

Multi Commander Documentation

Alpha/Test Version of the autogenerated PDF documentation

Since the documentation is autogenerated from the online documentation there is still some formatting/styling and other issues.

This documentation is auto generated from the documentation located online at <http://multicmd.com/docs/>

What Is Multi Commander

Multi Commander is a file manager for Windows that are a replacement for Windows Explorer. It provides a efficient way of working with folders and files.

It is using the popular and efficient Dual-Panel layout. And it is designed to be very keyboard friendly.

Mouse is supported, but the strength is that you can do everything with the keyboard alone. And that allows you to work fast and efficient.

Multi Commander offers a wide range of features.

- Support both Windows Explorer styled keyboard/mouse setup and the popular Commander Styled keyboard/mouse setup
- Support Drag & Drop
- Copy/Move operation support filtering and plugin for special situation.
- Zip, 7Zip, Tar, Gz can be browsed like a folder.
- FTP, Portable Device , Registry are exposed in the file system
- Multi-Rename tool with powerful rename operations.
- View text/logs files of ANY size with minimum memory usage (even of the log file are many gigabytes)
- And very very much more.

System Requirements

Multi Commander works on most system from Windows XP and Windows Server 2003 and newer.

Client Operation System : Windows 8.1, Windows 8, Windows 7, Windows Vista , Windows XP, 32bit and 64bit

Server Operation System :Windows Server 2012/R2, Windows Server 2008/R2, Windows Server 2003, 32bit and 64bit

Not all features are available on all operation system.

Minimal Hardware : 32bit CPU with SSE2 support

Screen Resolution : Minimum resolution of 800x600 is recommended

Download and Installation

Download

The official download location is <http://multicommander.com/downloads> if you download it from another location make sure that you really trust that source. Because you never know if somebody have manipulated the package.

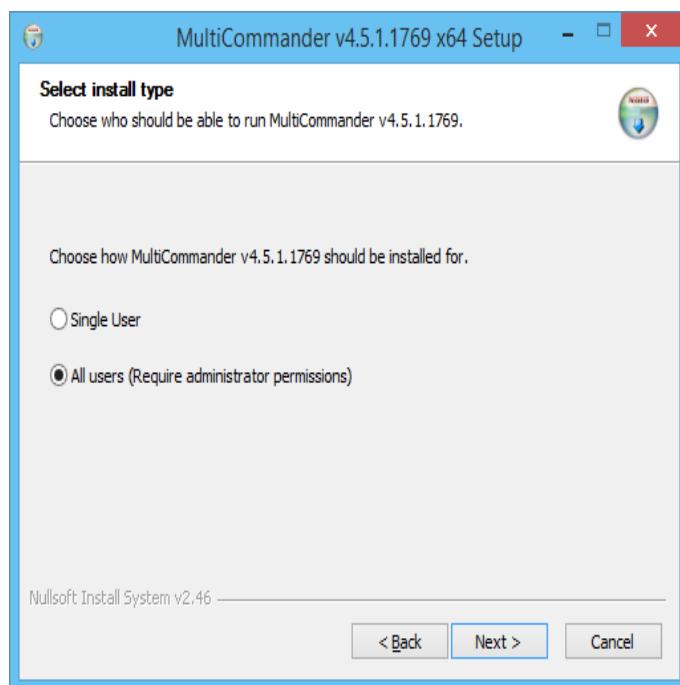
This is especially important with the portable version since that is only a .zip. Never download the portable version from another source.

Installation

Full Installer (.exe)

The full installer is named **MultiCommander_x64_{[version]}.exe** or **MultiCommander_win32_{[version]}.exe**

You will be given two chooses at the beginning of the installation : **Single User** or **All Users**.



If you choose **Single User** the installer will by default select a installation location that are under your private account. By doing this the installer will not require any administer rights for the installation.

If you select **All Users** then the installer will by default select the shared program files folders as the install location. By installing it here all user accounts on the machine will have access to the program.

This installation option require administrator rights and the installer will restart itself and ask for admin permission if needed.

All user that then run Multi Commander will get its own configuration. The configuration is always stored under the user account that is located under the **Application Data Path**.

Eg **C:\Users\<username>\AppData\Roaming\MultiCommander\Config**

Hint

By typing "**:goconfig**" in the command line field inside Multi Commander, The active Explorer Panel will then take you to the current configuration folder

Portable (.zip)

The portable package is a zip archive that you only need to unpack to an folder. Just make sure that you keep the folder structure from the zip package.

The portable version is the same as the installed version. The difference is that the portable version is configured to store all settings and configuration in the folder that Multi Commander is run from.

The portable version do no save anything to the registry or to the user account folder. Everything is saved in the Multi Commander folder. So the program must have write permissions to the folder that you start Multi Commander from. So this version is ideal to be places on a USB memory stick.

Upgrading

Update Tool

When Multi Commander starts it will do a check if a new version is available. If a new version is found you will be asked if you want to upgrade and if you select that you want to upgrade then the update tool is started. You can also do a manual check by selecting **Menu > Help > Check for Updates** or Manually launch the update tool with **Menu > Help > Launch MultiUpdate**

The Update tools can also be started manually by running **MultiUpdate.exe** that is located in the installation folder of Multi Commander.

The Update tool will download the correct update package and will then replace all files that need to be replaced. The tool works for both the installed and portable version.

When doing a upgrade be sure to close any running instances of Multi Commander before running the tool. The Update tool can now replace files that are in use.

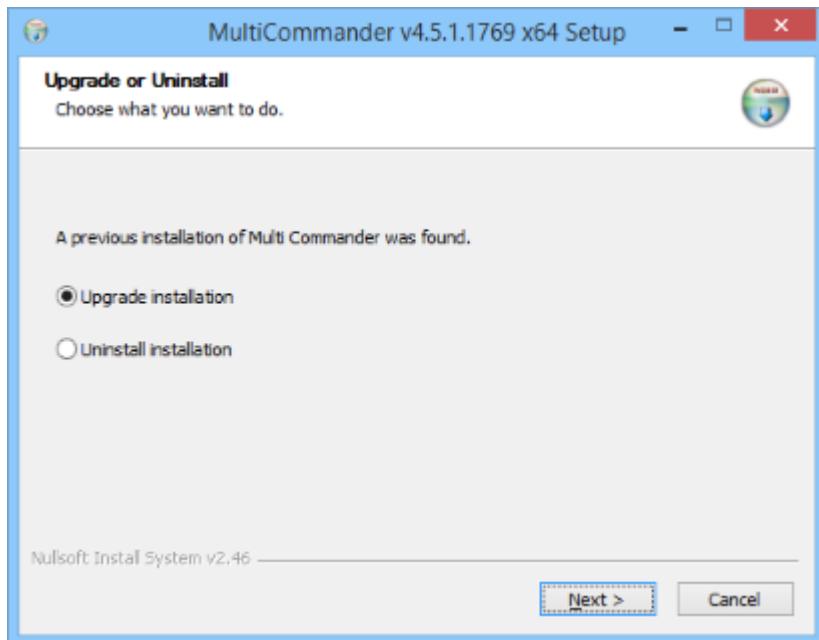
If you have configured Multi Commander to exit to the tray icon. Then Multi Commander is not really closed if you just press the X button. Then right click on the tray and select exit to really exit the program.

The update tool can also download beta version. and it also has some advanced options that can be set if you have issues with it.

Check the document page for [Multi Update](#) for more information about this tool.

Upgrade Using Installer

You can also upgrade using the install package. If the install package detects that you already have Multi Commander installed you will be given two options.



Upgrade Installation or Uninstall Installation. If you select Upgrade it will replace all existing files with the new one.

License

This version (5.x) of Multi Commander can be used for free (without paying for a license) for personal, commercial and government use.

You are allowed to install this version of Multi Commander on multiple computers.

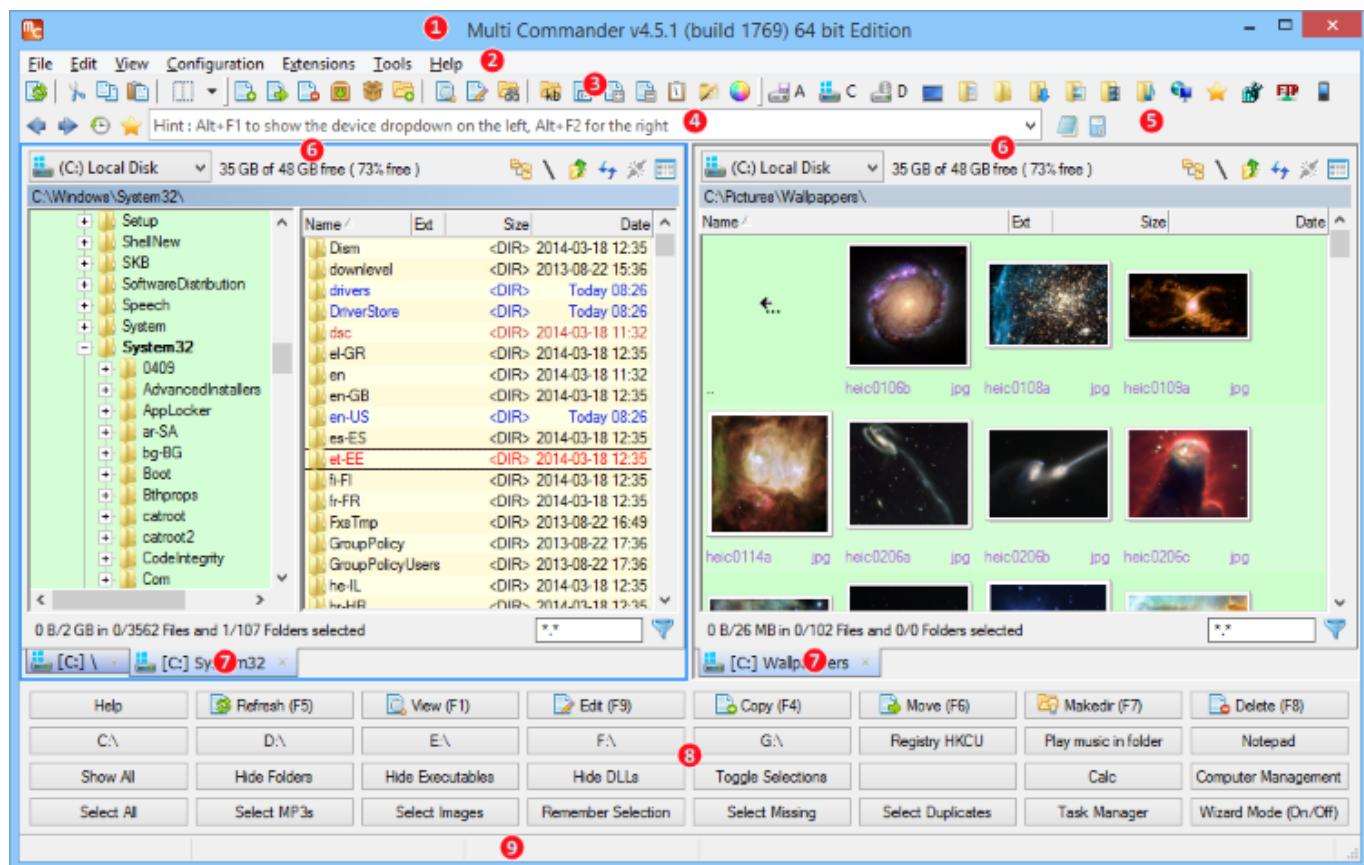
Multi Commander is not allowed to be used in situations and systems in which the failure of the Multi Commander software could lead to death, personal injury or severe physical or environmental damage.

Multi Commander, and Mathias Svensson (the author of Multi Commander) is not responsible for what you do with Multi Commander after you have installed it. It is up to you to be carefull. And always keep backup of you important data.

MultiCommander is a software that are designed to copy/move and delete files and using this software without proper knowledge of what you are doing, you might destroy and/or lost data. So always keep backups of important data.

Mathias Svensson also reserves the right to change this license in future version in any way.

Main Application Layout



1. [Title bar](#)
2. [Menu bar](#)
3. [Tool bar](#)
4. [Command line field](#)
5. [Quick Launch Bar](#)
6. [Presentation area](#)
7. [Tab area](#)
8. [Button panel](#)
9. [Status bar](#)

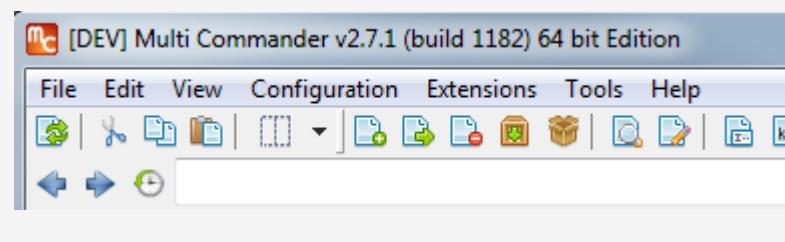
1. Title Bar

The topmost line of the MultiCommander window is the **Title Bar**; it shows the *version number* and *build number* of MultiCommander.

You can also add custom text to be shown at the beginning of the title. This is done using the `-T` command line parameter when invoking MultiCommander.exe

Hint

MultiCommander.exe -T=DEV .



2. Menu Bar

The second line (when visible) shows the **Menu Bar**. The default entries are: **File**, **Edit**, **View**, **Configuration**, **Extensions**, **Tools** and **Help**. You can create and modify additional entries for user-defined commands by following the procedures outlined in [Create User Defined Commands](#).

Clicking on any of these brings up a submenu, [as described of menu here](#).

Hint

The display of the Menu Bar can be toggled by the shortcut key **Ctrl+M**.

3. Toolbar Area

The Toolbar area contains buttons for some of the most used commands that you find in the menu. It also contains Buttons for available drives to open in the Explorer Panel. The Command Line field and the QuickLaunchBar are also here.

Read more about

[CommandLine field](#)

[QuickLaunchBar](#)

[Drive Bar](#)

[File Commands bar](#)

4. Command Line Field

5. Quick Launch Bar

6. Presentation Area

The next part of the display is the **presentation area**. This displays one or more GUIs that are controlled by extensions and plug-ins, such as the Explorer Panel and File Search.

7. Tab Area

8. Button Panel / Log Panel

Under the main presentation area there is by default a button panel. This panel can be toggled on and off with **Ctrl+B**. You can also have a Log Panel shown here. By default this is not visible, but it can be toggled by pressing **Ctrl+L**.

Read more about the [Button Panel](#)

Hint

Toggle button panel on/off with **Ctrl+B**.

Toggle Log panel on/off with **Ctrl+L**

9. Status Bar

The status bar shows various status information. For example, when copying files you will see the progress of the copy process here

Menu Bar



The menu entries are: **File**, **Edit**, **View**, **Configuration**, **Extensions**, **Tools** and **Help**.

You can add additional entries for user-defined commands by following the procedures outlined in [Create User Defined Commands](#).

Clicking on any of these brings up a submenu, as described below.

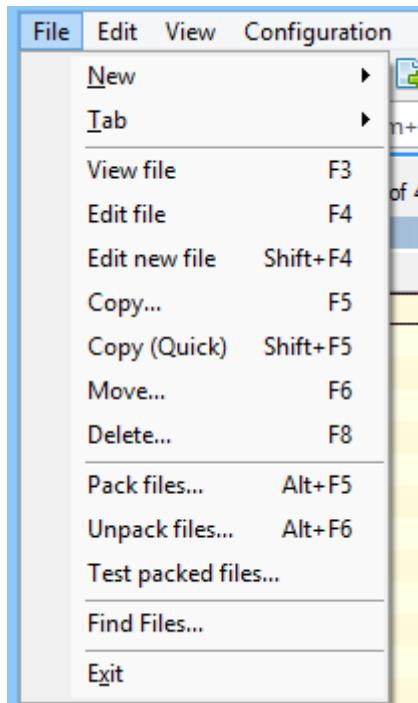
Menus

- [Menu - File](#)
- [Menu - Edit](#)
- [Menu - View](#)
- [Menu - Configuration](#)
- [Menu - Extensions](#)
- [Menu - Tools](#)
- [Menu - Help](#)

Hint

The display of the Menu Bar can be toggled by the shortcut key **Ctrl+M**.

Menu - File



displays a submenu of **File**-related items.

New

displays a submenu where you can create a new:

- *MultiCommander window*
- *MultiCommander window (as Administrator)*
- *Explorer Panel tab (in the active panel)*

Tab

displays a submenu where you can:

- *Close the current tab* (in the active panel)
- *Close all tabs but this one* (in the active panel)
- *Duplicate tab*
- *Move tab*

This displays another submenu where you can:

1. *Swap the location of the left/right panels*
 2. *Move the active panel to the other side*
- *Lock tab*

View file

allows you to view the selected file (using the **MultiData** viewer).

Edit file

allows you to edit the selected file(s) (using **Notepad**).

Copy...

displays the **Copy files and folders** dialog. (See [Copy/Move Operations](#).)

Copy (Quick)

(Shift+F5) creates a copy of the file/folder in focus in the same folder with ".bak" added to the end. Only the file/folder in focus is copied, even if there are other files/folders selected.

Move...

displays the **Move files and folders** dialog. (See [Copy/Move Operations](#).)

Delete...

displays the **Delete Files** dialog. (See [Delete Operations](#).)

Pack files...

displays the **Pack files** dialog.

Unpack files...

displays the **Unpack archives** dialog (but only if the file in focus is an archive.)

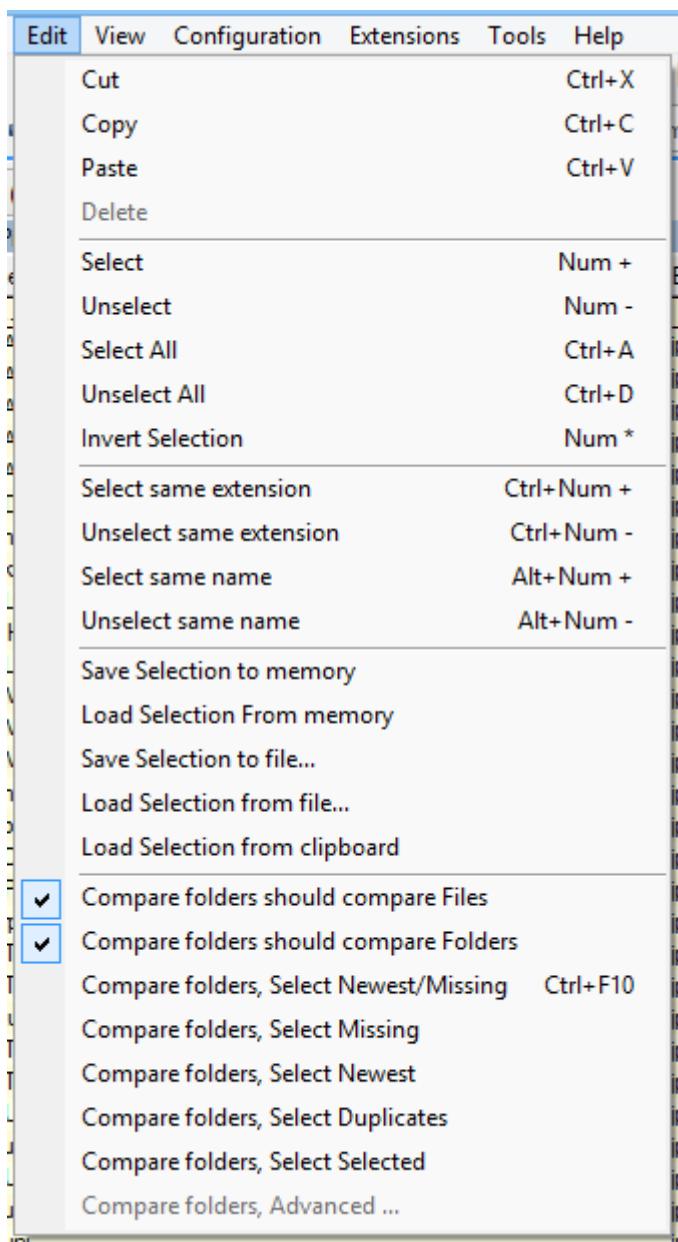
Test packed files...

tests the selected packed files for certain types of errors.

Exit

closes **MultiComander**.

Menu - Edit



displays a submenu of **Edit**-related items.

Cut

Designates the currently selected files/folders/text to be **moved** when **Paste** is selected.

Copy

Designates the currently selected files/folders/text to be **copied** when **Paste** is selected.

Paste

Copies or **Moves** the files/folders/text that were previously designated by **Edit->Copy** or **Edit->Cut**.

Delete

Deletes selected text.

Select

displays the **Expand Selection** dialog.

Unselect

displays the **Shrink Selection** dialog.

Select All

selects all files and folders in the active panel.

Unselect All

deselects all files and folders in the active panel.

Invert Selection

inverts all selections in the active panel, i.e. switches all *selected* files and folders to *unselected*, and all *unselected* files and folders to *selected*.

Select Same Extension

selects all files in the active panel having the **same extension** as the file currently in focus.

Unselect Same Extension

deselects all files in the active panel having the **same extension** as the file currently in focus.

Select Same Name

selects all files in the active panel having the **same filename** as the file currently in focus.

Unselect Same Name

deselects all files in the active panel having the **same filename** as the file currently in focus.

Save Selection to Memory

saves a list of the currently selected files/folders. The list is saved internally, and will be lost if MultiCommander is closed before the list is reloaded.

Load Selection from Memory

selects all files/folders designated in the currently saved **selection list**.

Save Selection to file...

displays a **Save-As** dialog to let you designate a file in which to save a list of the currently selected files/folders.

Load Selection from file...

displays an **Open** dialog to let you choose a file from which to load a list of files/folders to select. The list can be one that was previously saved, or a manually-created list of file and folder specifications.

Compare folders should compare files

is a checkbox option that toggles whether compare folders should compare the **files** within the folders.

Compare folders should compare folders

is a checkbox option that toggles whether compare folders should compare the **subfolders** within the folders.

Compare folders, Select Newest/Missing

compares the folder in the active panel with the one in the opposite panel, and selects (depending on the current options)

any files/folders in either panel that are either **missing** from the opposite panel, or are **newer** than the file/folder in the opposite panel with the same name.

Compare folders, Select Missing

compares the folder in the active panel with the one in the opposite panel, and selects (depending on the current options) any files/folders in either panel that are **missing** from the opposite panel.

Compare folders, Select Newest

compares the folder in the active panel with the one in the opposite panel, and selects (depending on the current options) any files/folders in either panel that are **newer** than the file/folder in the opposite panel with the same name.

Compare folders, Select Duplicates

compares the folder in the active panel with the one in the opposite panel, and selects (depending on the current options) any files/folders in either panel that are **duplicates**.

