



Apple® IIcs

FTPTool Set Reference Manual



Frank McCarthy

Dedicated to the memory of Joe Kohn 1947-2010

FTPTool Set is Freeware and Copyright © 2017 Ewen Wannop

FTPTool Set and its supporting documentation may not be printed,
copied, or distributed for profit.

Distributing and/or archiving is restricted while in an electronic form.

Any “free” distribution must be given permission by Ewen Wannop
in advance -- please contact via email by sending mail to:

spectrumdaddy@mac.com

There is no guarantee that the right to redistribute this material
will be granted. The contents of this document may not be
reprinted in part or in whole.



Contents



Introduction	4
Using the FTPTool Set	
Requirements	5
Programming	6
The Routines	6
The Calls	
FTPBootInit	7
FTPStartup	7
FTPShutDown	7
FTPVersion	8
FTPReset	8
FTPStatus	9
FTPLogin	10
FTPLogout	11
FTPSendFile	12
FTPReceiveFile	14
FTPDeleteFile	16
FTPRenameFile	17
FTPDeleteDir	18
FTPMakeDir	19
FTPListDir	20
Appendix	
The Parameters	22
The Flags	22
Sample Code	23
Response Buffer	24
FTP Tool Set Error Codes	24
FTP Tester	24
Links	24



Introduction



Introduction to the FTPTool Set

FTP Tool Set

Andrew Roughan suggested to me some time ago, that it would be useful to him to have the ability to send software builds out to a remote FTP server as and when needed.

Although not exactly answering his request, and taking rather longer than I would have liked, I have now partially answered it by adapting the core FTP routines from SAFE2 into this FTPTool Set.

The FTPTool Set has been released into the Public Domain, so that anyone who wishes, can use it to send or receive files by FTP from within their own applications.

Full programming details for using the Tool are given within this manual.

It is up to you to do the rest!

If you have any questions about the use of this toolset, please contact:

spectrumdaddy@mac.com

FTPTool Set is Freeware and Copyright © 2017 Ewen Wannop



Using the FTPTool Set

Requirements

Drop TOOL131 into the Tools folder in the System folder and reboot.

The FTPTool Set requires at least these basic tools to be active. They would normally have already been started by your application:

Control Manager, Event Manager, Integer Math Tool Set, Memory Manager, Miscellaneous Tool Set

The FTPTool Set also requires 'Marinetti 3.0b3' or later to be installed and configured.

A simple application, 'FTP.Tester' is provided for you to see how the FTPTool works in practice, and to help you understand the kind of input to the tool that is required. You will also find the 'SOS.FTP NDA' in the archive, which also uses the FTPTool.

The FTPTool set is provided for you to incorporate FTP routines into your own application. If you need more powerful FTP functionality, please use the SAFE2 application that you will also find on my web site.

To obtain the FTPTool Set, SAFE2, and any of my other software, please go to my web site:

<http://speccie.uk>

Programming

The FTPTool Set must be started by your application after any other required Tool sets have been started.

Your application should first load **Tool131**, and then call **FTPStartUp** and **FTPLogIn**. If successful, an **ftpUserID** will be returned, which must be used for all the other calls.

When you are finished with the Tool set, call **FTPLogOut** and **FTPShutDown**, then **Unload Tool131**,

At any time, up to eight applications can be logged into **Tool131**.

Press **Escape** to stop a transfer that is in progress.

The Routines

Housekeeping Routines

FTPBootInit	Initialises the FTP Tool Set; called only by the Tool Locator must not be called by an application
FTPStartUp	Called when application starts
FTPShutDown	Called when application quits
FTPLogIn	Starts up the FTP Tool Set for use by an application
FTPLogOut	Shuts down the FTP Tool Set by the application
FTPVersion	Returns the version number of the FTP Tool Set
FTPReset	Resets the FTP Tool Set; called only when the system is reset must not be called by an application
FTPStatus	Indicates if the FTP Tool Set is active

Global Routines

FTPSendFile	Sends a single file to an FTP Host
FTPReceiveFile	Receives a single File from an FTP Host
FTPDeleteFile	Deletes a File on an FTP Host
FTPRenameFile	Renames a File or Directory on an FTP Host
FTPDeleteDir	Deletes a Directory on an FTP Host
FTPMakeDir	Makes a new Directory on an FTP Host
FTPListDir	Lists the contents of a Directory on an FTP Host

\$0183 FTPBootInit

Initialises the FTP Tool; called only by the Tool Locator.

