Creating Parameterized Message Test Clients from Traffic

This topic provides an overview of how to create parameterized test clients (SOAP Client, REST Client, EDI Client, or Message Responder tools) from traffic that is captured in traffic logs. Sections include:

Prerequisites

- Before you can start creating parameterized test clients or Message Responders from traffic, your team must have a Data Repository Server installed and running. For details, see Installing a Remote Data Repository Server
- Message contents must be well-formed (e.g. if XML, it must be well-formed; if EDI, it must be a valid EDI message, etc.); otherwise, auto-creation of tests from the traffic might fail. SOAP messages and/or Message Responders must have only one top-level XML element.
- Message Grouping OptionsNote that the Data Repository does not support parameterization of JSON arrays with mixed types. If a JSON array does have mixed types, SOAtest or Virtualize will assume that all elements in the array are the same type as the first element.

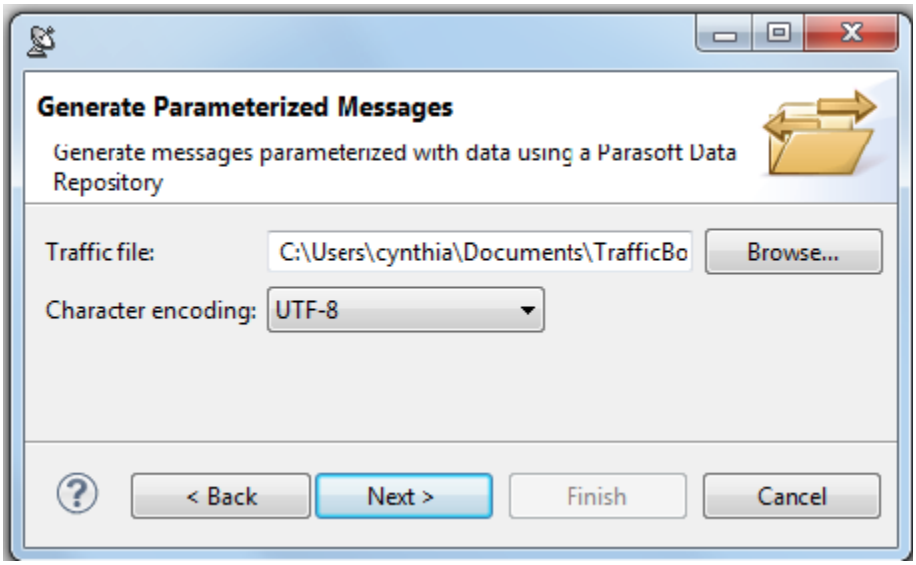
Monitoring the Console View

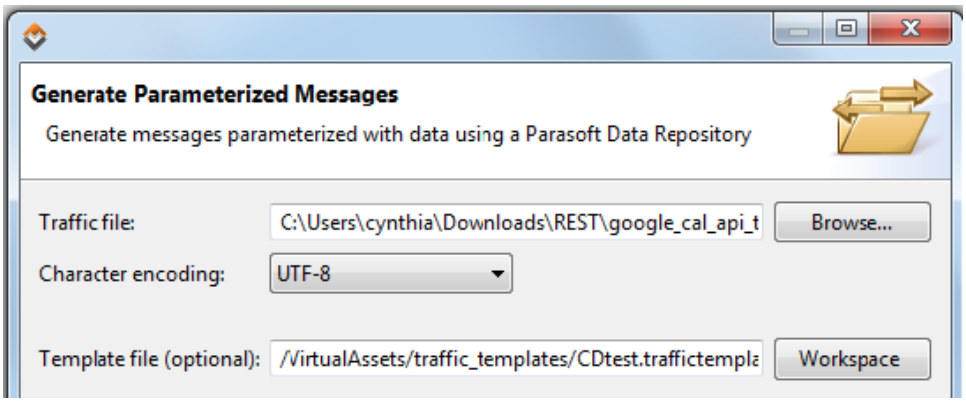
It's helpful to keep the Console view visible as you are creating tests and/or message responders from traffic. This view will display any warnings, errors, and informational messages that are generated while processing the traffic file.

