

HTTPJMSMQ Traffic

This topic provides an overview of how to generate tests from recorded HTTP, JMS, or MQ traffic. Sections include:

- [Generating Tests from Traffic Overview](#)
- [Supported Traffic Log Formats](#)
- [Using Parameterized vs. Fixed Messages](#)



How does method this compare to the "Generate tests from JMS/TIBCO/Sonic" wizards?

This method supports HTTP and MQ as well as JMS (the other options support only JMS). It lets you record multiple connections at once (the others support a single JMS connection at one time). Also, it consistently leaves messages on the queues (if you select that option).

However, this method is a two step process (capture the traffic in a file, then create tests from that file); the other wizards require only one step.

With the other wizards, tests get created automatically when the initial wizard is completed. They also provide TIBCO-aware and Sonic-aware configuration panels that automatically configure some of the provider-specific settings. The recording wizard shows a generic JMS configuration interface.

Generating Tests from Traffic Overview

SOAtest can create tests for application behavior that is captured in traffic logs—including:

- Traffic files produced by SOAtest recording proxies (e.g., for JMS, MQ, HTTP) as outlined in [Recording HTTP, JMS, and MQ Traffic for Test Generation](#).
- Traffic files produced by Parasoft Virtualize or Parasoft CTP.
- Logs produced from other sources; for example:
 - Message traces or logs captured at the network level using network sniffing tools.
 - HTTP logs obtained by having your application log its traffic.

Supported Traffic Log Formats

SOAtest supports the following traffic log formats (the file extension does not matter).

HTTP Traffic logs

For the HTTP parser type, SOAtest can recognize and parse HTTP traffic with POST, GET, and PUT methods. DELETE is not supported. SOAtest expects the log to contain the HTTP headers for a message, followed with message body content:

```
HTTP request 1 headers
HTTP request 1 body
HTTP response 1 headers
HTTP response 1 body
HTTP request 2 headers
HTTP request 2 body
HTTP response 2 headers
HTTP response 2 body
...
```

SOAtest also expects there to be two line endings (Windows or Unix) between a message body and the next set of HTTP headers.

This format is similar to what would be captured by the SOAtest recording wizard or other HTTP traffic sniffing tools that can record HTTP traffic as it gets transmitted over the HTTP socket connection.

HTTP chunking and traffic files that are recorded with chunked traffic are supported. However, raw HTTP traffic files that include chunked messages or "continue" headers are not supported.

JMS and MQ Traffic Logs

SOAtest supports the creation of tests from JMS and MQ traffic files produced by the SOAtest traffic recording wizard.

Message Formats

For parameterized messages, the traffic files can include any message format that SOAtest supports (including EDI, JSON, fixed length, and custom message formats). For fixed messages, only XML and JSON are supported.

Using Parameterized vs. Fixed Messages

SOAtest provides two options for generating tests from traffic logs:

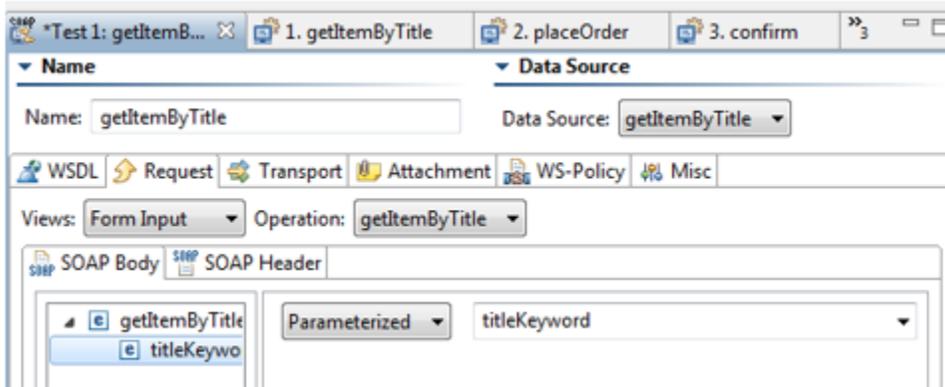
- **Generate parameterized messages** creates test clients that are automatically parameterized with values from a Parasoft Data Repository (which is also automatically created when you use this option). This option allows you to rapidly add and modify request message values using a graphical interface designed for manipulating large, complex data sources. Data values are stored and manipulated independently of the test client and .tst.
- **Generate fixed messages** creates one test client for each request detected. All request message data is stored and manipulated within the test client. These test clients can later be parameterized.

