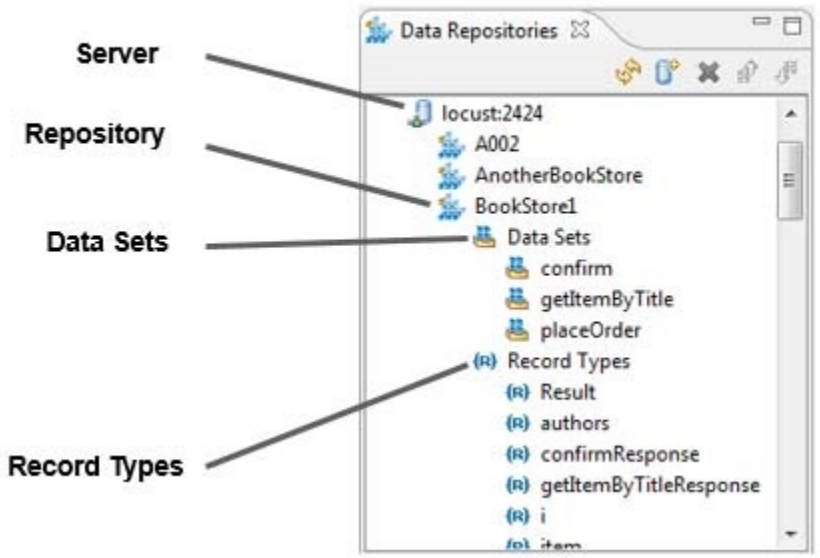
Viewing and Modifying the Repository Structure and Contents

This topic explains how you can use the Data Repositories view and the related editor to review and modify the repository structure as well as update the included records.

Sections include:

Defining the Structure

The Data Repositories view is designed to help you review and extend the available servers, repositories, data sets and record types. From this view, you can add servers and repositories (as described in earlier topics) as well as define data sets and record types.



Adding a Data Set

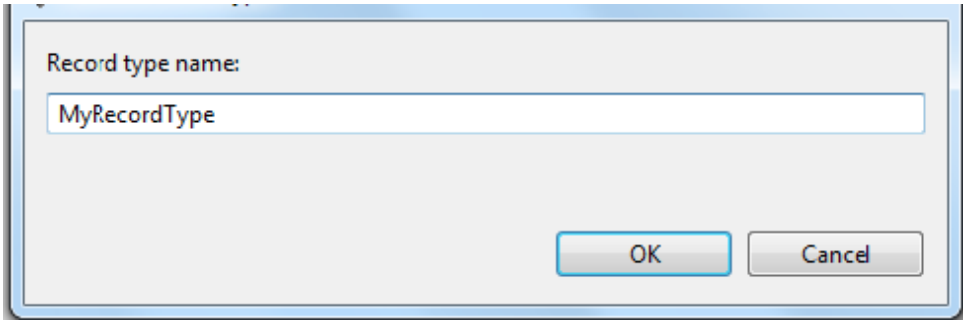
The data on each repository server can be logically grouped into any number of subsets, which are called data sets. All data sets draw from the library of record types defined for the given repository. To add a data set:

1. Right-click the repository's **Data Sets** node, then choose **Create Data Set**.
2. In the dialog that opens, enter a data set name.

Adding a New Record Type

At least one record type must be defined for each data set. To add a record type:

1. Right-click the repository's **Record Types** node, then choose **Create Record Type**.
2. In the dialog that opens, enter a record type name. You can enter a new name, or use the name of an existing record type that is already defined for this repository (and shown in the Data Repositories view).



Adding Record Types

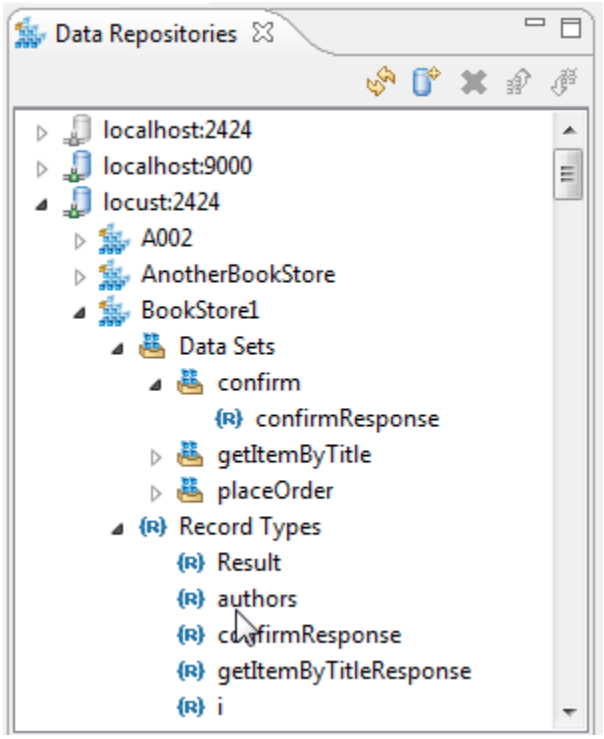
You can define any number of record types that will be describe the structure of records on this repository. To add a record type:

