Using the URL Adapter with the FTP Protocol and Related Protocols

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.

With the URL adapter, it is possible to get and put messages via the FTP protocol. The input and output of this adapter is always of type **Blob** to support all possible data types (e.g. binary for images, text with HTML code, XML messages, etc.).

Example File (Builder project Add-ons/URL):

<bridgeDocPath>\Add-ons\URL\uml\urlFtp.xml

FTP commands like **RNFR**, **RNTO**, etc. are also supported (see also http://www.faqs.org/rfcs/rfc959.html).

Differences between FTP, FTPS, and SFTP

A lot of times these protocols get confused and mixed up due to the similarity in function and names. The Bridge supports the following file transfer protocols:

FTP	FTPS	SFTP
File Transfer Protocol	FTP Secure and FTP-SSL	SSH File Transfer Protocol
Classic insecure FTP	FTP over TLS/SSL	Extension of SSH
Plain FTP	Plain FTP over TLS/SSL channel	Has nothing in common with original FTP
Username and password are sent as clear text over the network	User name and password are encrypted	User name and password are encrypted
Transfer is not encrypted	Transfer is encrypted	Transfer is encrypted
Uses TCP port 21	Uses TCP port 21 or 990	Uses TCP port 22

FTP via SSH is not supported by the Bridge.

Transfer Mode

The transfer mode by default is set to **binary**, only the **list** command is transferred in ASCII mode.

In order to change the transfer mode, use the cURL option **CURLOPT_QUOTE** (see chapter Setting cURL Options on the URL Adapter). Using this option, the transfer mode is changed before any method is executed. Using the command **TYPE Ib** in the command array, these commands are executed after any method like put, get, or list.

Streaming FTP

When reading or writing files, the examples load the whole file into memory. This might cause problems if these files are big.

In order to avoid consuming too much memory the URL adapter offers the option to read and write data in little chunks. This mechanism is called streaming, since, after reading a small data chunk from the FTP server, it is immediately written to a file before doing the next read. For sending data, the mechanism works analogously. The above example model contains a use case **File Streaming** that shows how to receive and send big files without using much memory.

On this Page:

Related Pages:

- SFTP
- FTPS

- Setting cURL Options on the URL Adapter
- URL Adapter Reference

Tagged Values

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

Tagged Value	Description	Values
protocol	Transport protocol.	ftp
method	FTP method.	**GET**, POST, PUT, LIST (PUT will internally be to POST)
port	Machine port number the service is binding to. This port number can be given at service level only.	**21**
path	FTP path for the request.	
Advanced		
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.	