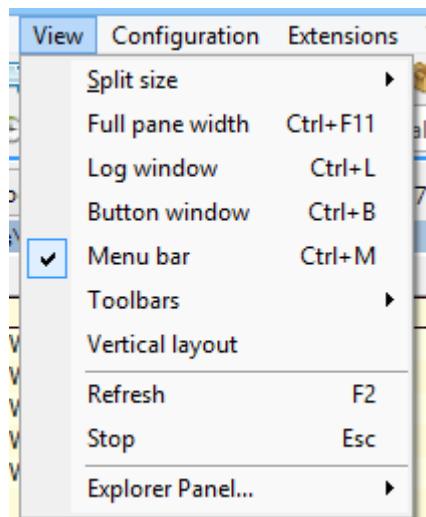
Compare folders, Select Selected

compares the folder in the active panel with the one in the opposite panel, and selects (depending on the current options) any files/folders in either panel that are **selected** in both panels.

Compare folders, Advanced

This is not yet enabled.

Menu - View



displays a submenu of **View**-related items.

Split size

displays a dialog that lets you choose one of the following **split-sizes** for the explorer panels: **0/100, 25/75, 50/50, 75/25, 100/0**.

Full pane width

expands the currently active panel to **full pane width**. (Shortcut key is **Ctrl+F11**.)

Log window

is a checkbox option that toggles showing the **Log window**. (Shortcut key is **Ctrl+L**.)

Button window

is a checkbox option that toggles showing the **Button window**. (Shortcut key is **Ctrl+B**.)

Menu bar

is a checkbox option that toggles showing the **Menu bar**. (Shortcut key is **Ctrl+M**.)

Toolbars

shows a list of selectable toolbars. Some of these are not really separate lines, but are subsections of the tool list line. Uncheck the boxes for those you don't wish to have visible:

- **Command Line Bar**

selects whether to display the *commandline* toolbar. When present, this is a separate line. The tools shown here are **Back**, **Forward**, **History**, **Commandline**, and **user-selected tools** (if any). (For details, see [Command Line bar](#).)

Standard

selects whether to show the *standard* tools in the **Tool Bar**. These are **Refresh**, **Cut to Clipboard**, **Copy to Clipboard**, **Paste from Clipboard**, and **Split**. (See [Standard Command Bar](#).)

File Commands

selects whether to show the *file command* tools in the **Tool Bar**. These are **Copy Selected Files**, **Move Selected Files**, **Delete Selected Files**, **Pack Selected Files**, **Unpack Selected Files**, **View File**, **Edit File**, **Rename Current File/Folder**, **Calculate folder size**, **Select**, **Unselect**, and **Copy the file/folder pathname to the clipboard**. (See [File Commands](#).)

Drives

selects whether to show the *drives* that are present on the system. (For details on the use of this subset, see [Drive Bar](#).)

Vertical layout

toggles between **vertical split** and **horizontal split**.

Refresh

refreshes the display. (Shortcut key is **F2** for Commander style, **F5** for Windows Explorer style.)

Stop

stops the current activity. (Shortcut key is **Esc**.)

Explorer Panel

selects a format or mode to control for the Explorer Panel. The choices are

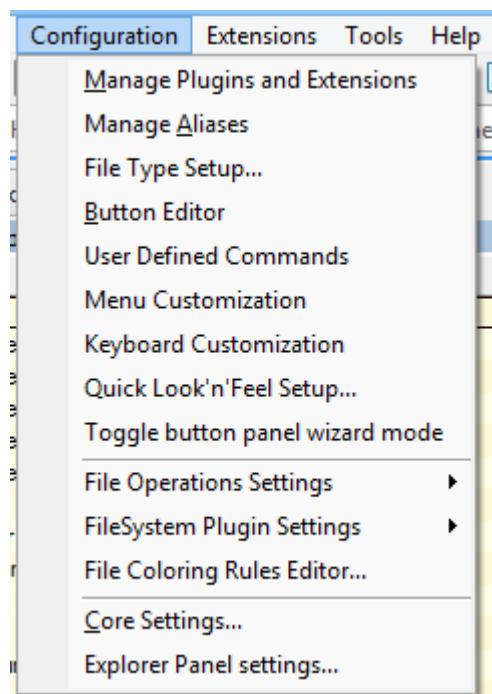
1. **Date format**

choices are *Standard*, *Week days*, or *File age*.

View Mode

choices are *Details*, *List*, or *Thumbnail List (Beta)*.

Menu - Configuration



displays a submenu of **Configuration**-related items.

Manage Plugins and Extensions

displays the **Extension Manager** dialog. This lets you manage **Application Extensions**, **File System Plugins**, **File Operations Plugins** and **File Properties Plugins**.

Manage Aliases

displays the **Alias Manager** dialog. This lets you create and modify aliases. (See [Managing Aliases](#).)

File Type Setup

displays the **File Type Setup** dialog. This lets you configure viewers/editors/launchers to be used for various file types. (See [FileType Setup](#).)

Button Editor

displays the **Button Editor** dialog. This lets you configure the Button Panel. (See [Using The Button Editor To Customize The Button Panel](#).)

User Defined Commands

displays the **User Commands** dialog. This lets you define or modify User Commands. (See [Create User Defined Commands](#).)

Menu Customization

displays the **User Menu Editor** dialog. (See [User-Menu Editor](#)).

Keyboard Customization

displays the **Customize Keyboards Setup** dialog. This lets you modify the keyboard shortcut commands in the menu. (See [Customize Keyboard Shortcuts](#).)

Quick Look'n'Feel Setup

displays the **Quick Look'n'Feel Setup** dialog. This lets you quickly change the look and feel of the file manager. (See [Change To Windows Explorer Look and Feel.](#))

Toggle Button Panel Wizard Mode

is a checkbox option that toggles whether the Button Panel is in **Wizard mode**. (See [Button Panel - Wizard Mode.](#))

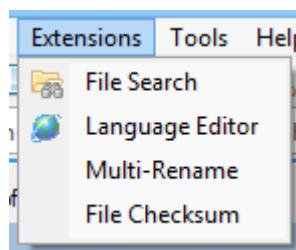
Core Settings

opens a new tab that lets you modify the MultiCommander **Core Settings**. (See [Core Settings.](#))

Explorer Panel Settings

opens a new tab that lets you modify the **Explorer Panel Settings**. (See [Explorer Panel Settings.](#))

Menu - Extensions



displays a submenu of **Extensions**-related items.

File Search

displays the **Find Files** dialog. (See [File Search.](#))

Language Editor

opens a new tab that lets you edit various MultiCommander messages.

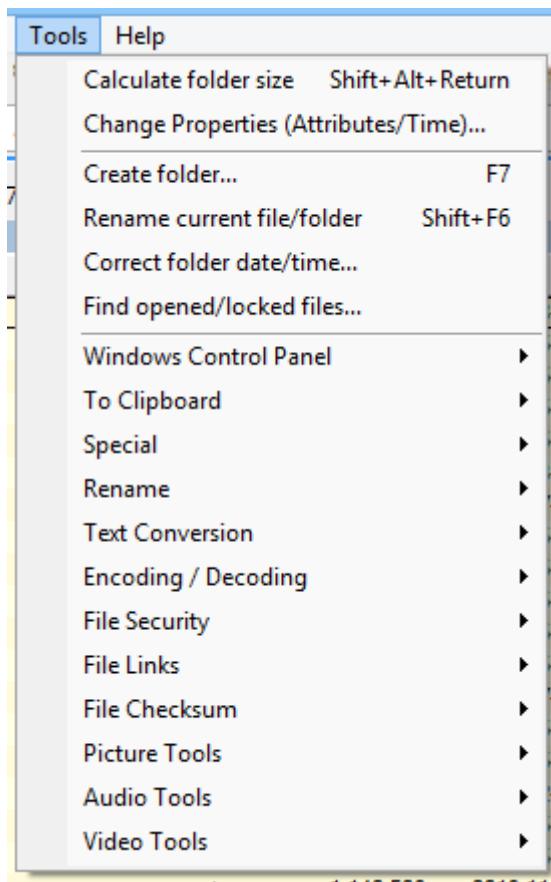
Multi-Rename

displays the **Multi-Rename** dialog.

File Checksum

opens a new tab that lets you verify checksums. (See [File Verifier \(CRC/SFV/MD5\).](#))

Menu - Tools



displays a submenu of **Tools**-related items.

Calculate Folder Size

calculates folder sizes. (See [Calculate Folder Size](#).)

Change Properties

displays the **Change attributes and/or time** dialog. This allows you to change the properties and date/time of the selected files and folders.

Create Folder...

displays the **Create new folder** dialog.

Rename Current File/Folder

lets you modify the name of the file or folder that is currently in focus.

Correct Folder Date/Time...

displays the **Correct Folder Date/Time** dialog. (See [Correct Folder Date/Time](#).)

Find Opened/Locked Files...

displays the **Find opened/locked Handle/File/DLL** dialog. (See [Find Opened/Locked Files](#).)

To Clipboard

displays a submenu of items that can be copied to the clipboard:

1. Files/Folders path and name
2. Files/Folders name
3. Files/Folders path

4. File version
5. File/Folder Date
6. Name, Size, Date
7. Path, Name, Size, Date
8. Comma-Separated List
9. Comma-Separated List, with full paths
10. File content as text

Special

lets you select the **oldest duplicates based on version** in name.

Rename

displays a submenu of renaming styles:

1. Uppercase
2. Lowercase
3. Capitalize
4. Replace text in name
5. Similar name, different extension

Text Conversion

displays a submenu of text conversions:

1. To Unix format (LF)
2. To Windows format (CRLF)
3. To Mac format (CR)
4. Tab to space
5. Reformat XML to Structured
6. Reformat XML to Compact

Encoding/Decoding

displays a submenu of text encoding/decoding options:

1. Base64 encode
2. Base64 decode

File Security

displays a submenu of **file security** options:

1. Take Ownership (Admin)
2. Remove all file/folder permissions (Admin)
3. Add file/folder permissions for user (Admin)

File Links

displays the **Create Links to File/Folder** dialog.

Picture Tools

displays a submenu of **Picture tools**:

1. Convert picture format...
2. Rotate picture...
3. Strip EXIF Metadata from pictures
4. Export EXIF Metadata from pictures
5. Adjust EXIF Date/Time Information...

For more information, see [Picture Tools](#).

Audio Tools

displays a submenu of **Audio tools**:

1. Remove Metadata Tags from audio files
2. Export Metadata from audio files
3. Show MP3 File Information

For more information, see [Audio Tools](#).

Video Tools

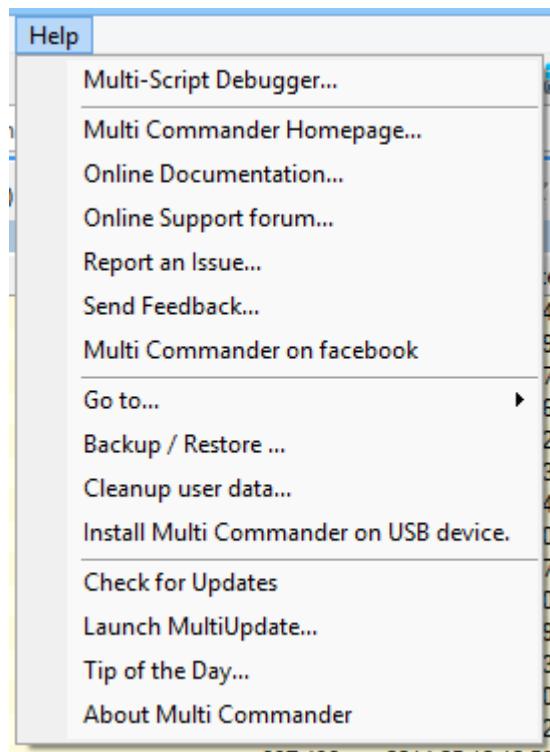
displays a submenu of **Vidio tools**:

1. Scan for Media Information

displays the **Scan Media Information** dialog, which lets you scan the selected files/folders for media information.
For more information, see [Video Tools](#).

Menu - Help

displays a submenu of **Help**-related items.



Multi-Script Debugger...

displays a dialog where you can adjust various settings. (Under development.)

MultiCommander Homepage

displays the **MultiCommander Homepage** in your default web browser.

Online Documentation...

displays the **MultiCommander Documentation** web page in your default web browser.

Online Support forum

displays the **MultiCommander Support Forum** web page in your default web browser.

Report an Issue...

displays the **MultiCommander Send Issue Report** web page in your default web browser.

Send Feedback...

displays the **Send Feedback** dialog so that you can submit feedback about Bugs, Problems or Feature requests.

Go to...

displays a submenu of places you can **Go To**:

1. Config folder
2. Log folder
3. Application folder

Backup/Restore...

displays the [**Backup/Restore**](#) dialog, where you can backup or restore your MultiCommander configuration.

Cleanup user data...

displays the **Cleanup leftover User Data** dialog.

Install MultiCommander on USB device.

displays the **Install MultiCommander to USB Device** dialog.

Check for Updates

checks to see if an **updated version** is available for MultiCommander.

Launch MultiUpdate...

prompts for execution of the **MultiUpdate** plugin, which checks for the current version of MultiCommander. (See [MultiUpdate](#).)

Tip of the Day...

opens the **Tip of the Day** window, where you can see the current Tip of the Day, and cycle through the complete set.

Register...

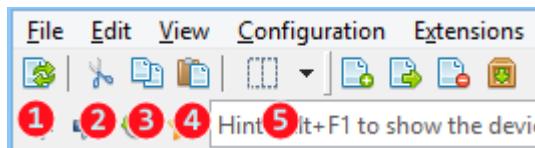
opens the **MultiCommander - Register License** dialog.

About MultiCommander

opens the **About MultiCommander** window. This contains Build information and links to Support websites.

Standard Command Bar

The *Standard Command Toolbar* is a subset of the toolbar that is shown if the corresponding checkbox in View->Toolbars is checked. It has a number of standard commands the extension can support. If the extension in the currently active panel supports any of these commands the commands will be enabled; otherwise they are disabled.



1. Refresh

Button 1 refreshes the content in the active panel.

2. Cut

Button 2 cuts the selected content to the clipboard.

3. Copy

Button 3 copies the selected content to the clipboard.

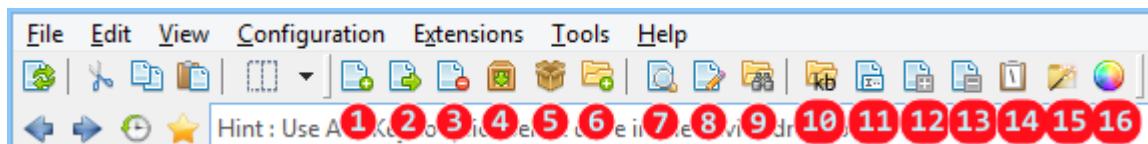
4. Paste

Button 4 pastes the clipboard into the active panel.

5. Resize Panel

Button 5 provides a dropdown list where you can select what sizes you want to split the panel into. If you click on the icon itself it reverts to a 50/50 split.

File Commands



The *File Commands* toolbar contain tools related working with files. Most of them only work if the active panel is a Explorer Panel. The toolbar is displayed whenever **View->Toolbars->File Commands** is checked. and it can also be disabled altogether in Explorer Panel settings.

1. Copy

Displays a dialog for launching a *Copy* operation. (See [Copy/Move Operations](#).)

2. Move

Displays a dialog for launching a *Move* operation. (See [Copy/Move Operations](#).)

3. Delete

Launches a *Delete* operation. (See [Delete Operations](#).)

4. Pack Selected Files

Opens the *Pack Files* dialog. (See [Pack Files](#).)

5. Unpack Selected Files

Opens the *Unpack Archives* dialog. (See [Unpack Files](#).)

6. Create Folder

Show the Create folder dialog.

Right click and a popup menu with a list of folders to create is shown. This list can be customized by selecting "Customize..."

7. View File

Causes the current file to be *viewed* using the default file viewer for that filetype. (See [File Type Setup](#).)

8. Edit file

Causes the current file to be *edited* using the default editor for that filetype. (See [File Type Setup](#).)

9. Find Files

Find Files

10. Calculate Folder Size

Causes *folder sizes* to be calculated. (See [Calculate Folder Size](#).)

11. Rename Current File/Folder

Renames the current file or folder using the operating system renaming protocol.
Right click and a popup menu with some Quick Rename options is shown.

12. Expand selection

Lets you specify a *filter* for selecting additional files/folders.

Right click and a popup menu with some dynamic selection items are shown (build from what items are shown in the current path.)

Also a some entire of a pre defined list of types are show in the list. And it is possible to add more or change them by selecting "Customize..."

13. Shrink selection

Lets you specify a *filter* for de-selecting files/folders.

Right click and a popup menu with some dynamic selection items are shown (build from what items are shown in the current path.)

Also a some entire of a pre defined list of types are show in the list. And it is possible to add more or change them by selecting "Customize..."

14. Copy PathName

Copies the file/folder *pathname* to the clipboard.

Right click and a popup menu with command for other copy path alternatives are shown.

15. Show/Hide Hidden and System files

Show hidden and system files.

Right click and a popup menu with option to only show/hide hidden files or show/hide system files.

16. File Coloring

A popup menu is show where the Rule based file coloring can be configured.

It is possible to enable/disable the file coloring from the menu. Also a list of all the file coloring profiles are shown so what file coloring profile to use can be easily changed

Drive Bar



The **Drive toolbar** is the rightmost subset of the toolbar, and is displayed whenever **View->Toolbars->Drives** is checked and if it is enabled in the Explorer Panel Settings.

The Drive toolbar shows a list of drives that the user can choose from. It also shows common folder locations like desktop, pictures, download and more. And all the virtual filesystem (like REG, FTP: WPD:) are also here.

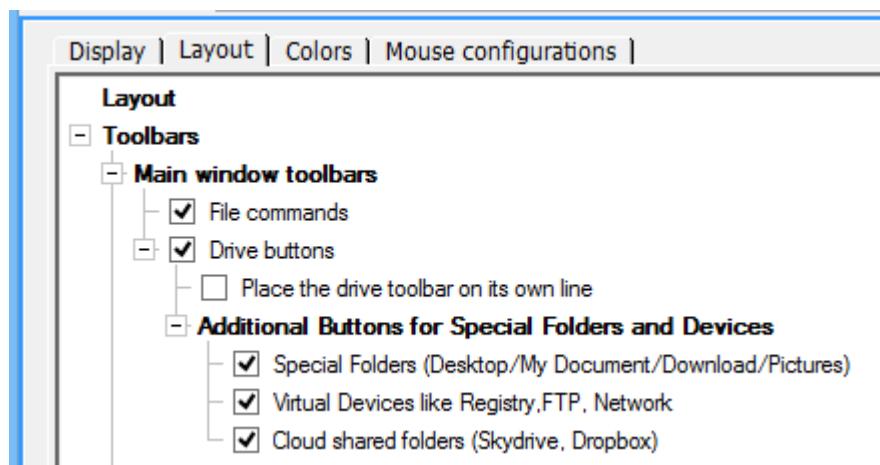
Clicking on one of these icons causes the current tab to switch to that location.

You can also right click on a drive button to show the Windows Shell Context Menu (The Right click menu) for that drive.

Add more buttons

It is possible to configure which of the special folders and virtual devices to show on the drive toolbar.

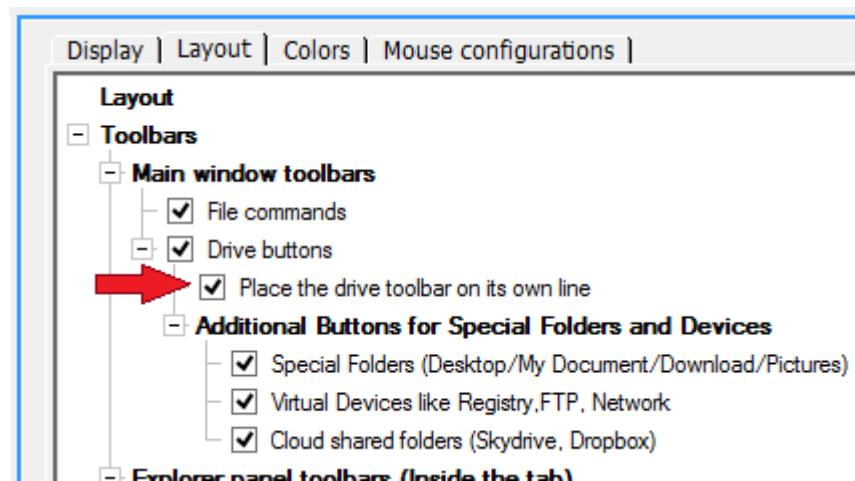
To configure the drive toolbar go to **Menu > Configuration > Explorer Panel Settings** and click on the **Layout** tab



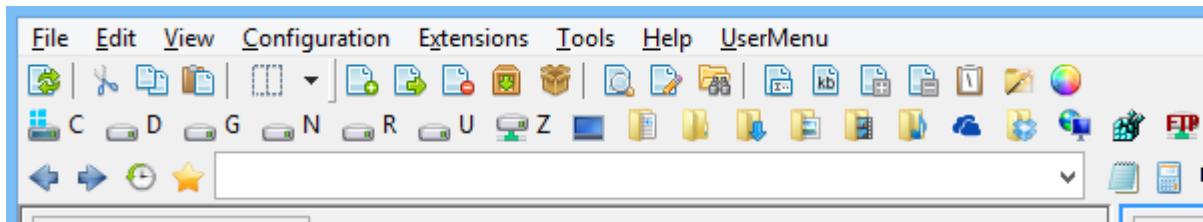
If you enable all of them the toolbar might be very long and the system will try to auto rearrange it to its own line if it does not fit in the Window.

On some systems this causes the toolbar area to rearrange itself incorrectly and it might give you a bad look.

That can be fixed if you in settings enable that the drive toolbar should be placed on its own line from the beginning



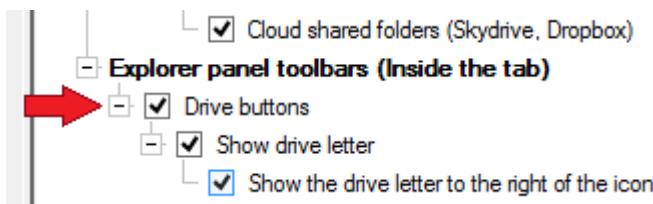
Enabling that option and the drive toolbar will be placed on its own line and if there are a lot of driver buttons they will all fit better.