1. Right-click the **Record Types** Data Repositories view node, then choose **Create Record Type**.
2. Specify a name for the record type.

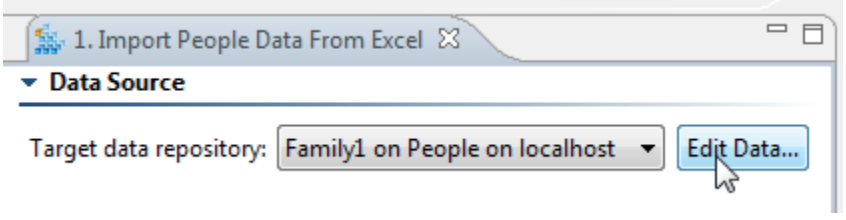
Navigating through the Repository Records

To start navigating through the repository records, do one of the following:

- Double-click the appropriate data set or record type node in the Data Repositories view.

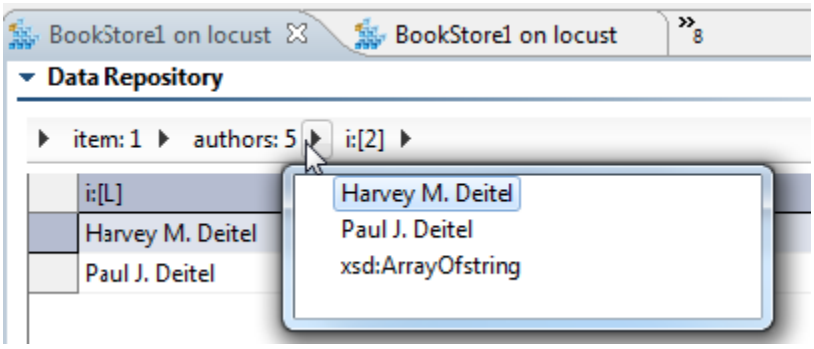


- Click the **Edit Data** button at the top right corner of the Data Repository tool configuration panel (next to where your Target data repository is selected).

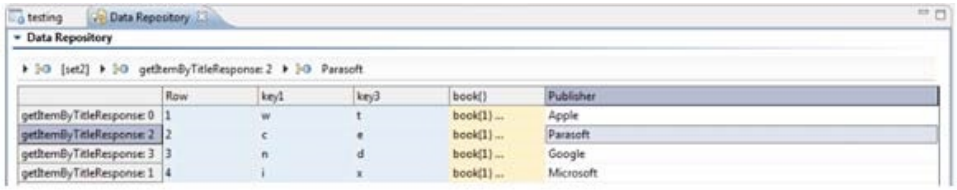


Once the Data Editor is open, you can navigate through the available records by using the breadcrumbs at the top of the editor, or by drilling

down into the table itself.



Any column marked with yellow is a column that you can drill down into (yellow is used to represent complex columns such as primitive lists or hierarchical record lists). To drill down, you can click or press the space bar.



As you navigate through the records, note that the following colors and symbols have special meaning:

Color	Meaning
Yellow	A complex column that you can drill-down into.
White	Stores a primitive value that you can edit here.
Purple	Determines the row number, which is used by SOAtest in determining how to iterate over data source row. Not applicable for Virtualize.

Symbol	Meaning
[R]	Record list
[L]	Literal / primitive list

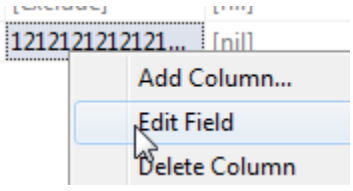
Editing and Extending the Repository

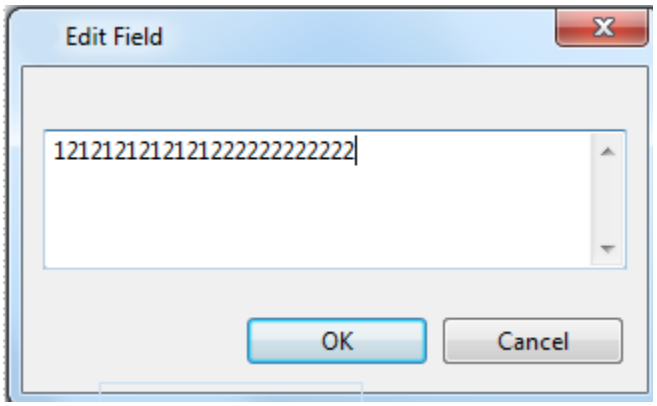
From the Data Editor, you can edit record values, rows, and columns.

