_Creating_Parameterized_Message_Tests_Clients_from_Traffic

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.

2. Complete the first page of the Traffic wizard:
   1. Specify the location of the traffic file.
   2. Change the character encoding if needed
   3. 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.

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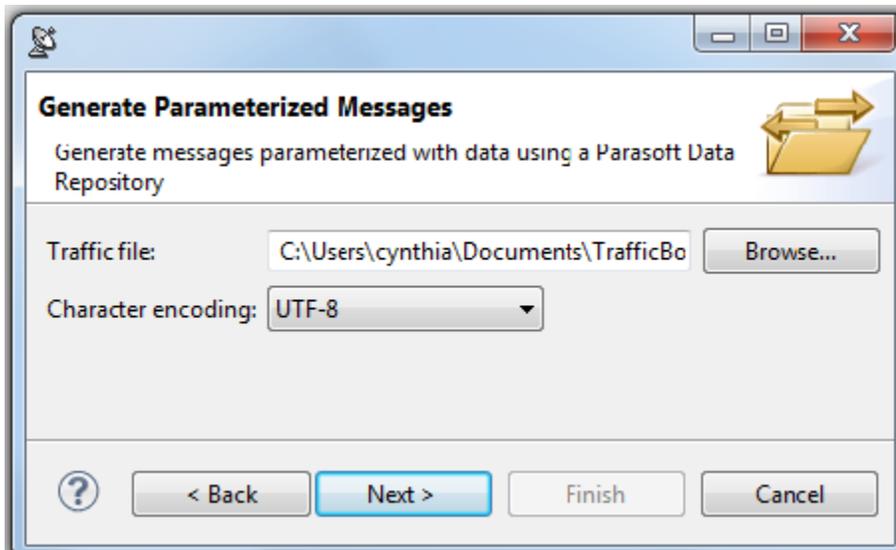
3. In the Parasoft Data Repository Settings page, specify which data repository should store the data used to parameterize the test clients or message responders and click **Next.**

- 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**.

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

    SOAtest

    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.
2. If you want to configure any conversion options that are available for the selected formats (e.g., for EDI or custom formats), click the **Conversion Options** button to the right of that format and make the desired changes.
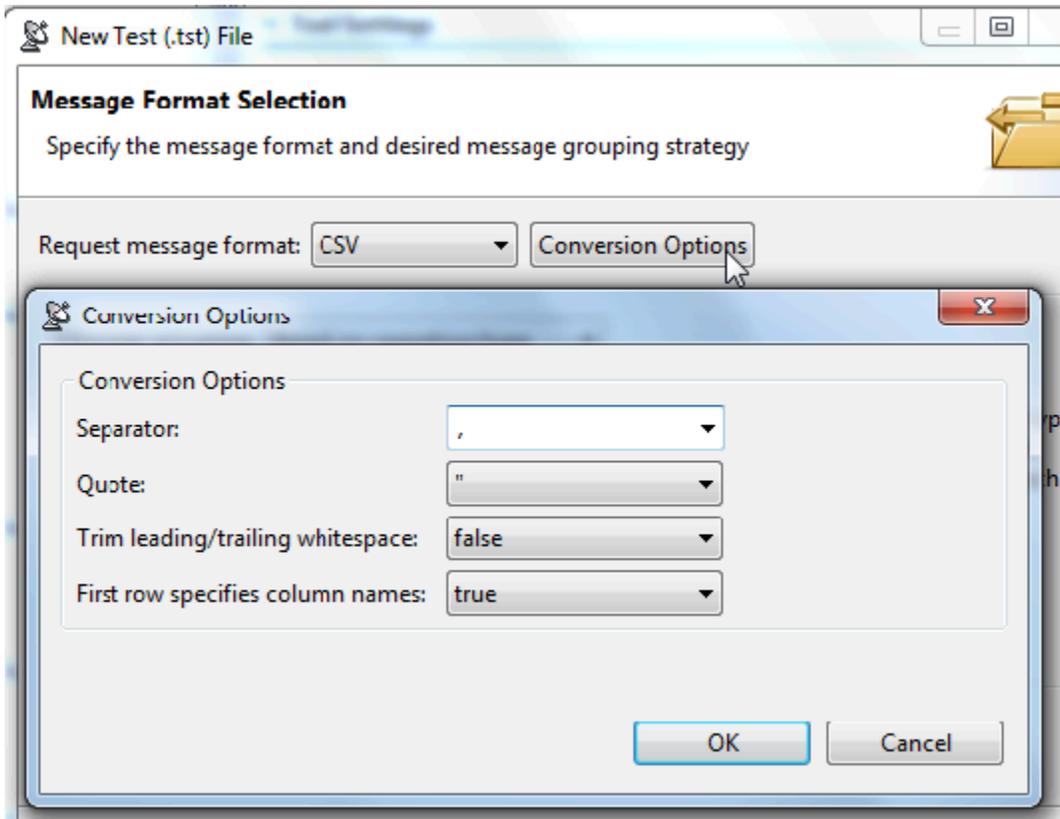
3. Specify the desired message grouping option (see the box below for details), then click **Next**.

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.

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).

6. Add, modify, reorder, and remove grouping criteria using the available controls. See Customizing Grouping Criteria for details on configuring grouping criteria.

   If you change the criteria, be sure to click **Regroup** before proceeding.
7. 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** option enables you to update the existing records' corresponding fields with data from the traffic and add new records for new record types.

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

   (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.
9. 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.

   - (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.

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

.

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