Warning

An Application must never make this call.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

C Call must not be made by an application.

\$0283 FTPStartUp

Must be called when the application starts.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

C `extern pascal void FTPStartUp ()`

\$0383 FTPShutDown

Must be called when the application Quits.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

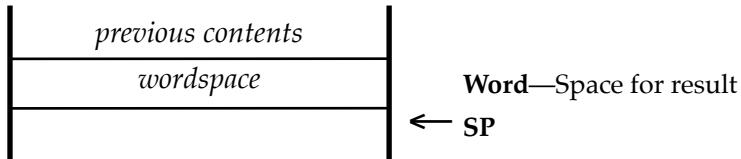
C `extern pascal void FTPShutDown ()`

\$0483 FTPVersion

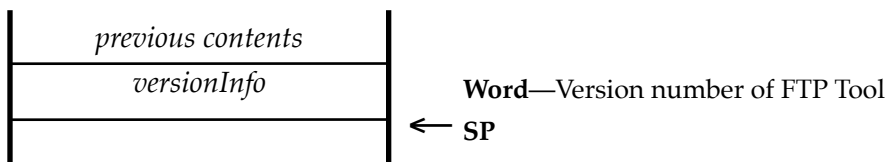
Returns the version number of the FTP Tool.

Parameters

Stack before call



Stack after call



Errors None

C `extern pascal Word FTPVersion ()`

\$0583 FTPReset

Resets the FTP Tool; called only when the system is reset.

Warning

An Application must never make this call.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

C Call must not be made by an application.

\$0683 FTPStatus

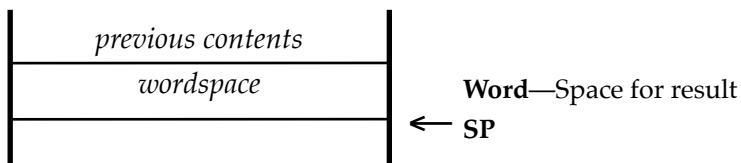
Indicates whether the FTP Tool Set is active.

FTPStatus returns *TRUE* if FTPStartup has been called and FTPShutDown has not been called. The routine returns *FALSE* if FTPStartUp has not been called.

Note: Because a number of apps may be logged in concurrently, *FALSE* will only be returned after the last app has called FTPShutDown.

Parameters

Stack before call



Stack after call



Errors None

C `extern pascal Boolean FTPStatus ()`

\$0783 FTPLogIn

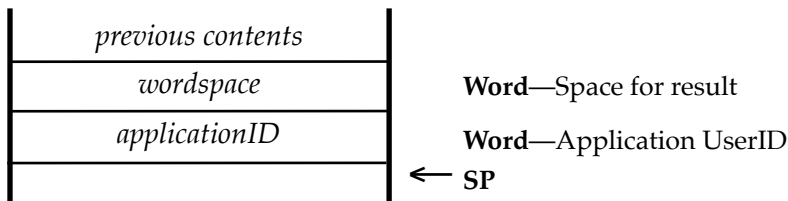
Starts up the FTP Tool for use by an application.

Important

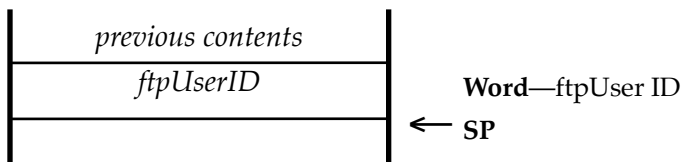
Your Application must make this call before it makes any other FTP Tool calls. The returned *FtpUserID* must be used as input for the other calls that require it.

Parameters

Stack before call



Stack after call



Errors	\$83FF	ftpBadCall	Application ID not valid
	\$83FC	ftpBusy	More than 8 applications already logged in

C

```
extern pascal Word FTPLogIn (applicationID);  
Word      applicationID, ftpUserID
```

Note: The *applicationID* is the UserID of the calling application. If successful, a new *FtpUserID* for use by the Tool will be created and returned. This *FtpUserID* should then be used for the other FTP Tool commands that require it.

Up to eight applications can LogIn to the Tool at any one time. These must be balanced by a LogOut call when they Quit. This is different to the way other Tools operate, but is required so the *ftpUserID*'s for the TCP/IP calls can be correctly assigned for each application that has logged in.

