

# Length of a File on the Internet

[BrentDT](#)

*Originally published in NL 136*

[Length of a File on the Internet](#) | [Notes from Brent](#) | [Modifications](#) | [Demo](#)

---

## Notes from Brent

*Brent says to consider the following if you are using this code to implement a file updater.*

I want to point out that using the last modified date of a file would be a better indicator of a file's freshness than its size. The size may not change between updates, especially if an update involves correcting typos in text (the "teh" for "the" variety) or changing BMP files.

But comparing the date returned by the server to the local date and time doesn't always ensure problem-free updating because there is no guarantee that the local date and time are accurate. And there's no guarantee that the server is accurate, for that matter. It would be most appropriate to compare server time to server time between updates, thus requiring the storing of the most recently retrieved file date. But if it is the first update check, the only option is to use the local time for comparison.

## Modifications

Here are the modifications to my demo to query the file date, instead of its length:

```
---[ FIND ]---  
 5 As ULong, _ 'HTTP_QUERY_CONTENT_LENGTH  
  
---[ REPLACE WITH ]---  
 11 As ULong, _ 'HTTP_QUERY_LAST_MODIFIED  
  
---[ FIND ]---  
 Print Buffer$; " Bytes"  
  
---[ REPLACE WITH ]---  
 Print Buffer$
```

## Demo

Text here.

```
'Retrieving HTTP Content-length Demo
'By Brent D. Thorn, 8/2005

remoteFile$ =
"http://babek.info/libertybasicfiles/images/lbfor2.gif"

BUFFER.SIZE = 32 'size of buffer to hold length string

Open "wininet" For DLL As #inet

'It would be a good idea to test for an Internet
'connection before proceeding. This demo assumes
'you have one already.

' Register a new user agent. Use proxy settings
' set up in Internet Options Control Panel.
CallDLL #inet, "InternetOpenA", _
    "My User Agent" As Ptr, _
    0 As Long, _ 'INTERNET_OPEN_TYPE_PRECONFIG
    _NULL As Long, _
    _NULL As Long, _
    0 As ULong, _
    hInet As ULong
If hInet = 0 Then [ErrExit]

' Open a request to a remote file. This does not
' download the file, but simply requests its
' headers, one of which should be "Content-length."
Struct pdw, Context As ULong
CallDLL #inet, "InternetOpenUrlA", _
    hInet As ULong, _
    remoteFile$ As Ptr, _
    _NULL As Long, _
    0 As Long, _
    0 As ULong, _
    pdw As Struct, _
    hRequest As ULong
If hRequest = 0 Then [ErrExit]

' Query Content-length from the headers and copy
' the number of bytes to a string buffer. One could
' call this function with an empty buffer and it
' will fill pdw.BufferLength.struct with the required
' buffer size.
Struct pdw, BufferLength As ULong
pdw.BufferLength.struct = BUFFER.SIZE
```

```
Buffer$ = Space$(BUFFER.SIZE)+Chr$(0)
Struct pdw2, Index As ULong
pdw2.Index.struct = 0
Call1DLL #inet, "HttpQueryInfoA", _
    hRequest As ULong, _
    5 As ULong, _ 'HTTP_QUERY_CONTENT_LENGTH
    Buffer$ As Ptr, _
    pdw As Struct, _
    pdw2 As Struct, _
    ret As Long
If ret = 0 Then [ErrExit]

Buffer$ = Left$(Buffer$, pdw.BufferLength.struct)

Print Buffer$; " Bytes"

[ErrExit]
    ' If you get an error, find GetLastError on MSDN. There
    ' you will find a link to a list of error codes.
Call1DLL #kernel32, "GetLastError", ret As ULong
If ret Then Print "Error ";ret

    ' Free the handles we created and exit.

If hRequest Then _
Call1DLL #inet, "InternetCloseHandle", _
    hRequest As ULong, ret As Long

If hInet Then _
Call1DLL #inet, "InternetCloseHandle", _
    hInet As ULong, ret As Long

Close #inet

End
```

[Length of a File on the Internet](#) | [Notes from Brent](#) | [Modifications](#) | [Demo](#)