

FTP Client Engine Library for Visual dBase

Programmer's Manual

(FCE4DB)

Version 4.0

November 11, 2023.

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

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	Using the FCE Library	Page 5
1.3	Documentation Set	Page 5
1.4	Example Program	Page 6
1.5	Installation	Page 7
1.6	Uninstalling	Page 7
1.7	Pricing	Page 7
1.8	Updates	Page 7
2	Library Overview	Page 8
2.1	Dynamic Link Libraries	Page 8
2.2	Keycode	Page 8
2.3	INCLUDE Files	Page 8
2.4	Visual dBase 5.6 and 5.7	Page 8
2.5	Visual dBase 7.0 and above	Page 8
2.6	Passing Strings to FCE	Page 8
2.7	Null Terminated Strings	Page 9
2.8	FTP Parameters	Page 9
2.9	Error Display	Page 9
3	Compiler Issues	Page 10
3.1	Compiling dBASE Programs	Page 10
3.2	Compiling dBASE Projects	Page 10
3.3	Compiling to an Executable	Page 10
4	Example Programs	Page 11
4.1	FCEVER	Page 11
4.2	GET	Page 11
4.3	LIST	Page 11
4.4	FTP	Page 11
4.5	FTP2	Page 11
4.6	WINFTP	Page 12
4.7	FIELDS	Page 12
4.8	PROXY	Page 12
4.9	MGET	Page 12
4.10	MPUT	Page 12
4.11	SPEED	Page 13
4.12	FTIME	Page 13
5	Revision History	Page 14

1 Introduction

The **FTP Client Engine for Visual dBASE (FCE4DB)** library is a toolkit that allows software developers to quickly develop FTP client applications in Visual dBASE and dBASE Plus.

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 dBASE 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 using FCE in a dBASE programming environment.

The **FTP Client Engine for Visual dBASE** component library supports all Win32 (dBASE 7.0, 7.5 and dBASE Plus) versions of dBASE. **FCE4DB** includes several example programs, including standard .PRG programs and .WFM forms based programs that demonstrate FTP processing to help software developers easily build software applications using **FCE4DB**.

FCE4DB runs under all versions of 32-bit and 64-bit Windows through Windows 11. A Win32 DLL is provided with FCE4DB (a Win64 DLL is available).

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, Xbase++, PowerBASIC, etc.) capable of calling the Windows API.

When comparing the **FTP Client Engine** library (FCE) against our competition, note that:

- **FCE4DB** is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
- **FCE4DB** does NOT depend on ActiveX or Microsoft Foundation Class (MFC) libraries or similar "support" libraries.
- **FCE4DB** 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), PowerBASIC (FCE4PB), Visual FoxPro (FCE4FP), Visual Basic (FCE4VB), and Xbase++ (FCE4XB). All versions of FCE use the same DLLs (FCE32.DLL and FCE64.DLL). However, the examples provided for each version are written for the specified computer language.

The latest 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.
- 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 Win32 (dBASE 7.0, 7.5 and dBASE Plus) versions of dBASE.
- 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++, Visual FoxPro, COBOL, Access and Excel.
- Supports Windows through Windows 11.
- License covers all programming languages.
- Royalty free distribution with your compiled application.
- Can be purchased with or without source code {Ansi C}.
- 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 Using the FCE Library

The **FTP Client Engine** component library (**FCE**) has been tested on multiple computers running Windows XP through Windows 11.

The FCE4DB library has been tested with all versions of Win32 (dBASE 7.0, 7.5 and dBASE Plus) of dBASE. The FCE32.DLL functions may be called by any Windows application program capable of calling the Windows API provided that the proper declaration file is used. FCE64.DLL is available to use with Win64 applications.

The SETUP installation program will copy the DLL to the Windows directory. Refer to Section 1.5 "Installation". After SETUP is run, the FCE4DB files are copied to the directory specified (default \FCE4DB). Three sub-directories are created, as follows:

```
DOCS - All documentation files
APPS - All example code
DLLS - All DLL's
```

1.3 Documentation Set

The complete set of documentation consists of three manuals. This is the first manual (FCE_4DB) in the set.