Editing Record Values

You can edit the record values in any column with a white or light-purple background. Cut/copy/paste is supported. Changes are saved in real time; you do not need to explicitly save your changes.

Note that if a field contains large data, you can double-click that field to open the value in a editor dialog. Alternatively, you can right-click that field and choose **Edit** to open that same editor dialog.

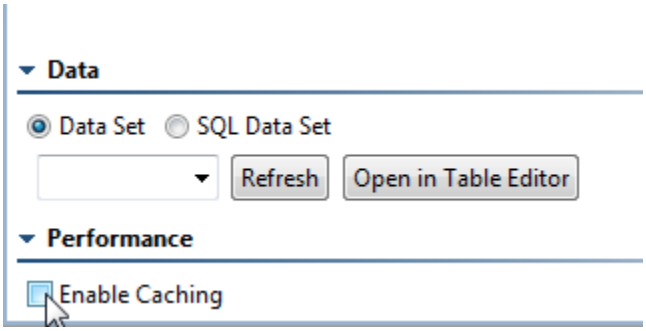




Controlling How Repository Changes are Reflected in Deployed Virtual Assets

Data source caching settings control whether repository data changes are immediately reflected in deployed virtual assets.

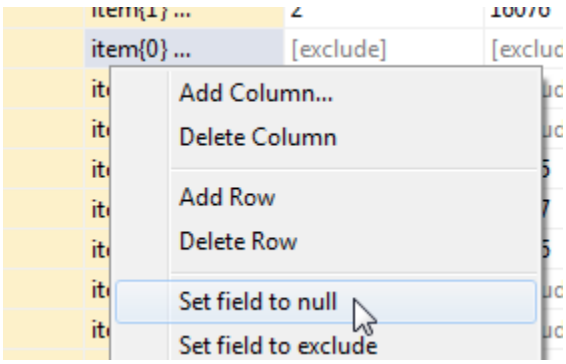
If the data source's **Enable caching** option is disabled (the default), updates to the repository data will be immediately reflected in any deployed virtual assets which use that data.



If you enable caching, virtual assets will need to be redeployed in order for the repository data changes to take effect at runtime. Enabling caching for load testing is recommended for performance optimization.

Setting Values to Null or Exclude

If you want to set a value to nil or exclude (so it will not appear in the message when an element is populated from this data source), use the available right-click options (**Set field to null, Set field to exclude**). Or, you can enter special string values: `[null]` will be treated as a value that has been set to null, and `[exclude]` will be treated as a value that has been set to `exclude`.



Adding Rows

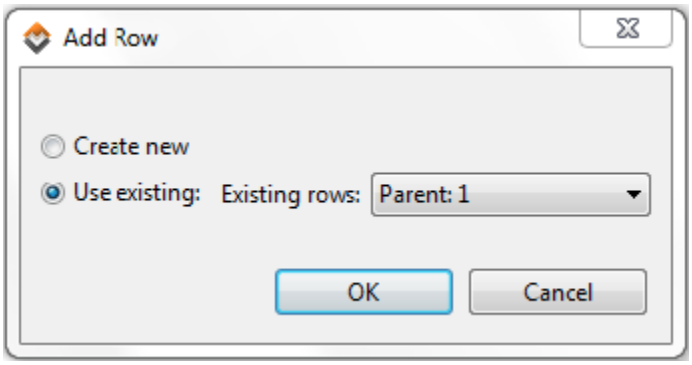
When adding rows, you can either create a new row or add a reference to an existing row. If you add a reference to an existing row, you essentially link this new row to the original row; if the original row is updated, those changes will be propagated across all referencing rows.

To create a new row:

1. Right-click the table, then choose **Add Row**.
2. In the dialog that opens, select **Add New**, then click **OK**.

To create a reference to an existing row:

1. Right-click the table, then choose **Add Row**.
2. In the dialog that opens, select **Use Existing**, specify which existing row you want to use, then click **OK**.