Using the Wizard

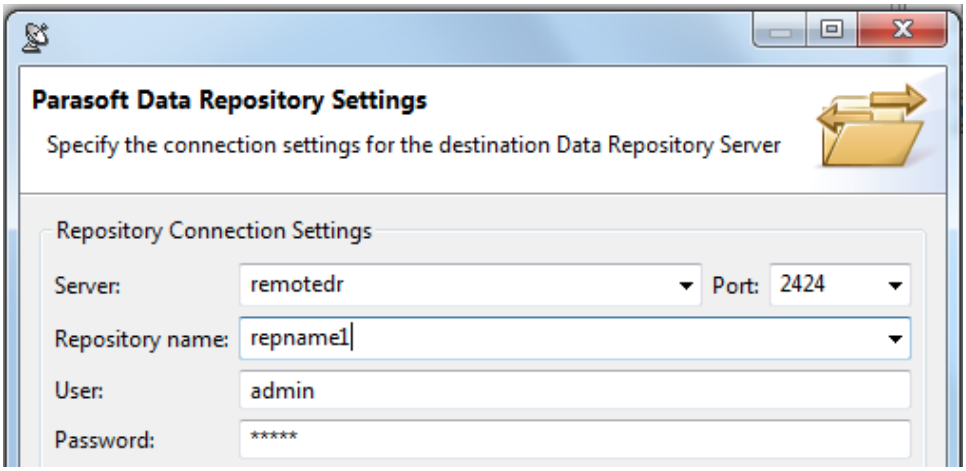
1. Choose the **Traffic> Generate Parameterized Messages** option in one of the available creation wizards. See the following chapters for additional details:Adding a New .tst File to an Existing Project
   Adding a New Test Suite.

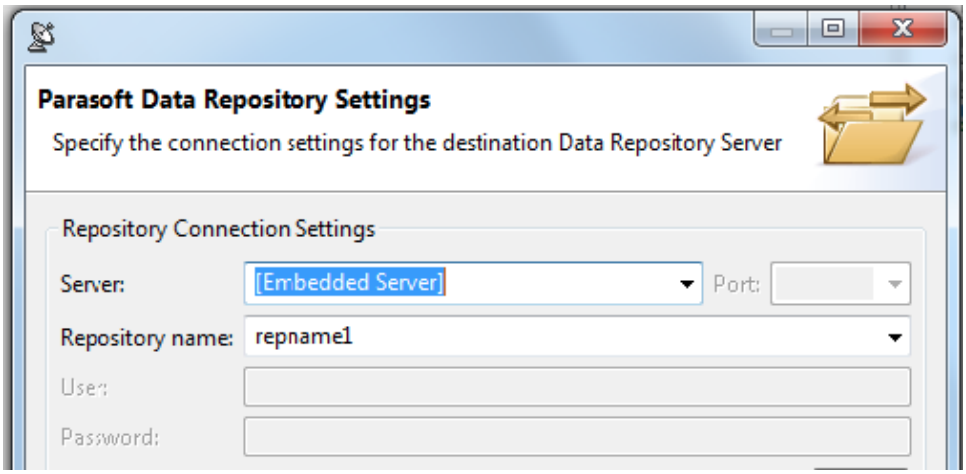
   Adding Projects, Virtual Assets, and Responder Suites.
2. Complete the first page of the Traffic wizard:
   a. Specify the location of the traffic file.
   b. Change the character encoding if needed
   c. If you want to populate the wizard with a previous group of settings saved in a template, enter the location of that template.See Using Configuration Templates to Reuse and Share Wizard Settings for details about creating and using templates in SOAtest and Using Configuration Templates to Reuse and Share Wizard Settings for details about creating and using templates in Virtualize.
   d. Click **Next**.

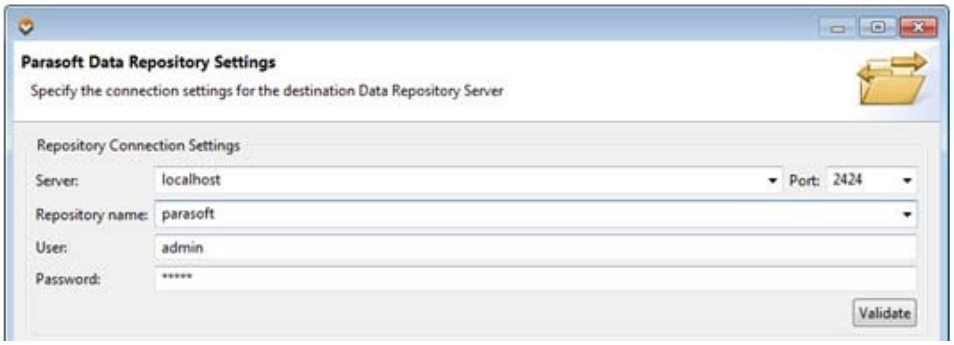


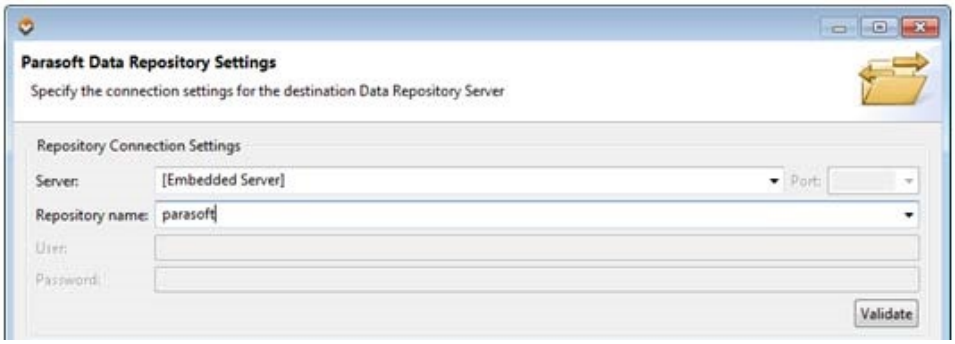


3. In the Parasoft Data Repository Settings page, specify which data repository should store the data used to parameterize the test clients or message responders and click **Next.**



