saving the schema definition in a local JSON file or in the NoSQL data store that the schema is used for.

After saving your schema definition, you can configure your driver to use the schema definition when connecting to and working with the data store. For detailed information, see the Installation and Configuration Guide for your driver.

To save a schema definition:

- Choose one:
  - To save the schema definition in a local JSON file, select File > Save, specify a name and location for the file, and then click Save.
  
  **Important:**
  If the data store already contains a schema definition, that schema definition is permanently overwritten when you select Publish Schema Map.

  - Or, to save the schema definition in the data store that you are connected to, select Connection > Publish Schema Map.

  **Note:**
  This option is not available for DocumentDB.

If you are not connected to a data store, the Schema Editor prompts you to connect to a data store. For information about how to specify connection information, see Connecting to a Data Store on page 35.
If you are not already connected to a data store when you try to sample data or open a schema definition from a data store, the Schema Editor prompts you to provide your connection information and connect to the data store.

**To connect to a data store through ODBC:**

1. In the connection prompt, do one of the following:
   - Select *DSN List* and then select a DSN from the drop-down list.
   - Or, select *Connection String* and then type an ODBC connection string in the field.

   ![Note:](Note)
   For more information about connection string syntax, see the Installation and Configuration Guide for your driver.

2. Click *Connect*.

**To connect to a data store through JDBC:**

1. In the connection prompt, in the *Connection URI* field, type a JDBC connection URL.

   ![Note:](Note)
   For more information about connection URL syntax, see the Installation and Configuration Guide for your driver.

2. Click *Connect*. 
Configuring Logging

To help troubleshoot issues, you can configure the Schema Editor to run with logging enabled.

⚠️ Important:
Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

Set the **Log Level** option to enable logging and specify the amount of detail included in log files. The following table lists the logging levels available in the Schema Editor, in order from least verbose to most verbose.

<table>
<thead>
<tr>
<th>Log Level Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>Disables all logging.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Logs error events that occur while the Schema Editor is running.</td>
</tr>
<tr>
<td>INFO</td>
<td>Logs general information that describes the progress of the Schema Editor.</td>
</tr>
<tr>
<td>TRACE</td>
<td>Logs all Schema Editor activity.</td>
</tr>
</tbody>
</table>

**To enable logging:**

1. From the menu bar at the top, select **Help > Settings**.
2. From the **Log Level** drop-down list, select the desired level of information to include in log files.
3. To include your connection information in the log, select the **Log Connection String** check box.
4. In the **Log Location** field, specify a full path for the log file. The log file must be saved in `.log` format.
5. Click **OK**.
6. Restart the Schema Editor to make sure that the new settings take effect.

The Schema Editor produces a log file in the directory specified in the Log Location field.
To disable logging:

1. From the menu bar at the top, select **Help > Settings**.
2. From the **Log Level** drop-down list, select **OFF**.
3. Click **OK**.
4. Restart Schema Editor to make sure that the new settings take effect.
Third-Party Trademarks

Debian is a trademark or registered trademark of Software in the Public Interest, Inc. or its subsidiaries in Canada, United States and/or other countries.

Linux is the registered trademark of Linus Torvalds in Canada, United States and/or other countries.

Mac, macOS, Mac OS, and OS X are trademarks or registered trademarks of Apple, Inc. or its subsidiaries in Canada, United States and/or other countries.

Microsoft, Windows, Windows Azure, Azure, Windows Server, Windows Vista, and the Windows start button are trademarks or registered trademarks of Microsoft Corporation or its subsidiaries in Canada, United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Red Hat, Red Hat Enterprise Linux, and CentOS are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in Canada, United States and/or other countries.

SUSE is a trademark or registered trademark of SUSE LLC or its subsidiaries in Canada, United States and/or other countries.

Ubuntu is a trademark or registered trademark of Canonical Ltd. or its subsidiaries in Canada, United States and/or other countries.

Amazon DynamoDB, Amazon, and DynamoDB are trademarks or registered trademarks of Amazon Web Services, Inc. or its subsidiaries in Canada, United States and/or other countries.

Couchbase and Couchbase Server are trademarks or registered trademarks of Couchbase or its subsidiaries in Canada, United States and/or other countries.

MongoDB and Mongo are trademarks or registered trademarks of MongoDB, Inc. or its subsidiaries in Canada, the United States and/or other countries.

All other trademarks are trademarks of their respective owners.
Third-Party Licenses

The licenses for the third-party libraries that are included in this product are listed below.

RapidJSON License

Tencent is pleased to support the open source community by making RapidJSON available.

Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip. All rights reserved.

If you have downloaded a copy of the RapidJSON binary from Tencent, please note that the RapidJSON binary is licensed under the MIT License.