Note that a text field will replace the drop-down list if a large number of rows are available; in this case, you can use the autocomplete functionality to specify the record ID (this is what's shown in the row headers and breadcrumb).

New rows are always added at the bottom of the table.

Adding Columns

Columns can have the following types:

- **Primitive:** A primitive type. Strings, numbers and booleans are all represented (and treated) as a single primitive type in the repository. When Form Input is parameterized with a primitive field, it will use the value in the type that the schema intended for it.
- **Primitive list:** An array of primitive types. For example, this might be used to contain a list of aliases that a person uses (Jonathan, Jon, Jonny, etc.). These are noted with [L].
- **Record list:** A series of complex records that are hierarchical and have multiple fields/columns. For example, this might contain data about a person's children—complex records that have a set of columns such as first name, last name, birth date, etc.). These are noted with [R].

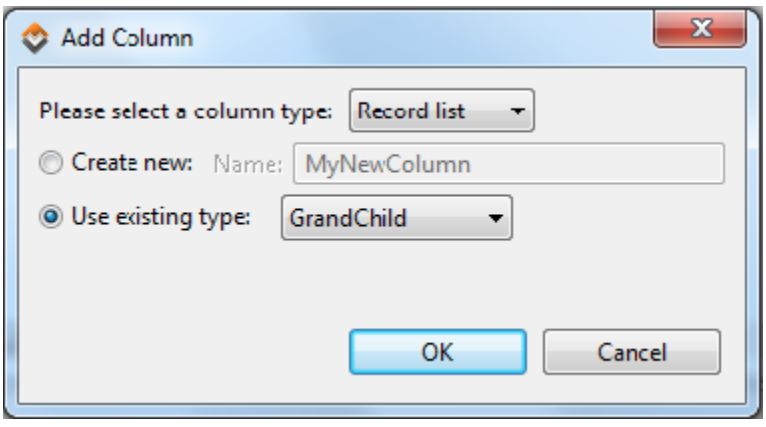
You can add "regular" columns, as well as "key columns": special columns that are used for Virtualize responder correlations. Key columns must be of primitive type.

Note that the specially-marked (in light purple) row column determines how SOAtest iterates through the data source rows. This column is required if you will be using this repository data in SOAtest.

Regular Columns

To create a new regular column:

1. Right-click the table, then choose **Add Column**.
2. In the dialog that opens, select a column type, either specify a new name or an existing type (for record lists), then click **OK**.



New columns are always added at the right of the table.

Key Columns

To create a new key column:

1. Right-click the table, then choose **Add Key Column**.
2. In the dialog that opens, specify a name for the new key column, then click **OK.**

Specifying Record Identities

A record identity is the subset of a record type's fields which uniquely identifies that record type. For example, a bank customer record type might have 15 different fields, but its identity might use only social security number and account number.

Identities enable you to correlate imported traffic data with existing data repository records. This matching helps determine when existing records can be reused and when new ones need to be created. Maximizing reuse simplifies data updating and management. If a record is reused across 1000 ancestors you can update it once and the changes will be automatically propagated to all related records. Otherwise, you would need update all 1000 records individually.

Each set of identity field values must be unique. For example, the following is valid:

Identity Column 1	Identity Column 2	Identity Column 3
1	2	3
1	2	4
1	3	4
1	3	5

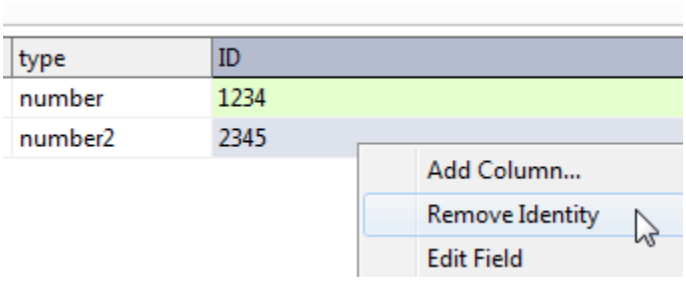
The following is not valid:

Identity Column 1	Identity Column 2	Identity Column 3
1	2	3
1	**2**	**4**
1	3	4
1	**2**	**4**

To mark an identity:

1. Select the fields you want to use as the identity.
2. Right-click the selection, then choose **Create Identity**.

The selected field(s) will then be marked in green. If you later decide to change the identity, remove it (right-click> **Remove Identity**), then add a new identity.



Note that fields themselves will not be removed; only the identity status will be impacted. Identity fields cannot be deleted.