

Accessing URL Resources via the URL Adapter

i This page explains the **URL Adapter** in Bridge context. If you were looking for the same information regarding the **PAS Designer**, refer to **URL Adapter** in the Designer guide.

On this Page:

- [GET Resource](#)
- [POST Resource](#)
- [Tagged Values](#)

GET Resource

With the `<<URLAdapter>>` it is also possible to access URL resources. Getting e.g. an XML file follows the same rules as a "normal" GET request. The output is an object called "response" of type blob. The value of response is the XML data encoded in a base64 string.

If these XML data should be provided as a SOAP message to the client we have to use the `xmlToClass()` operation. The mapping between the XML data and SOAP classes is defined in a class diagram.

The following activity and deployment diagram shows a Service which gets data from <http://www.e2ebridge.com/bridgeFiles/address.xml> and sends back the following data structure:

```
<address id="120"> <street>Lautengartenstr. 12</street> <city>Basel</city>
</address>
```

Related Pages:

- [Setting cURL Options on the URL Adapter](#)
- [URL Adapter Reference](#)

The following rules are valid:

- XML attributes to class are mapped to class attributes.
- XML elements are mapped to class associations.

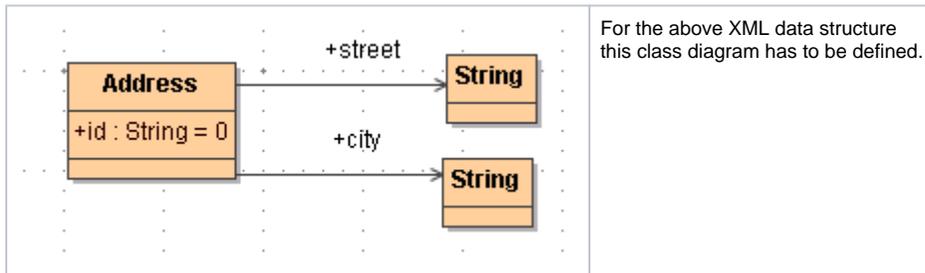


Figure: Get XML Data Via HTTP/GET

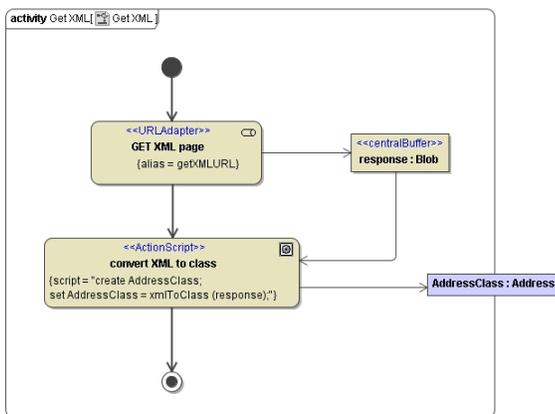


Figure: `<<URLAlias>>` for Getting XMLData

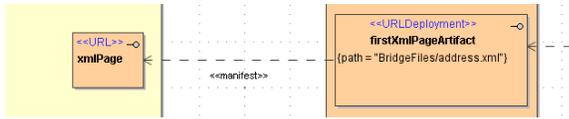


`<<URLAlias>>`

getXMLURLAlias

```
{host = "www.e2ebridge.com",
path = "BridgeFiles/address.xml",
protocol = http}
```

Figure: Component Diagram for Getting XMLData

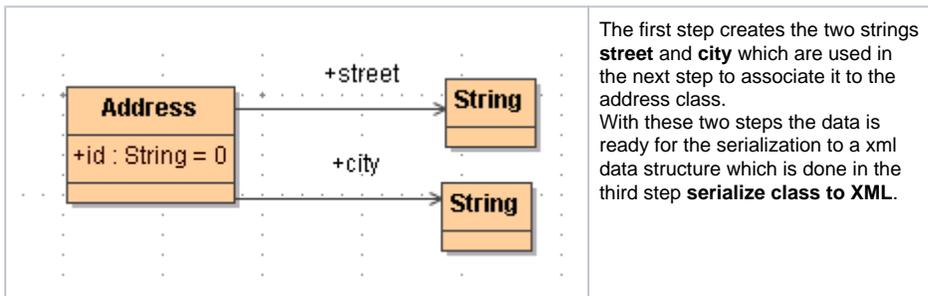
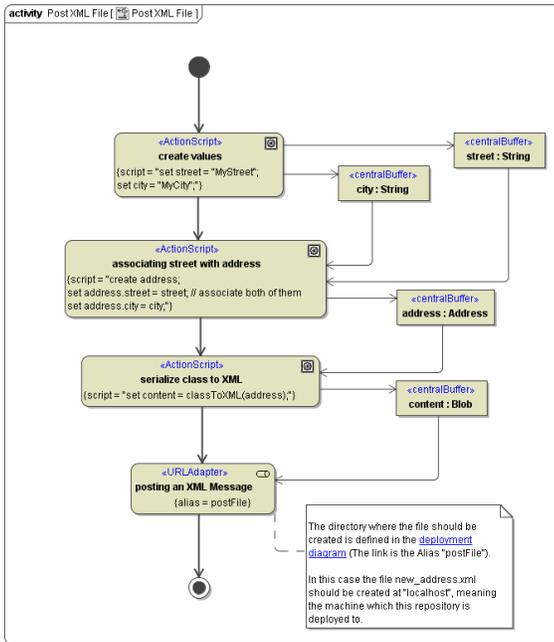


For more information on setting the parameters of the URL adapter dynamically refer to [URL Adapter Reference](#).

POST Resource

The following activity diagram shows how to create e.g. a .xml file with the URL adapter. Analogous to creating this file a HTTP/POST of xml data would be possible. The corresponding component diagram is shown above (refer to the component diagram above).

Figure: Posting (Writing) an XML File



The first step creates the two strings **street** and **city** which are used in the next step to associate it to the address class. With these two steps the data is ready for the serialization to a xml data structure which is done in the third step **serialize class to XML**.

The operation `classToXML()` returns a blob and expects as input parameter a class defining the xml data structure.

For more information on setting the parameters of the URL adapter dynamically refer to [URL Adapter Reference](#).

Tagged Values

Find below a list of relevant tagged values, if the URL adapter is used with the HTTP protocol. Default values used when an option is not explicitly set are written in bold.

Tagged Value	Description	Values	
protocol	Transport protocol.	http , https	
method	HTTP method.	get , post, put	
port	Machine port number the service is binding to. This port number can be given at service level only.	80	
path	HTTP path for the request.		
Advanced			
followRedirects	Maximum number of redirects to follow.	any integer	
options	Native cURL options.	cURL Options	
Authentication			
user	Username/password.		
Proxy			
proxyType	Type of the proxy.	HTTP, SOCKS5	
proxyURL	URL of the proxy server.		
proxyUser	Proxy user.		
SSL			
sslCAInfo	File name containing additional certificates for the connection verification (e.g. additional root CAs).		
sslCertificateFile	File name containing the client certificate.		
sslCertificateType	Type of the certificate.	PEM, DER	
sslPrivateKeyFile	File name containing the private key.		
sslPrivateKeyPassword	Password for the private key.		
sslPrivateKeyType	Type of the key.		
sslVerifyHost	Whether to verify the host information from the SSL connection.	On	Verification on.
		Off	Verification off.
sslVerifyPeer	Whether to verify the peer information from the SSL connection.	On	Verification on.
		Off	Verification off.