Other Internet Commands

The Internet library has a number of additional commands for working with web forms, ftp commands, custom settings and troubleshooting. These commands are documented in more detail the LiveCode Dictionary.

Launching the User's Browser with a URL

To launch the default browser with a URL, use the **launch URL** command.

```
launch URL “http://www.livecode.com/"
```

Note: To render web pages within LiveCode, instead of launching an external browser, use the revBrowser. See the section on revBrowser for more information.

Working with Web Forms

To post data to a web form, use the **post** command. To encode data to make it suitable for posting, use the **libUrlFormData** function.

To create multi-part form data (as described in RFC 1867) use the **libUrlMultipartFormData** function.

To add data to a multipart form one part at a time, use the **libUrlMultipartFormAddPart** function.

This can be useful if you need to specify the mime type or transfer encoding for each part.

Working with FTP

For details on basic uploading and downloading using FTP, see the section above.

The following commands provide additional capabilities when working with the ftp protocol:

- **libURLSetFTPStopTime** : Sets the timeout value for FTP transfers.
- **libURLSetFTPMode** : Switches between active and passive mode for FTP transfers.
- **libURLSetFTPListCommand** : Switches between sending LIST or NLST formats when listing the contents of an FTP directory.
- **libURLftpCommand** : sends an ftp command to an ftp server.
- **libURLftpUpload** : uploads data. See the section above for more details.
- **libURLftpUploadFile** : uploads a file, without loading the entire file into memory. See the section above for more details.
- **libURLDownloadToFile** – downloads data to a file, without loading the entire data into memory. See the section above for more details.

HTTP methods and http URLs

The basic operations used by the HTTP protocol are called methods. For **http** URLs, the following HTTP methods are used under the following circumstances:

- GET: when an **http** URL in an expression is evaluated
- PUT: when you put a value into an **http** URL
- POST: when you use the **post** command
- DELETE: when you use the **delete URL** command with an **http** URL

Note: Many HTTP servers do not implement the PUT and DELETE methods, which means that you can't put values into an **http** URL or delete an **http** URL on such servers.

It's common to use the FTP protocol instead to upload and delete files; check with your server's administrator to find out what methods are supported.

HTTP headers

When LiveCode issues a GET or POST request, it constructs a minimal set of HTTP headers.

For example, when issued on a Mac OS system, the statement:

```
put URL "http://www.example.org/myfile" into myVariable
```

results in sending a GET request to the server:

```
GET /myfile HTTP/1.1 Host: 127.0.0.0 User-Agent: LiveCode (MacOS)
```

You can add headers, or replace the Host or User-Agent header, by setting the **HTTPHeaders** property before using the URL:

```
set the HTTPHeaders to "User-Agent: MyApp" & return
    & "Connection: close"
put URL "http://www.example.org/myfile" into myVariable
```

Now the request sent to the server looks like this:
GET /myfile HTTP/1.1 Host: 127.0.0.0 User-Agent: MyApp
Connection: close

The **ftp** URL scheme can be used to create a new file to an FTP server.

As with the **file** and **binfile** schemes, putting something into the URL creates the file:

```
put dataToUpload into URL
“ftp://jane:pass@ftp.example.com/newfile.dat"
```

Tip: You can create an FTP directory by uploading a file to the new (nonexistent) directory. The directory is automatically created. You can then delete the file, if you wish, leaving a new, empty directory on the server:

```
-- Create an empty file in the nonexistent directory:
put empty into URL "ftp://jane:pass@example.com/newdir/dummy"

-- Delete unwanted empty file to leave new directory:
delete URL “ftp://jane:pass@example.com/newdir/dummy"
```

Additional Transmission Settings

The following commands provide additional customization options for the Internet library:

- **libUrlSetExpect100** : Allows you to set a limit to the size of data being posted before requesting a continue response from the server.

- **libURLSetCustomHTTPHeaders** : Sets `the header`s to be sent with each request to an HTTP server. See also the section on HTTPHeaders above.

- **libURLFollowHttpRedirects** : Specify that GET requests should follow HTTP redirects and GET the page redirected to.

- **libUrlSetAuthCallback** : Sets a callback for handling authentication with http servers and proxies.

Troubleshooting

The following commands and functions can be useful when debugging an application that uses the Internet library.

- **resetAll** : Closes all open sockets and halts all pending Internet operations.

Caution: The **resetAll** command closes all open sockets, which includes any other sockets opened by your application and any sockets in use for other uploads and downloads. Because of this, you should avoid routine use of the **resetAll** command.

Consider using it only during development, to clear up connection problems during debugging.

- **libURLErrorData** : Returns any error that was caused during a download that was started with the load command.
- **libURLVersion** : Returns the version of the Internet library.
- **libURLSetLogField** : Specifies a field for logging information about uploads and downloads on screen.
- **libURLLastRHHeaders** : Returns the headers sent by the remote host in the most recent HTTP transaction.
- **libURLLastHTTPHeaders** : Returns the value of the httpHeadersproperty used for the previous HTTP request.