

Package 20001 - Internet Tools
 FTP Utility (*web/ftp)
 Details

Purpose of Program

This utility is used to send files to a server which is running an FTP server program, and will work if run from the ProvideX *WEB directory only.

Calling Sequence:

```
CALL "*WEB/FTP;SEND", ftpserver$, user$, password$, timeout,  

  files_to_send$,download_dir$, case$, result$  

CALL "*WEB/FTP;RECEIVE",ftpserver$, user$, password$, timeout,  

  files_to_receive$, download_dir$, case$, result$
```

Where the parameters are:

<i>ftpserver\$</i>	(Input)	Contains the name of the FTP server. It should be in the form of SERVER;SOCKET (eg. ftp.pvx.com;21.) If no socket number is specified, then a default FTP number of 21 is used. Maximum length of the <i>ftpserver\$</i> is 128 characters.
<i>user\$</i>	(Input)	Contains the user ID. Max length of the <i>USER\$</i> is 64 characters.
<i>password\$</i>	(Input)	Contains the user password. Maximum length of the <i>password\$</i> is 64 characters.
<i>Timeout</i>	(Input)	If non-zero, then this parameter specifies the timeouts for sending directly FTP Server. If zero, then the timeout value is set to 60 seconds. (Only values greater than zero have an affect with ProvideX version 4.15 higher)
<i>files_to_send\$</i>	(Input)	Contains the paths and filenames of the files to send to an FTP Server. <i>files_to_send\$</i> cannot be null. Maximum length of <i>files_to_send\$</i> is 32,000 characters
<i>files_to_receive\$</i>	(Input)	Contains the paths and filenames of the files to receive from an FTP Server. <i>files_to_receive\$</i> cannot be null. Maximum length of <i>files_to_receive\$</i> is 32 characters
<i>upload_dir\$</i>	(Input)	Contains the path and the directory name on an FTP Server to use to store. If null, then the upload directory is the root directory of the FTP Server.
<i>download_dir\$</i>	(Input)	Contains the path and the directory name on a client machine to use to store files. If null, then the upload directory is the working directory of the machine.
<i>case\$</i>	(Input)	"U" - convert all file & directory names to upper case "L" - convert all file & directory names to lower case "" - does not change file & directory names
<i>result\$</i>	(Output)	If an error is encountered while sending files to an FTP server, then the program loads this variable with a textual description of the problem and exits with an error. If one of the sending file names is invalid, then all that can be will be sent; those which cannot be sent will be returned in <i>result\$</i> field.

Usage Details:

files_to_send\$ / files_to_receive\$:

Wild characters are accepted in file names.

? = any one character.

* = any number of any characters.

Arguments can contain more than one file name. File names should be separated using one of the following:

\$8A\$

\$0A\$

\$0D0A\$

;

File transfer Options:

A number of options may be specified after a file name; options should be separated from the filename by a "|" (pipe). The allowable options are:

Transfer type

A = ASCII mode.

B = Binary mode (default).

Overwrite (if the file already exists on the FTP Server)

0 = Always overwrite (default).

1 = Never overwrite.

2 = Overwrite if the file being sent is newer.

Recursion

R = Recursion ON.

"" = Recursion OFF (default).

Example :

```
file_name1|A1 ; c:\tmp\*.*|BOR
```

In this example file_name1 will be transferred in ASCII; it will not be transferred if it currently exists. The directory c:\tmp and all files with it will be transferred in binary and will always overwrite any existing file of the same name. It will also recurse through all sub-directories of c:\tmp and transfer all files in every subdirectory.