\$0883 FTPLogOut

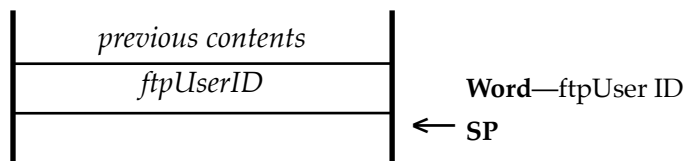
Shuts down the FTP Tool.

Important

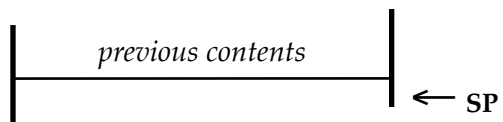
If your Application has started up the FTP Tool, the application must make this call before it quits.

Parameters

Stack before call



Stack after call



Errors \$83FF ftpBadCall User ID not valid

C extern pascal FTPLogOut (ftpUserID);
 Word ftpUserID

\$0983 FTPSendFile

Sends a file from a local folder to a Host server.

Parameters

Stack before call

<i>previous contents</i>	
<i>ftpUserID</i>	Word —FtpUserID returned from <i>FTPStartUp</i>
... <i>userPassword</i> ...	Long —pString of Host <i>password</i>
... <i>userName</i> ...	Long —pString of Host <i>username</i>
... <i>filePath</i> ...	Long —GS/OS String of path to Local file
... <i>hostPath</i> ...	Long —GS/OS String of path to Host file
... <i>IPAddress</i> ...	Long —pString of IP Address of Host
... <i>responseBuffer</i> ...	Long —Pointer to Thermometer response buffer
<i>flags</i>	Word —Settings flags
← SP	

Stack after call

<i>previous contents</i>	
← SP	

Errors	\$8301	ftpTransferFailed	The Transfer or Command failed
	\$8303	ftpBadFileName	GS/OS filename not accepted
	\$8304	ftpBadFileType	Not a binary or text file
	\$8305	ftpOpenPortFailed	Could not open Data Port
	\$8307	ftpBadDomainName	Domain name not found
	\$830A	ftpUserCancelled	User cancelled transfer
	\$830E	ftpUserPassBad	Username or password not recognised

C

```
extern pascal FTPSendFile (ftpUserID, userPassword, userName,
filePath, hostPath, IPAddress, responseBuffer, flags);
```

```
Long   userPassWord, userName, filePath, hostPath,
responseBuffer
```

```
Word   ftpUserID, IPAddress, flags
```

flags

Controls the behaviour of the Tool Set

debug	bit 15	Set Debug On (default Off)
<i>reserved</i>	bits 14-4	
overwrite	bit 2	When set, deletes the destination file before transfer (Default <i>Resumes</i> transfer)
fileType	Bit 1	For Binary files = 0, ASCII files = 1
active	Bit 0	When set, connects using an <i>Active</i> connection (default <i>PASV</i> connection)

responseBuffer	word	0 = Don't display a Thermometer 1 = Handle passed of a Thermometer Ctl 2 = A Pointer passed to a routine that will be called
	long	Handle to the Ctl, or a Pointer to routine to be called
	word	Passed <i>Scale</i> for progress calculation
	word	Returned <i>Position</i>

Notes:

Only Text, Binary, or unforked files may be sent to an FTP host. Batch processing is not supported. If necessary, compress any forked files or directories before sending.

If sending a Text file, set the ASCII flag.

If *Debug* is On, the received and sent FTP commands will be saved to the file '**:FTP.Log*' at the top level folder of the local boot drive.

The *responseBuffer* controls the behaviour of an optional progress Thermometer. A flag indicates the function, followed by either a Handle to a Thermometer Ctl, or a Pointer to a routine that will be called. If a Pointer to a routine to be called is passed, the *Scale* must first be set. The routine will be called regularly during the transfer, when the current position can be picked up. It is up to the app to decide what to do with the data when called.

\$0A83 FTPReceiveFile

Receives a file from a Host server to a local folder.

