

FTP Client Engine Library for PowerBASIC

Programmer's Manual

(FCE4PB)

Version 4.0

January 9, 2024

*This software is provided as-is.
There are no warranties, expressed or implied.*

Copyright (C) 2024
All rights reserved

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815

Email: info@marshallsoft.com
Web: www.marshallsoft.com

MARSHALLSOFT is a registered trademark of MarshallSoft Computing.

TABLE OF CONTENTS

1	Introduction	Page 3
1.1	Features	Page 4
1.2	Documentation Set	Page 5
1.3	Example Program	Page 6
1.4	Installation	Page 7
1.5	Uninstalling	Page 7
1.6	Pricing	Page 7
1.7	Updates	Page 7
2	Library Overview	Page 8
2.1	Dynamic Link Libraries	Page 8
2.2	Keycode	Page 8
2.3	Win32 STDCALL and DECLSPEC	Page 8
2.4	Adding FCE to Your Program	Page 8
2.5	FTP Parameters	Page 9
2.6	Error Display	Page 9
3	Compiler Issues	Page 10
3.1	Compiling Example Programs	Page 10
3.2	PowerBASIC Console Compiler (PBCC)	Page 10
3.3	PowerBASIC DLL Compiler (PBDLL)	Page 10
3.4	PowerBASIC Windows Compiler (PBWIN)	Page 10
3.5	Compiling FCE Source	Page 10
4	Example Programs	Page 11
4.1	FCEVER	Page 11
4.2	GET	Page 11
4.3	LIST	Page 11
4.4	FIELDS	Page 11
4.5	FTP	Page 12
4.6	PROXY	Page 12
4.7	PROXY2	Page 12
4.8	MGET	Page 13
4.9	MPUT	Page 13
4.10	WinVER	Page 13
4.11	WinGET	Page 13
4.12	FTIME	Page 13
5	Revision History	Page 14

1 Introduction

The **FTP Client Engine for PowerBASIC** (FCE4PB) library is a toolkit that allows software developers to quickly develop FTP client applications in PowerBASIC.

The **FTP Client Engine (FCE)** uses the Windows API to provide direct and simple control of the FTP protocol. The FCE component library can be used for both anonymous and private FTP sessions.

A straightforward interface provides the capability to easily build PowerBASIC FTP software applications to connect to any FTP server, navigate its directory structure, list files, upload files, rename files, delete files, append files, and download files using the FTP protocol.

This **FTP Client Engine Programmers Manual** provides information need to compile and run programs in a PowerBASIC programming environment.

The **FTP Client Engine for PowerBASIC** supports and has been tested with all versions of the PowerBASIC Console Compiler (PBCC), PowerBASIC DLL Compiler (PBDLL), and PowerBASIC Windows Compiler (PBWIN). **FCE4PB** includes several example programs that demonstrate FTP processing to help software developers easily build software applications using FCE4PB.

FCE4PB runs under all versions of 32-bit and 64-bit Windows through Windows 10.

The **FTP Client Engine SDK DLL's** (FCE32.DLL and FCE64.DLL) can also be used from any development environment (Visual Basic, C++, Delphi, Visual FoxPro, COBOL, Visual FoxPro, dBase, Xbase++, etc.) capable of calling the Windows API.

When comparing **FTP Client Engine** against our competition, note that:

- FCE4PB is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
- FCE4PB does NOT depend on ActiveX or Microsoft Foundation Class (MFC) libraries or similar "support" libraries.
- FCE is fully threadable.
- The FCE functions can be called from applications not capable of using controls.

MarshallSoft also has versions of the FTP Client Engine Library for C/C++ (FCE4C), Delphi (FCE4D), Visual FoxPro (FCE4FP), Visual Basic (FCE4VB), Visual dBASE (FCE4DB), and Xbase++ (FCE4XB). All versions of FCE use the same DLLs (FCE32.DLL/FCE64.DLL).

