URL

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.

All URL adapter parameters, that are specified on the <<URLAlias>> in the component diagram, can also be supplied dynamically via action script. Find below a list of all relevant parameters of the URL adapter and their description.

With the URL adapter, it is possible to get and put messages via file transfer protocols: **FTP**, **FTPS** and **SFTP**. These protocols often get confused and mixed up due to the similarity in function and names. Please refer to Using the URL Adapter with the FTP Protocol and Related Protocols for a distinction and more information on the usage with the URL adapter.

On this Page:

Related Pages:

- Providing the SOAP Adapter with URL Parameter
- Using the URL Adapter with the FTP Protocol and Related Protocols

Tagged Values

<<URLAlias>>

Name	Description	Allowed Values / Examples	
Protocol (protocol)	Transport protocol.	one of: ldap, file, ftp, ftps, sftp, gopher, **http** (default), https, telnet	
Port (port)	Machine port number the service is binding to. This port number can be given at service level only.	a valid port number (default: **80**)	
Path (path)	HTTP path for the SOAP request.		
Host (host)	Host name. Default is "localhost" if no value is specified.	a valid hostname (default: **localhost**)	
Method (method)	HTTP method of the URL adapter call. Default is "POST" if no value is specified.	one of: get, **post** (default), read, write, put, list	
Advanced			
Follow Redirects (followRedirects)	The maximum number of redirects to follow.	any positive integer or 0	
Ignore Http Errors (ignoreHttpErrors)	If true, HTTP error codes > 300 will not cause an exception in the model. This implies, that the response body is accessible even if HTTP errors occur. The default value is false.	true	Ignore HTTP error codes > 300.
		false	Do not ignore HTTP errors (default).
Options (options)	Native cURL options as listed on Setting cURL Options on the URL Adapter . Use one of the following syntax rules: • values separated by `','` in one line • values separated by `' '` in one line • list of tagged values	valid cURL options, e.g. `CURLOPT_TIMEOUT=20`	
Request Http Header Roles (requestHttpHeaderRoles)	Builder 7.12.0 Runtime 2020.12 In the context of HTTP based adapters (URL, REST, SOAP), enable automatic header generation for the listed headers. These definitions overwrite the default behavior, and **X-Transaction-Id**, **X-Request-Id**, **X-Sender-Host** and/or **X-Sender-Service** will be substituted by this definition. **requestHttpHeaderRoles** can hold a list of definitions in format `<http header name>:<role>`, that will automatically be generated for each adapter call on this alias. `<role>` can be one of the listed allowed values (one list entry per line). Refer to HTTP Header Support > Overwriting the Standard HTTP Headers for more details on header roles.	client_host	Provide the client host in a header `<http header name>` instead of **X-Sender-Host**.
		client_service	Provide the client service in a header `<http header name>` instead of **X-Sender-Service**.

		correlation_id	Provide the correlation ID in a header <`http header name`> instead of **X-Request-Id**.
		transaction_id	Provide the transaction ID in a header <`http header name`> instead of **X-Transaction-Id**.
		passthrough	Pass a present header <`http header name`> to the called service.
		passthrough=<`request header name`>	Pass an incoming header <`request header name`> to the called service under the name of <`http header name`>. This is equivalent to renaming a header.
Digest Algorithm (digestAlgorithm)	Builder 7.12.0 Runtime 2021.1 Generates a HTTP **digest** header using the specified algorithm. When applied, a **digest** header is generated using the specified algorithm, and sent with the request . The generated header conforms with RFC3230 and RFC5843. Only one value is supported (no multi-value header).	None	No header generated.
		MD5	Generate header using MD5 algorithm.
		SHA	Generate header using SHA algorithm.
		SHA-1	Generate header using SHA-1 algorithm.
		SHA-256	Generate header using SHA-256 algorithm.
		SHA-512	Generate header using SHA-512 algorithm.
Authentication			
User (user)	Username/password.		
Proxy			
Proxy URL (proxyURL)	URL of the proxy server.		
Proxy User (proxyUser)	Proxy user.		
Proxy Type (proxyType)	Type of the proxy	one of: `HTTP`, `SOCKS5`	
SSL			
Ssl CA Info (sslCAInfo)	Name of the file containing additional certificates for the connection verification (e.g. additional root CAs).		
Ssl Certificate File (sslCertificateFile)	Name of the file that contains the client certificate.		
Ssl Private Key File (sslPrivateKeyFile)	Name of the file that contains the private key.		

Ssl Private Key Password (sslPrivateKeyPassword)	Password for the private key.		
Ssl Certificate Type (sslCertificateType)	Type of the certificate.	one of: **PEM** (default), DER, P12	
Ssl Verify Host (sslVerifyHost)	Whether to verify the host information form the SSL connection.	**On**	Verification on (default).
		Off	Verification off.
		Existence	Limit verification to the mere existence of the host.
Ssl Verify Peer (sslVerifyPeer)	Whether to verify the peer information from the SSL connection.	**On**	Verification on (default).
		Off	Verification off.
Ssl Private Key Type (sslPrivateKeyType)	Type of the private key.	one of: **PEM** (default), DER, ENG	

<<URLAdapter>>

Since Runtime 2022.6 the Runtime uses **Transfer-encoding: chunked** for POST requests. You can override this behavior by setting

- Transfer-Encoding: <empty string>
- Content-Length: <length of the body>"

Tagged Value	Type	Description	Allowed Values	
alias		Specify the URL alias the adapter should connect to.	any valid URL alias	
digestAlgorithm		Builder 7.12.0 Runtime 2021.1 Generates a HTTP **digest** header using the specified algorithm. When applied, a **digest** header is generated using the specified algorithm, and sent with the request . The generated header conforms with RFC3230 and RFC5843. Only one value is supported (no multi-value header).	None	No header generated.