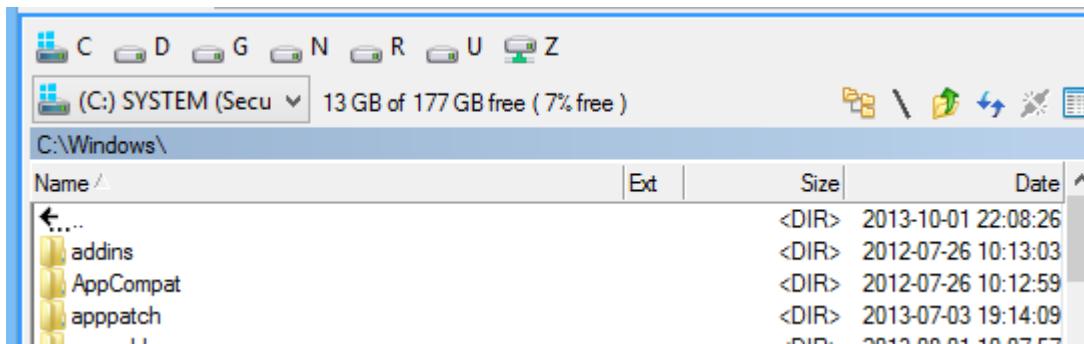
Drivebar Inside Explorer Panel

It is also possible to place the drivebar inside the Explorer Panel.

That is enabled by checking the option "Drive buttons" under the "Explorer Panel toolbars" section



There will now be a drive buttons bar inside the explorer panel.



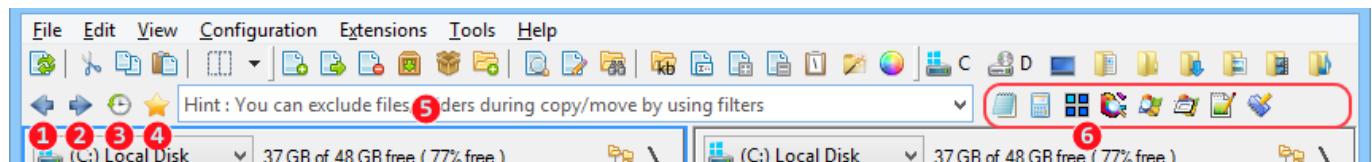
The drive bar that are inside the Explorer Panel is limited in the way that it does not support showing special folders and virtual devices.

Other configuration

If you have a drive bar active (application global or inside the Explorer Panel) you can remove the Device Dropdown list. However if you are a user that like to use hotkey for fast access then it is not recommended.

Command Line bar

The fourth line in the Main Application Layout is the **Command Line Bar**, shown below. This toolbar can be removed from the display by unchecking the check box in **View->Toolbars**.



The buttons are enabled if the active extension (panel) supports back/forward/history. If the active extension does not support these commands, the buttons are disabled.

1. Back

Button **1** tells the extension to go back one step. If this button is held down for 2 seconds, a list of previous locations that you can select from is displayed. If the Active extension is the Explorer Panel it will go back to the previously visited path.

2. Forward

Button **2** tells the extension to go forward one step. This can only be done if you have previously gone backwards. If this button is held down for 2 seconds, a list of all the forward locations is shown

3. History

Button **3** shows a list of the previous 40 locations that the currently active extension has visited. From this list a previous location can be accessed quickly.

The default keyboard shortcut for Back is ALT+Left Arrow, Forward is ALT+Right Arrow and History is Alt+Down Arrow. In the Explorer Panel the Back and Forward mouse button can also be used, if your mouse supports these buttons.

4. Favorites Window

The button shown as a *star* at the left end of the *Command Line*. Clicking the *star* button brings up the **Favorites** window. This window contains three sections:

- **Quick Paths (F1)**

Quick Paths are paths that can be selected by pressing *Ctrl+num*. There can be only 10 *Quick Paths* defined.

- **Favorites (F2)**

Favorites are paths that can be defined as a list of favorite paths. To select a Favorite path, choose the desired entry and press Enter.

- **Web (F3)**

Web paths are URLs that can be defined as a list of web addresses. To select a web address, choose the desired entry and press Enter.

5. Command line field

The Command line field (number **5**) allows you to enter commands for the active extension or plug-in. If the extension or plug-in does not handle the command the command is sent to Windows. If the command is a filename and the command is sent to Windows to be handled, Windows will launch that program. If the command is a URL, a browser is opened for that

URL.

If you type in a command for Windows to launch, you can insert a "*" in front if the command to run it as administrator. For example, "***cmd**" will launch the command prompt as administrator.

The Explorer Panel supports commands such as the command prompt's "**cd**" command. Thus if an ExplorerPanel is currently active you can type "**cd ..**" , "**mkdir myfolder**" and the Explorer Panel will create this folder.

Special Internal Commands

The Command line field supports some special internal commands:

*<external program>	Start an external program as administrator, e.g. "*cmd" starts an admin prompt at the current path
!<cmd / path>	Normally Multi Commander will go to a path if it is entered, Typing ! in front of the path will force the path/cmd to be opened by Windows Shell
cd <path>	Change directory
mkdir <path>	Create Folder
cf <file>	Create a new empty file
alias <name>=<expanded to>	Create a new alias, eg "alias d=d:\bin"
alias2 <name>	Create a new alias for the current path, eg "alias2 d"
topmost 1	Change Multi Commander from being always on top. Use 1 to enable, other to disable
addlanguage	Used to add support for another language so that you can create new translations
importcmd	Used to import a script command quickly
dbg folog	Enable full file operations logging
dbg fslog	Enable full file system logging
dbg applog	Enable full application logging
:save	Save all autoload settings for all tabs. This is the same save that is done before exit.
:showhidden	Change the settings to show all system and hidden files
:hidehidden	Change the settings to hide all system and hidden files
:goapp	Go to the path from which Multi Commander is started
:userdata	Go to the path under the user account where Multi Commander saves data
:gomcreg	Go to the registry path used by Multi Commander
:golog	Go to the path where Multi Commander stores the log files
:goconfig	Go to the path where Multi Commander reads/writes the settings files.
:box	Go to the location where you have your DropBox folder (if you have DropBox installed)
:sky	Go to the location where you have your SkyDrive folder (if you have SkyDrive installed)
:desk	Go to the desktop
:dl	Go to the Windows Download folder (not available on Windows XP/2003)
:ctc	Clear Thumbnail Cache - Will remove all thumbnail cache information stored in memory
:rdrive	Refresh all devices in Device dropdown and Device Toolbar.
:hex2dec <num>	Convert a hex value to decimal. Eg ":hex2dec ff"
:winerr <num>	Show the error message of an internal windows error code. Eg ":winerr 5"

6. Quick Launch Buttons

The space following the *Command Line Field* is used for **Quick Launch Buttons**. If you right-click in that area, you will get a menu with the following entries:

For further information, see [Quick Launch Bar](#).

- **Allow - Insert button by dropping file/folder**

If selected, this allows the use of drag'n'drop to insert a file or folder.

- **Allow - Rearrange of buttons**

If selected, this allows the buttons to be rearranged using drag'n'drop.

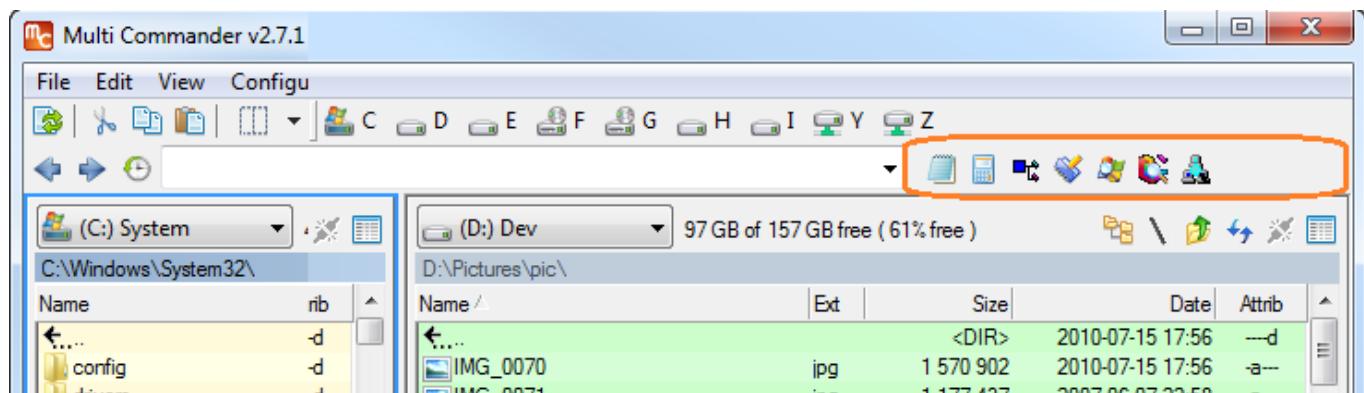
- **Insert User Defined Command**

brings up the *User Defined Command* menu, where you can select an existing User Defined Command (or create a new one) to be inserted.

- **Insert Separator**

inserts a separator line.

QuickLaunch Bar



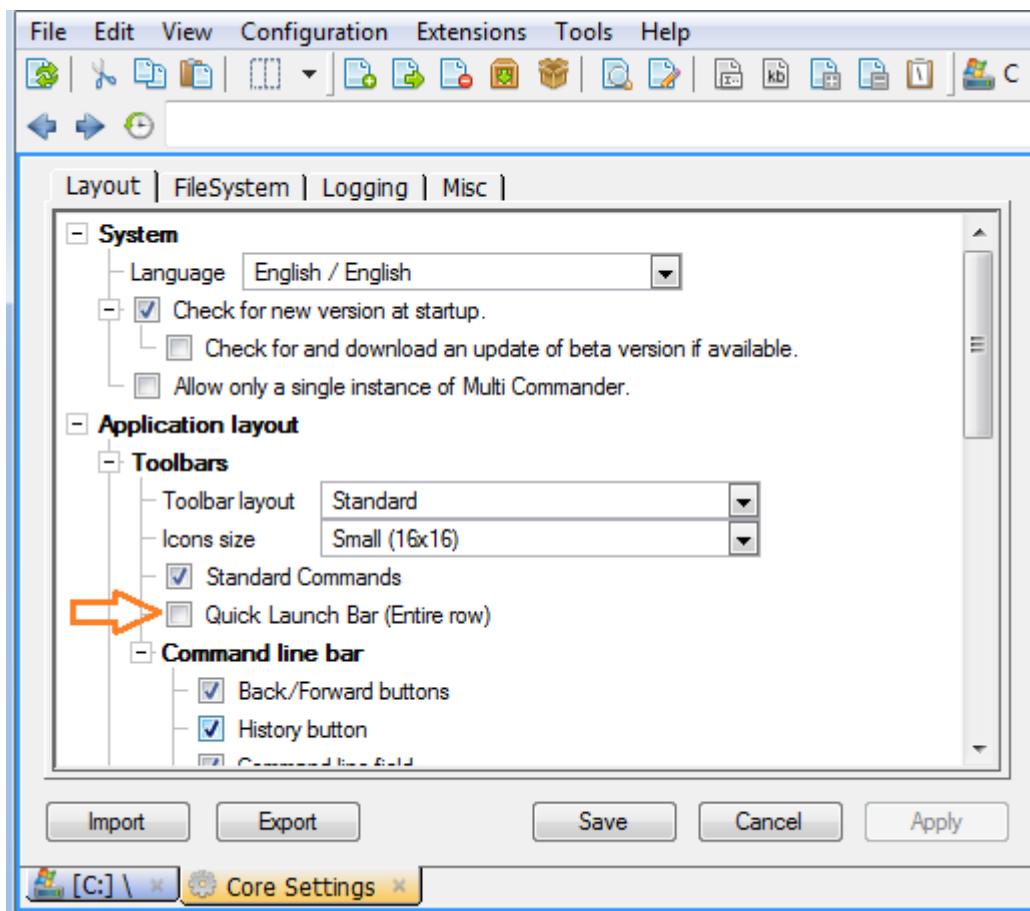
The **Quick launch bar** is located after the *command line field* and the more buttons that are added the shorter the command line field will be.

Here is a list of some of the features of the Quick Launch bar:

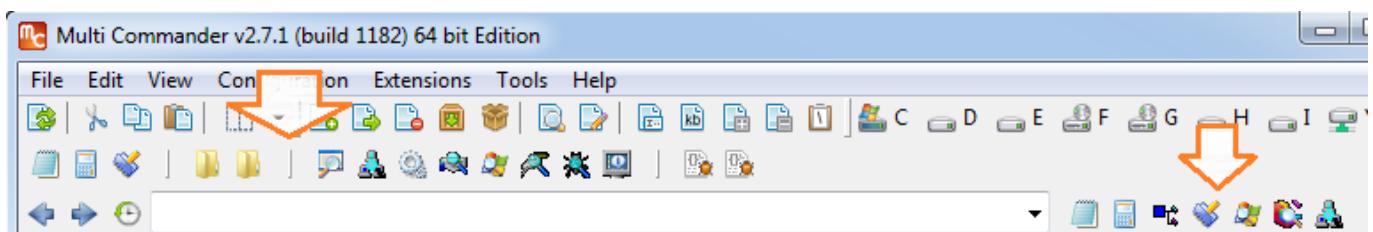
- If you drop a file or folder on the Quick Launch bar area, a button for it will be automatically added for it.
- You can rearrange the buttons in the Quick Launch bar by dragging them around.
- If a file is dropped on a program button, that file is sent as a parameter to the program it is dropped on. For example, if you drop a text file on the notepad button, the text file will be opened in notepad.
- If you click on a folder button, that folder will be opened inside Multi Commander.
- If you (left-) click on a program button, that program will be started. If you right-click on it, a menu will be opened that allows you to:
 1. Start it normally
 2. Start it as Administrator (Vista/Win7 +)
 3. Go to the folder containing the program file
 4. Remove the button from the Quick Launch bar
 5. Customize the way the program is opened, or what icon or tooltip to show for it, etc.
 6. Insert a user-defined command button on its left
 7. Insert a separator to its left. (This lets you group Quick Launch buttons.)

Entire Quick Launch Row

To add an entire Quick Launch row, go to *Configuration > Core Settings > Application Layout > Toolbars > Quicklauch bar (Entire row)* and click on the checkbox. (Clear the checkbox to remove the extra row.)



You will now have two Quick Launch Bars.



Enable/Disable

If the Quick Launch Bar items are unchecked under Core Settings, then they are not created or loaded during startup. But if they are checked in Core Settings, they can still be hidden if *Configuration > Core Settings > Application Layout > Toolbars > CommandLine Bar > Quick Launch bar* is unchecked.

You should only use that option to temporarily hide them. If you want to permanently remove them, then uncheck them under Core Settings. If they are just hidden they are still using memory resources and are loaded and created during startup. To minimize resource usage, disable them under Core Settings.

Status Bar



At the very bottom of the MultiCommander window is an area where the application can show status information about background tasks like folder sizing and copy operations and more.

The progress status for copy/delete/move operation is shown to the right in the status bar area. If you double-click on the progressbar the progress dialog window for that operation will be brought to the foreground and shown.

When an Administrator Mode is active you will see an indication of that in the status bar. The Administrator mode will turn itself off when it has been idle for a couple of minutes, but if you right-click on the status text for it a popup menu will be shown where you can turn off Administrator Mode without waiting.

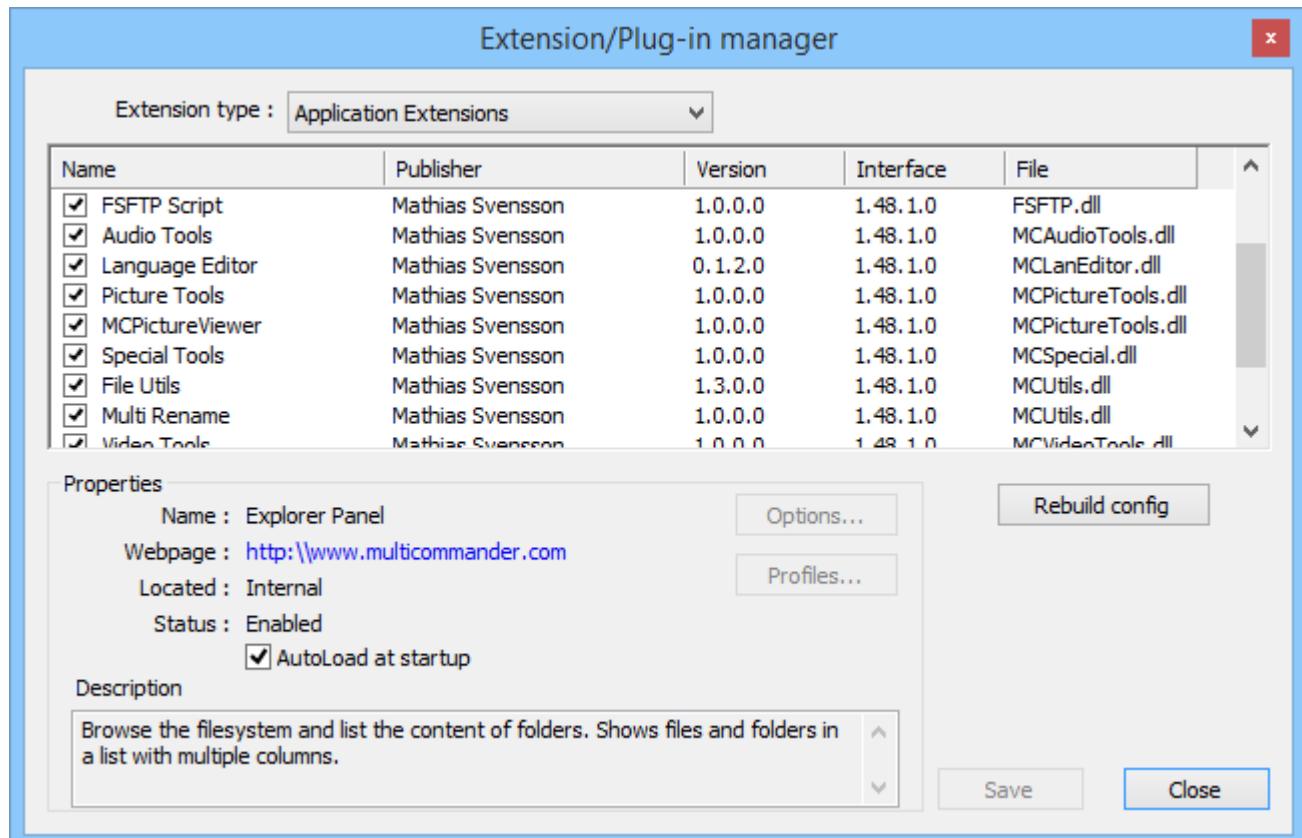
Extensions and Plug-ins

Multi Commander is built around **extensions** and **plug-ins**.

Some of the extension are built directly into MultiCommander and some of them are located as external DLL files.

This allows for Multi Commander to grow with new features in an easy way and for other developers to create plugins to extend Multi Commander.

With the **Manage Plugins and Extensions** selection in the *Configuration* menu you can enable and disable extensions and plugins.



There are 4 types of extensions and plug-ins:

Application Extensions

These are extensions that have GUIs and create new tabs with functions in Multi Commander. Examples of Application Extensions are the Explorer Panel and File Search.

FileSystem Plug-ins

All filesystems and archives that Multi Commander supports are from plug-ins. There are plug-ins for Zip, 7-Zip, RAR, FTP and the Registry.

File Operations Plugins

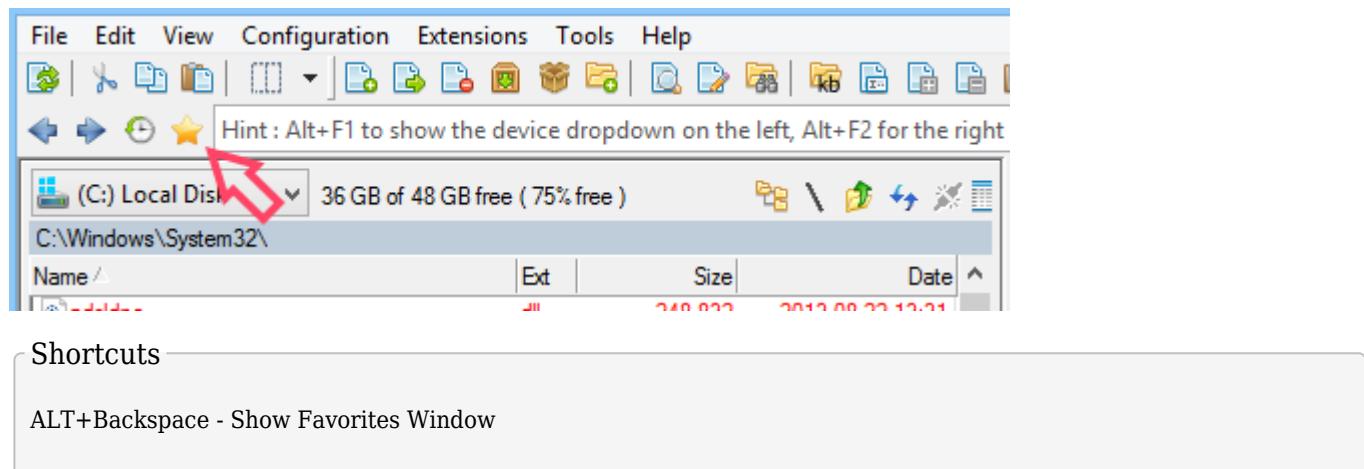
These are plugins that are used when copying and moving files. These plug-ins can change how and where files are read and written.

FileProperties Plug-ins

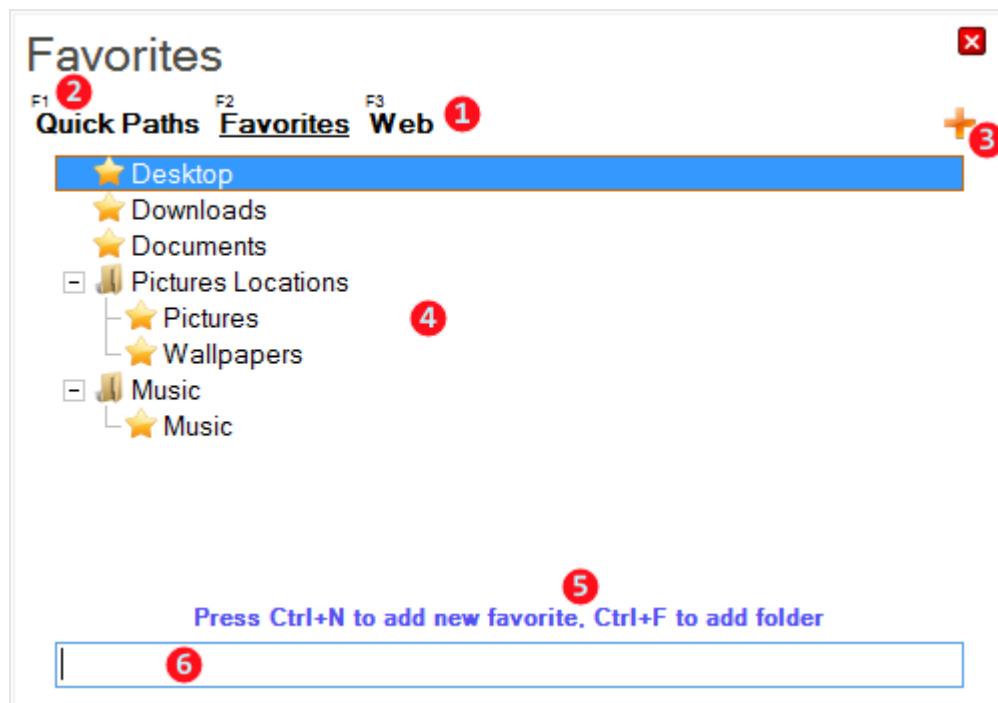
These plugins allow for more data to be shown about files in new columns in the Explorer Panel. For Example the file system plugin FS-Registry also has a FileProperties Extension allowing you to see the special registry information in the ExplorerPanel.