All versions of the FTP Client Engine Library (FCE) can be downloaded from our web site at <http://www.marshallsoft.com/ftp-client-library.htm>

1.1 Features

Some of the many features of the **FTP Client Engine** component library are as follows:

- Supports both 32-bit and 64-bit Windows through Windows 11.
- Connect to any (anonymous or private) remote FTP server.
- Get a list of files (names or long format) on the server.
- Navigate the server directories.
- Specify ASCII or BINARY transfer mode.
- Download files (with wildcard support).
- Upload files (with wildcard support).
- Delete files.
- Rename files.
- Append files.
- Create, delete and rename directories.
- Asynchronous and synchronous file transfer.
- Real-time upload/download data transfer rate.
- Real-time number bytes received/sent for async transfers
- Create and remove server directories.
- Support for PROXY servers.
- Supports active and passive mode (use with firewalls) file transfers.
- Supports multiple concurrent FTP sessions.
- Resume (restart) file uploads and downloads from any offset.
- Change files names while being uploaded or downloaded.
- Can parse long directory listings.
- Can specify the FTP or data port.
- Can set minimum and maximum response waits.
- Specify the allowable data port range.
- All operations can be aborted.
- Supports S/KEY password encryption.
- Use on Internet or your own intranet (LAN).
- Is native Windows code but can be called from managed code.
- Will run on machines without .NET installed
- Works with all versions of the PowerBASIC Console Compiler (PBCC), PowerBASIC DLL Compiler (PBDLL) and PowerBASIC Windows Compiler (PBWIN).
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Can be used with any program (in any language) capable of calling Windows API functions such as Visual C++, .NET, Visual Basic, Delphi, Xbase++, dBASE, COBOL, Access and Excel.
- License covers all programming languages.
- Royalty free distribution with your compiled application.
- Can be purchased with or without source code
- Updates are free for one year (Updates to source code are separate).
- Unlimited one-year email and phone support.
- Evaluation version is fully functional.

1.2 Documentation Set

The complete set of documentation consists of three manuals. This is the first manual (FCE4PB) in the set.

- [FCE4PB Programmer's Manual](#) (FCE_4PB.PDF)
- [FCE User's Manual](#) (FCE_USR.PDF)
- [FCE Reference Manual](#) (FCE_REF.PDF)

The FCE_4PB Programmer's Manual ([FCE_4PB.PDF](#)) is the language specific (PowerBASIC) manual dealing with compiler and programming issues such as installation and example programs. Read this manual first.

The FCE User's Manual ([FCE_USR.PDF](#)) discusses FTP in general as well as language independent programming issues such as application notes and includes purchasing and license information.

The FCE Reference Manual ([FCE_REF.PDF](#)) contains details on each individual FCE function.

Use Microsoft Word or Microsoft WordPad to print the document files. All manuals can also be viewed online at <http://www.marshallsoft.com/fce4pb.htm>

1.3 Example Program

The following PBCC example demonstrates the use of some of the **FTP Client Engine** library functions:

```
$INCLUDE "FCE32.PBI"
$INCLUDE "KEYCODE.PBI"

FUNCTION PbMain() AS LONG
Dim Code As Integer
Dim Srvr AS ASCIIZ * 50
Dim User AS ASCIIZ * 50
Dim Pass AS ASCIIZ * 50
Dim Temp As Ascii * 128

' Set FTP parameters

Srvr = "ftp.marshallsoft.com"
User = "anonymous"
Pass = "you@traveller.com"

' attach FCE
Code = fceAttach(1, %FCE_KEY_CODE)
If Code < 0 Then
    PRINT "Cannot attach FCE" & Str$(Code)
    Return
End If

' Connect to FTP server
PRINT "Connecting to ";Srvr;" as user '";User;"' ...";

Code = fceConnect(0, Srvr, User, Pass)
IF Code < 0 THEN
    PRINT "Error" & Str$(Code)
    Code = fceRelease()
    Exit Function
ELSE
    PRINT " We are connected!"
END IF

' close and exit
Code = fceClose(0)
Code = fceRelease()

END FUNCTION
```

In the example program above, **fceConnect** is called to connect to the FTP server as user "anonymous" and password "you@traveller.com".

Lastly, the connection to the FTP server is closed and FCE is released.

Refer to the FCE Reference Manual (FCE_REF) for individual function details. Access online at http://www.marshallsoft.com/fce_ref.pdf

1.4 Installation

- (1) Before installation of FCE4PB, a PowerBASIC compiler (PBCC, PBDLL, or PBWIN) should already be installed on your system and tested.
- (2) Unzip FCE4PB40.ZIP (demo version) or FCExxxx.ZIP (registered version; xxxx is the Customer ID).
- (3) Run the installation program SETUP.EXE that will install all FCE4PB files, including copying FCE32.DLL to the Windows directory.
- (4) You're ready to run! Compile and run FCEVER.BAS or WinVER.BAS first.

```
PBCC FCEVER
```

1.5 Uninstalling

Uninstalling FCE4PB is very easy.

