EZTWain

EZTW32.DLL is a free DLL that allows your Windows programs to communicate with TWAIN scanners. EZTW32.DLL, commonly referred to as "Easy TWAIN Library" or just plain "EZTwain", is widely referenced on the web. The dll has been widely used since its introduction in 1994. EZTW32.DLL has been placed in the public domain by Dosadi.

Licensing: Dosadi places EZTwain Classic in the public domain, and makes no claim or warranty of any kind regarding this software. If you use this package, you assume all responsibility for the results or lack thereof.

The most current version of EZTW32.DLL is 1.18, dated July, 2008. Dosadi continues to make this dll available, but offers no support for it. Dosadi does provide support for it's commercial TWAIN DLL, EZTWAIN Pro. Visit Dosadi.com for FAQ's, tools, libraries, and forums.

The downloaded eztw1.zip file includes declaration files for several languages, including Access (VBA), Clarion, C#, dBase, Delphi, LotusScript, Perl, PowerBASIC, PowerBuilder, Progress4GL, VB, VB.NET and VFP. There are no declaration files included for Liberty BASIC. This article is intended to bridge that gap.

There are at least 53 functions in EZTW32.DLL. Many of these functions control the scanner device itself. Others are needed to interface with the more complex GUI structuring demanded by other languages. A few provide basic information (width, height, bitmap depth) of the scanned image itself. Fortunately, only a few functions are required for allowing your Liberty BASIC program to communicate with a scanner and access the images. Only a few of the functions will be addressed in this article. Others will be addressed in a planned follow up article. Much of the translation of Liberty BASIC functions comes from the PowerBASIC translations published by Don Dickinson. You can view Don's original code at http://www.powerbasic.com/support/forums/Forum8/HTML/000867.html. Alyce Watson offers a Liberty BASIC package demonstrating three of the most basic functions. You can download eztwaindemo at http://alycesrestaurant.com/Graphics.htm.

Getting Started with EZTW32.DLL

Load the DLL. You may want to check to be sure you have the most current version (v1.18 as of August, 2008). TWAIN_EasyVersion returns the version as an integer. Divide that integer by 100 to get the version.

Use the TWAIN_IsAvailable function to detect if the system has an installed TWAIN source manager (scanner). TWAIN_IsAvailable returns a 0 if unable to determine the installation of a TWAIN source manager. The function does not check if the scanner is connected or turned on. A return of 1 indicates one or more scanners have been installed to the system.

Finally, allow the user to select which scanner will be used with TWAIN_SelectImageSource. A return

value of 1 indicates the user has made a selection. A return value of 0 may mean there was no source to be selected (in which case the TWAIN_IsAvailable would have returned a 0), or it may mean the user cancelled the dialog box without making a selection. The function allows the handle of the application (hApp) be passed, but will accept a 0. There is no need to pass a handle using Liberty BASIC.

The prerequisites -

- TWAIN_EasyVersion - Do you have the most current EZTwain dll?
- TWAIN_IsAvailable - Does the host system have one or more TWAIN source managers installed?
- TWAIN_SelectImageSource - Which scanner will be used?

```
    Open "EZTW32.dll" for DLL as #ezt

    EZTVersion = TWAINEasyVersion()
    Print "Version = ";EZTVersion

    IsAvailable = TWAINIsAvailable()
        Print "IsAvailable = ";IsAvailable
        If IsAvailable = 0 Then
            Print
"Unable to find installed scanner.  Program will end."
            End
        End If
        Print "One or more scanners found to be installed"

    ImageSource = TWAINSelectImageSource(0)
        Print "ImageSource = ";ImageSource
        If ImageSource = 0 Then
            Print
"No TWAIN Source Manager selected.  Program will end."
            End
        End If

    Close #ezt

End

Function TWAINEasyVersion()
    CallDLL #ezt, "TWAIN_EasyVersion", _
        TWAINEasyVersion as Long
End Function
'
'  Returns the version number of EZTWAIN.DLL, multiplied by 100.
'  So e.g. version 2.01 will return 201 from this call.
'
```

```
Function TWAINIsAvailable()
    CallDLL #ezt, "TWAIN_IsAvailable", _
        TWAINIsAvailable as Long
End Function
'
'  Call this function any time to find out if TWAIN is installed on th
e
'  system.  It takes a little time on the first call, after that it's
fast,
'  just testing a flag.  It returns 1 if the TWAIN Source Manager is
'  installed & can be loaded, 0 otherwise.
'

Function TWAINSelectImageSource(hApp)
    CallDLL #ezt, "TWAIN_SelectImageSource", _
        hApp as uLong, _
        TWAINSelectImageSource as Long
End Function
'
'  This is the routine to call when the user chooses the "Select Sourc
e..."
'  menu command from your application's File menu.  Your app has one o
f
'  these, right?  The TWAIN spec calls for this feature to be availabl
e in
'  your user interface, preferably as described.
'  Note: If only one TWAIN device is installed on a system, it is sele
cted
'  automatically, so there is no need for the user to do Select Source
.
'  You should not require your users to do Select Source before Acquir
e.
'
'  This function posts the Source Manager's Select Source dialog box.
'  It returns after the user either OK's or CANCEL's that dialog.
'  A return of 1 indicates OK, 0 indicates one of the following:
'    a) The user cancelled the dialog
'    b) The Source Manager found no data sources installed
'    c) There was a failure before the Select Source dialog could be p
osted
'  -- details --
'  Only sources that can return images (that are in the DG_IMAGE group
) are
'  displayed.  The current default source will be highlighted initiall
y.
'  In the standard implementation of "Select Source...", your applicat
```

```
ion
'  doesn't need to do anything except make this one call.
'
'  If you want to be meticulous, disable your "Acquire" and "Select So
urce"
'  menu items or buttons if TWAIN_IsAvailable() returns 0 - see below.
'
```