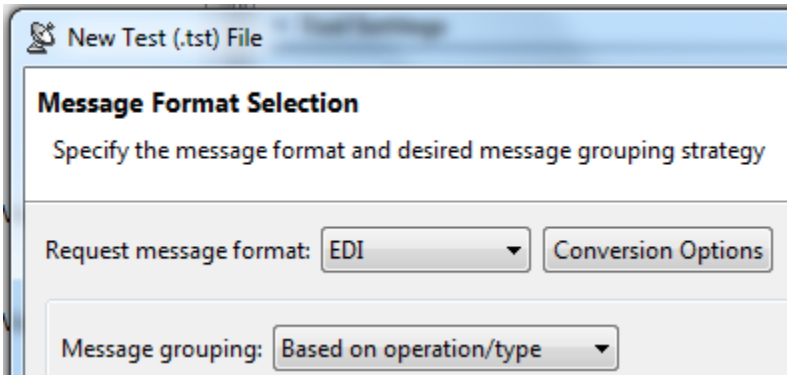


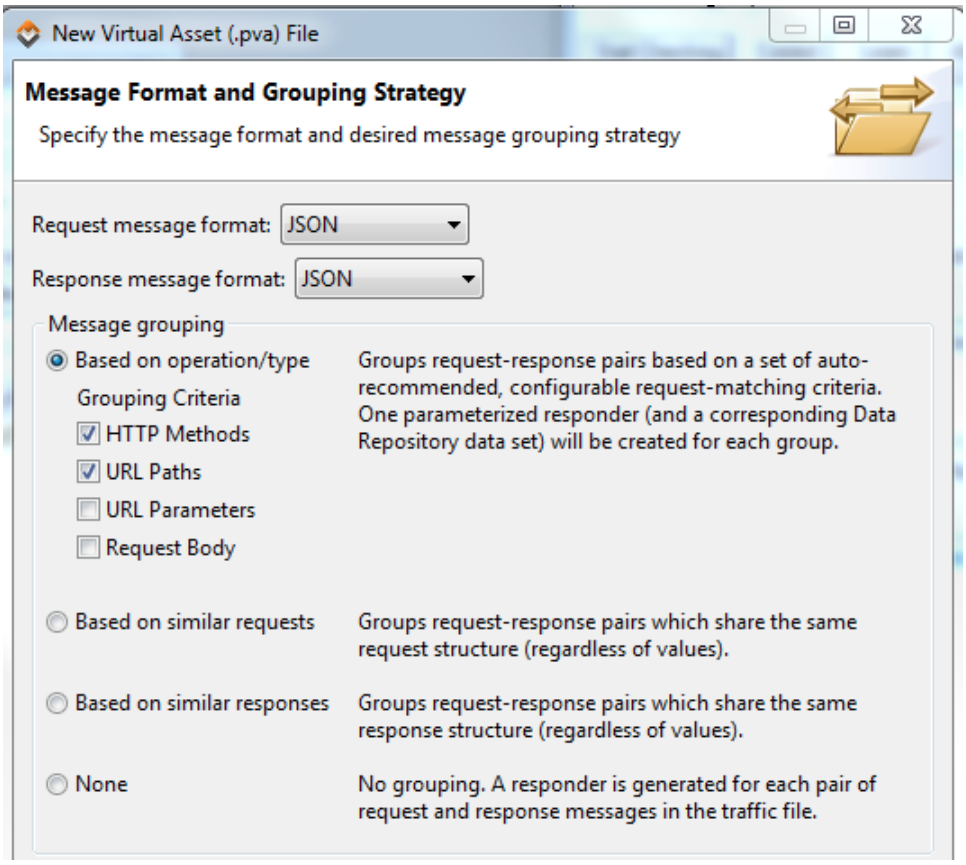




- In the **Server** field, specify which server you want to connect to (either the embedded server or a remote server). If you select the embedded server, the **Port**, **User**, and **Password** fields will be grayed out. If you select a remote server, the Port, User, and Password fields will be automatically populated (these can be adjusted if needed).
- In the **Repository name** field, select or enter the name of the repository you want to use. If you enter the name of a new repository, that repository will be created.
- When defining a repository connection, you can check the connection by clicking Validate.