First, run UNINSTAL.BAT, which will delete FCE32.DLL from your Windows directory, typically C:\WINDOWS for Windows 98/ XP/Vista/Windows 7/Windows 8 or C:\WINNT for Windows NT/2000.

Second, delete the FCE project directory created when installing FCE4PB.

1.6 Pricing

A developer's license for FCE4XB can be registered for \$115 (\$295 with ANSI C source code to the FCE DLL). Purchasing details can be found in Section 1.3, "How to Purchase", of the FCE User's Manual ([FCE USR](#)).

Registration includes one year of free updates and technical support.

1.7 Updates

When a developer license is purchased for FCE4PB, the developer will be sent a registered DLL plus a license file (FCExxxx.LIC, where xxxx is your Customer ID). The license file can be used to update the registered DLL for a period of one year from purchase. Updates can be downloaded from

<http://www.marshallsoft.com/update.htm>

After one year, the developer license must be updated to be able to download updates and receive technical support. The license can be updated for:

- \$33 if the update is ordered within one year of the original purchase (or previous update).
- \$55 if the update is ordered between 1 and 3 years of the original purchase (or previous update).
- \$77 if the update is ordered after three years of the original purchase (or previous update).

The update price includes technical support for an additional year. Note that the registered DLLs, (FCE32.DLL and FCE64.DLL) never expire. If source code was previously purchased, updates to the source code can be purchased for \$100 along with the license update. Refer to the file UPDATES.TXT located in the /FCE4PB/DOCS directory for more information.

Note that the registered DLL never expires.

2 Library Overview

The FTP Client Engine Library (FCE) has been tested on multiple computers running Windows 98, NT, 2000, 2003-2012, XP, Vista, Windows 7 and Windows 8.

2.1 Dynamic Link Libraries

The **FTP Client Engine** library is implemented as a dynamic link library (DLL). A DLL is characterized by the fact that it need not be loaded until required by an application program and that only one copy of the DLL is necessary regardless of the number of application programs that use it. Contrast this to the traditional static library that is bound to each and every application that uses it at link time.

An important advantage that DLL's have over other "popular" library formats such as VBX or OCX is that DLL's are callable by all Windows applications. Since DLL's are the building blocks of the Windows Operating System, they will not be replaced by a "newer technology".

2.2 Keycode

The FCE32.DLL has a keycode encoded within it. Your keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.PBI. The keycode for the evaluation version is 0. The developer will receive a new key code after purchasing the license.

The keycode is not the customer ID (which is a 4 or 5 digit number).

If the error message (value -74) is returned when calling **fceAttach**, it means that the keycode in the application does not match the keycode in the DLL. After registering, it is best to remove the evaluation version of the FCE DLL from the Windows search path.

2.3 Win32 STDCALL and DECLSPEC

FCE32 is written in ANSI C and is compiled using the STDCALL and DECLSPEC keywords. This means that FCE4PB uses the same calling conventions and file naming conventions as the Win32 API. In particular, function names are NOT decorated. There are no leading underscores nor trailing "@size" added to function names.

Any Windows application program may call the FCE32.DLL provided that the proper declaration file is used.

2.4 Adding FCE to A Program

In order to add FCE to a PowerBASIC program, add

```
$INCLUDE "FCE32.PBI"  
$INCLUDE "KEYCODE.PBI"
```

after any other \$INCLUDE statements. Then add

```
Code = fceAttach(%FCE_KEY_CODE)  
If Code < 0 Then  
    '... display error code to user here...  
    EXIT FUNCTION  
END IF
```

as the first executed FCE function. Also refer to the example programs.

2.5 FTP Parameters

There are two types of FTP connections: private and anonymous. However, some FTP servers do not accept anonymous connections.

Three parameters are necessary in order to connect to an FTP server, as follows:

- Host name (or IP address) of the FTP server.
- User name.
- User password.

For private connections, the users account name and password must be specified.

For anonymous connections, the user name is "anonymous" and the password is the user's email address.

These FTP parameters are hard coded in most of the examples. However, these parameters could be read from the keyboard, from a file, from a dialog box at runtime, etc., as well as being hard coded.

Refer to the FCE User's Manual ([FCE USR](#)) for more information regarding FTP protocol parameters.

2.6 Error Display

The error message text associated with FCE error codes can be displayed by calling **fceErrorText**. Each sample program contains examples of error processing.

Also see the file fceErrors.txt for a list of all FCE error codes.