- [FCE4DB Programmer's Manual](#) (FCE_4DB.PDF)
- [FCE User's Manual](#) (FCE_USR.PDF)
- [FCE Reference Manual](#) (FCE_REF.PDF)

The FCE_4DB Programmer's Manual ([FCE_4DB.PDF](#)) is the language specific (dBASE) manual dealing with compiler and programming issues such as installation and example programs.

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. Read this manual second.

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

All manuals can also be viewed online at <http://www.marshallsoft.com/fce4db.htm>

1.4 Example Program

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

```
#INCLUDE C:\TEMP\FCE32.CC
#INCLUDE C:\TEMP\KEYCODE.CC

FTPserver = "ftp.marshallsoft.com" + Chr(0)
FTPuser = "anonymous" + Chr(0)
FTPpass = "you@yourisp.com " + Chr(0)

#define TEMP_SIZE 128
#define DATA_SIZE 5000

Code = fceAttach(1, FCE_KEY_CODE)
if Code < 0
    ? "Cannot attach FCE ", Code
    return
endif
? "Connecting to ", FTPserver
Code = fceConnect(0, FTPserver, FTPuser, FTPpass)
if Code < 0
    TempBuffer = SPACE(TEMP_SIZE)
    Code = fceErrorText(0,Code,TempBuffer,TEMP_SIZE)
    ? Left(TempBuffer,Code)
    Code = fceRelease()
    return
endif
? "Connected. Now getting file list."
DataBuffer = SPACE(DATA_SIZE)
Code = fceGetList(0, FCE_FULL_LIST, DataBuffer, DATA_SIZE)
if Code <= 0
    TempBuffer = SPACE(TEMP_SIZE)
    Code = fceErrorText(0,Code,TempBuffer,TEMP_SIZE)
    ? Left(TempBuffer,Code)
else
    ? Left(DataBuffer,Code)
endif
Code = fceClose(0)
Code = fceRelease()
? "Logged off."
return
```

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

Next, a full file listing is requested. 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.

1.5 Installation

(1) Before installation of FCE4DB, a dBASE or dBASE Plus compiler (any version) should already be installed and tested.

(2) Unzip FCE4DB40.ZIP (evaluation version) or FCExxxx.ZIP (registered version; xxxx is the Customer ID).

(3) Run the installation program SETUP.EXE which will install all FCE4DB files, including copying FCE32.DLL to the Windows directory, which is normally C:\WINDOWS.

(4) Edit the example programs (.PRG and .WFM):

- Set the location of the include files KEYCODE.CC and FCE32.CC.
- Change the server name, user, and password as necessary.

(5) You're ready to run! Compile and run FCEVER.PRG as a test of your installation.

1.5 Uninstalling

Uninstalling FCE4DB is very easy.

First, run UNINSTAL.BAT, which will delete FCE32.DLL from your Windows directory, typically C:\WINDOWS for Windows 98/XP/2003-2012/Vista/Win7/Win8/Win10 or C:\WINNT for Windows NT/2000.

Second, delete the FCE project directory created when FCE4DB was installed.

1.6 Pricing

A developer's license for FCE4DB can be registered for \$119 (\$299 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 FCE4DB, 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:

- \$30 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).
- \$75 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 /FCE4DB/DOCS directory for more information.

Note that the registered DLL never expires.

2 Library Overview

2.1 Dynamic Link Libraries

The FCE4DB library component is implemented using a Win32 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

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.CC. The keycode for the evaluation version is 0. You will receive a new key code when registering.

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

2.3 INCLUDE Files

All example programs include two files: KEYCODE.CC and FCE32.CC. The file FCE32.CC contains all the necessary constants and function declarations for FCE4DB, while the file KEYCODE.CC contains your key code, as discussed in Section 2.2 above.

There are three recommended ways to handle these INCLUDE files in dBASE programs.

1. Copy the INCLUDE files to your compiler's INCLUDE directory.
2. Edit the INCLUDE statements (in each program) with their physical location.
3. Replace the INCLUDE statements (in each program) by their contents.

2.4 Visual dBASE 5.6 and 5.7

Visual dBASE 5.6 and 5.7 create 16-bit applications. FCE4DB no longer supports 16-bit applications.

2.5 Visual dBASE 7.0 and above

Visual dBASE (7.0 and above) and dBASE Plus create 32-bit applications that use FCE32.DLL. Copy the 32-bit CC file FCE32.CC to the compiler's INCLUDE directory.

