HTTP 1.1

This topic explains configuration options for using HTTP 1.1 with selected supporting tools and provisioning action tools.

Sections include:

- Configuring HTTP 1.1 Settings
- Error Handling

Configuring HTTP 1.1 Settings

When selecting HTTP 1.1 as the transport protocol, you can specify whether you want the client's requests to use Keep-Alive connections (required for NTLM and Digest HTTP authentication). It will also be reused for a single invocation of a test suite from the GUI or the command line. You will be able to add, modify, and remove custom HTTP headers to the SOAP request from the **Transport** tab of an appropriate tool. In addition, you will also be able to specify **HTTP Chunking**, which allows HTTP messages to be broken up into several parts. Chunking is most often used by the server for responses, but clients can also chunk large requests.

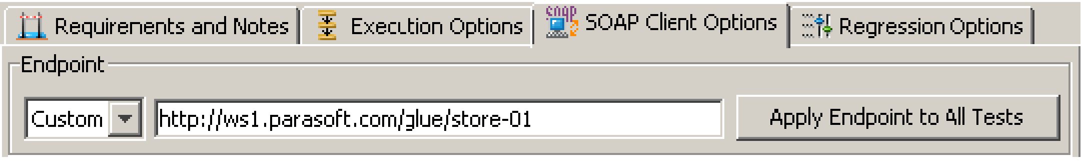
After selecting **HTTP 1.1** from the **Transport** drop-down menu within the **Transport** tab of an appropriate tool, the following options display in the left pane of the Transport tab:

- General
- URL Parameters
- Security
- HTTP Headers
- Cookies

General

General page options include:

- **Router Endpoint:** The endpoint is the URL of the service endpoint. By default, endpoints are set to the endpoint defined in the **WSDL**. Besides WSDL, there are three other endpoint options:
  - **Default:** When this option is selected, the endpoint will be the endpoint defined in the test suite that has the tool. To see the GUI for the endpoint defined in the test suite, click on the test suite node and click on the tool's Options tab::

    Endpoint

    Custom http://ws1.parasoft.com/glue/store-01

    Clicking on the **Apply Endpoint to All Tests** button will set endpoints of all tools in the test suite to the endpoint defined in the GUI.

  - **Custom:** Allows you to set any custom endpoint.
  - **UDDI serviceKey:** Describes what UDDI serviceKey is used to reference this server endpoint in the UDDI registry specified in the Preferences panel's WSDL/UDDI tab.

- **SOAP Action:** Specifies how the server processes the request. This field is disabled if the **Constrain to WSDL** check box is selected. *For SOAP Client only*

- **Method:** Specifies which method is used to processes the request. This field is disabled if the **Constrain to WSDL** check box is selected. The method to invoke can be specified as a fixed value, parameterized value, or scripted value.

  For details about parameterizing values, see Parameterizing Tests with Data Sources, Variables, or Values from Other Tests and Parameterizing Tools with Data Source Values, Variables, and Extracted Values.

  With fixed values, you can access data source values using `${var_name}` syntax. You can also use the environment variables that you have specified. For details about environments, see Configuring Testing in Different Environments and Configuring Virtualize Environments.

  For details about scripting values, see Extensibility and Scripting Basics

- **Message Exchange Pattern: Expect Synchronous Response:** Specifies whether a response body is expected. An HTTP response header is always expected. If this option is not selected, then the product sends a one-way message and waits for a notification header (typically "HTTP/1.1 202 Accepted").

- **Connection Settings:** Specifies Keep-Alive or Close connections for the transport protocol selected.
  - **Keep-Alive connection:** Adds a "Connection: Keep-Alive" header to request a keep-alive connection if the server supports it. This is required for NTLM and Digest HTTP authentication. For more information, see Error Handling.
  - **Close connection (default):** Outputs no additional HTTP headers and performs a regular HTTP 1.0 exchange. This is the default behavior for HTTP 1.0.

- **HTTP Chunking:** Sends HTTP messages in chunks. Rather than sending the entire message from memory, SOAtest and Virtualize read the message a block at a time, and then sends the block. A number is added before each block so that the receiving end knows how many bytes to expect.

- **Redirect Settings (for messaging clients only—e.g., REST, SOAP, Messaging, EDI clients):** Specifies whether to automatically follow HTTP redirects. Disable this option if you want to perform an action or validation on the original request/response traffic (instead of working only with the final request/response pair).
- **Compression Settings (for messaging clients only—e.g., REST, SOAP, Messaging, EDI clients):** Specifies whether to compress requests and decompress responses.
  - **Gzip request payload:** Gzips the request payloads being sent over the network. Data sent to attached tools will not be compressed. Note that compression does not apply to SOAP Clients configured to send attachments or in MTOM mode.
  - **Decompress gzip-encoded response payload:** Decompresses response payloads that have "Content-Encoding: gzip" as a header field. Attached tools will receive the uncompressed data.