Parameters

Stack before call

<i>previous contents</i>	
<i>ftpUserID</i>	Word —FtpUserID returned from <i>FTPStartUp</i>
.... <i>userPassword</i>	Long —pString of Host <i>password</i>
.... <i>userName</i>	Long —pString of Host <i>username</i>
.... <i>filePath</i>	Long —GS/OS String of path to Local file
.... <i>hostPath</i>	Long —GS/OS String of path to Host file
.... <i>IPAddress</i>	Long —pString of IP Address of Host
.... <i>responseBuffer</i>	Long —Pointer to Thermometer response buffer
<i>flags</i>	Word —Settings flags
	← SP

Stack after call

<i>previous contents</i>	
	← SP

Errors

\$8301	ftpTransferFailed	The Transfer or Command failed
\$8302	ftpBadPathName	Bad path to local directory
\$8303	ftpBadFileName	GS/OS filename not accepted
\$8304	ftpBadFileType	Not a binary or text file
\$8305	ftpOpenPortFailed	Could not open Data Port
\$8307	ftpBadDomainName	Domain name not found
\$8308	ftpGSOSError	GS/OS Error
\$830A	ftpUserCancelled	User cancelled transfer
\$830E	ftpUserPassBad	Username or password not recognised

C

```
extern pascal FTPReceiveFile (ftpUserID, userPassword, userName,
filePath, hostPath, IPAddress, responseBuffer, flags);
Long      userPassword, userName, filePath, hostPath,
responseBuffer
Word      ftpUserID, IPAddress, flags
```

flags Controls the behaviour of the Tool Set

debug	bit 15	Set Debug On (default Off)
<i>reserved</i>	bits 14-4	
overwrite	bit 2	When set, deletes the destination file before transfer (Default resumes transfer)
fileType	Bit 1	Binary files = 0, ASCII files = 1
active	Bit 0	When set, connects using an Active connection (default PASV connection)

responseBuffer	word	0 = Don't display a Thermometer
		1 = Handle passed of a Thermometer Ctl
		2 = A Pointer passed to a routine that will be called
	long	Handle to the Ctl, or a Pointer to routine to be called
	word	Passed <i>Scale</i> for progress calculation
	word	Returned <i>Position</i>

Note: Batch processing is not supported. If necessary, compress any forked files or directories before receiving.

If receiving a Text file, set the ASCII flag.

If *Debug* is On, the received and sent FTP commands will be saved to the file '**:FTP.Log*' at the top level folder of the local boot drive.

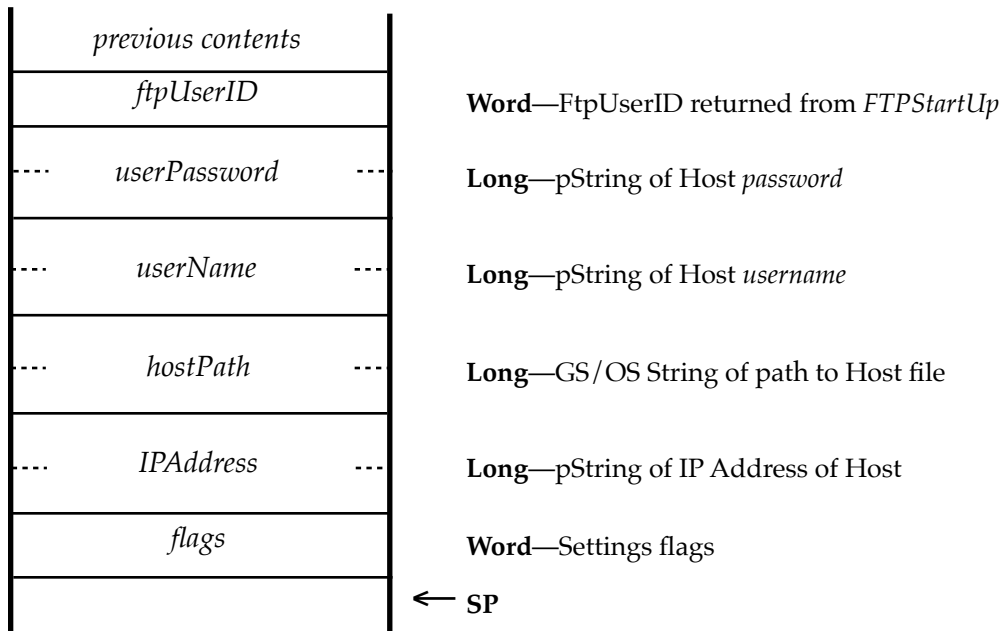
The *responseBuffer* controls the behaviour of an optional progress Thermometer. A flag indicates the function, followed by either a Handle to a Thermometer Ctl, or a Pointer to a routine that will be called. If a Pointer to a routine to be called is passed, the *Scale* must first be set. The routine will be called regularly during the transfer, when the current position can be picked up. It is up to the app to decide what to do with the data when called.