Acquiring the Image

- TWAIN_AcquireToFilename
- TWAIN_AcquireNative
- TWAIN_AcquireToClipboard

Having established the availability of and the selection of a scanner, it's time to scan the image. The easiest function to work with is TWAIN_AcquireToFilename. This function opens the scanner's dialog, allowing the user to scan the image. It is important to note that EZTwain does not automatically scan. The user initiates the scan through the scanner's own application window. Once scanned, the image is saved to disk in .bmp format. The saved file can then be loaded with Liberty BASIC's Loadbmp command, and drawn using Drawbmp.

Two arguments are passed to TWAIN_AcquireToFilename: hApp and zFile$. hApp is the handle of your application. As with TWAIN_SelectImageSource, it is perfectly acceptable to pass a 0 instead. zFile$ is the full destination path of the saved bitmap. If zFile$ is passed as null, EZTwain opens a Save File dialog to allow the user to choose the destination. A return of 0 indicates a successful save. The Liberty BASIC code to scan and save to disk using EZTwain follows. To conserve space and time, the preliminary steps of checking version, existence of, and selection of the scanner has been omitted. EZTwain will default to the last scanner chosen.

```
    Open "EZTW32.dll" for DLL as #ezt

    AcquireToFilename = TWAINAcquireToFilename(0, "")
    Print "AcquireToFilename = ";AcquireToFilename
    Select Case AcquireToFilename
        Case 0
            Print "Success!  Bitmap saved to disk."
        Case -1
            Print "Error.  Did you cancel without saving?"
        Case -2
            Print "Error saving file."
        Case -3
            Print "Error.  Unable to lock DIB."
        Case -4
            Print "Error writing to disk."
```

```
        Case Else
            Print "Error - unable to determine."
    End Select
    Close #ezt

End

Function TWAINAcquireToFilename(hApp, zFile$)
    CallDLL #ezt, "TWAIN_AcquireToFilename", _
        hApp as uLong, _
        zFile$ as Ptr, _
        TWAINAcquireToFilename as Long
End Function
'
'  Acquire an image and write it to a .BMP (Windows Bitmap) file.
'  The file name and path in pszFile are used.  If pszFile is NULL or
'  points to an empty string, the user is prompted with a Save File di
alog.
'  Return values:
'  0 success
'  -1 Acquire failed OR user cancelled File Save dialog
'  -2 file open error (invalid path or name, or access denied)
'  -3 (weird) unable to lock DIB - probably an invalid handle.
'  -4 writing BMP data failed, possibly output device is full
'
```