Favorites

The Favorites window will be shown if you press the Favorites button on the toolbar.



When the Favorites window is shown it will be shown with the section show that you had the last time it was shown.



Favorite is a path to location that can be accessed faster. It can be a path to a filesystem location, but also a path to a web address is possible.

The favorites window is automatically closed if it loses input focus. So you do not have to click the X button. Pressing ESC or clicking anywhere outside of the window will close it.

1. Favorites Sections

It is possible to have multiple sections of favorites for easier management.

With many sections it is possible to have different lists of favorites for different purposes.

Above the section name you will see F1, F2 and so on. This is the shortcut key to switch to that section.

2. Quick Paths

The first section is special. This section has 10 fixed favorites that are assigned to the QuickPath shortcut keys. The QuickPaths are accessible from almost anywhere by pressing **Ctrl+1**, **Ctrl+2** and so on up to **Ctrl+10**.

Right-click on a QuickPath item and select what you want to do: Clear/Modify/Copy/Set Current.

To set the current active path to an QuickPath item selected the item to assigned the path to and press **Ctrl+S**

3. Add Section

Press this to add a new section to the list.

Right click on existing section titles to rename or remove them

4. Favorites list

All favorites for the active section is listed here. With right click a menu is shown with actions that can performed in the list

It is also possible to create subsection where favorites can be place/move to. Subsection are good when having many favorites and favorites need to be better organized.

To Add a new Favorites to the list press **Ctrl+N** the favorites name and path is by default the path of the active panel. Press **Ctrl+F** to create a new subsection, **Ctrl+E** will edit and allow the current selected favorite item to be modified.

Pressing **Ctrl+Q** and the list will toggle between showing the favorites name and the target path.

Ctrl+C and **Ctrl+V** can be use to copy and paste favorites. It can be used to copy one favorite from on section and paste it into a new section.

It is also possible to drag and drop favroties to rearrange the order and placment of favorites inside the current section.

5. Help/Hint Texts

This field will show hints and help text

6. Filter

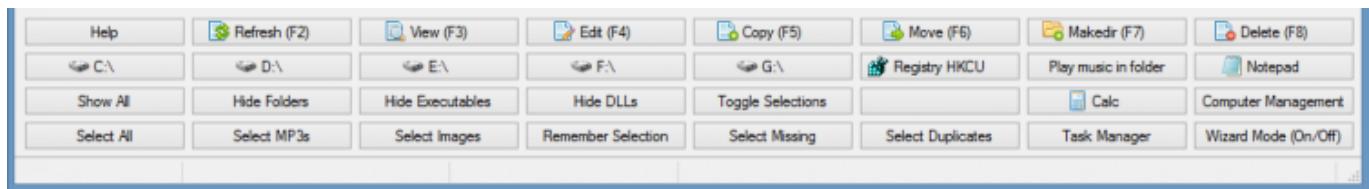
Typing here and the list result will be filtered. It is a way to minimize the list if there are a lot of favorites.

Browse Favorites like a FileSystem

The favorites is also exposed as a virtual file system under the virtual device **FAV:** and it is also found in the device dropdown or folder tree

[Read more about that here](#)

Button Panel



The button panel is shown at the bottom of the program. All of this buttons can be customized in the [Button Editor](#)

A button can be set to change the location that one of the panel are showing or it to call any internal or external command or script.

Since the button panel can run any of the build in script the buttons can be customized to be very powerful.

By default there are 4 rows of button and 8 button per row, every button can have 4 actions set to it giving you up to 128 commands for easy access.

The number of rows can also be change from 1 to 6 and the entire Button Panel can also be disabled.

When holding Ctrl, Shift or Ctrl+Shift (For mouse it is holding Right or Middle button pressed) the button panel is switch to its alternative buttons. Allowing easy access for the alternative buttons.

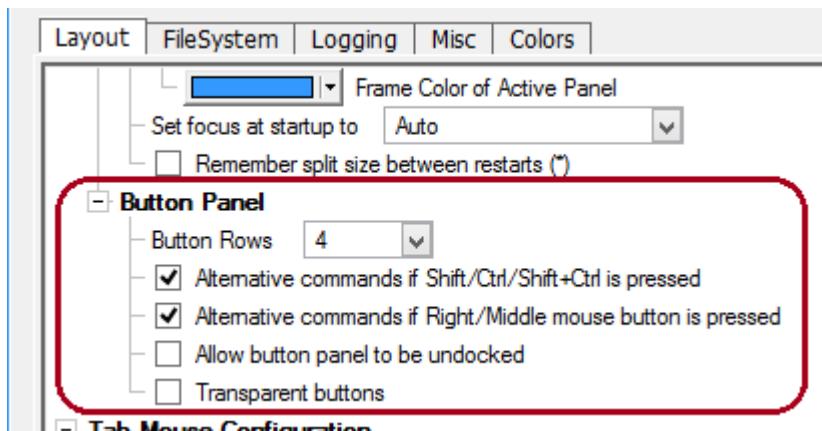
If alternative buttons are not used they can be disabled in settings

Shortcuts

Ctrl+B - Toggle the button panel on/off

Button Panels Settings

The settings for the button panel is found under the Core Settings. (Menu > Configuration > Core Settings)

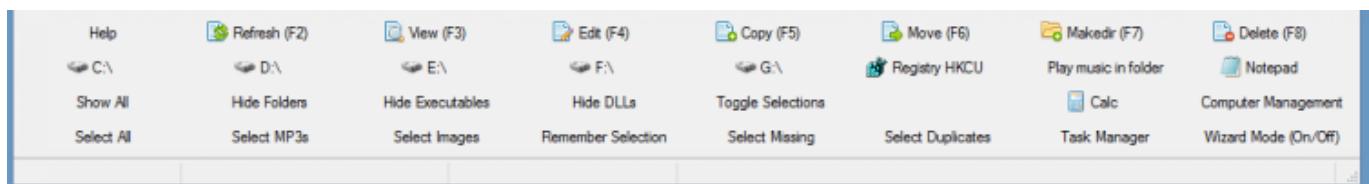


Option - Alternative Button

The two first button panel option enables/disable the switching of the button panel to show the alternative buttons actions. If both if this option is disabled there is no way to access the alternative buttons.

Option - Transparent/Flat

The Transparent option will remove the board and make the button transparent. Giving a flat look



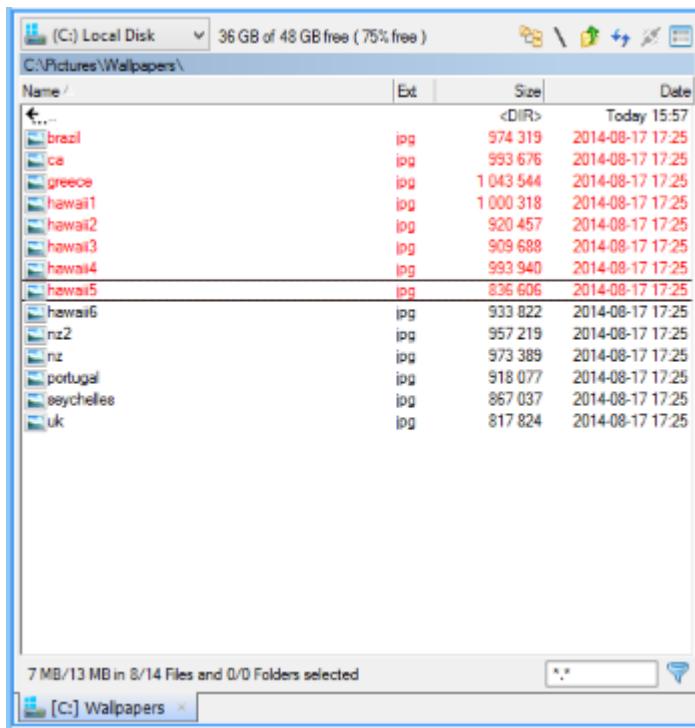
More

Read more about how to customize buttons with the [Button Editor](#) and how [button wizard](#) works

Explorer Panel

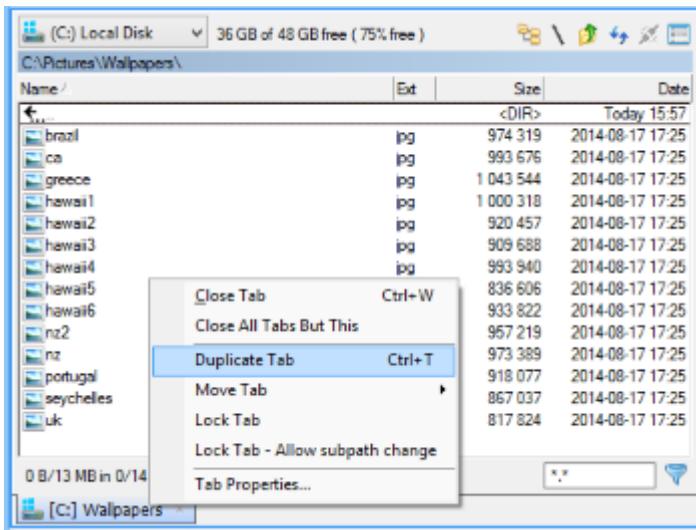
The **Explorer Panel** is the main panel in MultiCommander.

It is from here that you browse the file system and select files for copying, moving, renaming, deleting, viewing and so forth.



You can have multiple tabs that contains a Explorer Panel. Each Explorer Panel is a separate file browser.

The Explorer Panel also adds a couple of additional commands to the context menu on the Panel tab. It adds supports for *Duplicate Tab* and *Lock Tab*, *Tab Properties*..



By selecting "**Duplicate Tab**" from the Context Menu on the Tab or with the keyboard shortcut **CTRL+T**, a new Explorer Panel will be shown. It will display the same location that the Explorer Panel you had active is showing.

With "**Lock Tab**" the Explorer Panel will be locked at the location it currently displays. If you try to go to another location a new Explorer Panel will be opened for that location.

Tab Properties will show a window where you can customize the tab colors and you can from there also force a tab name,

[Read more about the Layout](#) of the Explorer Panel

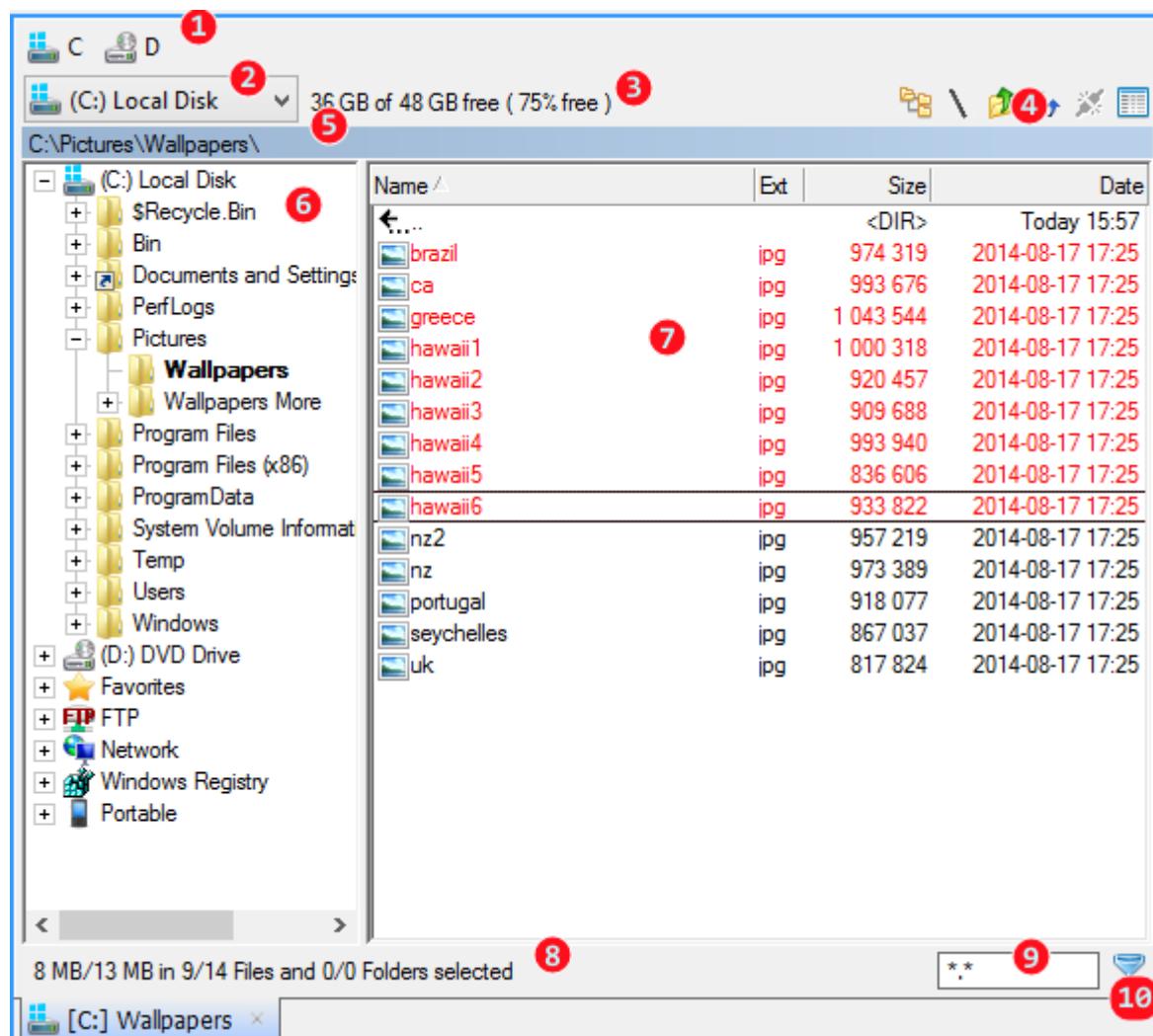
[Read more about the functionality of the Explorer Panel](#)

Explorer Panel - Layout

The layout of the Explorer Panel looks similar to the display you see in most file managers that you are already familiar with, although it has many more features than most of them.

Below is the description of the various parts of the display.

The image below shows an example of how the Explorer Panel can look with all of its user interface options enabled. Most of the user interface items can be configured and turned on/off in settings.



1. Device toolbar

Here you can get a list of available devices. This toolbar is very similar to the global device toolbar that you can have, except that this toolbar will only show devices. Special folders and virtual devices that the global toolbar can show is not shown in this Explorer Panel Device Toolbar.

You can enable/disable this toolbar in Explorer Panel Settings by checking/unchecking the "Drive buttons" options.

2. Device Dropdown list

This dropdown list shows not only devices. It shows Devices, Special Folders, Virtual Devices. When an item in the dropdown list is selected the explorer panel will go to that location and show the content of that path.

For fast access to drives you can press the letter E when the list is shown to make the Explorer Panel go to the E: drive. If the **SHIFT** key is pressed while selecting a location, that location is then opened in a new Explorer Panel tab.

In settings it is possible to configure what extended location types should be shown in the dropdown list.

Hint

Pressing **Alt+F1** the Device Dropdown in the left panel will be shown. **Alt+F2** and the one on the right is shown

3. Free Space area

Shows the available free space on the drive, and the total capacity of the volume.
It also shows the percentage of the total volume capacity that is available.

If you double click on the Free Space message, it will show *used* space instead of *available* space. You can also set which of the two ways you want as default by going to *Configuration->Explorer Panel settings->Layout->GUI Components->Show free/used space*.

4. Command Buttons

This buttons are some command that affect only the active explorer panel it is pressed on.

Folder Tree

Splits the panel to show the *folder tree* in the left portion. (shortcut key *Alt + Num** (See [Folder tree](#).)

Go To Root Button

Click on the backslash symbol ("\") to go to the root of the device (shortcut key *Ctrl + Back*).

Go Up

Click on this icon to go up to the parent level in the file structure (shortcut key *Back*).

Refresh / Reload

Click on this icon to force a refresh of the current path; hold down the **CTRL** key to force a flush of the cache as well as a forced reload of the current path. All selections will be lost.

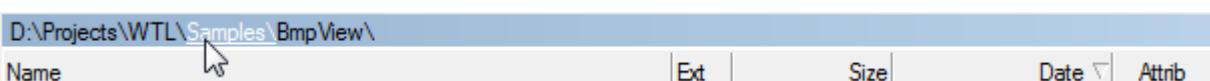
Disconnect

This icon will disconnect the current device. If the current device is an external USB drive it will be prepared for safe removal. If the device is an FTP connection then the FTP connection will be disconnected.

View Mode

This icon will display a menu of possible **View Modes**. Currently these consist of **Details**, **List**, and **Thumbnail List (Beta)**.

5. Path field



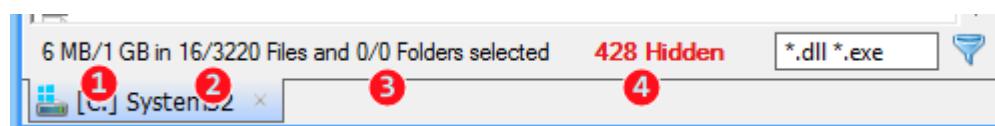
The path field shows the current path that the explorer panel displays. Sub folders in the path that is shown are clickable.
Clicking on a subpath causes the Explorer Panel to try to go to that path.

The path field have many options and feature. [Read more about the Path field](#).

6. Folder Tree

7. File list view

8. Selection Status bar



The Panel Status bar appears at the bottem of the panel. The various parts are described below:

1 - Select / Total Size

The size of all the selected files and folders (if folders are showing file size) and the total size of all visible files and folders.

2 - Selected / Total Files

Number of selected files and the total visible files.

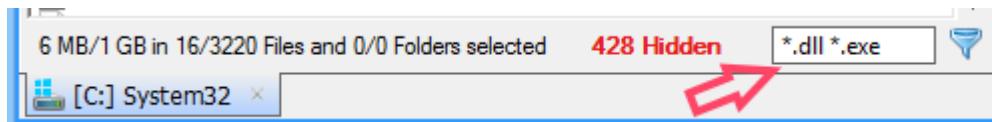
3 - Selected / Total Folders

Number of selected folders and the total visible folders.

4 - Hidden files / Folders

Number of files and folders that are hidden by the view filter.

9. View Filter



Filter the view to only show certain files or folders. Prefixing a filter with "-" will exclude those files. To create a filter to target folders, prefix it with "/".

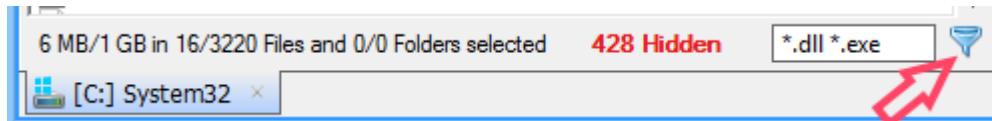
You can edit the View Filter from a keyboard shortcut by pressing **CTRL+F** (if using the default commander-styled setup).

Examples

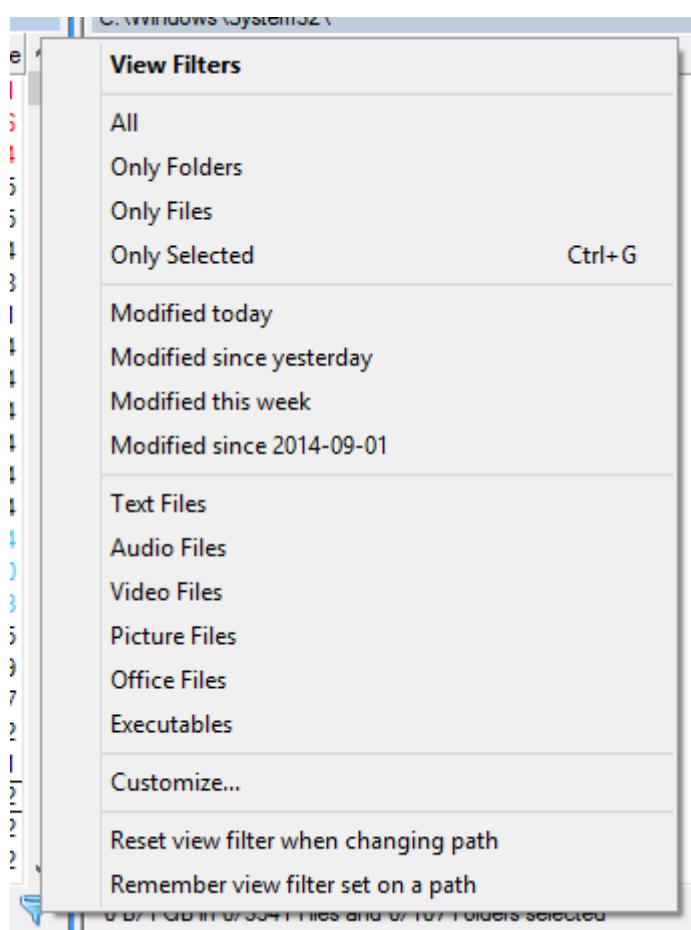
. -*.jpg	Show all files, except jpg files.
. -/backup	Do not show the "backup" folder
*.jpg *.tiff	Show only jpg and tiff files.

It is also possible to change the view filter from a [CustomCommand](#).

