WinSCP PUT command

https://winscp.net/eng/docs/scriptcommand_put#speed

Uploads one or more files from local directory to remote directory.

- Syntax
- Remarks
- Examples
- Converting to .NET Assembly

Syntax

```
put <file> [ [ <file2> ... ] <directory>/[ <newname> ] ]
```

Remarks

If only one parameter is specified, uploads the file to remote working directory. If more parameters are specified, all except the last one specify set of files to upload. Filename can be replaced with Windows wildcard1 to select multiple files. To upload all files in a directory, use mask *.

The last parameter specifies target remote directory and optionally operation mask to store file(s) under different name. Target directory must end with slash. To upload more files to current working directory use `./` as the last parameter.

With the /stdin command-line switch, `-` can be used as a source to stream the standard input of winscp.com to remote server. Streaming is supported with the SFTP protocol only.

When specific file name is used, command fails when the file does not exist. When wildcard is used, command succeeds, without doing anything, if the wildcard does not match any file, by default (you can change this using `option failonnomatch on` command).

You can have WinSCP generate a code template for `put` for you.

See also `synchronize`, if you need to transfer modified files only.

Switches:

Switch	Description
`-delete`	Delete source local file(s) after a successful transfer.
`-latest`	Uploads the latest file from the files selected by the `file`, `file2` … parameters (typically file masks) only.
`-resume`	Automatically resume transfer if possible.2 SFTP and FTP protocols only. Cannot be combined with `-append`. Has no effect when streaming (`-` source).
`-append`	Append source file to the end of target file. SFTP protocol only. Cannot be combined with `-resume`. Has no effect when streaming (`-` source).
`-preservetime`	Preserve timestamp.
`-nopreservetime`	Do not preserve timestamp.
`-permissions=<mode>`	Set permissions (SFTP and SCP protocols only). Has no effect when streaming (`-` source).
`-nopermissions`	Keep default permissions.
`-speed=<kbps>`	Limit transfer speed (in KB/s).
`-transfer=<mode>`	`binary|ascii|automatic` Transfer mode: binary, ascii (text), automatic (by extension).
`-filemask=<mask>`	`<mask>[;<mask2>...]` Sets file mask.
`-resumesupport=<state>`	`on|off|<threshold>` Configures automatic resume/transfer to temporary filename.
`-neweronly`	Transfer new and updated files only. See also `synchronize` command.
`-rawtransfersettings`	Allows configuring any transfer settings using raw format as in an INI file. E.g. to

`setting1=value1` `setting2=value2 …`	enable preserving of directory timestamps, use `-rawtransfersettings PreserveTimeDirs=1`. The switch should come only after other parameters.

Aliases: `send`, `mput`

Effective options: `confirm`, `reconnecttime`, `failonnomatch`

XML log elements: `upload`, `mkdir`, `chmod` (with `-permissions`), `touch` (with `-preservetime`)

Examples

```
put index.html
put -delete index.html about.html ./
put -permissions=644 index.html about.html /home/martin/public_html/
put d:\www\index.html about.*
put *.html *.png /home/martin/backup/*.bak
put -filemask=*.html -resumesupport=on *
```

Converting to .NET Assembly

When converting script to .NET Assembly, map `put` command to `Session.PutFiles` method.

Parameters mapping: Command parameter `file` maps to method parameter `localPath`. When multiple source file parameters are used, you need to call `Session.PutFiles` multiple times. Command parameter `directory/newname` maps to method parameter `remotePath`. You have to convert relative paths to absolute paths.

Switches mapping:

Switch	Mapping
`-delete`	Value `true` (`$True` in PowerShell) for method parameter `remove`.
`-latest`	See Uploading the most recent file.

`-resume`
`-append`
`-preservetime`
`-nopreservetime`
`-permissions`
`-nopermissions`
`-transfer`
`-filemask`
`-resumesupport`
`-speed=<kbps>`
	Converting transfer settings scripting switches to .NET assembly class `TransferSettings`.

`-neweronly`	Not supported. Use `Session.SynchronizeDirectories` instead.

To emulate the (default) `option batch abort` mode, call `TransferOperationResult.Check` on method's result. See also Capturing results of operations.

For example, following script snippet:

```
cd /home/martinp
lcd d:\
...
put -delete -filemask=*>1M -resumesupport=off *.txt *.xml web/
```

maps to following PowerShell code:

```
$transferOptions = New-Object WinSCP.TransferOptions
# -filemask=*>1M
$transferOptions.FileMask = "*>1M"
# -resumesupport=off
$transferOptions.ResumeSupport.State = [WinSCP.TransferResumeSupportState]::Off
# Absolute paths + $True for -delete + Two calls for two source parameters
# + calling Check on result to emulate the (default) "option batch abort" mode
$session.PutFiles("d:\*.txt", "/home/martinp/web/", $True, $transferOptions).Check()
$session.PutFiles("d:\*.xml", "/home/martinp/web/", $True, $transferOptions).Check()
```

1. The Windows wildcard supports `*` and `?` only. If you want to use a full syntax of file masks, use a `-filemask` switch.Back

2. Applies for individual files transfers only. It does not skip transfer of fully transferred files. For that, combine the switch with `-neweronly`.Bac