2.6 Passing Strings to FCE

Strings passed to FCE functions must be terminated with Chr(0) since this is the way the Windows API (written in ANSI C) handles strings.

For example:

```
Server = "ftp.marshallsoft.com" + Chr(0)
```

2.7 Null Terminated Strings

All strings returned from FCE functions are null terminated. That is, the end of the string is delimited by a Chr(0) character. These strings may be converted for dBase in one of two ways: (1) if the length of the string is known, use the dBase LEFT function: For example,

```
* get server IP address
TempBuffer = SPACE(TEMP_SIZE)
Code = fceGetString(0, FCE_GET_SERVER_IP, TempBuffer, TEMP_SIZE)
if Code > 0
    ? "Server IP ", LEFT(TempBuffer, Code)
endif
```

If the length of the null terminated string is not known, use the dBase AT function to find the position of Chr(0). For example,

```
Offset = fceMatchFile(ListBuffer, Offset, NameBuffer, 128, FTPpattern, 1)

P = AT(Chr(0), NameBuffer)
? "Downloading file " + Left(NameBuffer, P-1)
```

2.8 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.

2.9 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 Winsock and FCE error codes.

3 Compiler Issues

Programs using **FCE4DB** can be compiled with 32-bit dBASE (dBase 7.0 and above) or dBASE Plus.

3.1 Compiling dBASE Programs

Visual dBase programs end with the extension ".PRG". Before compiling any of the example programs, edit each file with your Internet TCP/IP parameters. Programs can be edited within any text editor, and compiled from the VDB (Visual dBase) command window with the COMPILE command (e.g.: COMPILE LIST.PRG) or executed from the VDB command window with the DO command (e.g.: DO LIST.PRG).

To open a program within the Visual dBase source editor, choose "File", then "Open". When the "Open File" dialog box appears, choose "Programs" for "Files of Type", then choose the program (*.PRG) to open. Lastly, choose "Open in Source Editor" for "Action" and push the "Open" button.

After editing the source program with your internet (or TCP/IP) parameters, you are ready to compile. From the dBase menu bar, choose "Build", then "Compile". To run choose, "Run". The dBASE command window must be displayed in order to see the output.

3.2 Compiling dBASE Projects

Visual dBase projects consist of several types of files such as forms, reports, data modules, etc. The project file itself ends with the extension of ".PRJ".

There is one example Visual dBase project WINFTP. Open WINFTP by choosing "File", then "Open Project" from the dBase menu bar. To compile WINFTP, choose "Build" from the menu bar, then "Rebuild All". This will create WINFTP.EXE, which can be executed by choosing "Execute winftp.exe" from the "Build" menu bar pull down, or from the Windows command line prompt.

3.3 Compiling to an Executable

dBASE programs end in ".PRG". They can be compiled to an executable using the dBASE BUILD command.

For example, to create FCEVER.EXE from FCEVER.PRG in the C:\FCE4DB\APPS directory, type the following in the dBASE command window:

```
BUILD PROJECT C:\FCE4DB\APPS\FCEVER FROM C:\FCE4DB\APPS\FCEVER
BUILD EXE C:\FCE4DB\APPS\FCEVER FROM C:\FCE4DB\APPS\FCEVER
```

4 Example Programs

Several example programs are included in FCE4DB. Refer to Section 3 above for help in compiling the example programs.

All example programs end with ".PRG", except for the FORM example (WINFTP), which ends with ".WFM".

4.1 FCEVER

The FCEVER ("FCE Version") program (FCEVER.PRG) displays the DLL version, build number, and registration string. This is the first program to try. Its purpose is to display the current version of the FCE DLL as well as to verify that FCE32.DLL is being found and loaded by Windows. It does not require a TCP/IP (Internet) connection. Run FCEVER from the dBASE command window with "DO".

4.2 GET

GET is an FTP client that connects to our FTP server anonymously and downloads the file "fce-new.txt". After compiling, start GET from the command line.

4.3 LIST

The LIST program (LIST.PRG) connects to the MarshallSoft FTP site at "ftp.marshallsoft.com" as user "anonymous" with password "msc@traveller.com". Change the password to your email address before compiling. Run LIST.PRG from the command window.

After connecting, LIST gets the list of all files on the server at the base FTP directory level, then logs off.