URL Parameters

URL Parameters page options include:

- **URL Parameters:** Allows you to add parameters to the URL for a GET request. After clicking the **Add** button, you can specify **Parameter s/Value** pairs in the dialog that opens. If a data source is available, you can parameterize the values as well.

Security

Security> Client side SSL page options include:

- **Use Client Key Store:** Specifies the key store used to complete the handshake with the server.

Security> HTTP Authentication page options include:

- **Perform Authentication:** To set up basic, NTLM, Digest, or Kerberos authentication, select the **Perform Authentication** check box, then select **Basic, NTLM, Kerberos,** or **Digest** from the **Type** drop-down list.
  - For **Basic, NTLM,** or **Digest**, enter the **Username** and **Password** to authenticate the request.
  - For **Kerberos**, enter the **Service Principal** to authenticate the request. If the correct username and password, or the correct service principal, are not used, the request will not be authenticated.
- **Use Global Preferences:** Alternatively, you can select **Use Global Preferences** if you have set Global HTTP Authentication Properties within the Security Preferences. For more information, see Security Settings.

Security> OAUth Authentication page options include:

- **Perform Authentication:** Enabling this option indicates that OAuth Authentication should be performed. An Authentication field containing OAuth specific information will be added to the HTTP Header.
- **Consumer Key and Secret Configuration:** The Consumer Key and Consumer Secret are the credentials that the client uses to validate itself with the server. The Consumer Key is unique to each client using it. Both of these are required at all steps.
- **OAuth Authentication Mode:** Specifies what step of the OAuth Scenario you'd like to perform.
  - **Obtain Request Token:** Requests the Request Token from the server using the Consumer Key and Secret.
  - **Scope:** Restricts what information may be accessed. This information in embedded into the Consumer Key.
  - **Exchange Request Token for Access Token:** Exchanges the Request Token plus the verification code for the Access Token.
- **Request Token:** Specifies Temporary Request Token credentials obtained from the server (used to exchange for the Access Token).
- **Request Token Secret**: Specifies Temporary Request Token credentials obtained from the server (used to exchange for the Access Token).
- **Verification Code:** Specifies the verification code provided by the server; this confirms that the resource owner will grant permission.
  - **Sign Request for OAuth Authentication:** Uses the specified Access Token and Access Token Secret to give the client access to the user's private resources.
- **OAuth Parameters:** Allows you to specify additional parameters on the OAuth Token— for example, the timestamp and nonce.

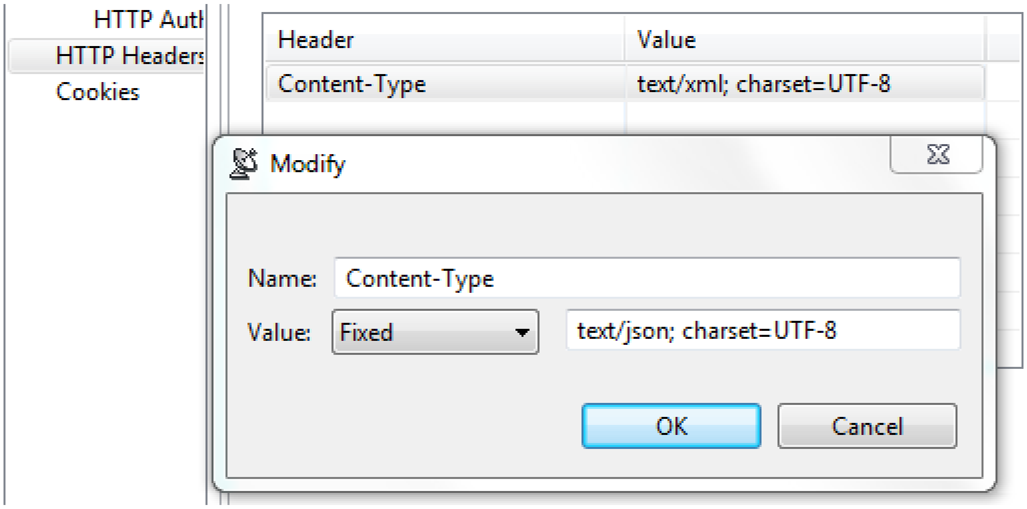
For details on using OAuth authorization, see Using OAuth Authentication.

HTTP Headers

HTTP Headers page options include:

- **Add:** Click to add a custom HTTP header.
- **Modify:** Click to modify the selected HTTP header. A dialog box will display, allowing you to modify the Name and Value of the header. If the tool is using a data source, values for the header can be accessed from the data source.
- **Remove:** Click to delete the selected HTTP header.