4. Complete the Message Format and Grouping Strategy page, then click **Next**.

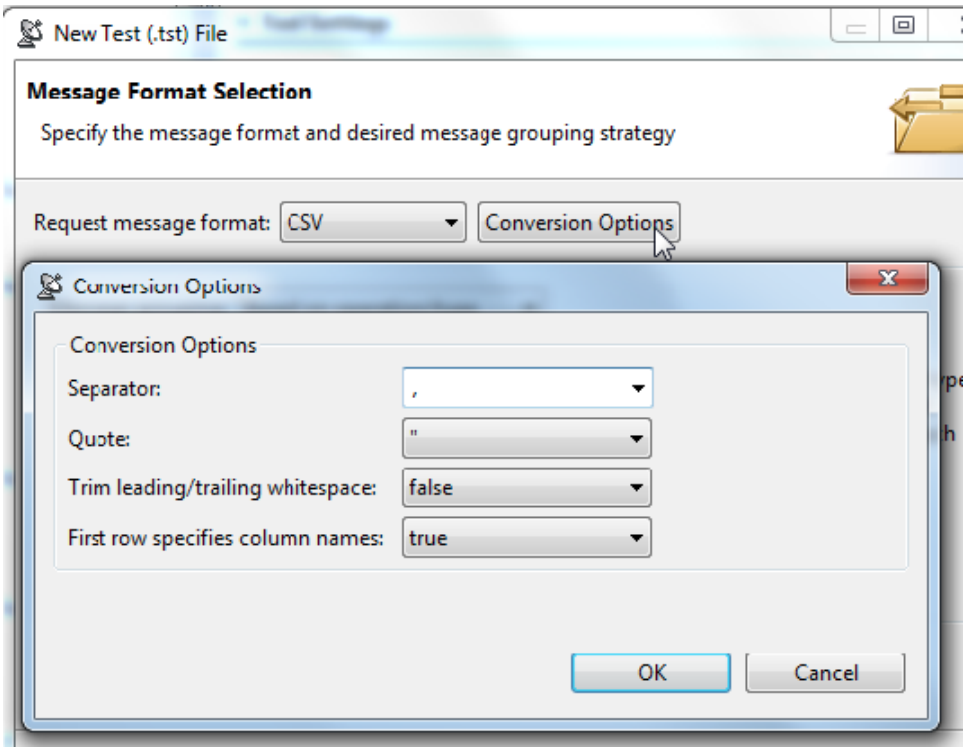


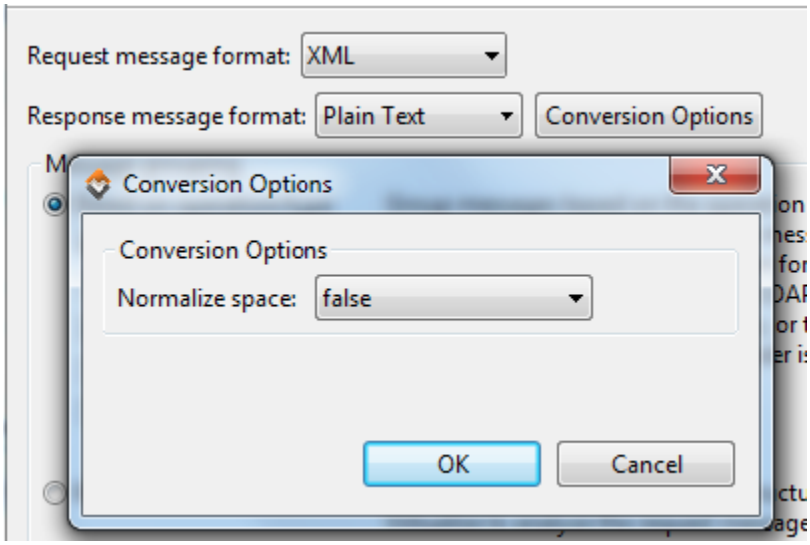


a. Verify that **Request message format** and **Response message format** are set to the correct format. If not, select the appropriate format.

SOAtest and Virtualize will attempt to identify the message format of the request and response based on the first message in the traffic file. All requests in a single traffic file are expected to have one format, and all responses in that same file are expected to have one format. The request format may be different than the request format. If the message format is not detected, Plain Text will be selected.

b. If you want to configure any conversion options that are available for the selected formats (e.g., for EDI or custom formats), click the **Con version Options** button to the right of that format and make the desired changes.





c. Specify the desired message grouping option (see the box below for details), then click **Next**.

Message Grouping Options

Available options are:

Based on operation/type: Group messages based on the operation or message type. This is useful for service traffic that contains messages that are distinctly identifiable either by operation or by the format's message type (i.e., the name of the element under the SOAP Body, the name of the root element in plain XML messages, or the message type of a specified message format). A responder is generated for each operation/type discovered within the traffic file.If you select this option, Virtualize will recommend grouping heuristics to apply, based on its analysis of the traffic file. You can change the pre-selected heuristics. To learn more about the heuristics, see Understanding Heuristics for Grouping by Operation - Type.**Based on similar requests:** Group messages based on request message structure. This tells Virtualize to analyze the request message structures and group the request/response into responders so that each responder will contain responses that correlate with requests that have a similar structure. Messages are considered "similar" when they have an identical DOM tree model, even if they have different values. This option is used to optimize and simplify the rules for correlating requests to responses within each Message Responder.