4.4 FTP

The FTP program (FTP.PRG) is a generic FTP program that can perform the most basic FTP functions such as listing and downloading files. Note that there are many FCE functions not implemented in the FTP.PRG example. The FCE Reference Manual (FCE_REF) contains details on each individual FCE function.

Run FTP.PRG from the dBASE command window.

4.5 FTP2

The FTP2 program (FTP2.PRG) is identical to FTP except that it uses **fceDriver** when downloading and uploading files. Refer to Section 4, "Theory of Operation", in the FCE User's Manual ([FCE_USR](#)).

Run FTP2.PRG from the dBASE command window.

4.6 WINFTP

The WINFTP program (WINFTP.PRG) is similar to the FTP program, except that it uses the dBase form. To compile WINFTP.FRM, choose "Open Project" from under "File" on the dBASE menu bar.

Note that code at the beginning of WINFTP.WFM

```
#INCLUDE C:\TEMP\FCE32.CC
#INCLUDE C:\TEMP\KEYCODE.CC

#define TEMP_SIZE 128
#define DATA_SIZE 35000
```

must be re-inserted manually after modifying the file with the dBase form designer.

4.7 FIELDS

The FIELDS program (FIELDS.PRG) connects to the MarshallSoft FTP site at "ftp.marshallsoft.com" as user "anonymous" with password "msc@traveller.com". Change the password to your email address before compiling. Run FIELDS from the command window.

The FIELDS program is similar to the LIST program. After connecting, FIELDS gets the list of all files on the server at the base FTP directory level, and lists each field as a separate string.

4.8 PROXY

The PROXY example program connects to a FTP server through a proxy server using the "USR@SERVER" protocol. Edit PROXY.PRG with your FTP server name (or IP address), user name, and password before compiling.

Refer to the FCE User's Manual (FCE_USR) for a discussion of proxy servers and proxy protocols; Section 3.7 "Proxy Servers" and Section 3.8 "Proxy Protocols" of the FCE User's Manual ([FCE_USR](#)).

4.9 MGET

The MGET example program downloads files according to a wildcard pattern (using '?' and '*' characters). Edit MGET.PRG with your FTP server name (or IP address), user name, and password before compiling.

4.10 MPUT

The MPUT example program uploads files according to a wildcard pattern (using '?' and '*' characters). Edit MPUT.PRG with your FTP server name (or IP address), user name, and password before compiling.

4.11 SPEED

The SPEED example program connects to the MarshallSoft FTP server at `ftp://ftp.marshallsoft.com` and downloads a test file from the MarshallSoft FTP server and displays the time required. Use this program to see how long it takes to download files from FTP servers.

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.

5 Revision History

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

Version 1.2: October 15, 1999

- The initial release of the dBase version of FCE.

Version 2.0: May 11, 2000

- WriteBufferSize default set 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.

Version 2.1: February 8, 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 16, 2001.

- Default write buffer size increased from 512 to 1024 (WIN32 only).
- 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 and PROXY example programs.

Version 2.3: December 12, 2002.

1. Added FCE_SET_BLOCKING_MODE to control blocking (default ON) while connecting.
2. Size of command buffer in fceCommand increased from 64 to 128 bytes.
3. "WARNING: 226/250 not seen" written to log file rather than returning error.
4. Added FCE_GET_ERROR_LINE.
5. fceCommand sends CRLF with command in one network write.
6. Added fceFileLength function.
7. fceExtract handles line # 0.
8. Added FCE_GET_QUEUE_ZERO (returns # times fceQueueLoad returns 0).
9. Fixed problem with long server offsets (replaced "REST %d" with "REST %ld") in 16-bit version.
10. Changed to 32 channels & 128 data ports (random time bits no longer used).
11. fceGetList returns error if receive buffer is too small.

Version 2.4: June 1, 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 and MGET example programs (using wildcards).

Version 2.5: July 28, 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.PRG example program.
- Added fceShortToByte and fceByteToShort functions

Version 2.6: February 16, 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 22, 2008

Beginning with this version, FCE4DB no longer supports 16-bit applications.

- 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.PRG example program.

Version 3.0: October 30, 2009

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

Version 3.1: July 14, 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 5, 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 6, 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 18, 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: November 11, 2023

- 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> for the latest version of our FTP software.