If you have downloaded a copy of the RapidJSON source code from Tencent, please note that RapidJSON source code is licensed under the MIT License, except for the third-party components listed below which are subject to different license terms. Your integration of RapidJSON into your own projects may require compliance with the MIT License, as well as the other licenses applicable to the third-party components included within RapidJSON.

A copy of the MIT License is included in this file.

Other dependencies and licenses:

Open Source Software Licensed Under the BSD License:

The msinttypes r29
Copyright (c) 2006-2013 Alexander Chemeris
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Open Source Software Licensed Under the JSON License:

json.org
Copyright (c) 2002 JSON.org
All Rights Reserved.

JSON_checker
Copyright (c) 2002 JSON.org
All Rights Reserved.

Terms of the JSON License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Terms of the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX

ORACLE AMERICA, INC. ("ORACLE"), FOR AND ON BEHALF OF ITSELF AND ITS SUBSIDIARIES AND AFFILIATES UNDER COMMON CONTROL, IS WILLING TO LICENSE THE SOFTWARE TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY SELECTING THE "ACCEPT LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND/OR BY USING THE SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THE TERMS AND AGREE TO THEM. IF YOU ARE AGREING TO THESE TERMS ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE LEGAL AUTHORITY TO BIND THE LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO BE BOUND BY THE TERMS, THEN SELECT THE "DECLINE LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND YOU MUST NOT USE THE SOFTWARE ON THIS SITE OR ANY OTHER MEDIA ON WHICH THE SOFTWARE IS CONTAINED.

1. DEFINITIONS. "Software" means the software identified above in binary form that you selected for download, install or use (in the version You selected for
download, install or use) from Oracle or its authorized licensees, any other
machine readable materials (including, but not limited to, libraries, source files,
header files, and data files), any updates or error corrections provided by Oracle,
and any user manuals, programming guides and other documentation provided
to you by Oracle under this Agreement. "General Purpose Desktop Computers
and Servers" means computers, including desktop and laptop computers, or
servers, used for general computing functions under end user control (such as
but not specifically limited to email, general purpose Internet browsing, and office
suite productivity tools). The use of Software in systems and solutions that
provide dedicated functionality (other than as mentioned above) or designed for
use in embedded or function-specific software applications, for example but not
limited to: Software embedded in or bundled with industrial control systems,
wireless mobile telephones, wireless handheld devices, kiosks, TV/STB, Blu-ray
Disc devices, telematics and network control switching equipment, printers and
storage management systems, and other related systems are excluded from this
definition and not licensed under this Agreement. "Programs" means (a) Java
technology applets and applications intended to run on the Java Platform,
Standard Edition platform on Java-enabled General Purpose Desktop
Computers and Servers; and (b) JavaFX technology applications intended to run
on the JavaFX Runtime on JavaFX-enabled General Purpose Desktop
Computers and Servers. "Commercial Features" means those features identified
in Table 1-1 (Commercial Features In Java SE Product Editions) of the Java SE
documentation accessible at
"README File" means the README file for the Software accessible at

2. LICENSE TO USE. Subject to the terms and conditions of this Agreement
including, but not limited to, the Java Technology Restrictions of the
Supplemental License Terms, Oracle grants you a non-exclusive, non-
transferable, limited license without license fees to reproduce and use internally
the Software complete and unmodified for the sole purpose of running Programs.
THE LICENSE SET FORTH IN THIS SECTION 2 DOES NOT EXTEND TO
THE COMMERCIAL FEATURES. YOUR RIGHTS AND OBLIGATIONS
RELATED TO THE COMMERCIAL FEATURES ARE AS SET FORTH IN THE
SUPPLEMENTAL TERMS ALONG WITH ADDITIONAL LICENSES FOR
DEVELOPERS AND PUBLISHERS.

3. RESTRICTIONS. Software is copyrighted. Title to Software and all associated
intellectual property rights is retained by Oracle and/or its licensors. Unless
enforcement is prohibited by applicable law, you may not modify, decompile, or
reverse engineer Software. You acknowledge that the Software is developed for
general use in a variety of information management applications; it is not
developed or intended for use in any inherently dangerous applications,
including applications that may create a risk of personal injury. If you use the
Software in dangerous applications, then you shall be responsible to take all
appropriate fail-safe, backup, redundancy, and other measures to ensure its safe
use. Oracle disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Oracle or its licensors is granted under this Agreement. Additional restrictions for developers and/or publishers licenses are set forth in the Supplemental License Terms.

