

## Downloading a File

- [Alyce](#)  
by Alyce Watson

## Table of Contents

[Downloading a File](#)

[Valid URL?](#)

[Doing the Download](#)

[What do we do now?](#)

[Downloading Other File Types](#)

[Parsing the HTML](#)

[DEMO](#)

The API call URLDownloadToFileA can be used to download files from the internet and save them on the user's computer.

## Valid URL?

We can download just about any type of file that is online, but before we attempt to download an HTML file, we'll check to see if the URL is valid. URL stands for Universal Resource Locator, and it is the address on the internet of the web page we want to download.

The function to test for a valid URL is called IsValidURL and it is part of "URLmon.DLL". The validation function is for HTML files, not for other types of files online. This DLL is not one of the standard DLLs that is recognized by Liberty BASIC, so we must open it and give it a handle in order to use it, and close it when it is no longer needed.

```
open "URLmon" for dll as #url
'code to call on the DLL
close #url
```

The API call is encapsulated in a Liberty BASIC function below. There are three arguments. The first and third arguments are ignored and must be set to 0. The second argument is a string containing the URL to check. The URL for the Liberty BASIC site is <http://www.libertybasic.com/> URLs always begin with "http", which stands for HyperText Transfer Protocol." The return from the IsValidURL function is 0 if the URL is valid. Here is an extract from the demo program.

The IsValidURL function requires a unicode (also called "wide") character string. Liberty BASIC uses ASCII strings, so the URL string requires conversion before it is sent to the function. This conversion can be accomplished with the API function from kernel32.dll called MultiByteToWideChar. I've adapted some code provided by Stefan Pendl that uses this function to convert strings.

The function is as follows. If the function succeeds, it returns the unicode string. If it fails, it returns an empty string, and the demo program traps that error. Each argument is commented.

```
function MultiByteToWideChar$(String$)
    'converts any string into unicode
    CodePage = 0 : dwFlags = 0
    cchMultiByte = -1
    lpMultiByteStr$ = String$
    cchWideChar = len(String$) * 3
    lpWideCharStr$ = space$(cchWideChar)

    call dll #kernel32, "MultiByteToWideChar", _
    CodePage as ulong, _      'CP_ACP=0, ansi code page
    dwFlags as ulong, _      'use 0, flags for character translation
    lpMultiByteStr$ as ptr, _ 'the ascii string to convert
    cchMultiByte as long, _  'len of string, -1 for null-
terminated string
    lpWideCharStr$ as ptr, _ 'buffer for returned ansi string
    cchWideChar as long, _  'size in wide characters of string buffer
    result as long
    'returns number of wide characters written to buffer

    if result = 0 then
        MultiByteToWideChar$ = ""
    else
        MultiByteToWideChar$ = left$(lpWideCharStr$, result * 2)
    end if
end function
```

To call the function:

```
u$ = MultiByteToWideChar$("http://www.libertybasic.com/index.html")

if u$ = "" then
    print "Unable to convert URL to unicode string."
    end
end if

'if result = 0, URL is valid
result = ValidURL(u$)
```

If the result is equal to 0, the URL is valid. The function itself:

```
Function ValidURL(urlfile$)
    open "URLmon" for dll as #url
    call dll #url, "IsValidURL",_
    0 as long,_
        'ignored, must be 0
    urlfile$ as ptr,_
        'urlfile to check
    0 as ulong,_
        'ignored, must be 0
    ValidURL as long
    close #url
end function
```

## Doing the Download

Once we've determined that a URL is valid, we can download the file. We'll use another function from "URLmon.DLL" called URLDownloadToFileA. As before, the DLL must be opened, used, then closed.

```
open "URLmon" for dll as #url
'code to call on the DLL
close #url
```

The URLDownloadToFileA function is set up to work either with or without an ActiveX control. Liberty BASIC cannot use ActiveX controls natively, so three of the arguments for use with ActiveX controls must be set to null, or 0. There are two arguments we'll need that require strings as PTRs. The "urlfile\$" argument is the URL, or web address of the file to download. The "localfile\$" argument is the disk filename where the downloaded file will be saved. Downloaded files must be saved to disk.