Based on similar responses: Group messages based on response message structure. This tells Virtualize to analyze the response message structures and group the request/response pairs into responders so that each responder will contain responses that have a similar structure. Messages are considered "similar" when they have an identical DOM tree model, even if they have different values.

None: No grouping. A responder is generated for each response message in the traffic file. Use this option if you want every request/response pair in separate Message Responders.

5. In the Message Grouping Review page, review the information about the operations and messages found.

   The types of columns shown depends on the grouping strategy applied.

   Each table row represents the criteria for defining a group. One group will be generated for each table row. One responder will be generated for each group.

   The correlation criteria will be processed in the order in which they appear in the table (from top to bottom). URL paths and parameters will be parameterized against a field from the record type. The fields will have an automatically-generated name and will be visible in the Data Reuse page (later in the wizard).For details on how these groupings were created, see Understanding Heuristics for Grouping by Operation - Type.
6. Add, modify, reorder, and remove grouping criteria using the available controls. See Customizing Grouping Criteria for details on configuring grouping criteria.For details on configuring grouping criteria Customizing Grouping Criteria in SOAtest and Customizing Grouping Criteria in Virtualize.See Customizing Grouping Criteria for details on configuring grouping criteria.If you change the criteria, be sure to click **Regroup** before proceeding.

7. If all the **Autoconfig** boxes are checked and you want Virtualize to automatically configure Message Responders for the specified groups, *you can skip this following step*.

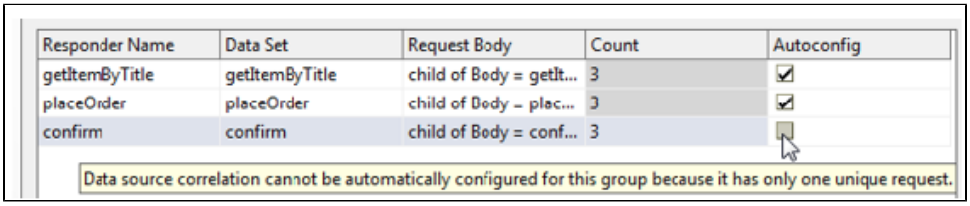
   If you want to perform any of the following tasks, disable **Autoconfig** for each message group you want to customize, click **Next**, then configure the request mapping as described in Customizing Request Matching and Correlations:

   Customize which parameter values should be used to determine the response messages of the virtual asset

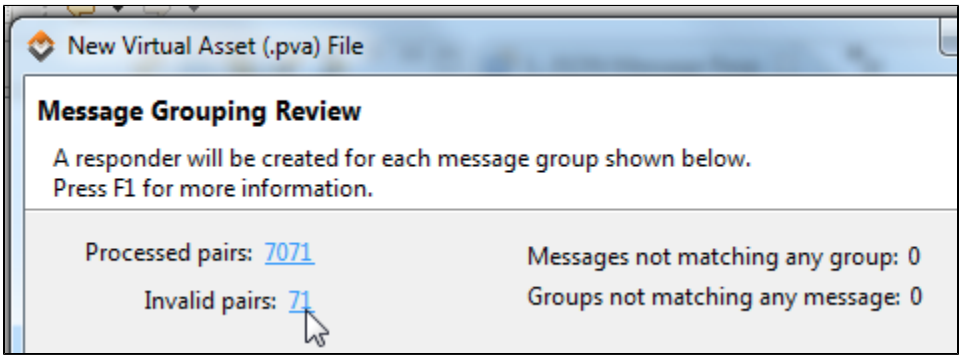
   Modify the automated request/response pair matching