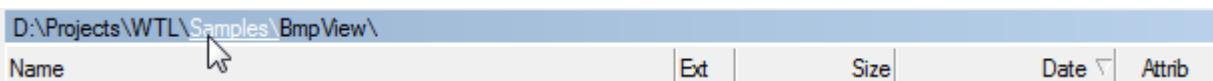
10. Reset Filter and Quick Filters



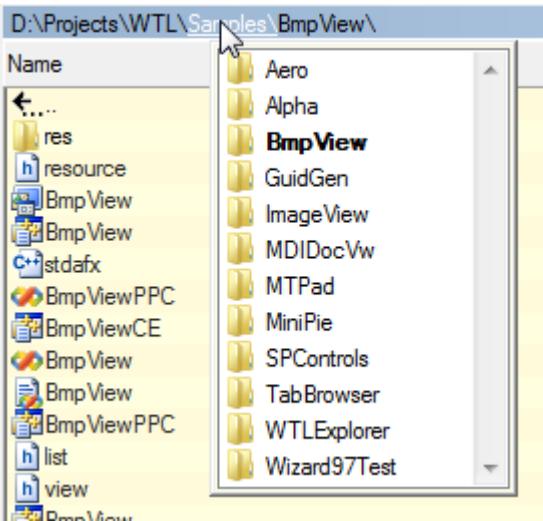
Click to toggle quick filter. Right click to show a popup menu with pre defined set of View Filters. By selecting "Customize..." it is possible to edit the list of quick filters and it is also possible to add new filters to the list.



Path field



The path field shows the current path that the explorer panel displays. Sub folders in the path that is shown are clickable. Clicking on a subpath causes the Explorer Panel to try to go to that path.



If you hold down the **CTRL** key while clicking on a sub folder a list of all the sub-folders under that path will be shown. You can then select a sub-folder that you want to go to from that list.

Special Feature

A special feature is that if you hold down both **SHIFT** and **CTRL** keys while clicking on a sub-folder in the path field, and then select a sub folder from the folder list, it will try to go to the same sub-path under that folder as the one where you are currently located. E.g. If you are currently located in the path

"**D:\Projects\MyApp\Version_1.0.0\Source\Includes**" and you click on the "**MyApp**" sub path while holding done **Shift+Ctrl**, then in the folder list that is shown you Select "**Version_2.0.0**", it will try to go to "**D:\Projects\MyApp\Version_2.0.0\Source\Includes**". If that path does not exist it will go to the closest path that does exist.

Editing the Path Field



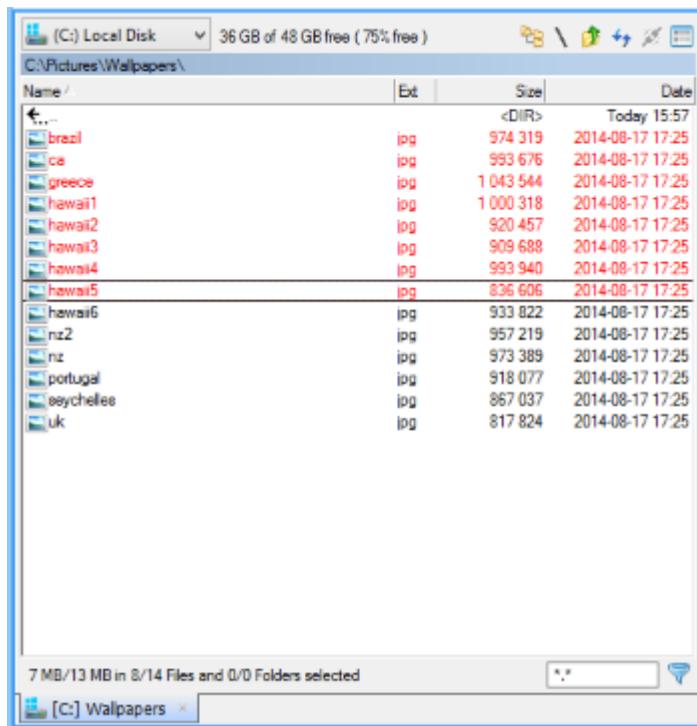
By double clicking on the empty space in the path field, it will transform into an editable path that can be changed. If the path is changed the ExplorerPanel will try to go to the new path.

Shortcuts

Ctrl+E - Will change the path field to be editable and input focus will be moved to it

Explorer Panel - Navigation

The **Explorer Panel** is the primary panel in MultiCommander. It is from here that you browse the file system and select files to copy, move, rename, delete or view.



All References to Keyboard and Mouse commands below assume that you are using the default Commander-Styled setup, unless otherwise specified.

Commander-Styled-Setup vs WindowsExplorer-Styled Setup

By default MultiCommander uses the **Commander-Styled** setup, meaning that mouse and keyboard shortcuts are similar to the file manager named NortonCommander that existed before Windows on MS-DOS. This setup has become very popular and is used in many other file managers on many platforms. It is popular because it is optimized for keyboard usage and provides the ability to do everything from the keyboard in a efficient way.

If you are more comfortable with the way **Windows Explorer** operates, it is possible to configure MultiCommander to work like Explorer. You will however lose some of the efficiency that you get from the Commander-Styled setup.

[Read more about how to change MultiCommander to the WindowsExplorer-Styled Setup](#)

Navigate the Filesystem

You can use both the keyboard and mouse to navigate in the Explorer Panel.

The files/folders that are currently in focus are highlighted by way of a border drawn around them. These are the files/folders that will be affected by your action if no other files/folders are selected. You can navigate up and down the list with the up and down arrow keys. The *HOME* and *END* keys will change the focus to the first or last item in the list.

If there are more files and folders than will fit on the screen you can use the page up and page down keys to scroll up and down in steps.

To choose a sub folder to browse you either have to **double-click** it or press the **RETURN** key if the folder is already in focus. Explorer Panel will then try to show you the content of that path. You are also able to browse archives if there are file system plugins installed that expose that archive format as a file system. For example .zip and .rar files can be browsed as if they were unpacked into normal folders.

To go back to the parent folder you can either double-click on the ".." icon or press the **BACKSPACE** key. Holding down **CTRL** while pressing **BACKSPACE** will send you to the root of the current volume.

Holding **CTRL** and pressing the **LEFT** or **RIGHT** arrow key can synchronize the current path or folder in focus to the panel on the left or right. For example if you are in the left panel and focus is on a folder and you press the **CTRL+RIGHT ARROW** key, the path for the folder in focus will be sent to the right panel, and if the right panel is an Explorer Panel it will now show that path location. If the **CTRL+LEFT ARROW** key is pressed when the active panel also is the left one, the Explorer Panel on the left will get the current path location of the right panel and show that location in the left panel.

If you hold **CTRL** while pressing the **PAGE DOWN** or **PAGE UP** key you will go to the next or previous sibling folder of the current location. For example If you have some **A to Z** folders in **C:\MyFolders** and you are currently located in **C:\MyFolders\D**, then when you press the **CTRL+PAGE DOWN** keys the Explorer Panel will now go to the **C:\MyFolders\E** folder. If instead **CTRL+PAGE UP** were pressed, it would have gone to the **C:\MyFolders\C** folder. This however requires the sorting in the Explorer Panel to be set to the name column. It will go to whatever previous or next sibling folder that the current sorting determines.

If you have an Explorer Panel both in the source and target panel and want to show the same location in the target as the source panel has, then press the keyboard shortcut key **CTRL+.** (dot) in the source panel. The target and source panel will then show the same location.

Hint

Go To Sibling

A quick way to go to a sibling folder is by way of the shortcut keys

- **Ctrl+Page Up** goes to the *preceding* sibling folder (in sort order)
- **Ctrl+Page Down** goes to the *next* sibling folder.

Sorting

Name	Ext	Size	Date	Attrib
..[.]	<DIR>	2008-09-16 19:03:07	--d	
file1	>DIR<	2008-09-16 19:03:07d	

To sort the files and folders shown, click on the header of the column that you want to sort the contents by. Clicking on it again will reverse the sorting. You will see an arrow in the header of the column that currently has sorting activated, indicating if sorting is *ascending* or *descending*.

Sorting can also be changed using shortcut keys: **CTRL+F1** to sort by first column, **CTRL+F2** for second column and so on. Selecting the column that already is the sort column will reverse the sorting on that column.

If the configuration option **Sort Files Logical** for the Explorer Panel is checked, files and folders will be sorted so that numeric subfields are treated as numbers. E.g. instead of files being shown as

```
file_0011.txt
file_1.txt
file_10.txt
file_2.txt
```

They will be sorted as

```
file_1.txt
file_2.txt
file_10.txt
file_0011.txt
```

Select files and folders

Name	Ext	Size	Date	Attrib
...[.]	<DIR>		2008-09-16 19:03:07	--d
[res]	<DIR>		2008-09-16 19:03:07	--d
BmpView	cpp	1 700	2007-06-10 22:16:58	-a-
BmpView	dsp	5 129	2007-06-10 22:16:58	-a--
BmpView	dsw	539	2007-06-10 22:16:58	-a--
BmpView	rc	13 439	2007-06-10 22:16:58	-a---
BmpView	sln	914	2007-06-10 22:16:58	-a--

There are many ways to select files and folders in the explorer panel, and MultiCommander also remembers what files and folders you had selected if you return to a location where you previously were since the start of MultiCommander. (If MultiCommander is closed this information is lost).

Mouse (Commander-Styled Setup)

Right click on the file or folder that you want to select or deselect. You can also hold down the right mouse button and drag the mouse up and down to change the selection on multiple files and folders. When changing selections on multiple files or folders, the first selection performed determines whether it should select or deselect all the files/folders that are under the mouse cursor until the mouse button is released.

Mouse (Windows Explorer-Styled Setup)

Selection works as it does in Windows Explorer. You Select files/folders with the *left mouse button*, and to select multiple files and folders you hold down the *shift* key to select/unselect a range of items. Holding down *CTRL* will let you toggle the selection of items without deselecting already selected items.

Mouse Selection Rectangle

If the option **Selection Rectangle** is on in settings for the Explorer Panel, you can also drag out a rectangle on the display. All files and folder that are touched by the rectangle will be selected. To use the selection rectangle you must start a drag operation *outside* any file or folder name.

Expanding / Shrinking Selection by Filter

Using the shortcut key **NUM+** or "**Select**" from the Edit Menu, an "Expand Selection" dialog will be shown where you can insert a filter and all matching files and folders of that filter will be *selected*.

Using the shortcut key **NUM-** or "**Unselect**" from the Edit Menu, a "Shrink Selection" dialog will be shown where you can insert a filter and all matching files and folder of that filter will be *deselected*.

Invert Selection

By pressing the shortcut key **NUM*** or selecting "**Invert Selection**" from the Edit Menu, the selection of all the files and folders at the current location will be *inverted*. (Selected files and folders will be deselected, and unselected ones will be selected.)

Select All / Unselect All

Use the shortcut key **CTRL+A** or choose "**Select All**" from the Edit Menu to select *all* files and folders at the current location. Use **CTRL+D** or choose "**Unselect All**" from the Edit Menu to deselect all files and folders.

Quick Select Matching Extension or Filename

Using the shortcut key **CTRL + NUM+** or "Select same extension" from the Edit Menu, all files with the same file extension as the file currently in focus will be selected. Similarly, **CTRL + NUM-** deselects the files.

Using the shortcut key **ALT + NUM+** or "Select same name" from the Edit Menu causes all Files with the same name,

except those having the same file extension as the file currently in focus, to be selected. **ALT + NUM-** similarly deselects matching files.

Toggle Selection from keyboard shortcuts

You can toggle the selection of files and folder with the **INSERT** key; this key will toggle the selection of the current items in focus and then move down to the next item.

Toggling the selecting files and folders can also be performed with the **SPACE** key. If a selection is performed on a folder an action to calculate the folder size is also issued.

Quick Search

Quick Search allows you to start typing a name and the focus will change to files or folders starting with that name. Using the default commander-styled setup you need to hold down **ALT** and **CTRL** key while typing. This can be configured in the settings for the explorer panel.

When the Quick Search window is active you can press the **UP** and **DOWN ARROW** keys to move to the previous or next file or folder that matches the current quick search.

With Windows Explorer-Styled Setup you can just start typing without holding any key down. However, this setup will disable the feature that forwards all key-presses to the command line bar.

In the settings for the Explorer Panel you can configure the quick search window to be activated with **ALT+CTRL+<any letter>** or **ALT+<any letter>** or simply **<any letter>**. If configured to be activated by any letter typed, then the *Forward Keys to the command line* feature will be disabled.

Forward Keys

With the default Commander-Styled Setup, key presses are forwarded to the command line bar. This make it possible for you to start typing a command without having to first manually change input focus to the command line bar.

It is possible to change the input focus to the command line bar with the shortcut key **?** (This key can be different depending on keyboard layout. Press the key to the left of the '1' key on a normal keyboard. Laptop keyboards can be different). This key can be reconfigured in the keyboard customization manager.

Context Menu

The *context menu* is the *right click menu* that you see in Windows Explorer. In MultiCommander this can be shown in different ways and additional actions can be shown in the menu. If MultiCommander is set to Window Explorer mode the context menu is shown using the right click button. However for the default commander-styled setup you toggle the selections of files and folder with the right mouse button. But if you hold the right button down for 2 seconds it will show the context menu. Double clicking with the right button will also bring up the context menu.

Drag and Drop

You can drag and drop selected files using the left mouse button. If dropped files are dropped on the same drive the file operation will default to a *Move Operation* instead of a *Copy Operation*. To avoid this hold down the **SHIFT** key while dropping the files.

Copy / Paste

Copy and paste of files and folders can be performed just as in Windows Explorer, but there is also the function to copy the path of a file to the clipboard so that you can paste that path or filename into another program.

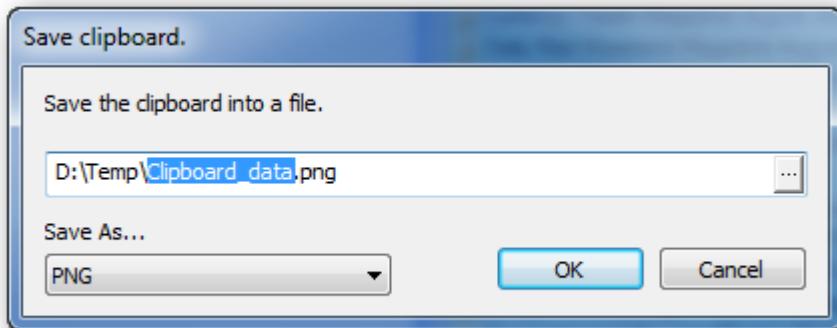
Copy to the Clipboard

CTRL+P will copy the complete path with the name of the selected file or folder to the clipboard.
CTRL+SHIFT+P will copy only the name of the selected files or folders to the clipboard.

SHIFT+ALT+P will copy only the path of the selected files or folders to the clipboard.

Paste from the Clipboard

If you have a path to a file or folder already in the clipboard you can paste that path into the Explorer Panel with **CTRL+V**. It will then show you that path so you can continue browsing that location.



If the data in the clipboard is not a file path, MultiCommander will check if it can identify what type of data it is. If it is an image (e.g. a screen capture) then you will be presented with a dialog to save the content from the clipboard as an image and you can choose what image format you want to save it as (**BMP/JPG/PNG**).

If the data is identified as text you will be presented with a dialog to save the content from the clipboard as a text file and you can choose to save the text content as **ASCII** or **Unicode**.

Keyboard shortcuts

- [Default Keyboard Shortcut \(Commander styled setup\)](#)
- [Default Keyboard Shortcut \(Windows Explorer styled setup\)](#)

Favorites/Bookmarks

There are several ways to have Favorites/Bookmarks in MultiCommander

- User Defined Command in the Button Panel
- Alias for the command line field
- User Command in Menu Bar
- Quick Launch Bar
- Favorites Window
- Quick Paths

User Defined Command in the Button Panel

Using **User Defined Commands** you can easily create a **Button Panel** button to change the current path. First activate *Wizard mode* by pressing the **Wizard Mode (On/Off)** button. (It is the last button to the right on the bottom row) or from the *configuration menu* by selecting **Toggle button panel wizard mode**

When **Wizard Mode** is active the button panel will have a red background around the buttons. You can now **drag and drop** a file or folder to any button and the *Button Panel Wizard* dialog will be shown. Click on **Go to file / path in Multi Commander** in the dialog and then press **OK**. You now have a button that when pressed will take you to that file or location.

Alias for the Command Line Field

An *alias* is a shortcut you type in the Multi Commander command line field that will expand into one or more longer commands. If you enter a path location in the Multi Commander command line field and then press return, Multi Commander will go to that location. So by creating an *alias* for a *path* you can go to that path by just typing the alias you assigned it. For people who like to be able to go to different paths quickly using the keyboard this is really useful.

You create an alias in the **Alias Editor** that you find in the **configuration** menu. But there are also two quick commands, **Alias** and **Alias2**, that can create an alias directly from the command line field. For example, if you wish to create a command line shortcut that takes you to your folder "D:\my music\mp3\" you can type **alias mp3="d:\my music\mp3"**. (The quotes are only needed if there are spaces in the the path. Now when you type **mp3** in the command line field and press return, Multi Commander will expand the "mp3" alias to the full path and execute that. Since it is a path it will go to that location.

But there is an even easier way to create this alias, and that is to use the **alias2** command. This command only takes the alias name as a parameter and assumes that the current path will be the target for that name. So if you are already standing in the *D:\my music\mp3* path and in the command line field you type **alias2 mp3**, you will create an alias named **mp3** that will take you to the path **D:\my music\mp3**.

[Read more about aliases](#)

User Defined Commands in the Menu Bar

The custom command for changing the path is MC.Explorer.GoTo. If you want a command that takes you to the MP3 folder we used in the alias example above you write

```
MC.Explorer.GoTo PATH="D:\My Music\MP3"
```

You can also specify LEFT/RIGHT (instead of PATH) if you want to control in which panel side the path should be opened.

After you have written your command and saved it, you can press the Menu button. From here you can add the newly created command to the menu bar.

[Read more about User Defined Commands](#)

Quick Launch Bar

You can Drag and Drop files and folders to the QuickLaunchBar. When a folder that is placed on the QuickLaunchBar is clicked, It will change the current panel to that path. You can also place a User Defined Command in the QuickLaunchBar by right-clicking on it and selecting *Insert User Defined Command*.

[Read more about QuickLaunch bar.](#)

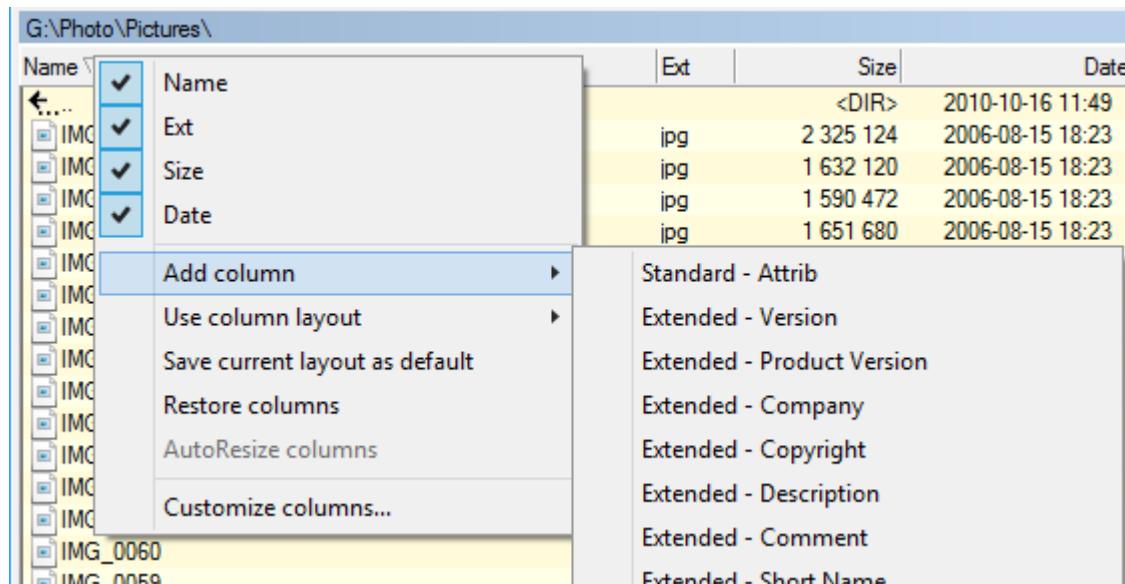
Explorer Panel - Customize Columns

Which columns that should be shown can be customized. It is possible to remove existing columns or add new one. There are many columns that will show a lot of extra information to choose from.

It is also possible to create new column layouts and then save that layout and have it switch to that layout automatically when browsing to some locations. Or have it switch the entry layout when pressing a hotkey or selecting a layout manually

Customize Columns

Right click on the column header to show the context menu that lets you customize the columns.



All existing column are shown in the popup menu first. To remove any of the existing column just click on that item in the menu and it will be removed.

Add Columns

Add columns by selecting what column to add from the list of columns under "**Add columns**". This list shows the most common columns, But there are a lot more columns available. To get to them select the "**More...**" item at the bottom on the column list. Then the "**Customize Columns..**" dialog will be shown. From this dialog all columns are available.

Use Layout

If you have created multiple column layouts. These layouts are listed there. Selecting one of them and the current column layout will change.

By default there are three layout available. "**Default**", "**Picture**" and "**Music**".

The "**Default**" layout is the layout that is used by default. If you want to modify the standard column layout it is the "**Default**" layout that should be changed.

The "**Picture**" and "**Music**" layout are alternative layout that will show extended file information for picture and music files when selected.

Save Current Layout as Default

This will save the current column layout and set it to the "**Default**" layout.

Select this option if columns are modified and the current layout (columns and size) should be used as the default layout for all Explorer Panels.

Restore Columns

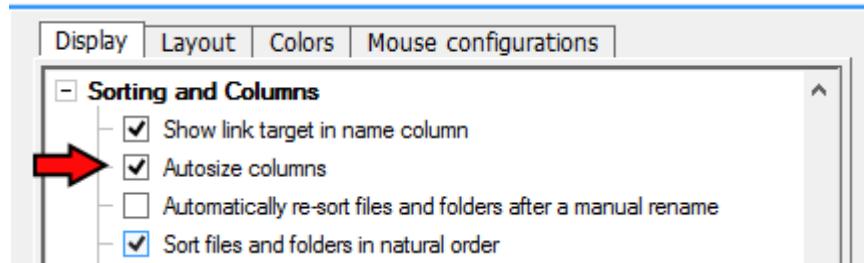
This restores the column to the saved state of the current column layout.

If "**Default**" layout is selected and columns are then modified. By selecting "**Restore Columns**" the column layout will now revert to what is saved in the "**Default**" layout

Is the same for any layout. If "Picture" layout is used and columns are manually changed and then "Restore Columns" is selected the column layout is reverted to the saved state of the "Picture" layout.

Autosize Columns

If the ExplorerPanel settings "Autosize columns" option is enabled this option will be **not** available.



If the option in the picture above is not enabled. Then you can use the "Autosize Columns" option in the column header menu to manually request for the columns to be resized.