\$OB83 FTPDeleteFile

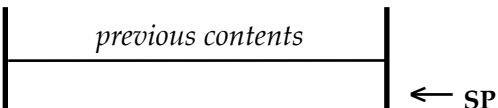
Deletes a file on a Host server.

Parameters

Stack before call



Stack after call



Errors	\$8301	ftpTransferFailed	The Transfer or Command failed
	\$8307	ftpBadDomainName	Domain name not found
	\$830B	ftpCommandFailed	Command failed
	\$830D	ftpFileNotFound	File not found on Host
	\$830E	ftpUserPassBad	Username or password not recognised

C extern pascal FTPDeleteFile (ftpUserID, userPassword, userName, hostPath, IPAddress, flags);

Long userPassword, userName, hostPath

Word ftpUserID, IPAddress, flags

\$0C83 FTPRenameFile

Renames or moves a file or directory on a Host server.

Parameters

Stack before call

<i>previous contents</i>	
<i>ftpUserID</i>	Word —FtpUserID returned from <i>FTPStartUp</i>
--- <i>userPassword</i> ---	Long —pString of Host <i>password</i>
--- <i>userName</i> ---	Long —pString of Host <i>username</i>
--- <i>hostPath1</i> ---	Long —GS/OS String of path to Host file/directory
--- <i>hostPath2</i> ---	Long —GS/OS String of path to new name/path
--- <i>IPAddress</i> ---	Long —pString of IP Address of Host
<i>flags</i>	Word —Settings flags
	← SP

Stack after call

<i>previous contents</i>
← SP

Errors	\$8301	ftpTransferFailed	The Transfer or Command failed
	\$8307	ftpBadDomainName	Domain name not found
	\$830B	ftpCommandFailed	Command failed
	\$830D	ftpFileNotFound	File not found on Host
	\$830E	ftpUserPassBad	Username or password not recognised

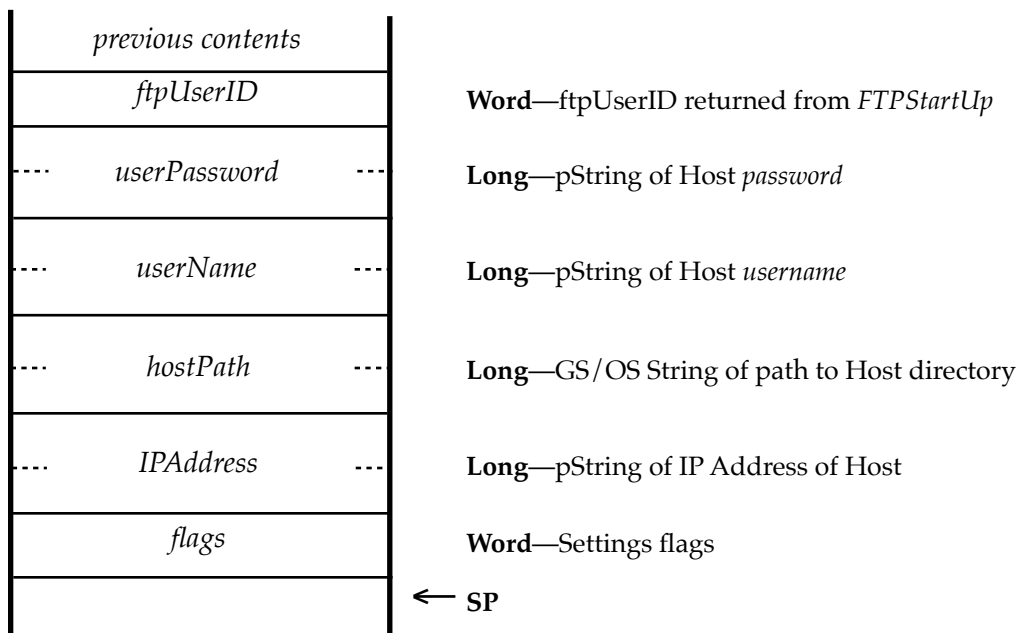
C extern pascal FTPRenameFile (ftpUserID, userPassword, userName, hostPath1, hostPath2, IPAddress, flags);
Long userPassword, userName, hostPath1, hostPath2
Word ftpUserID, IPAddress, flags

\$0D83 FTPDeleteDir

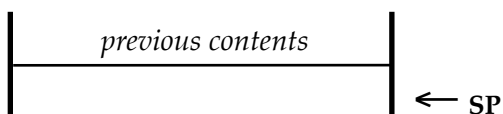
Deletes a directory on a Host server.