3 Compiler Issues

The **FTP Client Engine for PowerBASIC** supports and has been tested with all versions of the PowerBASIC Console Compiler (PBCC), PowerBASIC DLL Compiler (PBDLL), and PowerBASIC Windows Compiler (PBWIN). Be sure to read the comments at the beginning of each example program.

3.1 Compiling Example Programs

Most example programs are compiled at the command line with the PowerBASIC Console Compiler:

- PBCC FCEVER.BAS
- PBCC LIST.BAS
- PBCC GET.BAS
- PBCC FIELDS.BAS
- PBCC FTP.BAS
- PBCC PROXY.BAS
- PBCC MGET.BAS
- PBCC MPUT.BAS
- PBCC FTIME.BAS

Other example programs (WinVER.BAS and WinGET) are designed to be compiled using the PowerBASIC Windows (PW/WIN) compiler. See Section 4, "Example programs", for more information on the example programs.

3.2 PowerBASIC Console Compiler (PBCC)

Most of the example programs are written for the PowerBASIC Console Compiler (PBCC). Be sure to read the comments at the beginning of each example program. Also see Section 4.0 "Example Programs" below.

3.3 PowerBASIC DLL Compiler (PBDLL)

FCE functions can be called from programs written for the PowerBASIC DLL Compiler (PBDLL). The same DLL (FCE32.DLL) is used for both PowerBASIC and Visual Basic. FCE functions are defined in FCE32.PBI for PowerBASIC and in FCE32.BAS for Visual Basic. The Visual Basic version can be downloaded from www.marshallsoft.com/fce4vb.htm.

3.4 PowerBASIC Windows Compiler (PBWIN)

FCE functions can be called from programs written for the PowerBASIC Windows Compiler (PBWIN). There are no differences in how FCE functions are called in programs written in any of the Windows PowerBASIC compilers.

Refer to the WinVER.BAS and WinGET.BAS example programs.

3.5 Compiling FCE Source

FCE32.DLL is written in standard ANSI C (FCE32.C), and has been compiled using Microsoft Visual C/C++ with the STDCALL and DECLSPEC compiler keywords. Source code for the FCE library is provided in the registered version (if ordered) only.

For more information on the C/C++ version of FCE, download the latest version of FCE4C from our web site at <http://www.marshallsoft.com/fce4c.htm>

4 Example Programs

Several example programs are included in FCE4PB. Their primary purpose is to demonstrate how to call FCE functions. Be sure to read the comments at the beginning of each example program. Before running the example programs, edit each program with your FTP parameters (see Section 2.5).

All of the example programs (except WinVER.BAS and WinGET.BAS) are compiled using the PowerBASIC Console Compiler (PBCC). However, FCE4PB functions can also be called from PowerBASIC DLL Compiler (PBDLL) programs or PowerBASIC for Windows (PBWIN) programs.

Note that the INCLUDE files FCE32.PBI and KEYCODE.PBI are referenced by all PowerBASIC programs and should normally be in the same directory as the application code. However, the value of the keycode found in KEYCODE.PBI can be pasted directly into the call to fceAttach, in which case the file KEYCODE.PBI is not needed.

Before writing your own programs, compile and run the example programs.

4.1 FCEVER

The first example (PBCC) program is the program FCEVER (FCE Version) that displays the FCE library version number and registration string.

Compile with:

```
PBCC FCEVER.BAS
```

4.2 GET

The GET (PBCC) example program connects to our FTP server anonymously and downloads the file "fce-new.txt".

Compile with:

```
PBCC GET.BAS
```

4.3 LIST

The LIST (PBCC) example program connects to a specified FTP server and lists all files.

Note that LIST could be modified to take all parameters from any source (command line, file, etc.).

Compile with:

```
PBCC LIST.BAS
```

4.4 FIELDS

The FIELDS example (PBCC) program connects to a specified FTP server and lists all files by field. Before compiling, edit FIELDS.BAS with the appropriate FTP parameters.

Note that FIELDS could be modified to take all parameters from any source (command line, file, etc.).

Compile with:

```
PBCC FIELDS.BAS
```

4.5 FTP

FTP.BAS is an example (PBCC) FTP application that can be used to connect to an FTP server and upload, download, and delete files. FTP is a general purpose FTP client. All parameters are entered by the user at runtime.

Compile with:

```
PBCC FTP.BAS
```

4.6 PROXY

The PROXY example (PBCC) program connects to an FTP server through a proxy server using the "USER@SERVER" protocol, also known as the "remote user" protocol..