Customize Columns

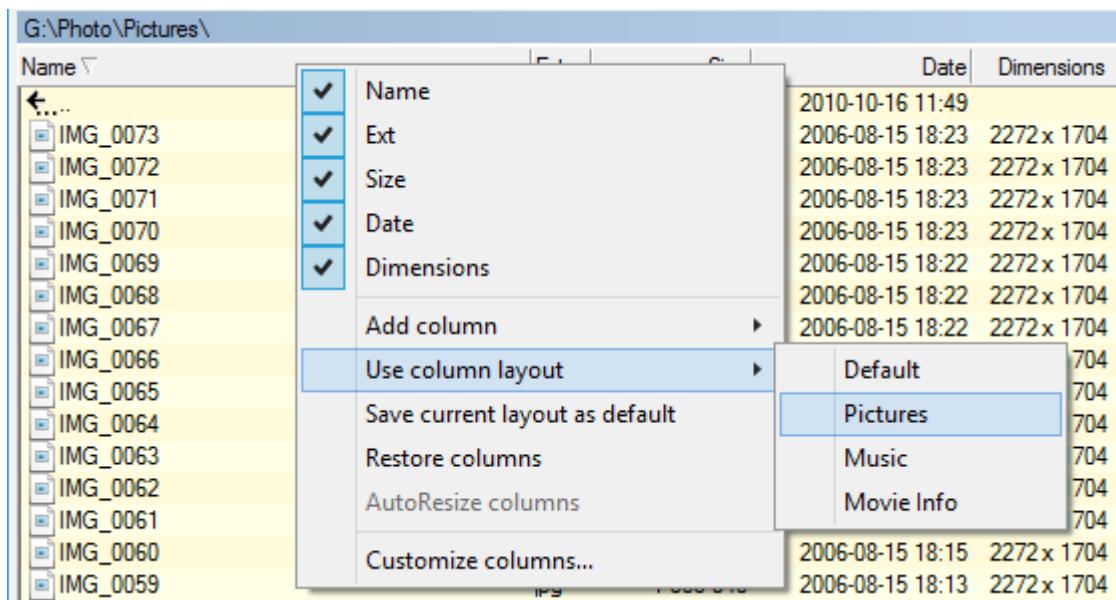
Selecting this will open up the Customize Column Layouts window.

There new layout can be created, existing once can be modified. It is also here where all the columns that does not fit in the Add Columns list are available.

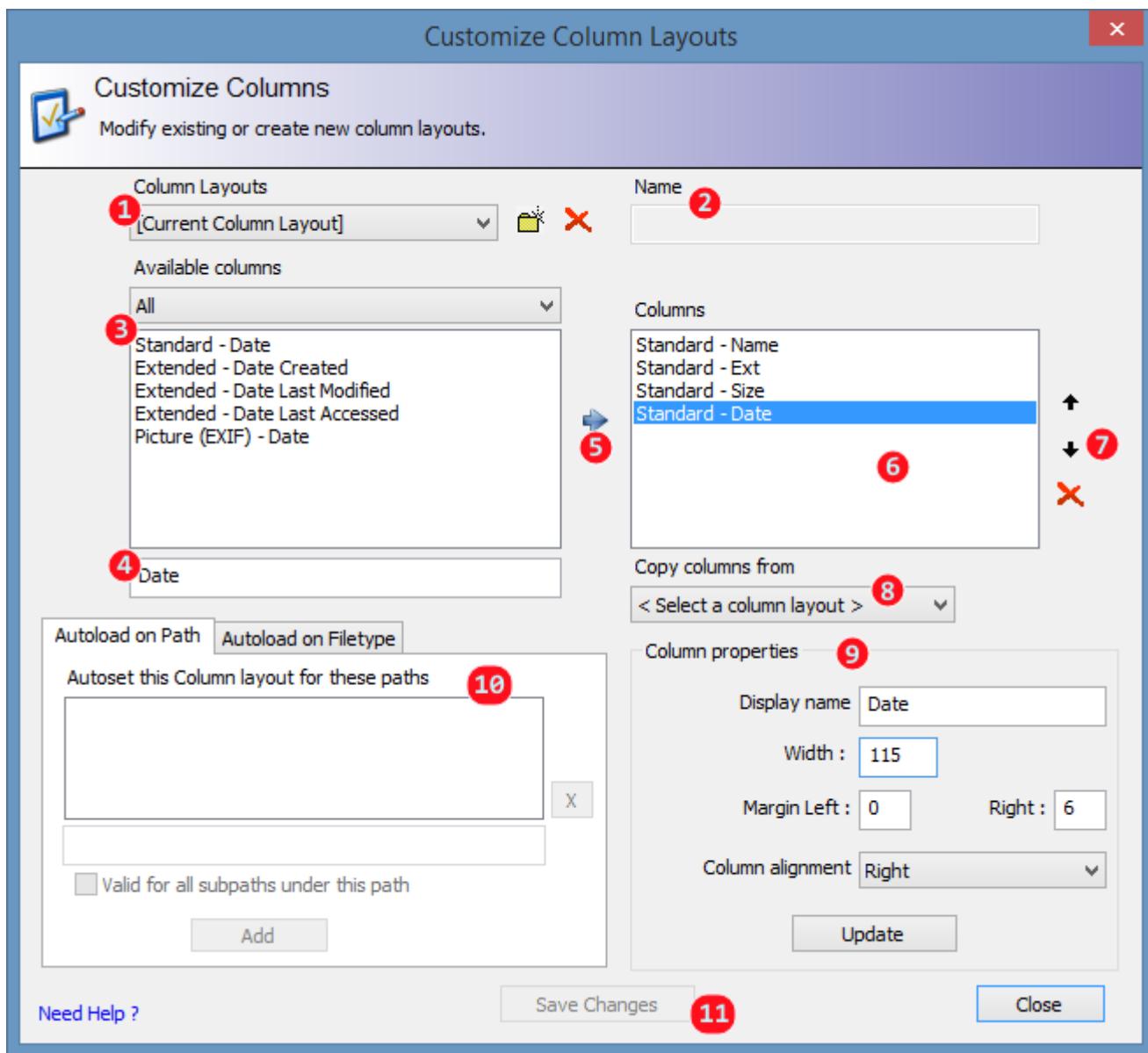
See [Customize Column Layouts](#) for more information

Explorer Panel - Customize Column Layouts

A **Column Layout** is a saved set of a column layouts that can be selected from the column header menu. A Column layout can also be set automatically if you go to a folder that was specified in the Column layout definition.



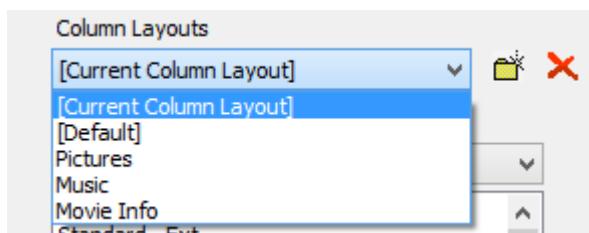
From the "Customize columns..." option you will get a dialog where you can create or modify and remove Column Layouts.



1. [Column Layouts](#)
2. [Column Layout Name](#)
3. [Available Columns](#)
4. [Column Filter](#)
5. [Add Column](#)
6. [Column List](#)
7. [Modify Column Order](#)
8. [Copy Column Layout From...](#)
9. [Column Properties](#)
10. [Autoload on Path](#)
11. [Save Changes](#)

1. Column Layouts

All available column layouts are listed in a dropdown list. After the list there are two buttons. The "Create New" and "Delete". Use them to create new layout or to delete existing.



The first item in the column layout list named "**[Current Column Layout]**" is special.
It is representing your current layout and modifying and saving that will only change the look for the current Explorer Panel. Not all of them and changes are not saved between restarts.

To change the standard column layout that all Explorer Panel tabs are using the **[Default]** column layout need to be modified. Changing this all Explorer Panel tabs will be update it look when closing the windows.

2. Column Layout Name

When creating a new column layout enter the name you want to identify that column layout as here. This name will be listed in the Use Column Layout submenu when right clicking on the column header.

3. Available Columns

This is the list of all the column available to choose from. From the dropdown list above the list it is possible to limit what column to list by type.

4. Column Filter

Since there are a lot of column it can be hard to find the column to add. In the filter field it is possible to filter the list of available columns.

Start typing and the available columns list will only show columns that name include what have been typed

5. Add Column

Press the Arrow button add the selected column to the list of column for the currently selected column layout.

6. Column List

This list shows all the columns that have been assign to the currently selected column layout.
The order of the column in the list is the order they will be shown in the Explorer Panel.

7. Modify Column Order

The up and Down arrows here allow column top be moved up and down the list. Select the column to move and then press the up/down button.

Press the X button to remove a column from the list.

8. Copy Column Layout From...

Often when creating a new column layout it is similar to another existing layout. This dropdown list will list all available column and selecting one of them the layout of that column layout is copied into the current column layout that are being modified.

9. Column Properties

All column can have some standard properties. These properties can be modified here.

Changing the Display name the column header name will now use this name instead.

Width is the default width of the column. However if autoresize columns settings is enabled this width might be ignored if needed.

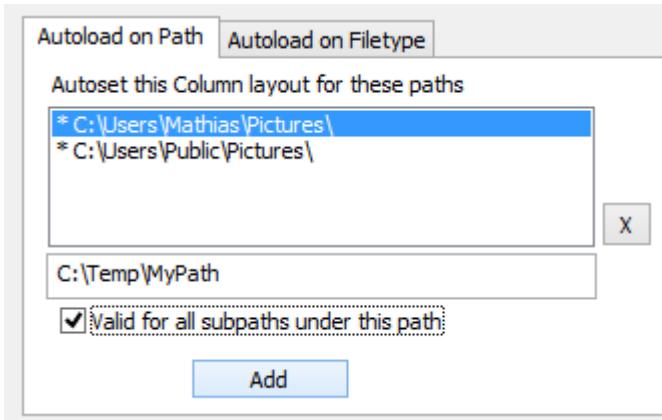
In Margin left/right it is possible to add extra blank space in the column to the left/right side.

In Column alignment the alignment of the content of the column can be changed. It can be right / left / center.

When any properties are changed then press the "Update" button so the properties are updated

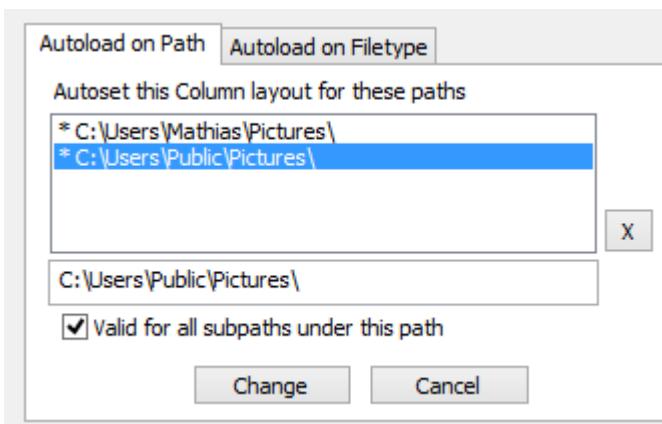
10. Autoload on Path

This allows for columns layout to automatically be used under path that you specify here



Enter the path that where the column layout should be autoloaded, Check the "Valid for subpaths under this path" if you want the Column layout also should be valid for subpath under this path. Then press the "Add" button.
Path in the list with a * in front of them has the subpath option checked on them

To modify a path in the list, double click on it.



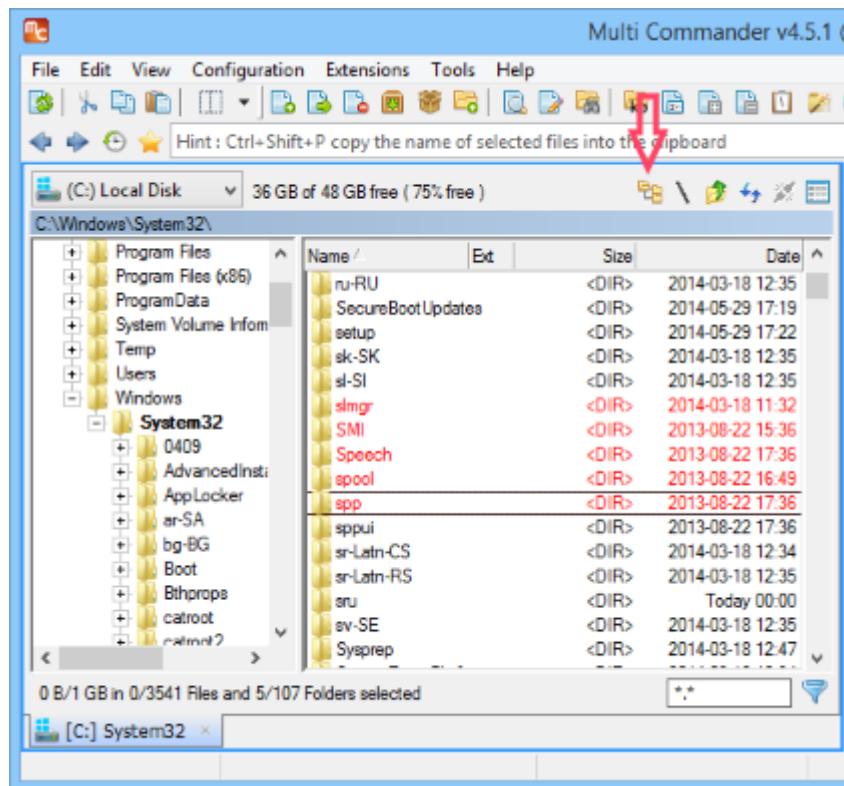
Change the path, or subpath settings and then click "**Change**"

Select a path in the list and press the X button to remove it

11. Save Changes

When all modification are done for the selected column layout press the "**Save Changes**" button. If this button is not press all change done is lost.

Folder tree



The first of the **Explorer Panel** command button Toolbar row is the **Toggle Folder Tree** command, This button will show/hide the Folder Tree.

Shortcuts

ALT+Num* - Show/Hide the folder tree panel

ALT+Num/ - Change the input focus between the folder tree panel and the file list view

The folder tree can be scrolled independently of the Explorer panel, and clicking on any node of the folder tree will change the Explorer panel to display that subfolder. The nodes are shown with "+" or "-" symbols to show whether the subtree is expanded or collapsed.

You can also drop files on a folder in the folder tree and then a copy operation. The copy operation will use the folder that is dropped on as the suggested target folder.

Pressing the refresh shortcut command (F2/F5 depending on setup) while a folder in the folder tree is in focus then that tree node will refresh.

NOTE

If you have "**Autosize columns**" set, the Folder Tree will obscure part of the file listing in the Explorer Panel. To turn this setting off, click on **Configuration > Explorer Panel Settings**, scroll down to "**Sorting and Columns**", and click on "**Autosize columns**" to clear the checkbox.

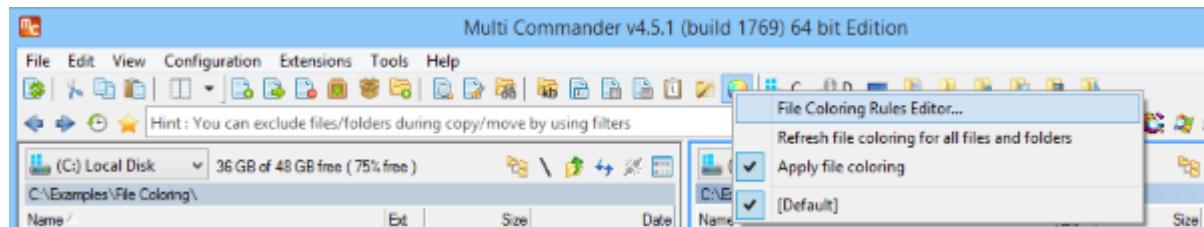
Rule Based File Coloring

With Rule Based File Coloring it is possible to create rules that will define what color a file or folder should be shown as.

It is also possible to have multiple color profiles that have their own setup of different rules and it is then possible to switch to a different profile from the toolbar button.

You can also create user commands that will switch to a specific profile and this command can then be assigned to a hotkey.

Where to Create the Rules

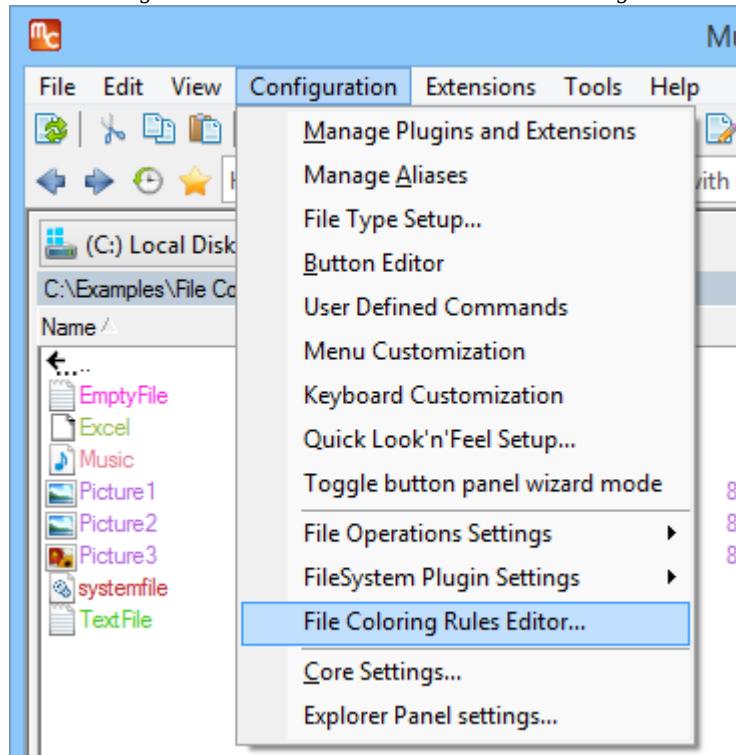


File Coloring Rules Editor is found by clicking on the ColorWheel icon on the toolbar.

There is also options to enable/disable the file coloring and what profiles are available. By default there is only a "[Default]" profile. But if you create own profiles they will also be listed here.

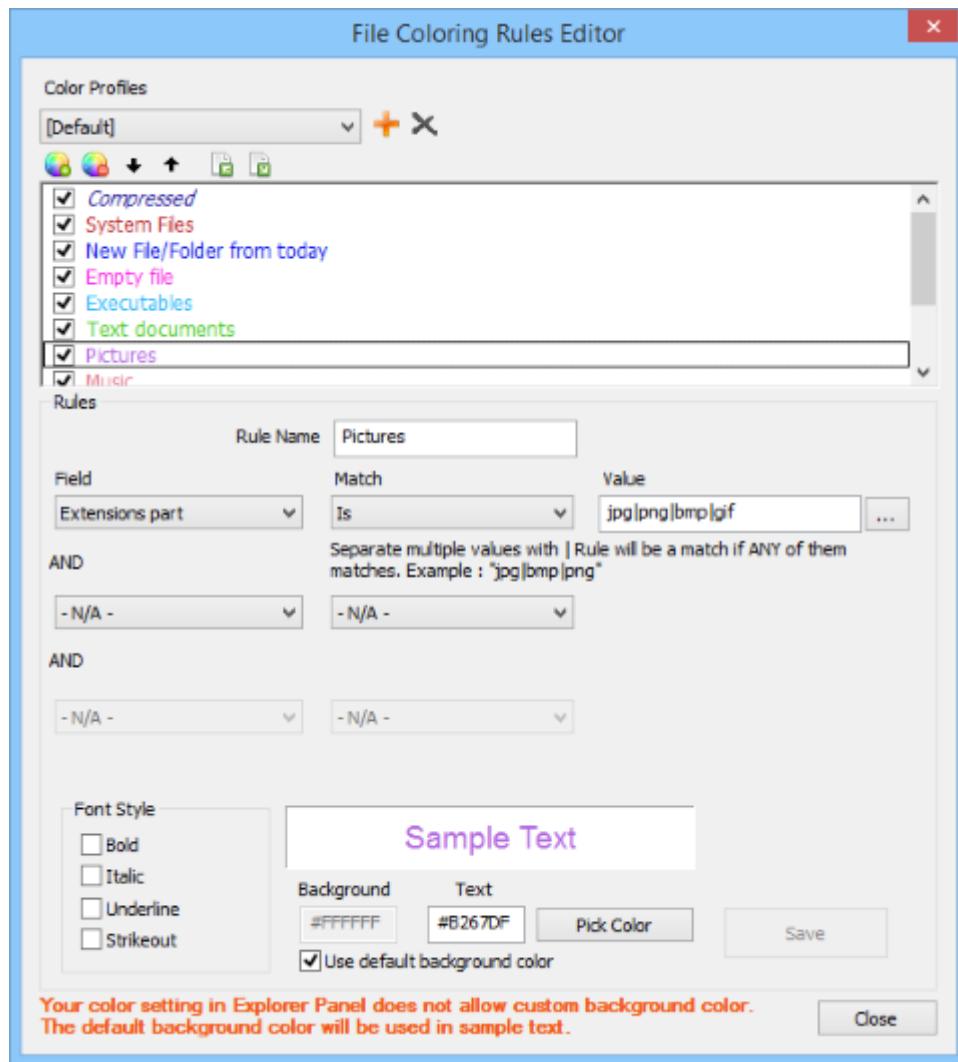
Clicking on "**Refresh file coloring for all files and folders**" Will remove all coloring information and reapply it for all files and folder.

File Coloring Rules Editor is also found under the configuration menu.



Rules Editor

In the File Coloring Rules Editor, It is possible to add/remove or modify existing rules. Entire new Color Profiles can also be created.



The list with rules will also work as a preview and the rule will be listed in the color that it will set if that rule is true.

Files and Folders are colored by the rules from top to bottom. And the first matching rule will be the one that decides the color.

If you have multiple rule that can get in conflict with each other then you can use the up and down button above the list to move rules up and down the list.

Every rule can have 3 criteria that must be true for the color to be set. Only if all parts of the rule are true are the color that is specified set.

The checkboxes in the list of rules shows if a rule is enabled or not. If you uncheck a rule then that rule will not be used when coloring files and folders.

You can also export rules into a file that you then later can import. When exporting you are ask if you want to export all the rules or only the active rules. If you disabled some rules then they will not be exported if you choose to export only active rules.