Parameters

Stack before call



Stack after call



Errors	\$8301	ftpTransferFailed	The Transfer or Command failed
	\$8307	ftpBadDomainName	Domain name not found
	\$830B	ftpCommandFailed	Command failed
	\$830D	ftpFileNotFound	File not found on Host
	\$830E	ftpUserPassBad	Username or password not recognised

C extern pascal FTPDeleteDir (ftpUserID, userPassword, userName, hostPath, IPAddress, flags);

Long userPassword, userName, hostPath

Word ftpUserID, IPAddress, flags

\$OE83 FTPMakeDir

Makes a new directory on a Host server.

Parameters

Stack before call

<i>previous contents</i>	
<i>ftpUserID</i>	Word —ftpUserID returned from <i>FTPStartUp</i>
--- <i>userPassword</i> ---	Long —pString of Host <i>password</i>
--- <i>userName</i> ---	Long —pString of Host <i>username</i>
--- <i>hostPath</i> ---	Long —GS/OS String of path to Host directory
--- <i>IPAddress</i> ---	Long —pString of IP Address of Host
<i>flags</i>	Word —Settings flags
	← SP

Stack after call

<i>previous contents</i>	
	← SP

Errors	\$8301	ftpTransferFailed	The Transfer or Command failed
	\$8307	ftpBadDomainName	Domain name not found
	\$830B	ftpCommandFailed	Command failed
	\$830D	ftpFileNotFound	File not found on Host
	\$830E	ftpUserPassBad	Username or password not recognised

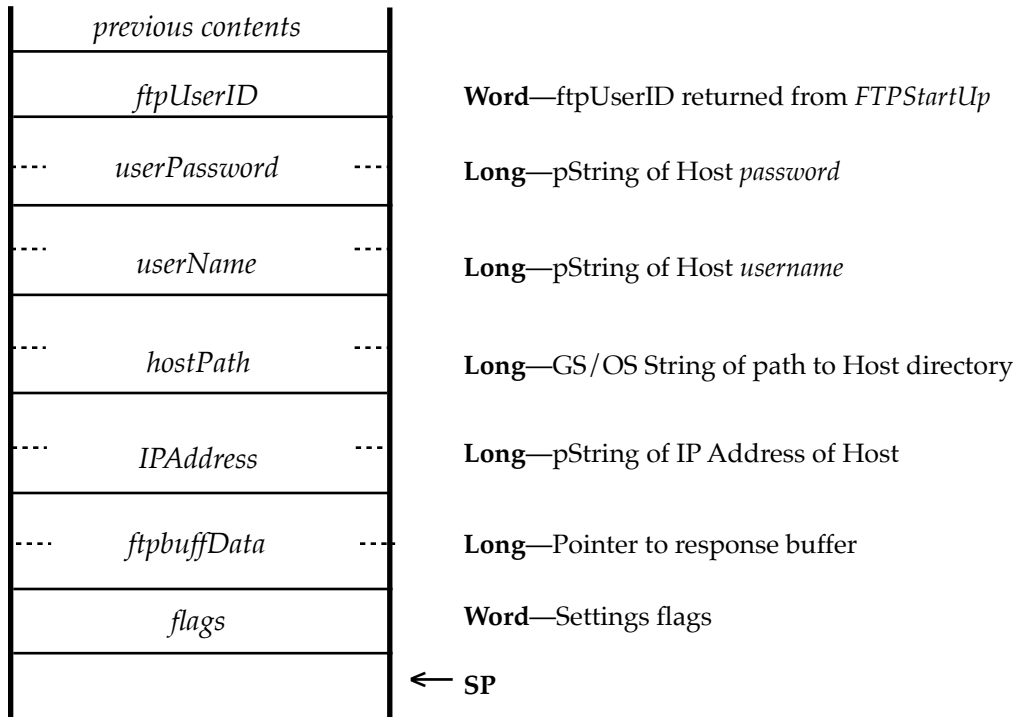
C extern pascal FTPMakeDir (ftpUserID, userPassword, userName, hostPath, IPAddress, flags);
Long userPassword, userName, hostPath
Word ftpUserID, IPAddress, flags