   Specify a WSDL or Schema

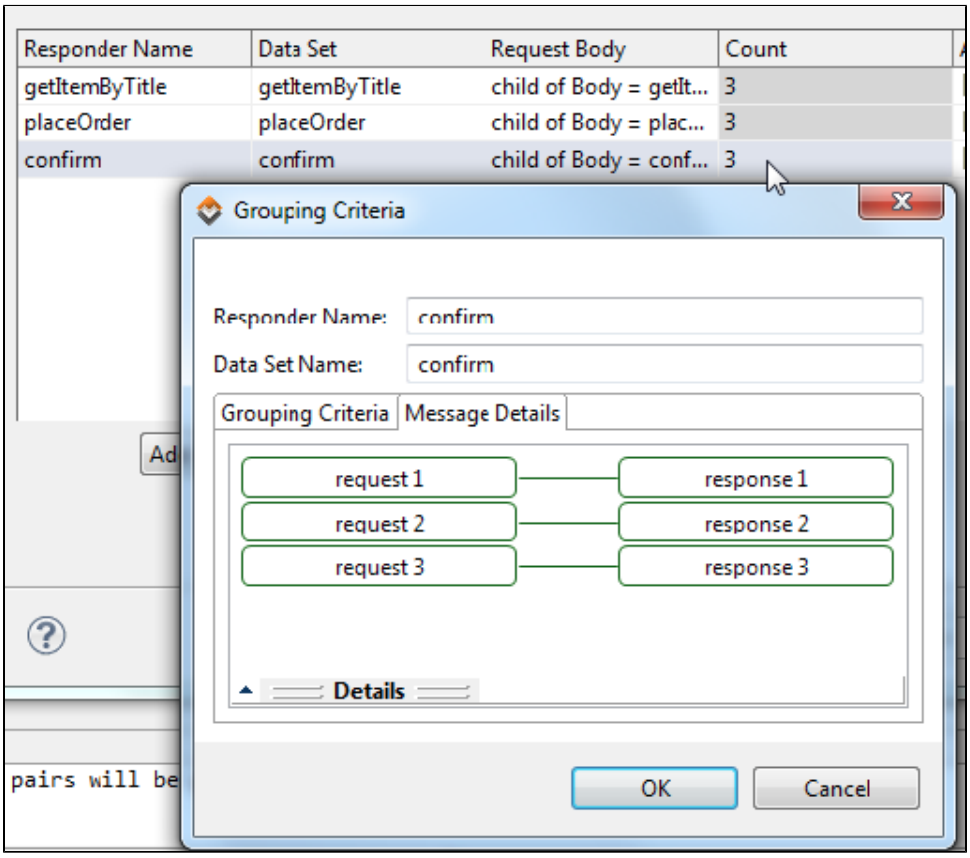
Auto-configuration is typically available when there are multiple requests within a message group and there are differences in the paths, the parameters, or the body. If the **Autoconfig** box is grayed out, this means that autoconfiguration is not available for this group; for more details on why a specific group cannot be automatically configured, see the tooltip for that item.



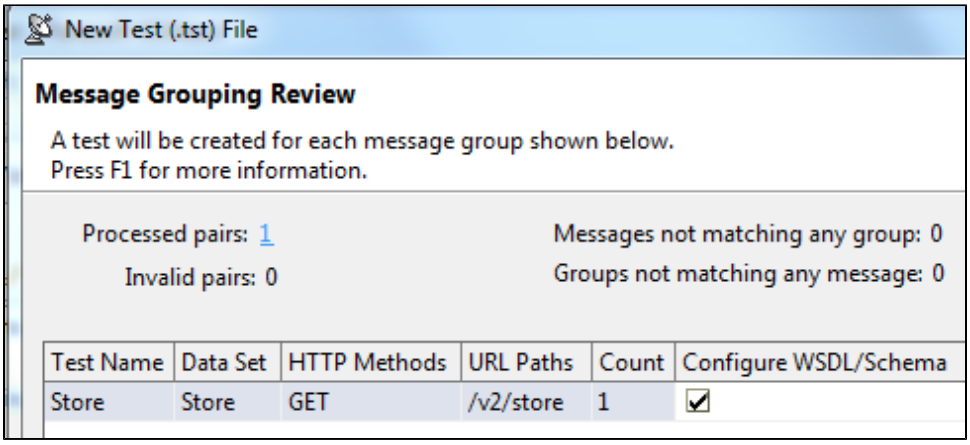
For more details on any of the items listed at the top of the panel (processed pairs, unprocessed pairs, messages that don't match groups, etc.), click the associated hyperlinks.

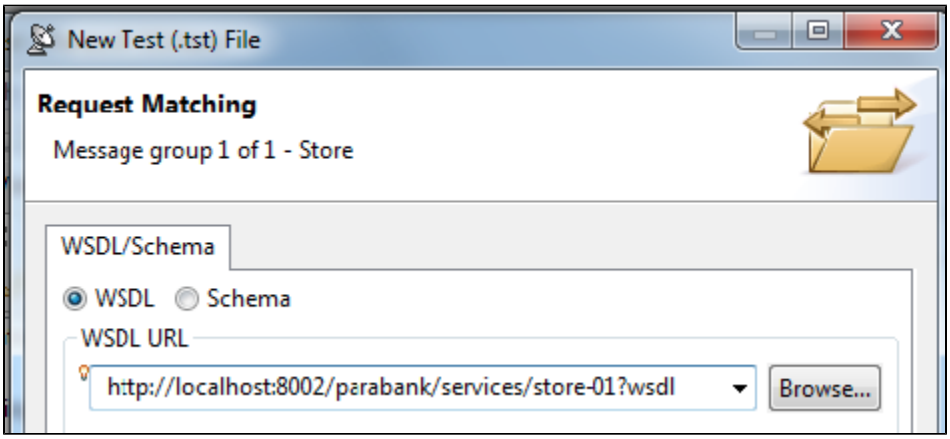


To review the messages associated with a particular responder—and/or to change the responder and data set name—click the related row in the **Count** column.



If you want to specify a WSDL/schema, enable **Configure WSDL/Schema**, then specify the appropriate values in the next page.





Should you specify a WSDL or Schema?

Advantages of specifying the WSDL or schema include:

The generated Form Input model will be based on that WSDL/Schema, which provides type richness when you are editing and maintaining the resulting Form Input.