Example

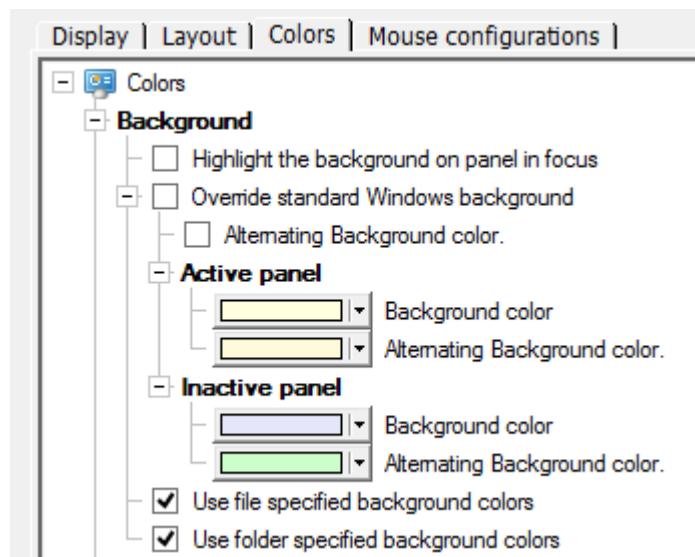
Example how the default color profile can color files and folders.



Coloring of Background

For the file coloring to be able to change the background color the configuration option "Use file/folder specified background colors" must be enabled.

(See the two checked items in the image below)



User Defined Commands

What file coloring profile to use can be controlled by using user commands.

Create a user defined command of "CustomCommand" and type use the MC.Explorer.SetColoringRules command.

Example

```
MC.Explorer.SetColoringRules Name="MyColorProfile"
```

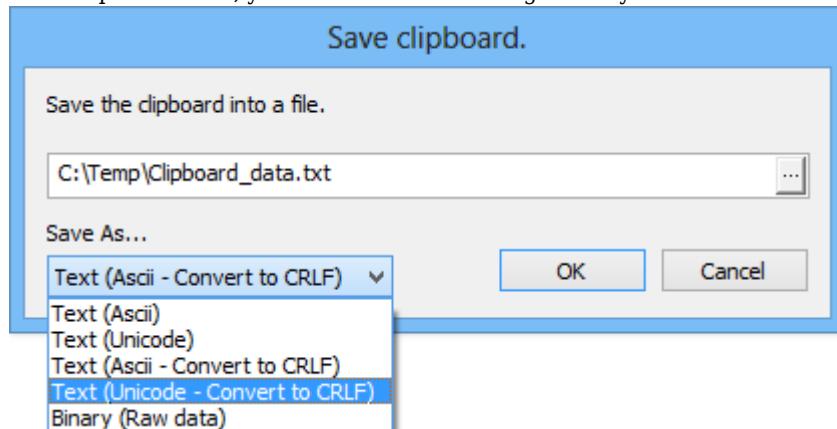
After the command is created it ispossible to assign a hotkey to that command.

Paste from Clipboard into Explorer Panel

If the data in the clipboard is text or picture data and you try to paste that into the Explorer Panel you will be asked to save the clipboard data as a text or image file.

Saving Text Data

If the data in your clipboard is *text* and you press Ctrl+V (or Menu > Edit > Paste) while the focus is in Multi Commander is in an Explorer Panel, you will be shown a dialog where you can save the text content from the clipboard to a file.



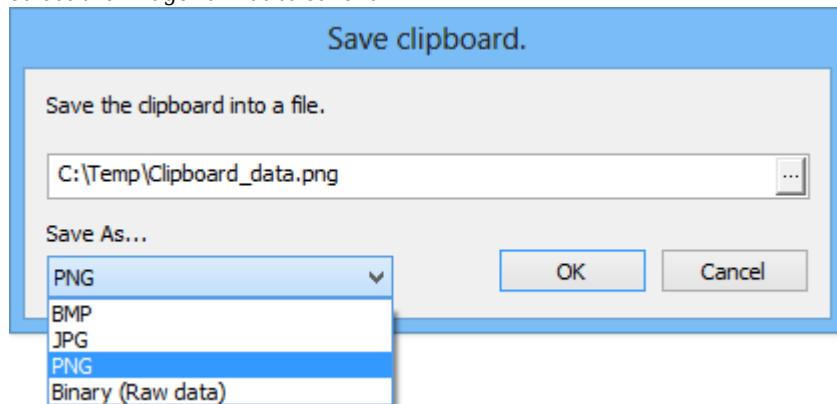
The dialog lets you choose how to store the text

- Ascii
- Unicode
- Ascii - convert to CRLF
- Unicode - convert to CRLF
- Binary (Raw data)

There is also some special handling done for text data. If the text data in your clipboard looks like a *path* for an existing folder or file, then instead of opening a dialog, MultiCommander will immediately change the current location to that path you have in your clipboard data. If the path in the clipboard has a file specified then the focus will be set to that file.

Saving Image Data

If the clipboard data is an *image* (for example if you pressed print screen to take a screenshot), you will have the option to select the image format to save it in



You can save the image in the following formats

- jpg
- png
- bmp

File Search

Multi Commander provides facilities to search for files and folders, along with a number of options you can use to control the search.

To perform a search, go to the path where you want the search to start and then press hot key for starting search or or select "File Search" in the Extensions menu or Find Files from the File menu.

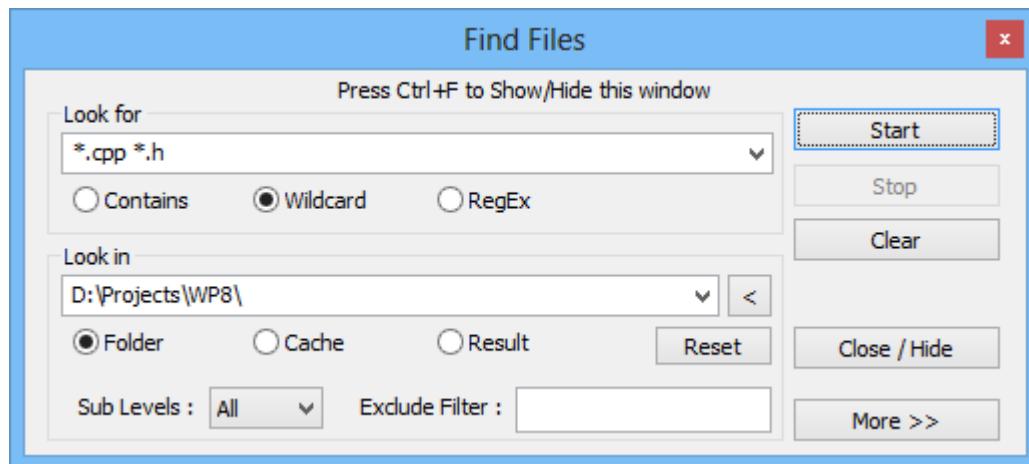
Shortcuts

Start an instance of the Find Files (Search) Panel.

Alt+F7 - For Commander Styled Setup

F3 - For Explorer Styled Setup

The search dialog will open in a new tab in the opposite panel from the one in which you are positioned.

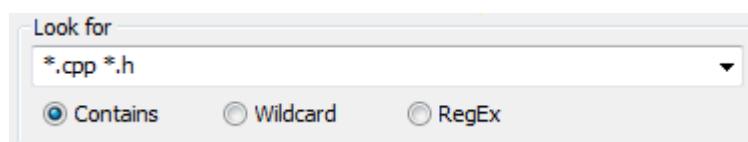


Specify what to look for and how you want to look for it.

You can specify multiple search terms by separating them with spaces.

(If you need to include a space in the search target, put quotes around it. e.g. "Images 23.jpg" .)

You also need to specify how to look for matches. There are three methods provided:



Contains means that files and folder names must *contain* any of the search terms you specify in order to match.

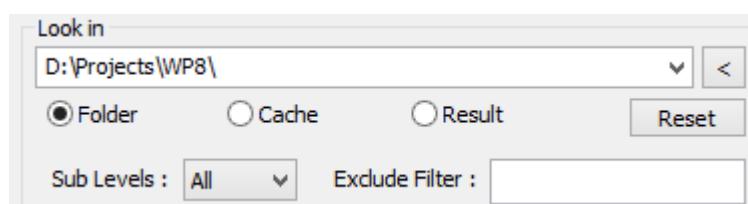
Wildcard means that the search terms that are specified contain *wildcard* characters, e.g. "*jpg" to find all files ending with ".jpg".

RegEx means that the specified search terms are *regular expressions*.

Using wildcard search you can also specify a exclude filter by including a - in front of the filter.

For example if you search for "*.cpp -base*"

This will find file named AlphaClass.cpp but it will not match BaseClass.cpp



Next you need to tell it where it should **Look In** when starting to search for matches. Normally this path is already set to the path that the file browser showed when you opened the search dialog. You can press the **Reset button** to reset the **Look In** path to the path that is currently shown in the opposite view.

(You can specify multiple location to search in by separating them with ; for example "D:\Project;F:\Downloads;C:\Temp")

You can also press the "<" button to add all local harddrive, or paths from open tabs.

You can also specify how many **Sub Levels** down in the file hierarchy it should search for files. By default it searches all the way down.

It is possible to exclude folders from the search in the **Exclude Filter**. There you can enter a space separated list of folder names that you do not want to look in when searching.

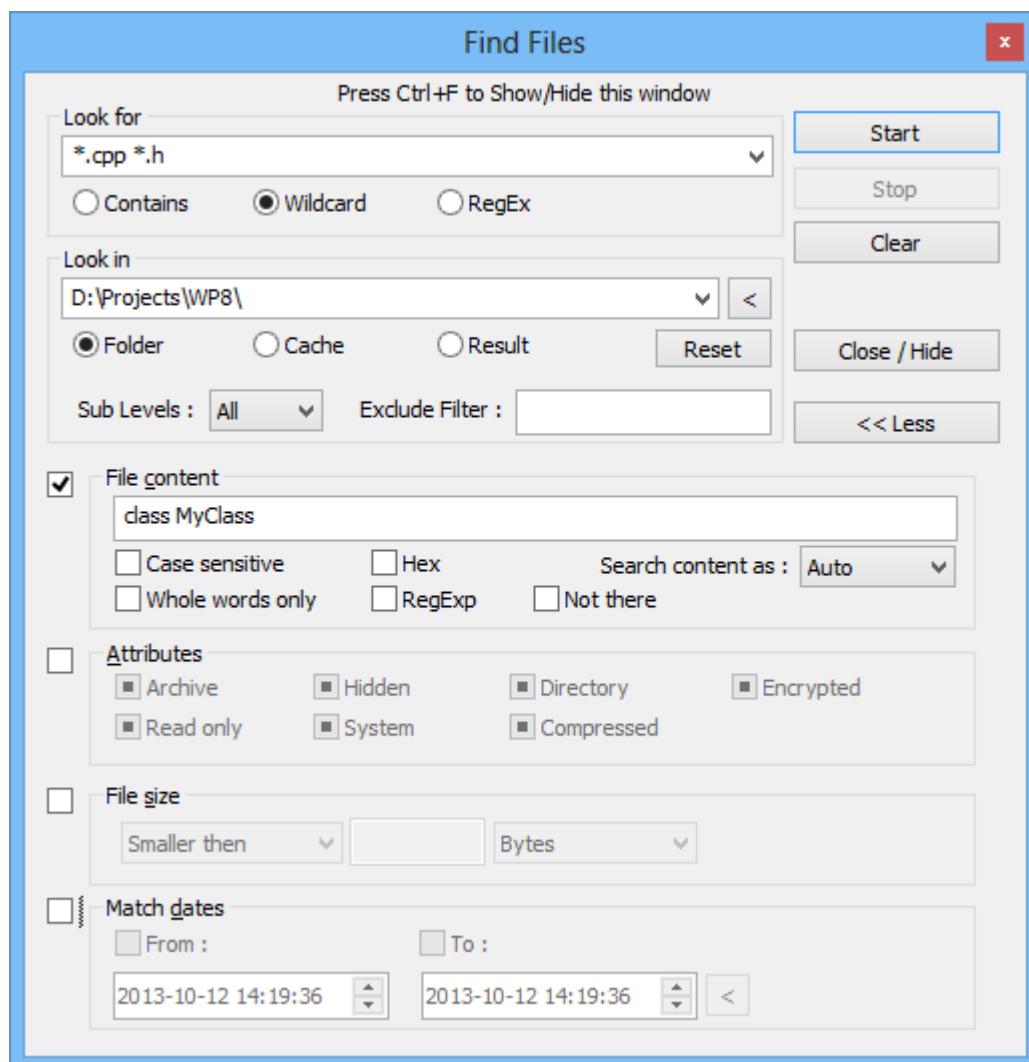
(Example if you enter ".svn tmp data" then it will not scan folder named that during search)

If you previously had done a search you can specify that it should look in the **Result** when doing this search.

This way you are doing a search on the previous search result. This is be a good way to filter out files if you get too many results.

More Options

By pressing the "More >>" button you can specify more advanced search options.



If you want to find files with a specific **File Content** you can include that in your search and you can specify how it should try to find the content. (Look at the screenshot for options)

It is also possible to match files on **Attribute** and **Size** and **Date**.

Date Match Help

Date matching has some predefined date ranges that it can set. By pressing the < Button in the Match dates you get a list of date ranges. If one of them is select the From and To date field is automatically change to match

Load / Save Presets

The Find Files window can load and save presets of already configured searches using the function keys. (F1 to F10).

To save a search to a preset press Ctrl+Function Key (e.g. Ctrl+F1).

To load a preset press a function key that has a preset saved to it, e.g. F1.

Hot keys in the Find Files Window

- Ctrl+F - Show / Hide the Find Files window.
- Alt+Z - Clear all search fields.
- Ctrl+X - Expand / Shrink the Find Files Dialog
- F1-F12 - Load Preset
- Ctrl+F1-F12 - Save Preset

Search Result

If you right click on items in the result you will get a list with options. Here you filter out files from the result based on path. This can be useful if the result you got was very large.

If you double click on an file in the result the opposite panel will go to that file and set that file as focus. You can also press View and Edit files directly from the search result panel. And it is also possible to copy files from the search result.

Scripting

File Search can also be scripted by create your own user defined command that will start a search and by assign a hotkey to that command you can then start search you do often by a simple press of a hotkey.

Search is controlled by CustomCommand script. [More information about the search script command is found here](#)

Scripting example

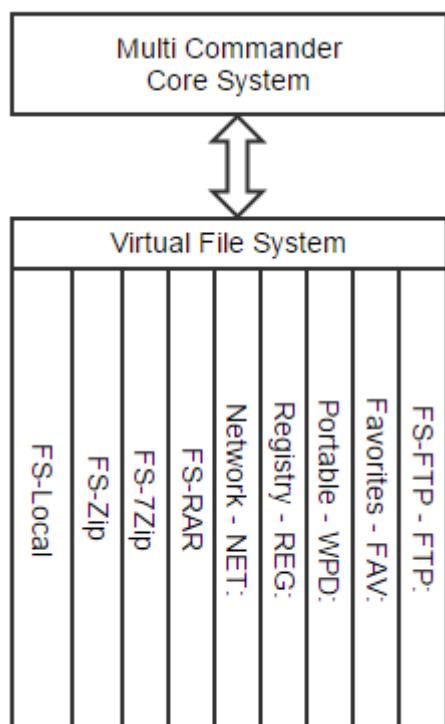
```
MC.FileSearch.Search SEARCHIN="${sourcepath}" SEARCHFOR="*.xml *.txt" AUTOSTART
```

Virtual FileSystem

Multi Commander is using a virtual file system layer.

The virtual file system then have different plug-ins that will extend the file system that Multi Commander shows. It can be plug-ins like FS-Zip that will make zip archives part of the virtual file system, Allowing zip files to be browsed and handled just like they are a folder. Plug-ins can also be like the FS-REG plug-in that will add a new file system prefix, like REG: that will allow the Windows Registry that is not even a filesystem to be browsed and handled like it is.

Not all file system plug-in support all file system function. It all depends on what the plug-in can support.



There are two types of file system plugins. Devices or Containers.

The devices file system plug-in will add new device prefixes to the virtual file system like FTP: , REG: , NET:
This will make things that are not normally a file system be presented in the file system, Like the Windows Registry

Containers is files that contains a file system and has a structure that can be represented like a file system.

Most common type of them are archives like Zip or RAR or 7-Zip. The Plug-in for them will handle the file as they are folders if opened inside Multi Commander. Handling archive like they are folder allow file to be inserted into the archive by copy file into the virtual folder. When viewing a file inside a archive that file will automatically first be copied to a temporary location before open.

FS-Local

The built in file system plug-in **FS-Local** is special. It has handled in a special way inside Multi Commander.

FS-Local handles all the normal file system on the machine, like all single letter device (A-Z) like C:,D:,E: and network file paths like \\Server\share. It is also the only file system plugin that can't be disabled.

Container/Archive Plug-ins

Included with Multi Commander is a couple of archive file system plugin that will handle different archives.

FS-Zip for Zip archive. Zip archives can be read/modify/write/create.

FS-RAR for rar archives, Only Read mode is supported. RAR archives can't be modified or created.

FS-7Zip handles the 7Zip archives. Read/Write/Create is supported.

FS-TarGZ handled both Tar and GZ archives.Read/Write/Create is supported.

FS-BZip2 handles BZip2 archives. Read/Write/Create is supported.

See more about [archive plug-ins](#)

Device Plug-ins

Included with Multi Commander is a couple of device plugin

FS-Favorites (FAV:) Allows the favorites in Multi Commander to be browsed like a file system.

FS-Portable (WPD:) exposes the Windows Portable Device interface that gives access to portable devices connected to the machine.

FS-Registry (REG:) make it possible to access the Windows Registry

FS-FTP (FTP:) adds FTP support

Configuration

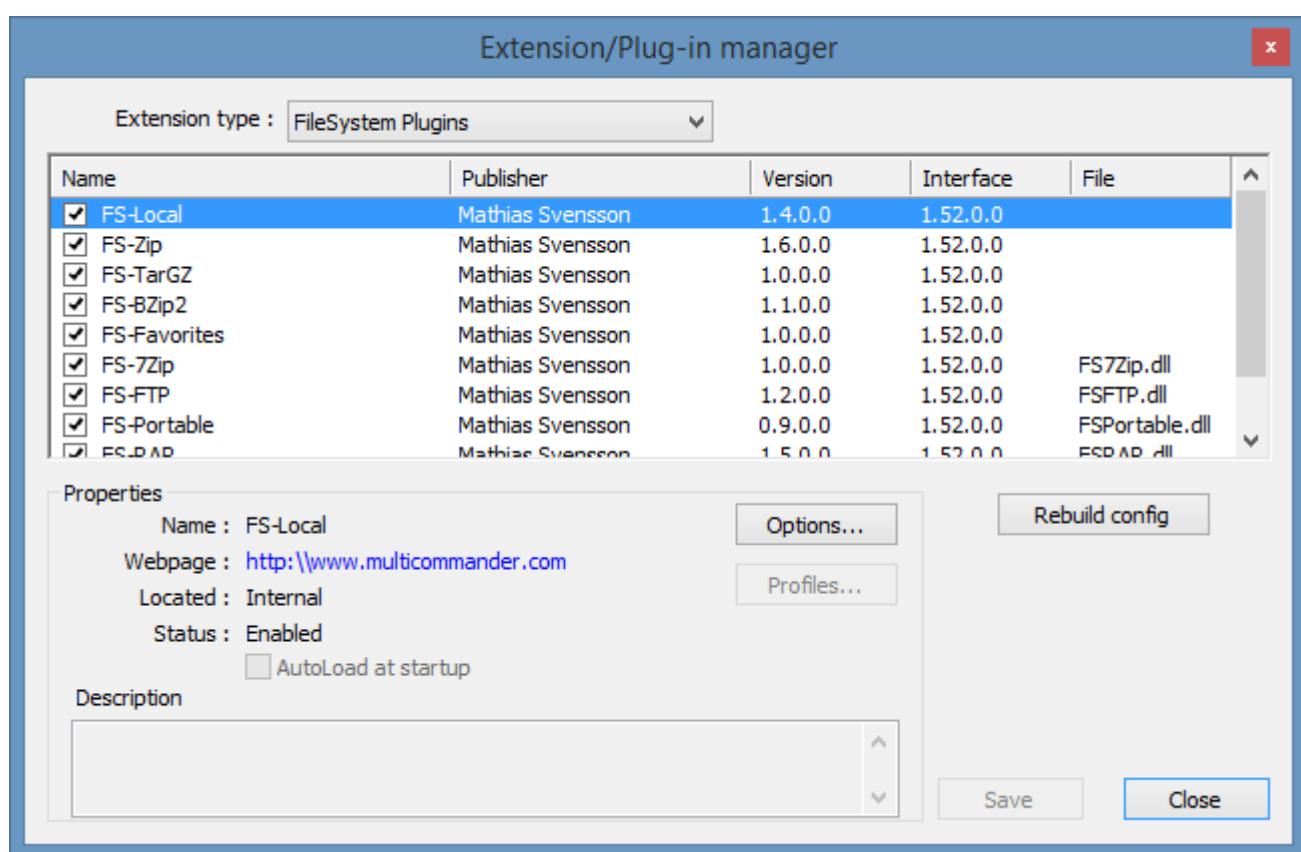
How the plug-in is connected into the virtual filesystem is configured in "Manage Plugins and Extensions".