Here is the API call, encapsulated in a Liberty BASIC function. The API call returns 0 if it is successful in downloading a file and saving it to disk. The demo calls the function like this:

```
result = DownloadToFile(  
"http://www.libertybasic.com/index.html", DefaultDir$ + "\test.htm")
```

The function itself:

```
Function DownloadToFile(urlfile$, localfile$)  
    open "URLmon" for dll as #url  
    call dll #url, "URLDownloadToFileA", _  
        0 as long, _           'null  
        urlfile$ as ptr, _    'url to download  
        localfile$ as ptr, _  'save file name  
        0 as long, _           'reserved, must be 0  
        0 as long, _           'callback address, can be 0  
    DownloadToFile as ulong  '0=success  
    close #url  
end function
```

## What do we do now?

The little demo below downloads an HTML file and saves it to disk. It then opens it, reads the contents into a string variable and displays the text in the mainwin. We can do anything with the downloaded files that we can do with a file on disk. If it is an image file, we can run a Paint program and pass the filename into the RUN command to display it.

```
RUN "mspaint " + chr$(34) + filename$ + chr$(34)
```

If it is an HTML file, we can use RUN to run Explorer with the file loaded.

```
RUN "explorer " + chr$(34) + filename$ + chr$(34)
```

We can open the file and parse the text to get information.

```
open "test.htm" for input as #f  
txt$ = input$(#f, LOF(#f))  
close #f
```

```
'parse the string of text now
```

## Downloading Other File Types

We can download other types of files from the internet. The demo program downloads the Liberty BASIC banner, which is an image file in JPEG format. We can download ZIP files, TEXT files, GIF image files, and so on.

Keep in mind that downloading the HTML file for a web page does not download the complete webpage. Most web pages include other types of files, such as images, style sheets (CSS files), music files, etc. If we want to get all files associated with a web page, we must open the HTML file that was saved to disk and read the text into a string variable. We must then parse this string to find the names of other files that are needed.

## Parsing the HTML

Once you have the file on disk, you may want to read the contents to check for images, links, etc. Jerry Muelver has written a demo on parsing, here: [Parsing Text](#).

## DEMO

```
'Download html from given
'URL to file on disk.
'
'Minimum availability Internet Explorer 3.0
'Minimum operating systems Windows NT 4.0, Windows 95
'

'check for valid URL for HTML page:
u$ = MultiByteToWideChar$("http://www.libertybasic.com/index.html")

if u$ = "" then
    print "Unable to convert URL to unicode string."
    end
end if

'if result = 0, URL is valid
result = ValidURL(u$)

if result = 0 then
    'download html from libertybasic.com
    downloadresult = DownloadToFile(
```

```
"http://www.libertybasic.com/index.html", DefaultDir$ + "\test.htm")
    if downloadresult <> 0 then print
    "Error downloading libertybasic.com HTML file.

    open DefaultDir$ + "\test.htm" for input as #f
    test$ = input$(#f, LOF(#f))
    close #f

    print test$ 'print HTML from file into mainwin
    print : print
    print
"The text above was downloaded from http://www.libertybasic.com/"
else
    print "Invalid URL:"; u$
end if

'now attempt to download the banner:
result = DownloadToFile(
"http://www.libertybasic.co
m/lb3banner.jpg", DefaultDir$ + "\banner.jpg")
if result <> 0 then
    print "Error downloading Liberty BASIC banner."
else
    run "mspaint.exe " + chr$(34) + DefaultDir$ +
"\banner.jpg" + chr$(34)
    print : print : print "Done"
end if

wait

Function DownloadToFile(urlfile$, localfile$)
    open "URLmon" for dll as #url
    calldll #url, "URLDownloadToFileA",_
    0 as long,_
        'null
    urlfile$ as ptr,_
        'url to download
    localfile$ as ptr,_
        'save file name
    0 as long,_
        'reserved, must be 0
    0 as long,_
        'callback address, can be 0
    DownloadToFile as ulong  '0=success
    close #url
end function

Function ValidURL(urlfile$)
    open "URLmon" for dll as #url
    calldll #url, "IsValidURL",_
    0 as long,_
        'ignored, must be 0
```

```
urlfile$ as ptr,_      'urlfile to check
0 as ulong,_           'ignored, must be 0
ValidURL as long
close #url
end function

function MultiByteToWideChar$(String$)
  'converts any string into unicode
CodePage = 0 : dwFlags = 0
cchMultiByte = -1
lpMultiByteStr$ = String$
cchWideChar = len(String$) * 3
lpWideCharStr$ = space$(cchWideChar)

call dll #kernel32, "MultiByteToWideChar", _
CodePage as ulong, _      'CP_ACP=0, ansi code page
dwFlags as ulong, _       'use 0, flags for character translation
lpMultiByteStr$ as ptr,_ 'the ascii string to convert
cchMultiByte as long, _  'len of string, -1 for null-
terminated string
lpWideCharStr$ as ptr, _ 'buffer for returned ansi string
cchWideChar as long, _  'size in wide characters of string buffer
result as long
'returns number of wide characters written to buffer

if result = 0 then
  MultiByteToWideChar$ = ""
else
  MultiByteToWideChar$ = left$(lpWideCharStr$, result * 2)
end if
end function
```