A disadvantage of allowing EZTwain to bring up the Save File Dialog is that the image cannot be easily retrieved later in the program. If the saved image file must be known, use Liberty BASIC's native Filedialog to obtain the user's selection first, then pass that along as zFile$. The user won't know the difference.

```
    Open "EZTW32.dll" for DLL as #ezt

    Graphicbox #demo.g, 0, 0, 300, 300
    Open "EZTwain Demo" for Window as #demo
    Filedialog, "Save Scanned Image as . . .", "*.bmp", fName$
    #demo "Trapclose XbyTrap"
    If fName$ = "" Then
        Close #ezt
        Close #demo
        End
    End If
    AcquireToFilename = TWAINAcquireToFilename(0, fName$)
    If AcquireToFilename <> 0 Then
```

```
        Notice "Error Saving File"
        Close #ezt
        Close #demo
        End
    End If
    Loadbmp "pic", fName$
    #demo.g "Down; Drawbmp pic; Flush"
Wait

Sub XbyTrap handle$
    Unloadbmp "pic"
    Close #ezt
    Close #demo
End
End Sub

Function TWAINAcquireToFilename(hApp, zFile$)
    CallDLL #ezt, "TWAIN_AcquireToFilename", _
        hApp as uLong, _
        zFile$ as Ptr, _
        TWAINAcquireToFilename as Long
End Function
'
'  Acquire an image and write it to a .BMP (Windows Bitmap) file.
'  The file name and path in pszFile are used.  If pszFile is NULL or
'  points to an empty string, the user is prompted with a Save File di
alog.
'  Return values:
'  0 success
'  -1 Acquire failed OR user cancelled File Save dialog
'  -2 file open error (invalid path or name, or access denied)
'  -3 (weird) unable to lock DIB - probably an invalid handle.
'  -4 writing BMP data failed, possibly output device is full
'
```

It is possible to obtain the image directly from the scanner without first writing to disk. The documentation says this can be accomplished with the TWAIN_AcquireNative function. hApp (or 0 if you prefer) and the type of bitmap (wPizTypes) are passed to the function. Supported values of wPizTypes are documented within the comments of the function. Passing 0 seems safest. The function then returns the handle of the device independent bitmap (hDIB). A problem with this approach is that the handle of the DIB is not necessarily the handle of the bitmap recognizable to Liberty BASIC, but more a pointer to where the bitmap lies in memory. It would be simple if the acquired image could then be loaded into the Liberty BASIC program using Loadbmp "pic", hDIB. Unfortunately, this is not the case. However, EZTwain does include a TWAIN_AcquireToClipboard function. A search of the Liberty BASIC newsletters led to an article by Alyce Watson and Dennis McKinney detailing how to retrieve the handle of a bitmap copied to the clipboard, Clipboard API Demos, and then loading that bitmap. The API call GetClipboardData

proved successful. As documented by Alyce, the clipboard must first be opened, the contents retrieved, and then the clipboard closed.

```
    Open "EZTW32.dll" for DLL as #ezt

    Graphicbox #demo.g, 0, 0, 300, 300
    Open "EZTwain Demo" for Window as #demo
    hDemo = hWnd(#demo)
    #demo "Trapclose XbyTrap"
    TWAINAcquireToClipboard = TWAINAcquireToClipboard(0, 0)

    CallDLL #user32, "OpenClipboard", _
        hDemo as uLong, _
        result as Long

    CallDLL #user32, "GetClipboardData", _
        _CF_BITMAP as Long, _
        hPic as uLong

    CallDLL #user32, "CloseClipboard", _
        result as Void

    Loadbmp "pic", hPic
    #demo.g "Down; Drawbmp pic; Flush"

Wait

Sub XbyTrap handle$
    Unloadbmp "pic"
    Close #ezt
    Close #demo
End
End Sub

' Copying image to clipboard
Function TWAINAcquireToClipboard(hApp, wPixTypes)
    CallDLL #ezt, "TWAIN_AcquireToClipboard", _
        hApp as uLong, _
        wPixTypes as uLong, _
        TWAINAcquireToClipboard as Long
End Function
'
'  Like AcquireNative, but puts the resulting image, if any, into the
system
```

```
'  clipboard.  Under Windows, this will put a CF_DIB item in the clipb
oard
'  if successful.  If this call fails, the clipboard is either empty o
r
'  contains the old contents.
'  A return value of 1 indicates success, 0 indicates failure.
'
'  Useful for environments like Visual Basic where it is hard to make
direct
'  use of a DIB handle.  In fact, TWAIN_AcquireToClipboard uses
'  TWAIN_AcquireNative for all the hard work.
'
```

If your Liberty BASIC program requires real-time scanning, then do check out EZTwain (EZTW32.dll). It's free, it's simple, and it works.