

Retrieving HTTPS Data

by Brent Thorn -

[BrentDT](#)

Originally in Liberty BASIC Newsletter #37

This demo reads a remote file into memory. Hopefully Yahoo doesn't mind my choice of URLs. Please note the strange hanging behavior at the "Close #inet" statement.

After doing a little testing, I found my hanging was the result of some ad-blocking software I have installed. When I disabled it the hanging ceased. It may be rare but you may want to keep this possibility in mind when using my code in your own programs. I can only suggest not closing #inet until after all your windows are closed at exit to avoid any possibility of a user's thinking your app is faulty.

Demo

```
' CODE BEGINS
'Retrieving HTTPS Data Demo
'By Brent D. Thorn, 9/2005

remoteFile$ = "https://login.yahoo.com/"

CHUNK.SIZE = 512 ' enough bytes to hold a single line of HTML

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
CallDLL #inet, "InternetOpenUrlA", _
```

```
hInet As ULong, _
remoteFile$ As Ptr, _
_NULL As Long, _
0 As Long, _
2147483648 As ULong, _ 'INTERNET_FLAG_RELOAD
_NULL As Long, _
hFile As ULong
If hFile = 0 Then [ErrExit]

' Start reading in the file by chunks.
Buffer$ = ""
chunk$ = Space$(CHUNK.SIZE)+Chr$(0)
Struct pdw, NumberOfBytesRead As ULong
Do
    CallDLL #inet, "InternetReadFile", _
        hFile As ULong, _
        chunk$ As Ptr, _
        CHUNK.SIZE As ULong, _
        pdw As Struct, _
        ret As Long

    If pdw.NumberOfBytesRead.struct = 0 Then Exit Do

    Buffer$ = Buffer$ + Left$(chunk$,
pdw.NumberOfBytesRead.struct)
    Loop Until 0

Print Buffer$

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

    ' Free the handles we created and exit.

If hFile Then _
CallDLL #inet, "InternetCloseHandle", _
    hFile As ULong, ret As Long

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

    ' Strangely, the next line causes LB to hang for around 45
```

```
' seconds on my computer, a 2.2 GHz running XP on dial-up.  
Close #inet  
  
End  
' CODE ENDS
```