Refer to the FCE User's Manual (FCE_USR) for a discussion of proxy servers, Section 3.6 "Proxy Server", of the FCE User's Manual ([FCE_USR](#)).

Before compiling, the proxy server and proxy port must be specified in PROXY.BAS.

Note that PROXY could be modified to take all parameters from any source (command line, file, etc.).

Compile with:

```
PBCC PROXY.BAS
```

4.7 PROXY2

The PROXY example (PBCC) program connects to an FTP server through a proxy server using the "USER after Logon" protocol, also known as the "double login protocol".

Refer to the FCE User's Manual (FCE_USR) for a discussion of proxy servers, Section 3.6 "Proxy Server" of the FCE User's Manual ([FCE_USR](#)).

Before compiling, the proxy server and proxy port must be specified in PROXY2.BAS.

Note that PROXY2 could be modified to take all parameters from any source (command line, file, etc.).

Compile with:

```
PBCC PROXY2.BAS
```

4.8 MGET

The MGET example (PBCC) program downloads files according to a wildcard pattern (using '?' and '*' characters). Note that MGET could be modified to take all parameters from any source (command line, file, etc.).

Compile with:

```
PBCC MGET.BAS
```

4.9 MPUT

The MPUT example (PBCC) program uploads files according to a wildcard pattern (using '?' and '*' characters). Note that MPUT could be modified to take all parameters from any source (command line, file, etc.).

Compile with:

```
PBCC MPUT.BAS
```

4.10 WinVER

The WinVER example program is a PowerBASIC PB/WIN program that displays the FCE version number, build number, and registration string. This program is the PBWIN equivalent of the PBCC program FCEVER.BAS.

Compile with the "PowerBASIC Compiler for Windows IDE"

4.11 WinGET

The WinVER example program is a PowerBASIC PB/WIN program that connects to our FTP server at <ftp://ftp.marshallsoft.com> anonymously and downloads the file "fce-new.txt". This program is the PBWIN equivalent of the PBCC program GET.BAS.

Compile with the "PowerBASIC Compiler for Windows IDE"

4.12 FTIME

The FTIME example program requests the "File Modification Time" for a file. However, not all FTP servers support the FTP MDTM command used to request the file modification time..

Compile with

```
PBCC FTIME.BAS
```

5 Revision History

The FTP Client Engine DLL (FCE32.DLL) is written in ANSI C. All language versions of FCE (C/C++, Delphi, Visual Basic, PowerBASIC, FoxPro, dBase, Xbase++, COBOL, and FORTRAN) use the same identical DLLs.

Version 1.2: August 20, 1999

The initial release of the PB version of FCE.

Version 2.0: June 8, 2000

- WriteBufferSize default reduced to 512.
- Added %FCE_GET_LINE_COUNT to fceGetString.
- Rename file being downloaded by specifying "oldname:newname" for file name.
- Added %FCE_SET_DATA_PORT to fceSetInteger.
- Password not used if specified password has zero length.
- fceSetLocalDir verifies that local directory is writable.

Version 2.1: January 25, 2001.

- Increased buffer size from 64 to 128 bytes for LocalDir and ServerDir.
- WriteBufferSize default increased to 512.
- Password not used if specified password has zero length.
- fceSetLocalDir() verifies that local directory is writable.
- Added %FCE_SET_APPEND_MODE.
- Allow 128 character filenames.
- Added %FCE_RENAME_DELIMITER to fceSetInteger().
- Corrected fceGetStatus(Chan, %FCE_CONNECT_STATUS)
- Added %FCE_SET_CLIENT_OFFSET and %FCE_SET_SERVER_OFFSET.
- Rename file being up/downloaded by specifying "oldname:newname" for file.
- Added %FCE_GET_LOCAL_IP to fceGetString.

Version 2.2: October 2, 2001.

- Default write buffer size increased from 512 to 1024.
- Added fceMatchFile function (used in multi-file uploads and downloads).
- Specify "User" as Chr\$(0) [in fceConnect] to skip USER and PASS processing.
- Performance improvements.
- Added MGET.BAS and PROXY.BAS example programs.

Version 2.3: January 3, 2003.

- Added %FCE_SET_BLOCKING_MODE to control blocking (default ON) while connecting.
- Size of command buffer in fceCommand increased from 64 to 128 bytes.
- "WARNING: 226/250 not seen" written to log file rather than returning error.
- Added %FCE_GET_ERROR_LINE.
- fceCommand sends CRLF with command in one network write.
- Added fceFileLength function.
- fceExtract handles line # 0.
- Added %FCE_GET_QUEUE_ZERO (returns # times fceQueueLoad returns 0).
- Changed to 32 channels & 128 data ports.
- fceGetList returns error if receive buffer is too small.