Example

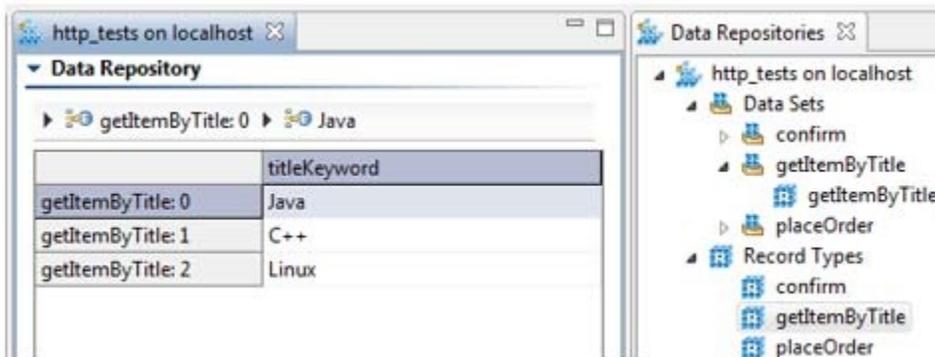
To understand the difference, consider the following example from a simple HTTP traffic file, including three messages for the getItemByTitle operation.

Parameterized Messages

With "generate parameterized messages," SOAtest configures the traffic file's three getItemByTitle requests with a single test client that is parameterized as follows.

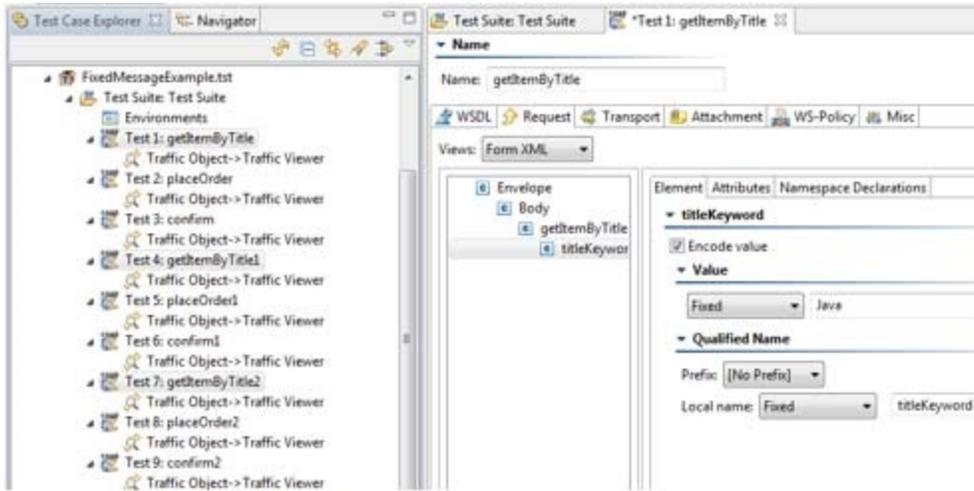


Additionally, it creates the following editable and extensible Data Repository.



Fixed Messages

With "generate fixed messages," SOAtest configures the traffic file's three getItemByTitle requests with three test clients. The following shows the test client that was created to search for books using the keyword *Java*.



Two similar test clients are also added to search for books using the keywords *Linux* and *C++*.