References - the filesystem (Operating System File System) Namespace

The filesystem namespace is new since MarvelClient Release 4.

The following table lists the available parameters for the filesystem Namespace.

Syntax for the filesystem Namespace is as follows:
command[_<parameters>]-path[?regex]

Example 1:
<filesystem:exists-{user#absolutefilepathtocheck}>
Checks whether the file <user:absolutefilepathtocheck> exists.

Example 2:
<filesystem:size_mb-{notes#data_directory}>
Determines the size of the IBM Notes data directory in Megabytes.

> ⚠️ **NOTE that some of the filesystem commands may take long(er) to execute, especially when targeted at network resources and/or recursing through many subdirectories and files.**

> ⚠️ **NOTE: You are advised to use an exclamation mark as execution type for filesystem references as to indicate that they are always executed again, even if the same filesystem reference were to be used more than once. Whilst for the time being it does not make a difference whether you use a colon, exclamation mark or pipe (:, !, |), the behaviour may change in a future release.**

> ⚠️ **NOTE: You must NOT hardcode any filepaths! <filesystem!size_mb-C:\test.txt>, for example, will fail because of the colon in the filepath (remember, the colon (:) is a reserved character in references). Passing a colon through a sub-reference works fine, however.**

The most important parameters from the filesystem Namespace			
Command	**Parameters**	**Target Type**	**Description**
exists	none	any	Returns "1" if target path exists, "0" otherwise
isdirectory	none	any	Returns "1" if target path is a directory , "0" otherwise
isfile	none	any	Returns "1" if target path is a file, "0" otherwise
isempty	none	any	For a directory: returns "1" if target path has no files or subdirectories in it, "0" otherwise For a file: returns "1" if target path is an empty file, "0" otherwise
size	mb - return size in megabytes; must be provided	file or directory	Returns the size of the target file or directory (including subdirectories) in megabytes (rounds down) *Example:* <filesystem!size_mb-{notes#data_directory}/{notes#um_path}> = size of the user's mailbox in megabytes Note that the forward slash "/" works operating system independent - you are advised NOT to use a backslash "\".
lastmodified	none	any	Returns the date and time the target path was last modified (yyyymmddThhnnss)
checksum	none	file	Returns an MD5 checksum for the file at the target path
checkwriteaccess	none	directory	Returns "1" if write access is permitted, "0" otherwise Creates a temporary file (mc_temp.tmp) in the target directory and deletes it afterwards to check effective write access. **NOTE If a file mc_temp.tmp already exists, MarvelClient will attempt to overwrite it.** Any path can be provided, MC will move one level up until an existing directory is found. *Example:* c:\notes\data\foo\bar.nsf is provided (NOT hardcoded, but through a sub-reference!), but no file bar.nsf or directory foo exists. Write access will be checked for c:\notes\data.

count	l - limit to current directory (no subdirectories) d - count directories only f - count files only x - use regex d and f cannot be combined, use one or the other (or none)	directory	Returns the number of filesystem objects found according to parameters/regex Counts the contents of the provided target path. If the path to a file is provided, the parent directory of the file is used. If a regex is used, will limit the count to files/directories which match the regex. The regex is applied to the filename (+extension) only, not the full or partial path. *Example 1:* <filesystem!count_fx-{notes#data_directory}?.*\.ntf> - will count all files with extension .ntf in data directory (recursively) *Example 2:* <filesystem!count_ld-{os#temp_directory}\test.txt> - will count all subdirectories in C:\Temp, but not recursively and no files *Example 3:* <filesystem!count-\\server\subdirectory> - will count all files and directories, recursively
find	l - limit to current directory (no subdirectories) d - find directories only f - find files only x - use regex a - return absolute paths d and f cannot be combined, use one or the other (or none)	directory	Returns a comma-seperated list of all files found (relative to target dir, or absolute if "a" is provided as a parameter). **NOTE: The maximum size for the string returned is 2 GB. It is not a good idea to even get close to a returned string of such size, i.e. you should not find all files in C:\.** Looks for all files in the provided target path which match the provided parameters/regex. If the path to a file is provided, the parent directory of the file is used. If a regex is used, will limit the search to files/directories which match the regex. The regex is applied to the filename (+extension) only, not the full or partial path. *Example 1:* <filesystem!find_fx-{notes#data_directory}?.*\.box> - list all files ending in .box within data directory (and subdirectories) *Example 2:* <filesystem!find-{os#system_root_directory}/System32> - lists all files and directories in the target directory (recursively) *Example 3:* <filesystem!find_x-{notes#idfile_path}?hugo.*\.nsf - lists all files in the directory where the user id is stored (and subdirectories) which start with hugo and have the extension .nsf
space	free_mb - free space available to the current user (honoring quota) total_mb - total space available to the current user totalfree_mb - total space on the disk (not honoring quota) See MSDN for details	directory	Returns the requested value (in megabytes) as a number. Calculate free/used space for disk of target path. If the path to a file is provided, the parent directory of the file is used. Any directory on a disk can be used.