4. DISCLAIMER OF WARRANTY. THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. ORACLE FURTHER DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

5. LIMITATION OF LIABILITY. IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR DATA USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, EVEN IF ORACLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. ORACLE'S ENTIRE LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED ONE THOUSAND DOLLARS (U.S. $1,000).

6. TERMINATION. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Oracle if you fail to comply with any provision of this Agreement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right. Upon termination, you must destroy all copies of Software.

7. EXPORT REGULATIONS. You agree that U.S. export control laws and other applicable export and import laws govern your use of the Software, including technical data; additional information can be found on Oracle's Global Trade Compliance web site (http://www.oracle.com/us/products/export). You agree that neither the Software nor any direct product thereof will be exported, directly, or indirectly, in violation of these laws, or will be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation.

8. TRADEMARKS AND LOGOS. You acknowledge and agree as between you and Oracle that Oracle owns the ORACLE and JAVA trademarks and all ORACLE- and JAVA-related trademarks, service marks, logos and other brand designations ("Oracle Marks"), and you agree to comply with the Third Party Usage Guidelines for Oracle Trademarks currently located at http://www.oracle.com/us/legal/third-party-trademarks/index.html. Any use you make of the Oracle Marks inures to Oracle's benefit.

9. U.S. GOVERNMENT LICENSE RIGHTS. If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation shall be only those set forth in this Agreement.
10. GOVERNING LAW. This agreement is governed by the substantive and procedural laws of California. You and Oracle agree to submit to the exclusive jurisdiction of, and venue in, the courts of San Francisco, or Santa Clara counties in California in any dispute arising out of or relating to this agreement.

11. SEVERABILITY. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

12. INTEGRATION. This Agreement is the entire agreement between you and Oracle relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

SUPPLEMENTAL LICENSE TERMS

These Supplemental License Terms add to or modify the terms of the Binary Code License Agreement. Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Binary Code License Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Binary Code License Agreement, or in any license contained within the Software.

A. COMMERCIAL FEATURES. You may not use the Commercial Features for running Programs, Java applets or applications in your internal business operations or for any commercial or production purpose, or for any purpose other than as set forth in Sections B, C, D and E of these Supplemental Terms. If You want to use the Commercial Features for any purpose other than as permitted in this Agreement, You must obtain a separate license from Oracle.

B. SOFTWARE INTERNAL USE FOR DEVELOPMENT LICENSE GRANT. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the README File incorporated herein by reference, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Oracle grants you a non-exclusive, non-transferable, limited license without fees to reproduce internally and use internally the Software complete and unmodified for the purpose of designing, developing, and testing your Programs.

C. LICENSE TO DISTRIBUTE SOFTWARE. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the README File, including, but not limited to the Java Technology Restrictions and Limitations on Redistribution of these Supplemental Terms, Oracle grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute the Software, provided that (i) you distribute the Software complete and unmodified
and only bundled as part of, and for the sole purpose of running, your Programs, (ii) the Programs add significant and primary functionality to the Software, (iii) you do not distribute additional software intended to replace any component(s) of the Software, (iv) you do not remove or alter any proprietary legends or notices contained in the Software, (v) you only distribute the Software subject to a license agreement that: (a) is a complete, unmodified reproduction of this Agreement; or (b) protects Oracle's interests consistent with the terms contained in this Agreement and that includes the notice set forth in Section H, and (vi) you agree to defend and indemnify Oracle and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software. The license set forth in this Section C does not extend to the Software identified in Section G.

D. LICENSE TO DISTRIBUTE REDISTRIBUTABLES. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the README File, including but not limited to the Java Technology Restrictions and Limitations on Redistribution of these Supplemental Terms, Oracle grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute those files specifically identified as redistributable in the README File ("Redistributables") provided that: (i) you distribute the Redistributables complete and unmodified, and only bundled as part of Programs, (ii) the Programs add significant and primary functionality to the Redistributables, (iii) you do not distribute additional software intended to supersede any component(s) of the Redistributables (unless otherwise specified in the applicable README File), (iv) you do not remove or alter any proprietary legends or notices contained in or on the Redistributables, (v) you only distribute the Redistributables pursuant to a license agreement that: (a) is a complete, unmodified reproduction of this Agreement; or (b) protects Oracle's interests consistent with the terms contained in the Agreement and includes the notice set forth in Section H, (vi) you agree to defend and indemnify Oracle and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software. The license set forth in this Section D does not extend to the Software identified in Section G.