Change Advisor (described in Change Management) is available to help you keep your assets in sync with evolving services and changing environment conditions.

If you notice that the generated Form Input and its data parameterizations do not match the original messages, this is a sign that the messages do not fully match the WSDL/Schema or that mapping the raw messages failed. If you are experiencing such issues, you should omit the WSDL/Schema to ensure that the generated Form Input model fully matches the traffic messages.

8. In the Data Reuse page, configure how you want the imported traffic to reuse or impact existing data.
   - The defined record identity is used to determine which data is new and which new records match existing records. If it has not already been specified for this data set, the identity can be added/modified from the data tree in this page.
   - The tree indicates identity fields with green arrow icons. Existing data sets are noted with annotations.
   - You can control how new data from the traffic file will extend and/or update existing repository data sets.

     Replace: Erase existing data then add the new data

     Append: Adds new records without first erasing the existing data.

   - You can also control whether matching data (data that matches existing record types, as determined by the identity) reuses existing record types or updates an existing record. The **Reuse** option enables you to reuse/share the existing records that match. The **Update** o ption enables you to update the existing records' corresponding fields with data from the traffic and add new records for new record types.

Virtualize Only Options
Replace: Erase existing data then add the new data. Merge: Import new data without modifying existing data. Update: Update matching records with new data and create new records as needed. Overwrite: Update matching records (with matching keys) with new data, do not create any additional records.

Additional details about specifying identities and choosing among the available data reuse/updating options in SOAtest is available at Configuring Data Reuse and Updating.

For Virtualize, see Configuring Data Reuse and Updating.

9. In the Export Template page, specify template export settings (if desired), then click **Next**.
    - If you want to save the settings you used in this wizard, you can export them into a template. Just check **Export configuration data into a reusable template** and indicate the desired file name and location.
      See Using Configuration Templates to Reuse and Share Wizard Settings in Virtualize and Using Configuration Templates to Reuse and Share Wizard Settings in SOAtest for details about creating and using templates.

      If you are creating the .pva in the Virtual Assets folder, which results in automated deployment, complete the Deploy Virtual Asset wizard page by specifying the desired name and deployment path for the virtual asset that will be created and click **Next**. The virtual asset will be deployed at the listed endpoint. For details, see Configuring Individual Virtual Asset Deployment Settings.

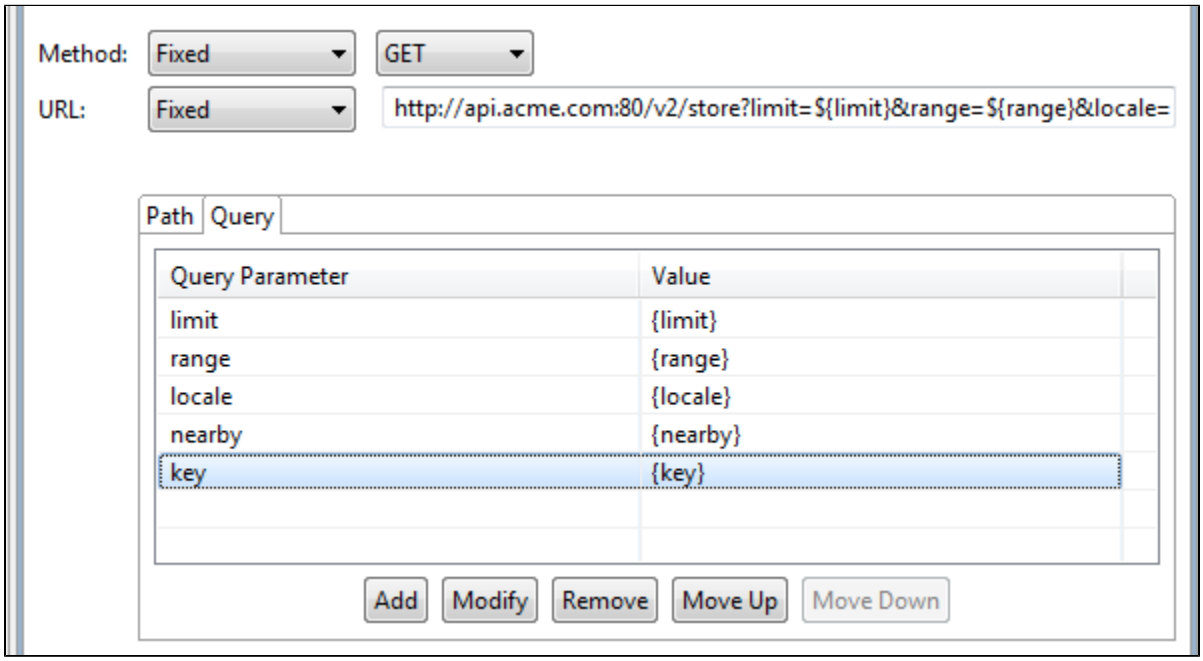
    (MQ and JMS only) Specify your connection settings in the next SOAtest wizard page. These settings will be applied to the tools created from this traffic. For details, see Configuring MQ Options and Configuring JMS Options.
10. Click **Finish**.

