

Adding and Editing Message Responders

- [Adding a Message Responder](#)
- [Configuring a Message Responder](#)

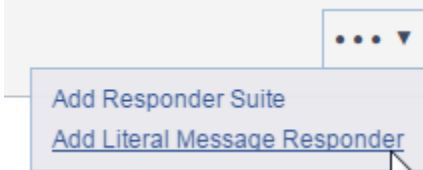
Directly from the Virtual Asset page, you can build, organize, and extend the Message Responder tools that specify which response a virtual asset should send for a given incoming request. Each responder responds to incoming request messages that match its correlation criteria. By customizing Message Responder options, you can customize the behavior of the virtual assets—with different request/response use cases specified manually or via data sources, error conditions, delays, and so forth.

This functionality is intended to provide a fast way to create new Message Responders which can be configured in "literal mode" (i.e., by specifying a literal message, such as a sample message you paste in and adapt). For more advanced functionality, open the related .pva in Virtualize Desktop.

Adding a Message Responder

To add a new Message Responder tool:

1. In the left pane, select the .pva or responder suite where you want the new Message Responder added.
2. Choose **Add Literal Message Responder** from the page-level action menu.



3. (Optional) Modify the name of the newly-created tool.
4. Configure the tool as described below.
5. Save the new tool configuration.

The new responder will be added at the end of the selected responder suite.

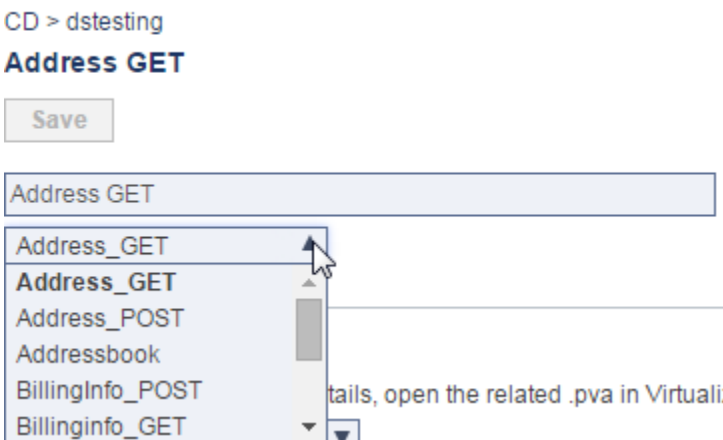
Configuring a Message Responder

Parameterizing Message Responders

Message Responder values can be parameterized using data sources or extracted values. See [Parameterizing with Data Source and Data Bank Values](#) for details.

To configure the response sent by a Message Responder tool:

1. If the containing virtual asset includes multiple data sources, you can review and change which data source is used to parameterize this responder. See [Parameterizing with Data Source and Data Bank Values](#) for details.



2. (Optional) In the Response area, add HTTP headers if desired.
 - a. Click **Add**.

- b. Specify a header name (this is case insensitive) and value. You can enter values in either table mode or literal mode.

Here is an example in literal mode:

HTTP headers: Literal

```
Content-Language: en-US
Access-Control-Allow-Origin: *
X-Target-Proxy: Transparent-PXYSGC05
Content-Type: application/json;charset=utf-8
Server: IBM_HTTP_Server
```

Here is that same example in table mode.

HTTP headers: Table

<input type="checkbox"/> Header	Value
<input type="checkbox"/> Content-Language	en-US
<input type="checkbox"/> Access-Control-Allow-Origin	*
<input type="checkbox"/> X-Target-Proxy	Transparent-PXYSGC05
<input type="checkbox"/> Content-Type	application/json;charset=utf-8
<input type="checkbox"/> Server	IBM_HTTP_Server

3. Set the appropriate payload format and media type (in the **Content type** and **MIME type** boxes).

Format: JSON

Content type: application/json

Input mode: JSON Editor
JSON Editor
Literal

4. Specify the payload in literal text editor, the JSON editor, or the XML editor (see [Editing JSON Messages](#) and [Editing XML Messages](#) for details and tips).

```

1  {
2  "array": [
3    1,
4    2,
5    3
6  ],
7  "boolean": true,
8  "null": null,
9  "number": 123,
10 "object": {
11   "a": "b",
12   "c": "d",
13   "e": "f"
14 },
15 "string": "Hello World",
16 "a": 1, "b": 2
17 }
```

↕ ↕ ↶ ↷

- object {8}
 - array [3]
 - boolean : true
 - null : null
 - number : 123
 - object {3}
 - string : Hello World
 - a : 1
 - b : 2

Tip: Using {var_name}

If you're familiar with Virtualize or SOAtest, you can use the standard $\{var_name\}$ notation to reference variables and data source values that are defined for the responder suite. When you are working with JSON, a special notation is used for parameterizing a number or boolean field within a JSON message: $\{number:<value>\}$ or $\{boolean:<value>\}$. For example, to parameterize a number field with the column Count, you would use $\{number:Count\}$.

5. Configure correlations as described in [Adding and Editing Message Responders](#).
6. (Optional) Specify a request URL template. Enter a URL that is typical/representative of the URLs that the application under test would serve (and the responder should simulate). If the responder is created through the traffic wizard, this field will be populated with the URL from one of the requests in the traffic corresponding to this responder.



The value specified here will be used to configure URL Paths and URL Parameters correlation settings for responder correlations and data source correlations.

7. (Optional) Specify a request message template. This template will be used to automatically populate the expected response when generating XPath parameters in Virtualize (e.g. in the message request XPath dialogs for data source correlation and responder correlation or in the multiple response XPath dialog).
- When a Message Responder is created from traffic logs, the Request Message Template is generated automatically. The largest request message identified in the Traffic Log is used for this purpose.
 - When a Message Responder is NOT created from traffic logs, the template will be empty. In this case, you can manually modify the Request Message Template through the Literal Editor or XML Editor (e.g., by copying in a sample request message that correlates with the response message you're configuring). See [Editing XML Messages](#) for details and tips.
 - If the request message template is empty, it will be updated when you first select a value from the XPath builder (described in [Specifying XPaths](#)).
 - The completed Request Message Template is used to populate the views in the Edit XPath Function dialogs available from the Message Responder in Virtualize desktop. The XML Data Bank and XML Transformer output in Virtualize desktop will also use the template if it is attached to the Message Responder as a Request output.

Request template:

URL:	<input type="text" value="mybookstore/current_products/v3/saleable_quantity_by_location"/>
Message:	<input type="text" value="Literal"/>
	<pre>1 <?xml version="1.0" encoding="UTF-8"?> 2 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"> 3 <SOAP-ENV:Body> 4 <addItemToCart xmlns="http://bookstore.parasoft.com/soap/1.0"> 5 <itemId xmlns=""></itemId> 6 <quantity xmlns=""></quantity> 7 </addItemToCart> 8 </SOAP-ENV:Body> 9 </SOAP-ENV:Envelope></pre>