

SQL Responder Action

This topic explains how to configure and apply the SQL Responder Action tool, which allows you to model complex interdependent environment scenarios by connecting multiple virtual assets. More specifically, it enables you to have a Message Responder update values in a virtual database that is represented by a .CSV-driven SQL Responder.

Sections include:

- [Understanding the SQL Responder Action Tool](#)
- [Configuring the SQL Responder Action Tool](#)
- [Resetting Data Modified by the Tool](#)

Understanding the SQL Responder Action Tool

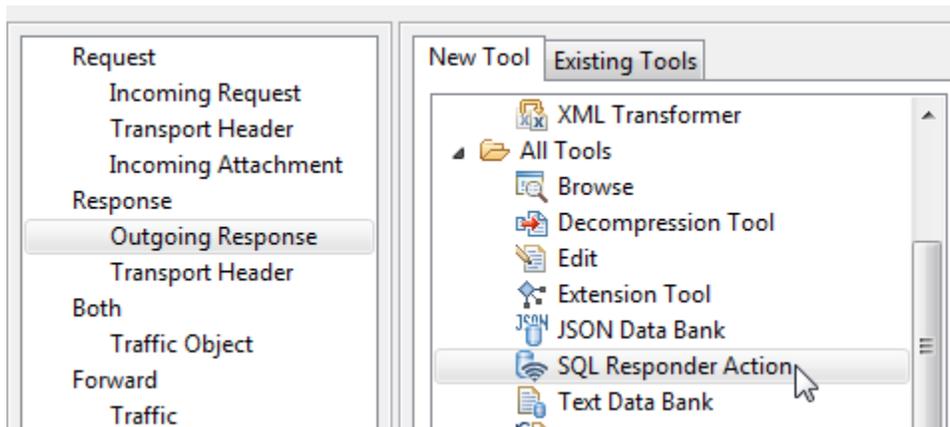
The SQL Responder Action tool enables you to establish an interconnected virtualization environment where an operation made by a Message Responder updates values in a virtual database that is represented by a .CSV-driven SQL Responder.

For instance, assume that you are working with a banking service that handles account transfers, deposits, and withdrawals. These operations change the values stored in a database. Now assume that you virtualized the service with a Message Responder and virtualized the database with a SQL Responder. How do you ensure that any new transactions performed from your virtual service update the values that were previously recorded for your virtual database?

By applying SQL Responder Action tools, you can create a synchronized environment so that when the virtualized service performs an operation, it will update the corresponding value in the virtual database. That updated virtual database value can then be accessed or operated on by yet another virtual asset (perhaps another service that retrieves account values and transaction histories)—creating a chain of interconnected operations.

Configuring the SQL Responder Action Tool

To add a SQL Responder Action tool, chain it as an outgoing response to the Message Responder that modifies data in the virtual database.



Next, configure the following options:

Target Tab

Configure the following settings in this tab:

- **Target column** is the virtual database column you want to modify.
- **Operation** describes how you want to modify the existing data in the virtual database.
- **Value** is the value that will be used to perform the specified operation. You can use specific fixed values—for instance, if you're transferring \$100 from one account to another, the value here would be 100. You can also use scripted or parameterized values—for instance, a value that was pulled from the incoming request and stored in an Incoming Request> XML Data Bank tool.
- **JDBC URL** points Virtualize to the appropriate set of files captured during the database recording. Type `j . t o` to open a list of valid JDBC URLs.

ResultSet Selector Tab

You add Query Parameters table values to specify which recorded result set you want updated with the value and operation specified in the Target tab. For instance, you might have result sets for customer_id 12212 and customer_id 12222. If you want this action to be performed only on queries for customer_id 12212, you would indicate this as follows:

Target	ResultSet Selector	Row Selector
▼ Query Parameters		
	Name	Value
1	customer_id	12212
2		

Note that the names and values that you specify here need to correspond to the actual values in the SQL query and result set (see the SQL Responder control panel to review the actual values).

The screenshot shows the SQL Responder interface. The main window has a name field set to "SQL Responder". Under the "Queries" section, a list of query templates is visible, including "SELECT id, date, headline, story FROM" and "SELECT id, first_name, last_name, add". Below the queries are "Add" and "Remove" buttons. The "Parameters" section shows a table with "customer_id" and the value "12212". A floating window titled "SQL Responder Action" is open, showing its name as "SQL Responder Action" and a "Query Parameters" table with 7 rows. The first row of this table has "customer_id" in the "Name" column.

Row Selector Tab

Specify which rows of the SQL query result set you want to operate on.

A single SQL query result set could include multiple rows. Using this control, you can specify if the operation should be performed on all rows or on specific rows. For instance, if the result set includes multiple accounts for customer_id 12212 and you want to operate only on account ID 12345, you would indicate this as follows:

Target	ResultSet Selector	Row Selector
▼ Parameters		
	Name	Value
1	ID	12345

The names and values that you specify here need to correspond to the actual row values (see the .csv file referenced in the SQL Responder control panel to review the actual values).

Target	ResultSet Selector	Row Selector
▼ Parameters		
	Name	Value
1	ID	12345
2		
3		

*SQL Responder Action Test 1: SQL Responder data_1cjk0u.csv

```

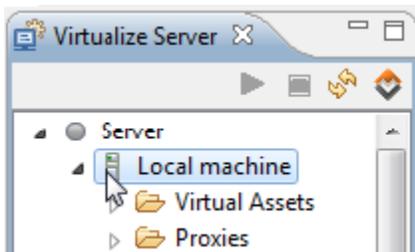
ID, CUSTOMER_ID, TYPE, BALANCE
"12345", "12212", "0", "-2700.00"
"12456", "12212", "0", "10.45"
"12567", "12212", "0", "100.00"
"12678", "12212", "1", "-100.00"
"12709", "12212", "0", "100.00"
"12900", "12212", "0", "0.00"
"13011", "12212", "0", "100.00"
"13122", "12212", "0", "1100.00"
"13233", "12212", "0", "100.00"
"13344", "12212", "1", "1231.10"
"13566", "12212", "2", "1000"
"13677", "12212", "2", "1000"
"54321", "12212", "0", "1351.12"

```

Resetting Data Modified by the Tool

To reset the originally-recorded values for any data that was modified by this tool:

1. In the Virtualize Server view, double-click **Local machine**.



2. In the page that opens, click the **Reset** button.

