

# Creating Parameterized Message Test Clients from Traffic

This topic provides an overview of how to create parameterized test clients, such as SOAP Clients, REST Clients, and EDI Clients, from traffic that is captured in traffic logs. Sections include:

- [Prerequisites](#)
- [Using the Wizard](#)
- [Completing the SOAtest Wizard: Advanced Topics](#)

## Prerequisites

- You 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. SOAP messages must have only one top-level XML element.
- The Data Repository does not support parameterization of JSON arrays with mixed types. If a JSON array does have mixed types, SOAtest 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 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 [Adding a New .tst File to an Existing Project](#) and [Adding a New Test Suite](#) for additional details.
2. Specify the following information in the Traffic wizard and click **Next**:
  - a. Specify the location of the traffic file.
  - b. Change the character encoding if needed.
  - c. (Optional) Specify the service definition file. Providing a service definition helps SOAtest create better message groupings.
  - d. 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.

The screenshot shows the 'New Test (.tst) File' dialog box with the following details:

- Title:** New Test (.tst) File
- Section:** Generate Parameterized Messages
- Sub-section:** Generate messages parameterized with data using a Parasoft Data Repository
- Traffic Section:**
  - Traffic file: test & Virtualize.app/Contents/Eclipse/examples/traffic/store.txt
  - Character encoding: UTF-8
- Service Definition (optional) Section:**
  - Definition type: WSDL
  - Definition file: /Applications/parasoft/soatest\_virtualize/2021.1/Parasoft SC
- Traffic Template (optional) Section:**
  - Template file: (empty)
- Buttons:** ? (help), < Back, Next > (highlighted), Cancel, Finish

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 choose a remote server, the **Port**, **User**, and **Password** fields will be automatically populated and can be adjusted as needed. If you choose Embedded Server, the **Port**, **User**, and **Password** fields will be grayed out.

- 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. Click **Next** and configure the message format and grouping strategy:

- Verify that **Request message format** is 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. Conversion options are available for some formats, such as EDI or custom formats. Click the **Conversion Options** button and make the desired changes.
- Choose one of the message grouping options:

- **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.
- c. If you specified a service definition or template file, you can enable the **Constrain** option in the Message Group Creation Options section. If this option is enabled, only the message groups defined in either the template or service definition file will be grouped. All other messages will be ignored. If this option is disabled, SOAtest will group all messages that appear in the traffic file. If a service definition and template file are provided and this option is enabled, groupings will be constrained according to the template file prior to being constrained by the service definition. As a result, a message will be ignored if it is defined in the service definition but not in the template file. The **Constrain** option is enabled by default.
5. Click **Next** and review the information about the operations and messages in the Message Grouping Review screen.

**Message Grouping Review**

A test will be created for each message group shown below.

Processed pairs: 0      Messages not matching any group: 0  
 Invalid pairs: 0      Groups not matching any message: 0

Test Name	Data Set	HTTP Methods	Request Body	Count	Service Definition
addItemToCart	addItemToCart	POST	child of Body = ...	3	<input type="checkbox"/>
getItemByTitle	getItemByTitle	POST	child of Body = ...	3	<input type="checkbox"/>
submitOrder	submitOrder	POST	child of Body = ...	3	<input type="checkbox"/>

Buttons: Add, Duplicate, Remove, Move Up, Move Down, Regroup

Navigation: ? < Back, Next >, Cancel, Finish

- The columns included are based 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).
- Add, modify, reorder, and remove grouping criteria using the available controls. See [Customizing Grouping Criteria](#) for details on configuring grouping criteria in SOAtest. If you change the criteria, be sure to click **Regroup** before proceeding.
- If you want to specify a WSDL/schema, enable the option in the **Configure** column and click **Next** to specify the appropriate values.

**New Test (.tst) File**

**Request Matching**  
Message group 1 of 1 - addItemToCart

**Service Definition**

WSDL  Schema

WSDL URL

file:///Applications/parasoft/soatest\_virtualize/2021.1/Parasoft SOAtest & Virtualize.app/Con

The service definition options will appear based on the type of traffic. For example, if the traffic is JSON, you will have the option to specify an OpenAPI/Swagger, RAML, or WADL file. For XML traffic, you will be able to specify WSDL or Schema.

**New Test (.tst) File**

**Request Matching**  
Message group 1 of 1 - Enabled

**Service Definition**

OpenAPI/Swagger

OpenAPI/Swagger URL

file:///Users/atrujillo/Downloads/soavirt\_openapi\_3-24-21.json

**i 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.

6. Click **Finish** to complete the wizard or **Next** to configure how the data reuse settings for the imported traffic.
  - a. 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.
  - b. The tree indicates identity fields with green arrow icons. Existing data sets are noted with annotations.
  - c. 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.
  - d. 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. 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](#).

7. Click **Next** and specify any additional configurations in the Final Options screen:
  - a. You can configure the wizard to create messages in Form or Literal mode. These modes present a Form Input view (see [Form Input](#)) or a Literal view (see [Literal](#)).
  - b. You can enable the **Export configuration data into a reusable template** option and specify a file name and location to save the settings you used in this wizard as a template. (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](#). See [Using Configuration Templates to Reuse and Share Wizard Settings in SOAtest](#) for details about creating and using templates.
  - c. 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 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:

Method: Fixed GET

URL: Fixed `http://api.acme.com:80/v2/store?limit=${limit}&range=${range}&locale=`

Path	Query
Query Parameter	Value
limit	{limit}
range	{range}
locale	{locale}
nearby	{nearby}
key	{key}

Add Modify Remove Move Up Move Down

Here is part of the corresponding repository:

Data Repository				
[Store] ▶ Store_Parameters: 0 ▶ 100				
	limit	locale	nearby	range
Store_Parameters: 0	20	en-US	55426	100

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](#).

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.

## 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](#)