These controls are used to override header fields. For example, you can override the Content-Type header field by specifying the desired name and value via these controls.



Header	Value
Content-Type	text/xml; charset=UTF-8

Name: Content-Type

Value: Fixed text/json; charset=UTF-8

The following header fields, which are added by default, can be overridden via these UI controls

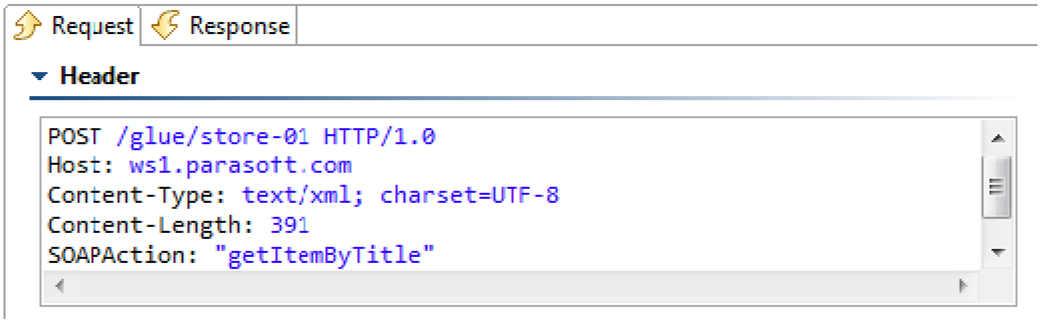
Host

The value will contain the host name and port number from the HTTP endpoint or resource URL.

Content-Type

The media type of the outgoing message. This header is only sent if the outgoing message contains a body which is controlled by the HTTP method. A body is sent for POST, PUT, and DELETE methods and not for GET, OPTIONS, HEAD, or TRACE.

The default value is determined based on the type of message being sent. The content-type of an SOAP message will vary depending on the SOAP version, "text/xml" for SOAP 1.1 or "application/soap+xml" for SOAP 1.2. Other XML messages will use "text/xml" by default. JSON messages will use "application/json". A message configured using the Table view will use "application/x-www-form-urlencoded". A message sent with MIME attachments will contain a "multipart" content-type with "start" and "boundary" parameters. Messages belonging to EDI, Fixed Length, CSV, or Custom message formats will have the media type for the message format.



```
POST /glue/store-01 HTTP/1.0
Host: ws1.parasoft.com
Content-Type: text/xml; charset=UTF-8
Content-Length: 391
SOAPAction: "getItemByTitle"
```

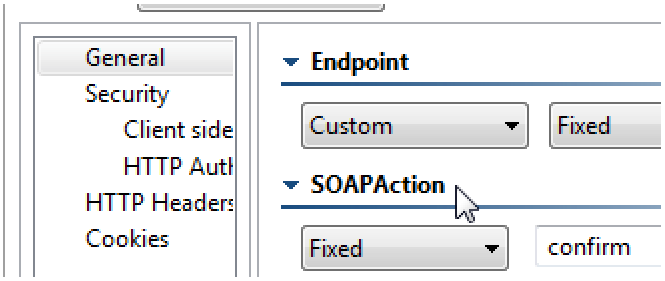
Content-Length

The size of the outgoing message in bytes. This header is not sent if "chunked" transfer encoding is enabled.

The following HTTP headers are configured conditionally. They are configured outside of this table or have values that must be dynamically generated.

SOAPAction

This HTTP header is sent for SOAP 1.1 only. It is set in the SOAPAction field of the **General** page



Authorization

This header is constructed automatically based on the HTTP Authentication and OAuth settings specified in your preferences (under Security> HTTP Authentication and OAuth). The value for NTLM, Digest, and Kerberos authentication will vary depending on various factors, including dynamically-generated challenge responses and security tokens.

Connection

This header is added to the message with the value of close if **Close connection** is enabled. This header is not sent if **Keep-Alive connection** is enabled (this is the default). Keep-Alive must be enabled for NTLM and Digest HTTP authentication.

Proxy-Authorization

This header is constructed based on the Proxy Authentication settings in the preferences and whether the server indicated that proxy authentication is required.

Cookies

Cookies page options include:

- **Reset existing cookies before sending request**: Allows you to clear the current cookies so that next HTTP invocations start a new session.

Error Handling

Under normal conditions, test cases using HTTP 1.1 Keep-Alive will reuse a single connection for the duration of a scenario. When a test case using HTTP 1.1 Keep-Alive times out while attempting to send or receive data, the client will issue a graceful close on the transport connection. The next test in the scenario will start a new connection and test execution will continue normally.