\$0F83 FTPListDir

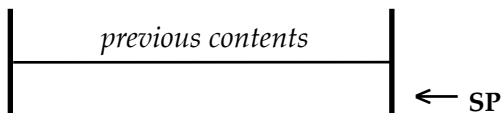
Lists a directory a file on a Host server.

Parameters

Stack before call



Stack after call



Errors	\$8301	ftpTransferFailed	The Transfer or Command failed
	\$8307	ftpBadDomainName	Domain name not found
	\$830B	ftpCommandFailed	Command failed
	\$830D	ftpFileNotFound	File not found on Host
	\$830E	ftpUserPassBad	Username or password not recognised

C extern pascal FTPListDir (ftpUserID, userPassword, userName, hostPath, IPAddress, ftpbuffData, flags);
 Long userPassword, userName, hostPath, ftpbuffData
 Word ftpUserID, IPAddress, flags

flags Controls the behaviour of the Tool Set

debug bit 15 Set Debug On
 (default Off)
 reserved bits 14-4

ftpbuffData Pointer to a 6 byte response buffer:

word Boolean = FALSE, Handle requested
 Boolean = TRUE, Handle supplied
 long Handle passed, or returned

The calling app can either pass a Handle through *ftpbuffData+2*, or request a Handle, which will be returned in *ftpbuffData+2*. The calling application must Dispose of the Handle when done. The returned data is an ASCII directory listing of the target directory. The format may vary depending on the host:

```
total 10
-rw-r--r-- 1 ewenwannop admin 28676 Apr 9 09:43 .DS_Store
-rw-r--r-- 1 ewenwannop admin 0 Aug 18 2016 .localized
-rw-r--r-- 1 ewenwannop admin 34423 Feb 10 2016 Ext Webcam.html
-rw-r--r-- 1 ewenwannop admin 34423 Feb 10 2016 Int Webcam.html
-rw-r--r-- 1 ewenwannop staff 478205 Apr 9 00:00 FTPTool Manual.pdf
drwx----- 62 ewenwannop admin 2108 Apr 7 10:20 IIgs Software
drwx----- 285 ewenwannop admin 9690 Feb 28 11:49 JuicedGS
drwx----- 6 ewenwannop admin 204 Apr 7 15:31 Movies
-rw-r--r-- 1 ewenwannop staff 6272 Mar 7 10:18 U2.BXY
-rw-r--r-- 1 ewenwannop staff 1757701 Mar 6 16:13 Uthernet II manual.pdf
```

Note: If *Debug* is On, the received and sent FTP commands, as well as returned data from the command, will be saved to the file '**:FTP.Log*' at the top level folder of the local boot drive.

Appendix

The Parameters

<i>applicationID</i>	Original UserID of the calling application
<i>filePath</i>	Full path to the local file to send or receive: <code>:boot.drive:folder:filename</code>
<i>flags</i>	Settings used internally by the Tool (see Send File for more details)
<i>hostPath</i>	Full Path to target file or directory on Host: <code>/folder1/folder2/file</code>
<i>hostPath1</i>	Full Path to target file or directory on Host to rename: <code>/folder/oldname</code>
<i>hostPath2</i>	Full Path to target file or directory on Host to rename: <code>/folder/newname</code>
<i>IPAddress</i>	IP Address of the target Host: www.syndicomm.com 192.168.0.7
<i>ftpbuffData</i>	Pointer to a response buffer, which can control progress display
<i>ftpUserID</i>	ftpUserID returned from the <i>ftpStartup</i> call
<i>responseBuffer</i>	Pointer to response buffer that controls an optional Thermometer
<i>userName</i>	UserName to sign in to the Host server. This must be provided, unless the Host does not require you to sign in
<i>userPassword</i>	Password to sign in to the Host server. This must be provided, unless the Host does not require you to sign in.