E. DISTRIBUTION BY PUBLISHERS. This section pertains to your distribution of the JavaTM SE Development Kit Software ("JDK") with your printed book or magazine (as those terms are commonly used in the industry) relating to Java technology ("Publication"). Subject to and conditioned upon your compliance with the restrictions and obligations contained in the Agreement, Oracle hereby grants to you a non-exclusive, nontransferable limited right to reproduce complete and unmodified copies of the JDK on electronic media (the "Media") for the sole purpose of inclusion and distribution with your Publication(s), subject to the following terms: (i) You may not distribute the JDK on a stand-alone basis; it
must be distributed with your Publication(s); (ii) You are responsible for downloading the JDK from the applicable Oracle web site; (iii) You must refer to the JDK as JavaTM SE Development Kit; (iv) The JDK must be reproduced in its entirety and without any modification whatsoever (including with respect to all proprietary notices) and distributed with your Publication subject to a license agreement that is a complete, unmodified reproduction of this Agreement; (v) The Media label shall include the following information: “Copyright [YEAR], Oracle America, Inc. All rights reserved. Use is subject to license terms. ORACLE and JAVA trademarks and all ORACLE- and JAVA-related trademarks, service marks, logos and other brand designations are trademarks or registered trademarks of Oracle in the U.S. and other countries.” [YEAR] is the year of Oracle's release of the Software; the year information can typically be found in the Software’s “About” box or screen. This information must be placed on the Media label in such a manner as to only apply to the JDK; (vi) You must clearly identify the JDK as Oracle's product on the Media holder or Media label, and you may not state or imply that Oracle is responsible for any third-party software contained on the Media; (vii) You may not include any third party software on the Media which is intended to be a replacement or substitute for the JDK; (viii) You agree to defend and indemnify Oracle and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys’ fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of the JDK and/or the Publication; and (ix) You shall provide Oracle with a written notice for each Publication; such notice shall include the following information: (1) title of Publication, (2) author(s), (3) date of Publication, and (4) ISBN or ISSN numbers. Such notice shall be sent to Oracle America, Inc., 500 Oracle Parkway, Redwood Shores, California 94065 U.S.A, Attention: General Counsel.

F. JAVA TECHNOLOGY RESTRICTIONS. You may not create, modify, or change the behavior of, or authorize your licensees to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun", “oracle” or similar convention as specified by Oracle in any naming convention designation.

G. LIMITATIONS ON REDISTRIBUTION. You may not redistribute or otherwise transfer patches, bug fixes or updates made available by Oracle through Oracle Premier Support, including those made available under Oracle's Java SE Support program.

H. COMMERCIAL FEATURES NOTICE. For purpose of complying with Supplemental Term Section C.(v)(b) and D.(v)(b), your license agreement shall include the following notice, where the notice is displayed in a manner that anyone using the Software will see the notice:

Use of the Commercial Features for any commercial or production purpose requires a separate license from Oracle. “Commercial Features” means those features identified Table 1-1 (Commercial Features In Java SE Product Editions)
of the Java SE documentation accessible at
http://www.oracle.com/technetwork/java/javase/documentation/index.html

I. SOURCE CODE. Software may contain source code that, unless expressly
licensed for other purposes, is provided solely for reference purposes pursuant to
the terms of this Agreement. Source code may not be redistributed unless
expressly provided for in this Agreement.

J. THIRD PARTY CODE. Additional copyright notices and license terms applicable
to portions of the Software are set forth in the THIRDPARTYLICENSEREADME
file accessible at
addition to any terms and conditions of any third party opensource/freeware
license identified in the THIRDPARTYLICENSEREADME file, the disclaimer of
warranty and limitation of liability provisions in paragraphs 4 and 5 of the Binary
Code License Agreement shall apply to all Software in this distribution.

K. TERMINATION FOR INFRINGEMENT. Either party may terminate this
Agreement immediately should any Software become, or in either party's opinion
be likely to become, the subject of a claim of infringement of any intellectual
property right.

L. INSTALLATION AND AUTO-UPDATE. The Software's installation and auto-
update processes transmit a limited amount of data to Oracle (or its service
provider) about those specific processes to help Oracle understand and optimize
them. Oracle does not associate the data with personally identifiable information.
You can find more information about the data Oracle collects as a result of your
Software download at

For inquiries please contact: Oracle America, Inc., 500 Oracle Parkway,
Redwood Shores, California 94065, USA.

Last updated 02 April 2013

Apache License, Version 2.0

The following notice is included in compliance with the Apache License, Version 2.0
and is applicable to all software licensed under the Apache License, Version 2.0.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing
lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least
one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software that is licensed under the Apache License, Version 2.0 (listed below):

**Apache Log4j**
Copyright 1999-2014 Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Dumbster SMTP test server
Copyright 2004 Jason Paul Kitchen
TypeUtil.java
Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

Jackson
Copyright © 2009-2011 FasterXML, LLC

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.