The following items will be created and configured:

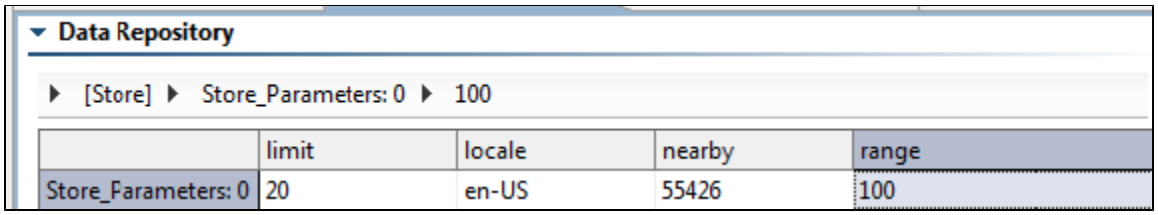
- One or more test clients with parameterized values. The tools created will be SOAP Clients, REST Clients, EDI Clients, or Messaging Clients, depending on the message format. The tools will default to Form Input / Form JSON view unless the message is XML or JSON and is so large that a performance impact is expected; in that case, Literal view is used.

  A Message Responder with parameterized elements as well as preconfigured responder correlation and data source correlation will be added. The tools will default to Form Input / Form JSON view unless the message is XML or JSON and is so large that a performance impact is expected; in that case, Literal view is used.
- (For new data repositories) A new Data Repository with applicable data sets and record types will be added. One data set will be added per message group identified by analyzing the traffic.
- (For existing data repositories) New data sets and record types will be added to the existing repository.
- A repository data source will be added for each added data set and each test client or message responder will be configured to use the associated data source.

For example, here is a sample REST Client parameterized with data repository values:



Here is part of the corresponding repository:



This parameterized, data-driven REST Client can now be run with a broad and varied scope of test values—without requiring any modification to the tool itself. Rather than edit the tool, you would modify or extend the associated data repository values.

If the .pva was created in the Virtual Asset folder, the virtual asset will be automatically deployed to the local Virtualize server as the wizard completes. Otherwise, it can be manually deployed to local or remote servers.

For details on how to edit and extend the data stored in the data repository, see Viewing and Modifying the Repository Structure and Contents.

Note that custom transport headers and any SOAP Headers (e.g. WS-Security Headers) that are present in the traffic file are not configured automatically into the generated assets or data repository data sets. You can specify them in the generated Message Responders

(see Message Responder Overview for details)

.

Deploying the Virtual Assets

If the .pva was created in the Virtual Asset folder, the virtual assets is automatically deployed to the local Virtualize server as the wizard completes. Otherwise, you can deploy it to local or remote servers whenever you are ready.

For a more detailed discussion of deployment procedures and options, see Deploying Virtual Assets - Overview.

Customizing the Virtual Assets

For details on how to customize the Message Responder's behavior, see Message Responder Overview.

Understanding Choice/Extension Type Support

If you do not enter a WSDL or schema file at the end of the wizard, Virtualize uses the data structure of recorded traffic to create the data repository. When the data structure of an element varies in the recorded traffic, it is likely that the type for that element is a choice in the underlying schema. However, the wizard does not explicitly support choice types; it interprets an element's data structure as a sequence of all possible child elements.

For example, assume an element whose actual schema is like this:

```
<element name="parent">
    <complexType>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="child1"/>
            <element name="child2"/>
            <element name="child3"/>
        </choice>
    </complexType>
</element>
```

Virtualize will represent the element with the following data structure:

```
<element name="parent">
    <complexType>
        <sequence>
            <element name="child1" minOccurs="0" maxOccurs="unbounded"/>
            <element name="child2" minOccurs="0" maxOccurs="unbounded"/>
            <element name="child3" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
</element>
```

Although the recorded traffic might have child elements appear in a varying order (e.g., the "parent" in one of the response message has "child1" and then "child2", while the "parent" in another response message has "child2" and then "child1"), Virtualize will parameterize message data in a fixed order. Thus, in this example, the elements "child1" and "child2" will always be in the same order within the response message.

Completing the Virtualize Wizard: Advanced Topics

The following topics provide additional details that will help you complete the wizard:

- Using Configuration Templates to Reuse and Share Wizard Settings
- Understanding Heuristics for Grouping by Operation - Type
- Customizing Grouping Criteria
- Customizing Request Matching and Correlations
- Configuring Data Reuse and Updating

Completing the SOAtest Wizard: Advanced Topics

The following topics provide additional details that will help you complete the wizard:

- Using Configuration Templates to Reuse and Share Wizard Settings
- Customizing Grouping Criteria
- Configuring Data Reuse and Updating

Video Tutorial: Creating Virtual Assets from Traffic Recorded with a Message Proxy

In this video you'll learn how to create a virtual asset from traffic recorded with a message proxy.

   </body> </html>