The Flags

Settings that controls the behaviour of the Tool Set

<i>debug</i>	bit 15	Set Debug On (default Off)
<i>reserved</i>	bits 14-4	
<i>overwrite</i>	bit 2	When set, deletes the destination file before transfer (Default resumes transfer)
<i>fileType</i>	Bit 1	Binary files = 0, ASCII files = 1
<i>active</i>	Bit 0	When set, connects using an Active connection (default PASV connection)

Sample Code

ORCA/M format

Startup:

```
pea    $0083                FTP Tool
pea    $0100
_LoadOneTool

_FTPStartUp

pha
PushWord MyID                Application ID
_FTPLogIn
PullWord UserID
```

Send File:

```
PushWord UserID                word
PushLong #UserPassword         pointer to pString
PushLong #UserName             pointer to pString
PushLong #FilePath             pointer to GS/OS Str    local file path
PushLong #HostPath             pointer to GS/OS Str    host file path
PushLong #IP_Address           pointer to pString      host IP Address
PushLong #ThermData            pointer to buffer       ThermCtl buffer
PushWord Flags                 word                    flags/settings
_FTPSendFile
```

Receive File:

```
PushWord UserID                word
PushLong #UserPassword         pointer to pString
PushLong #UserName             pointer to pString
PushLong #FilePath             pointer to GS/OS Str    local file path
PushLong #HostPath             pointer to GS/OS Str    host file path
PushLong #IP_Address           pointer to pString      host IP Address
PushLong #ThermData            pointer to buffer       ThermCtl buffer
PushWord Flags                 word                    flags/settings
_FTPReceiveFile
```

Shutdown:

```
PushWord UserID
_FTPLogOut

_FTPShutDown

pea    $0083                FTP Tool
_UnloadOneTool
```

ResponseBuffer Controls an optional Thermometer display

word	0 = Don't display a Thermometer 1 = Handle passed to a Thermometer Ctl 2 = A Pointer passed to a routine to be called
long	Handle to the Ctl, or a Pointer to routine to be called
word	Passed Scale for progress calculation
word	Returned position

ftpbuffData Pointer to a 6 byte response buffer:

word	Boolean = <i>FALSE</i> , Handle requested Boolean = <i>TRUE</i> , Handle supplied
long	Handle passed, or returned

FTP Tool Set error codes

ftpTransferFailed gequ \$8301	; Generic failure to send or receive file
ftpBadPathName gequ \$8302	; Bad file pathname
ftpBadFileName gequ \$8303	; Bad filename
ftpBadFileType gequ \$8304	; Bad file type
ftpOpenPortFailed gequ \$8305	; Could not open port
ftpHostTimedOut gequ \$8306	; Host timed out
ftpBadDomainName gequ \$8307	; Bad domain name
ftpGSOSError gequ \$8308	; GS/OS error
ftpVolumeFull gequ \$8309	; Local volume full
ftpUserCancelled gequ \$830A	; User cancelled transfer
ftpCommandFailed gequ \$830B	; Command failed
ftpFileNameExists gequ \$830C	; File exists
ftpFileNotFound gequ \$830D	; File not found
ftpBusy gequ \$83FC	; FTP Tool in use
ftpNotInit gequ \$83FD	; FTP Tool not started
ftpNoTCPIP gequ \$83FE	; TCP/IP not started
ftpBadCall gequ \$83FF	; Bad call

FTP.Tester

A simple application is provided for you to test out what you can expect from the various calls and commands.

The application has three menu choices, *Receive File*, *Send File*, and *Commands*. Selecting any of those three options will open a window where you can fill in the required boxes for that call, and then let you action that call.

If you check the 'Send Handle' box when requesting a directory listing, the app will send a Handle for the returned list. The default is for the Tool to create the Handle itself.

If you check the 'Debug' box, you will be able to check the *FTP.Log* file in the top directory of your *Boot Drive* folder to see the data sent and returned for the transfer. This can help you find any errors that you may have made.

If any of the calls you use returns an error, you will see the 'Error' value as non-zero. Check the Error Code list to see what the error number means.

Press *Escape* to cancel a transfer that is in progress.

Links

To obtain the FTPTool Set, or any of my other software:

<http://speccie.uk>

To subscribe to the Juiced.GS magazine:

<http://juiced.gs/>

To read all about the KFest conference:

<http://www.kansasfest.org/>

To shop for Apple II software:

<http://www.syndicomm.com/>