MD5	Generate header using MD5 algorithm.
SHA	Generate header using SHA algorithm.
SHA-1	Generate header using SHA-1 algorithm.
SHA-256	Generate header using SHA-256 algorithm.

			SHA-512	Generate header using SHA-512 algorithm.
headerParameters	Array of HeaderField	Supply additional header information for the URL adapter call in form of name and value pairs (see type **Header Field** in table URL Parameter Types below. The xUML Runtime assigns a correlation ID to each adapter call. This ID is stored in header field **X-Bridge-CorrelationID**. Adapter calls can be identified by this ID. Also, it is logged to the transaction log. If you set this header manually, the new header will be send to the target URL. Nevertheless, the correlation ID logged to the transaction log will still be the one generated by the Runtime!		
streaming		Specify a streaming method for the usage of file streaming with the URL adapter.	**none**	No file streaming (default).
			fromFile	Stream to a file.
			toFile	Stream from a file.

URL Adapter Parameters

Incoming Parameters

Name	Type	Direction	Restrictions		Description
			to listed protocol only	to listed method only	

url	string	in			This parameter contains the URL that will be accessed by the adapter. The URL contains protocol, port and path.
content	Blob	in		post, put, write	Use this parameter to supply the content to be used by the listed methods.
headerParameters	Array of HeaderField	in	http, https		Use this parameter to supply additional header information for the URL adapter call in the form of name and value pairs.
method	String	in			This parameter provides the method of the URL adapter call.
authentication	Authentication	in	ldap, ftp, ftps, sftp, http, https		This parameter provides an object of type **Authentication** containing the user and the password.
proxy	Proxy	in	ldap, ftp, ftps, sftp, http, https		Use this parameter to supply necessary proxy information.
followRedirects	Integer	in	http, https		Specify here the maximum number of redirects to follow.
ssl	SSL	in	https, ftps		Use this parameter to supply SSL information.
commands	Array of String	in	ftp, ftps		Use this parameter to specify ftp commands. All commands are executed after executing the ftp method. For a list of available ftp commands see RFC 959.
options	Array of Option	in	ldap, ftp, ftps, sftp, http, https		Use this parameter to specify cURL options for the URL adapter call.
streamSource	String	in	ftp, ftps, sftp, http, https	post, put	If you want to use file streaming with streaming method **fromFile**, supply the streaming source here. Then, **content** is not used.
streamTarget	String	in	ftp, ftps, sftp, http, https	get, post, put, list	If you want to use file streaming with streaming method **toFile**, supply the streaming target here. Then, **response** is not used.
ignoreHTTPErrors	Boolean	in	http, https		If true, HTTP error codes > 300 will not cause an exception in the model. This implies, that the response body is accessible even if HTTP errors occur. The default value is false.

Outgoing Parameters

Name	Type	Direction	Restrictions		Description
			to listed protocol only	**to listed method only**	
response	Blob	out		get, post, put, list, read	Contains the response content in relation to the used method.
httpStatus	Integer	out	http, https		Contains the HTTP status code of the response.
httpHeaderParameter	Array of HeaderField	out	http, https		DeprecatedThis attribute is deprecated as of Runtime 2020.11. Please use **httpHeaderMap** (see below) for new implementations as its implementation complies to the HTTP specification. Contains the HTTP headers of the response.

httpHeaderMap	Map of Entry	out	Map of Entry		Runtime 2020.11 Header information as a map. The map contains arrays of header value strings whereas the header name is the key of the map. • Header names are lowercase and treated case insensitive. • Multiple headers with the same name are treated as arrays. Refer to HTTP Header Support for more information on the standard xUML HTTP headers.

URL Parameter Types

Class	Attribute	Type	CURL Option	Description	Values/Example	
Authentication	password	String		Password needed for authentication.		
	username	String		Username needed for authentication.		
Entry	key	String		Key of the map entry.		
	value	Array of Any		List of values of the map entry. The dynamic type for **httpHeaderMap** is **String**.		
Header Field	name	String		Name of the header field.		
	value	String		Value of the header field.		
Option	name	String		Native cURL option.	`CURLOPT_CONNECTTIMEOUT`	
	value	String		Value of the cURL option.	`10`	
Proxy	authentication	Authentication	CURLOPT_PROXYUSERPWD	Proxy user.		
	type	String	CURLOPT_PROXYTYPE	Type of the proxy.	HTTP, SOCKS5	
	url	String	CURLOPT_PROXY	URL of the proxy server.		
SSL	caInfo	String	CURLOPT_CAINFO	Name of the file containing additional certificates for the connection verification (e.g. additional root CAs).		
	certificate	Certificate		Client certificate		
	key	Key		Private key of client certificate		
	verifyHost	String	CURLOPT_SSL_VERIFYHOST	Whether to verify the host information form the SSL connection.	**On**	Verification on (default).
					Off	Verification off.
					Existence	Limit verification to the mere existence of the host.
	verifyPeer	String	CURLOPT_SSL_VERIFYPEER	Whether to verify the peer information from the SSL connection.	**On**	Verification on (default).
					Off	Verification off.
Key	file	String	CURLOPT_SSLKEY	Name of the file containing the private key.		
	password	String	CURLOPT_KEYPASSWD	Password for the private key.		
	type	String	CURLOPT_SSLKEYTYPE	Type of the key.	PEM, DER, ENG	
Certificate	file	String	CURLOPT_SSLCERT	Name of the file containing the client certificate.		

	type	String	CURLOPT_SSLCERTTYPE	Type of the certificate.	**PEM**, DER, P12