Version 2.4: May 13, 2004.

- Added %FCE_SET_CONNECT_WAIT_IN_SECS.
- Added %FCE_SET_MAX_RESPONSE_WAIT_IN_SECS.
- Added %FCE_SET_MAX_LINE_WAIT_IN_SECS.
- %FCE_NOT_COMPLETED returned if code 226 (or 250) not returned by control socket (in 2 tries).
- Number data ports changed to 2048 per channel for 1 and 2 channels
- Number data ports changed to 512 data ports per channel for 3 to 8 channels
- Number data ports changed to 128 data ports per channel for 9 to 32 channels
- DataPort mask corrected to 0x7FFF
- Added fceGetLocalFList function to get list of files in local directory.
- Added fceGetLocalFSize function to get the size of a file in the local directory.
- Added new MPUT.BAS and MGET.BAS example programs (using wildcards).
- Added HELLO.BAS example program for PB/WIN.

Version 2.5: July 29, 2005

- LocalDir always stored with backslash as last character.
- fceWriteSocket makes up to 12 attempts to write.
- Add %FCE_AUTO_LOG_CLOSE and %FCE_CLOSE_LOG_FILE
- Adjusted wait time-outs.
- Added %FCE_NO_GREETING error.
- Improved operation of fceGetInteger(Chan, %FCE_GET_CONNECT_STATUS).
- Added GET.BAS example program.
- Added fceShortToByte and fceByteToShort functions.

Version 2.6: February 13, 2007

- Maximum PUT buffer size increased from 8K to 16K (16384)
- Recoded sleep wait in fceWriteSocket for improved upload performance.
- Added internal memory allocation debugging.
- Added %GET_FULL_RESPONSE
- Close control socket whenever fceConnect fails (fixes socket leak problem).
- Added %FCE_SET_FIRST_DATA_PORT and %FCE_SET_LAST_DATA_PORT to fceSetInteger.
- Maximum data port extended to 65535.
- Added %FCE_HIDE_PASSWORD to fceSetInteger.
- Added S/KEY authentication
- Added fceGetTicks()
- Added %FCE_STATUS_BEFORE_WRITE to fceSetInteger.

Version 2.7: July 28, 2008

- Fixed problem with non-blocking mode when connecting.
- Added %FCE_LOCAL_DIR_IS_CDROM to fceSetInteger, which allows the local directory to be a read-only device such as a CDROM.
- Added %FCE_DISABLE_SKEY to fceSetInteger, which disables S/KEY processing.
- Added MDTM.BAS example program.

Version 3.0: October 16, 2009

- Added support for 64 bit Windows (FCE64.DLL)
- Added fceIsConnected function
- Added fceToInteger function

Version 3.1: July 20, 2011

- Log file is now time-stamped
- Added diagnostics to fceFileLength
- Function fceGetLocalFList no longer counts subdirectories
- Added fceGetFileSize function
- Added fceGetFileTime function

Version 3.2: June 12, 2012

- Fixed Bug: Open control socket not always closed.
- Fixed Bug: Open listen socket not always closed.
- Added function fcePutDirFiles (uploads all files in directory)
- Added function fceGetDirFiles (downloads all files in directory)

Version 3.3: May 15, 2014

- Fixed problem with connecting w/o blocking.
- Fixed problem with server name being corrupted in the FCE log file.
- Added function fceGetSubDirs()
- Added %FCE_SET_DEBUG_LEVEL to fceSetInteger().

Version 3.4: November 23, 2015

- Added debug diagnostics to fceSocketStatus().
- Fixed problem with connecting w/o blocking.
- Fixed problem with server name being corrupted in the FCE log file.
- Replaced FCE_EOF with FCE_CANNOT_OPEN if FCE log file cannot be created.
- Automatically adjust sleep times for slow FTP servers.

Version 4.0: January 9, 2024

- Fixed problem: In some cases FCE functions returned 1 instead of -1
- Log file displays filename passed to fceGetFile and fcePutFile
- Data no longer written to log file
- Default write size buffer increased from 1024 to 4096
- Default is set to ASCII mode (as opposed to binary)
- Added "Not connected to server" error message.(FCE_NOT_CONNECTED)
- I/O buffer increased to 65536 bytes.

Check <http://www.marshallsoft.com/fce4vb.htm> for the latest version of our FTP client library software.