There you can enable/disable plug-ins and configure that file extension and/or bytemask and device prefix a file system plug-in should be connected with

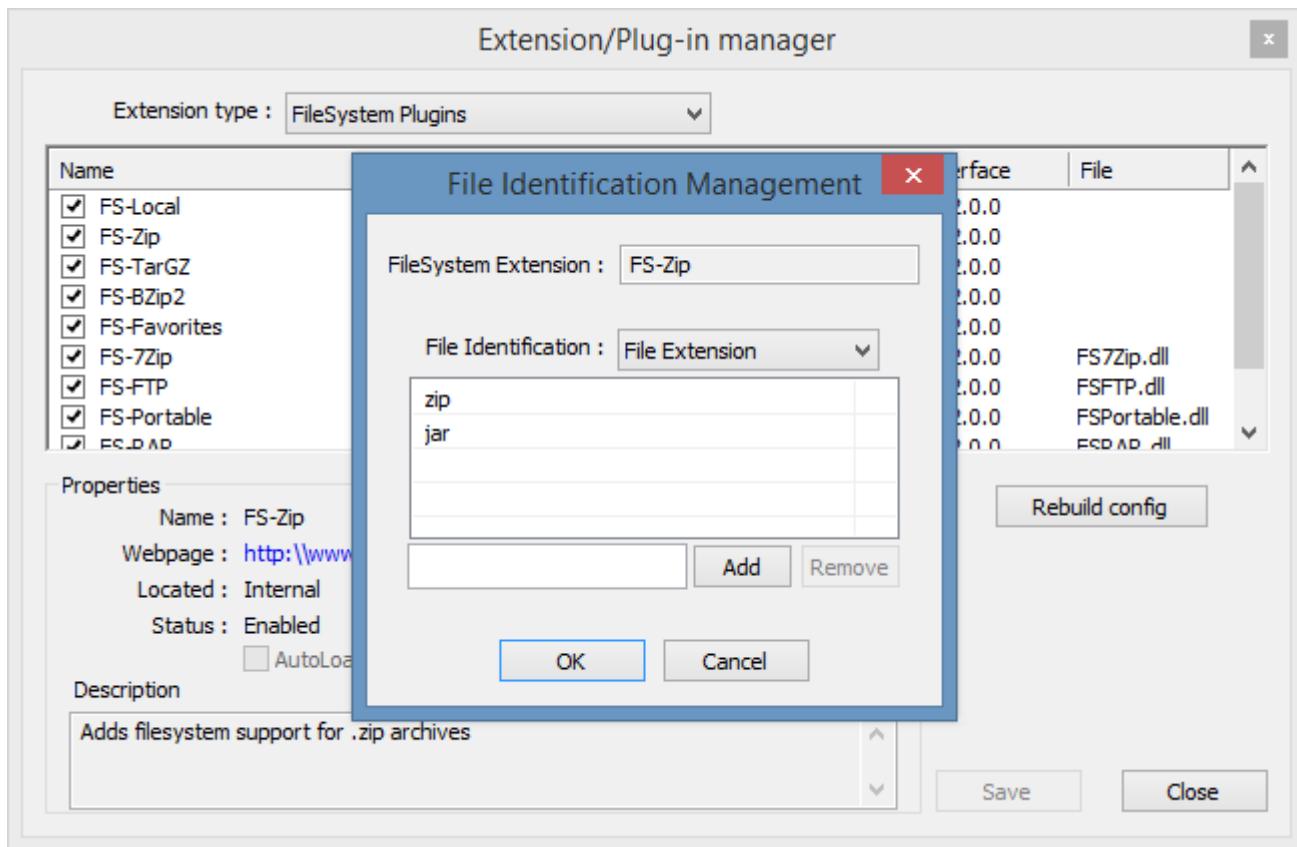
Changing how plugin are connected into the virtual file system are for **experts only**. You are able to configure the plug-ins in a way that will make Multi Commander act in a way you may not want.

Press the "Rebuild Config" will reset all extension/plugin setup

The "Manage Plugins and Extensions" window is found under Menu - Configuration and then select "FileSystem Plugins" in the Extension Type dropdown list



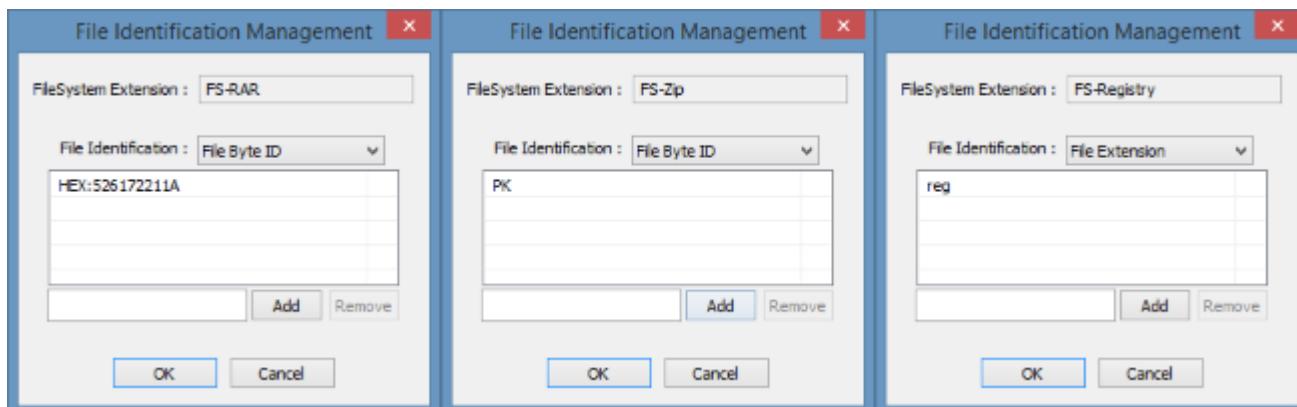
Press the Checkbox to the left to disable/enable a plugin. and to configure how it connects with the virtual file system press the **Options** button.



In the image above FS-Zip was select and it is connected to file extensions zip and jar

To remove a connection to a file extension select the extension to remove and press the remove button.

Adding a new file extension is done by adding the file extension (with out the dot) in the input field to the right of the add button, Then press the add button to add it.



In the "File Identification" dropdown list you can also insert a Byte Mark. The byte mark is also used when trying to identify if a file that should be opened by a container/archive plug-in.

The ByteMark is special bytes that the file always starts with.

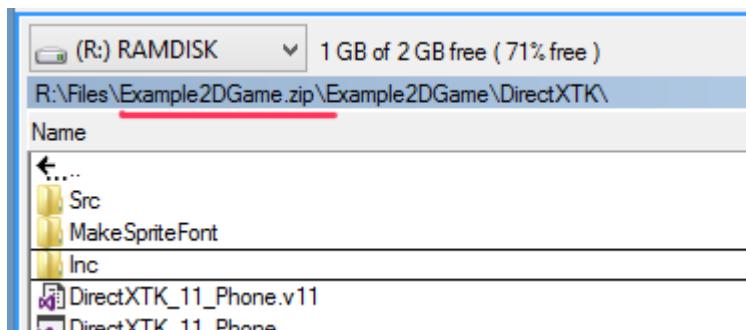
The Byte mark can be entered as a hex value (See FS-RAR in the iamge above) or as ASCII (See FS-Zip in image above). Enter the ByteMark in the input field and press the Add Button. Select a byte mark in the list and press the Remove button to remove.

Device prefixes for Device plug-ins are configured under the "File Extensions". Add the prefix without the ":" (See "reg" device for FS-Registry in image above)

FileSystem - Archives

Some of the virtual filesystem plug-ins will allow archive files like Zip, RAR, or 7Zip to be opened and handled by Multi Commander.

Multi Commander see the archive first as a normal file, but when opening it, it will open it as if it is a folder and continue to browse the content of the archive as it is a folder.



Since archives are handled as folder it is possible to browse into an archive and then select files and then press copy or drag them and only the selected files will be unpacked and copied.

If the virtual filesystem support writing it is also possible to copy files into the archive just as the archive was a folder. The archive will then be updated with the new files.

Delete and Rename are also support if the plug-in for that archive format support it. Not all archive plug-ins support write/modify operations.

Most operation that can be preformed on normal files can be performed on files inside an archive. The virtual filesystem will unpack the file to a temporary location if needed.

If viewing / editing a file inside an archive and the virtual filesystem layer will automatically extract and update the archive if needed.

Inside Multi Commander in many places it is possible to reference a files that is located inside an archive and the virtual file system will then automatically open the file inside the archive.

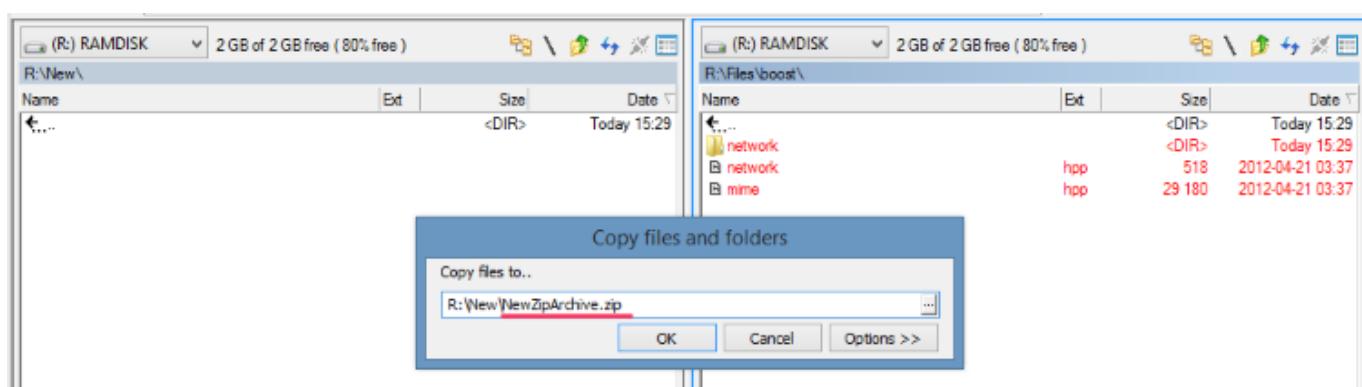
Pack / Unpack

With [Pack](#) and [Unpack](#) it is possible to quickly unpack files or to pack files.

Unpacking can also be done by browsing into the archive and selecting what to copy and then copy the content to the target location.

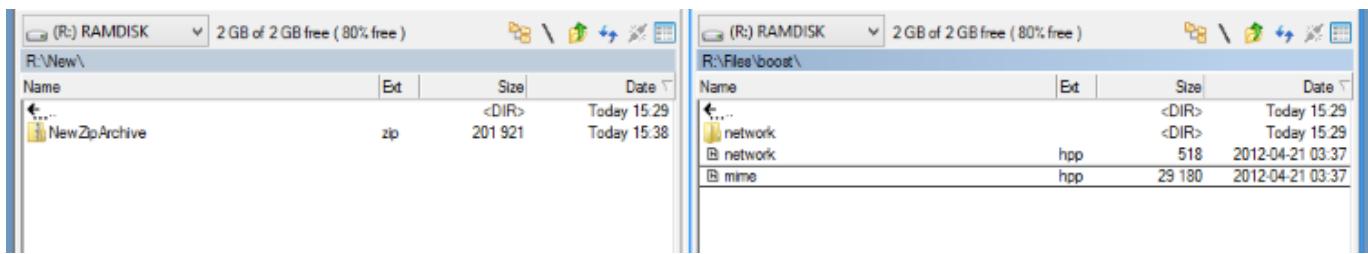
Create New Archive Automatically

Because the virtual filesystem will automatically open or create archives it is possible to create a new archive by copy files to an archive that does **not** exists. If it exists the existing archive will be updated instead.



In this image above the right panel have the folder "**Network**", and the file "**Network.hpp**" and "**mine.hpp**" selected. A copy operation of the selected files to the target panel to the left is then issued. In the CopyTo confirmation window a archive file name (**NewZipArchive.zip**) is added to the target path.

The copy operation will now copy the selected file into the new zip archive named NewZipArchive.zip



After the copy is completed there is now a new zip archive in the target location. "NewZipArchive.zip"

The Virtual Filesystem knows that the .zip extension is a virtual file system. Therefore when the target location is path that ends in a archive the virtual file system layer will then created that archive file if it can. If the file already exists the existing file will be updated.

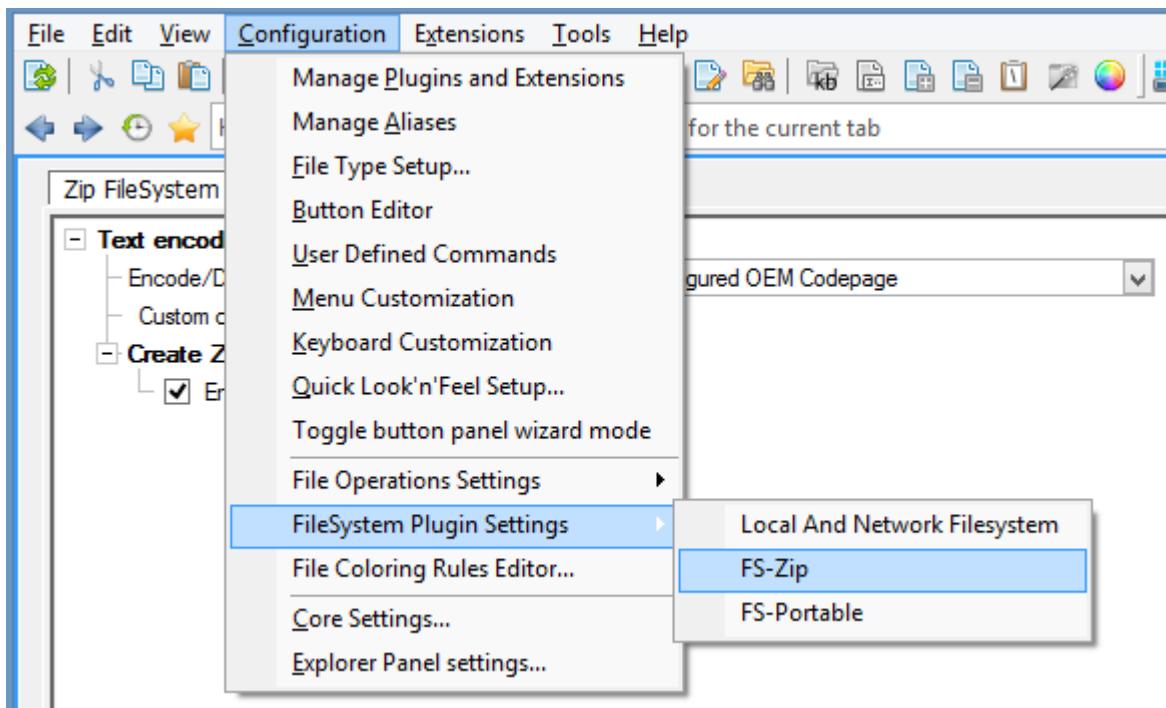
Just like if the target file was a folder

FS-Zip

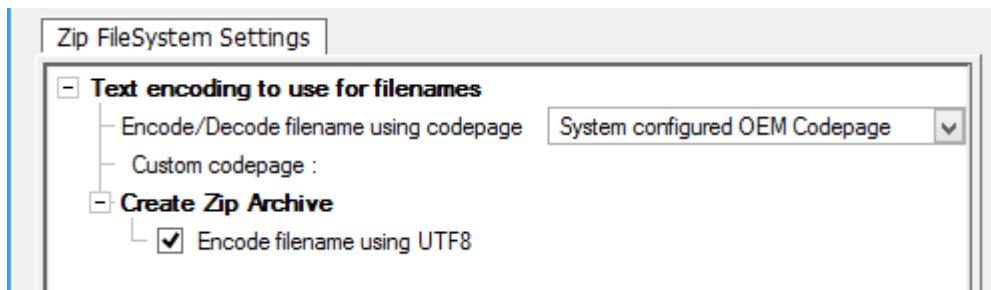
This virtual filesystem plug-ins adds support for opening and modifying and creating zip archives. Zip is a very old format and because of that it does not handle filenames very well that have characters that do not fit in the 7-Bit Ascii Table. The Zip standard have been updated to support UTF-8 encoded filename and FS-Zip do support that. But many other programs that creates/read zip files do not support UTF-8 encoding. This can make the filenames look differently if the zip archive are moved between computers with different language settings.

In the settings for the Zip FileSystem plugin it is possible to configure what codepage it should use when opening zip archives. That are not encoded using UTF-8

The settings page is found under the configuration menu



It is also possible to enable/disable if new zip archive created should use UTF-8 encoding

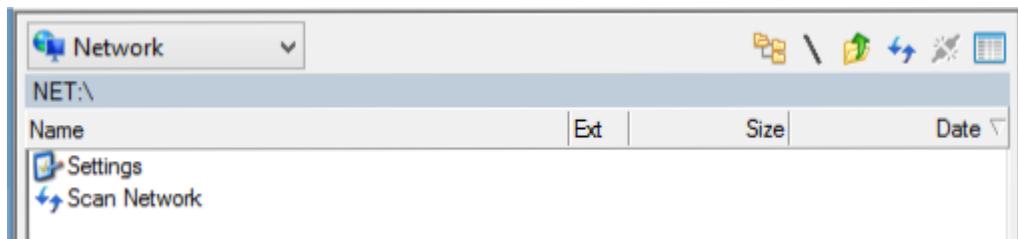


FS-Rar

This virtual filesystem plug-in provide **readonly** support for RAR files. It will allow for browsing and unpacking rar archives.

FileSystem - Network (NET:)

The virtual filesystem device extension **Network (NET:)** allows the local network to be scanned for computers. Found computers will then be listed.



To scan the network for computers execute the "**Scan Network**" command. A progress window will be shown during the scan. The scan can take a couple of minutes if the network is large. When finished all the found computers will be listed.

The list of computers are by default remembered between restarts of Multi Commander.

Opening a computer in the list will make Multi Commander browse that server path. The server path browsing is then handled by the normal [FS-Local](#) virtual file system plug-in. Just as if the path was entered manually.

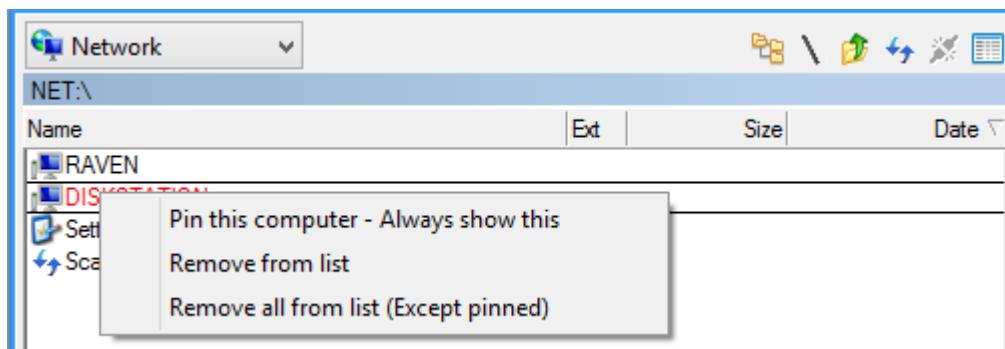
TS Client - Remote Desktop

Running on a computer that are logged in from Remote Desktop, the local drives can be accessed on the remote computer under \\TSCClient (If Remote Desktop is configured to bring with it drives to the remote computer).

The Network virtual filesystem plugin will detect if Multi Commander is running in a Remote Desktop session. And if it is and there are remote connected drives, Then a **TSCClient** entry will also be listed. It works the same as any other computer entry listed.

Network Commands

Right clicking on a computer in the list and a popup window is shown with some commands



Pin this Computer

This command will pin the selected computer. And if a new scan of the network is done this computer will not be removed from the list even if it was not found.

Remove from list

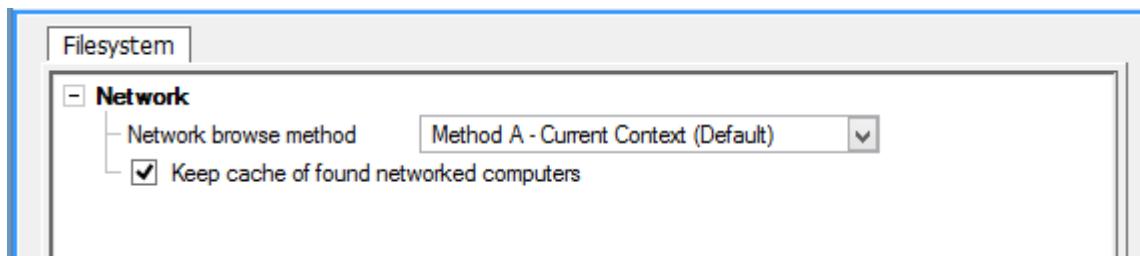
This command will remove the selected computer from the list.

Remove all from list (Except pinned)

This will remove all computers from the list. with the exception of the Pinned computers. They will not be removed.

Settings

Execute the "**Settings**" command to open up the settings page



Network browse method

There are three methods (Current Context, Global and Shell) to use to scan the network. If computer is not found in the list after a scan, Then try to change Network browse method and then scan the network again.

Keep cache of found networked computers

If checked founded computers will be remembered between restarts of Multi Commander.

Filesystem - Portable (WPD:)

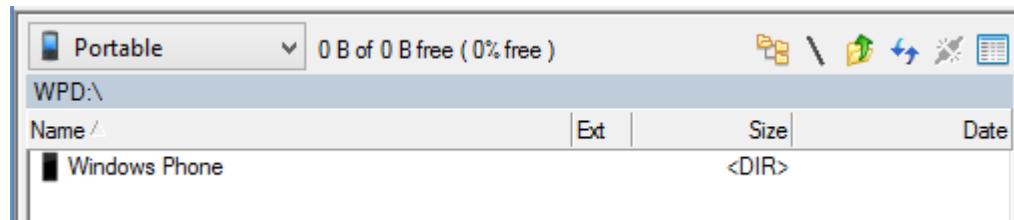
The virtual file system device extension **Portable (WPD:)** allows portable devices like cameras and phones to be accessed.

When connecting a phone, table or camera to the computer that device either allow full file system access and Windows will then provide a drive letter for the device and it is access as any other device. However many modern devices do not support this any more and they require that the device is accessed via a special protocol called MTP. Windows provide access to MTP device via the Windows Portable Device interface. And this file system extension will expose the WPD interface into Multi Commander providing access to MTP devices.

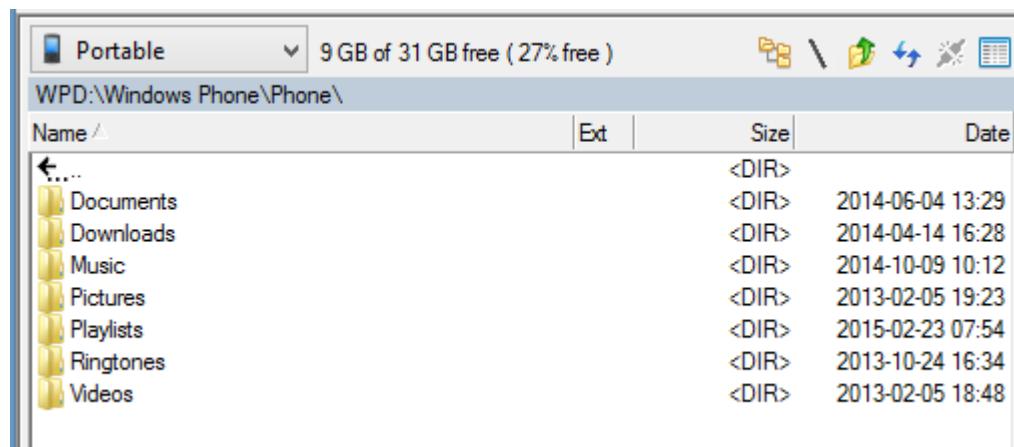
Since full file system access to the device is not allowed some file system operation might not be accessible. It may differ from device to device. Also transferring files to and from an MTP connected device is a lot slower.

Some phones and tables have a settings that will allow the device to be detected as a mass storage device instead. Windows will then get full file system access to the device and will provide a drive letter for the device. This way of accessing the device is a lot faster then using MTP/WPD and is recommended if available

Under WPD:\ all portable device are listed. If the device is now shown try to refresh the view by pressing the refresh button or F5/F2 (depending on setup)



The content of a portable device will be shown as a file system



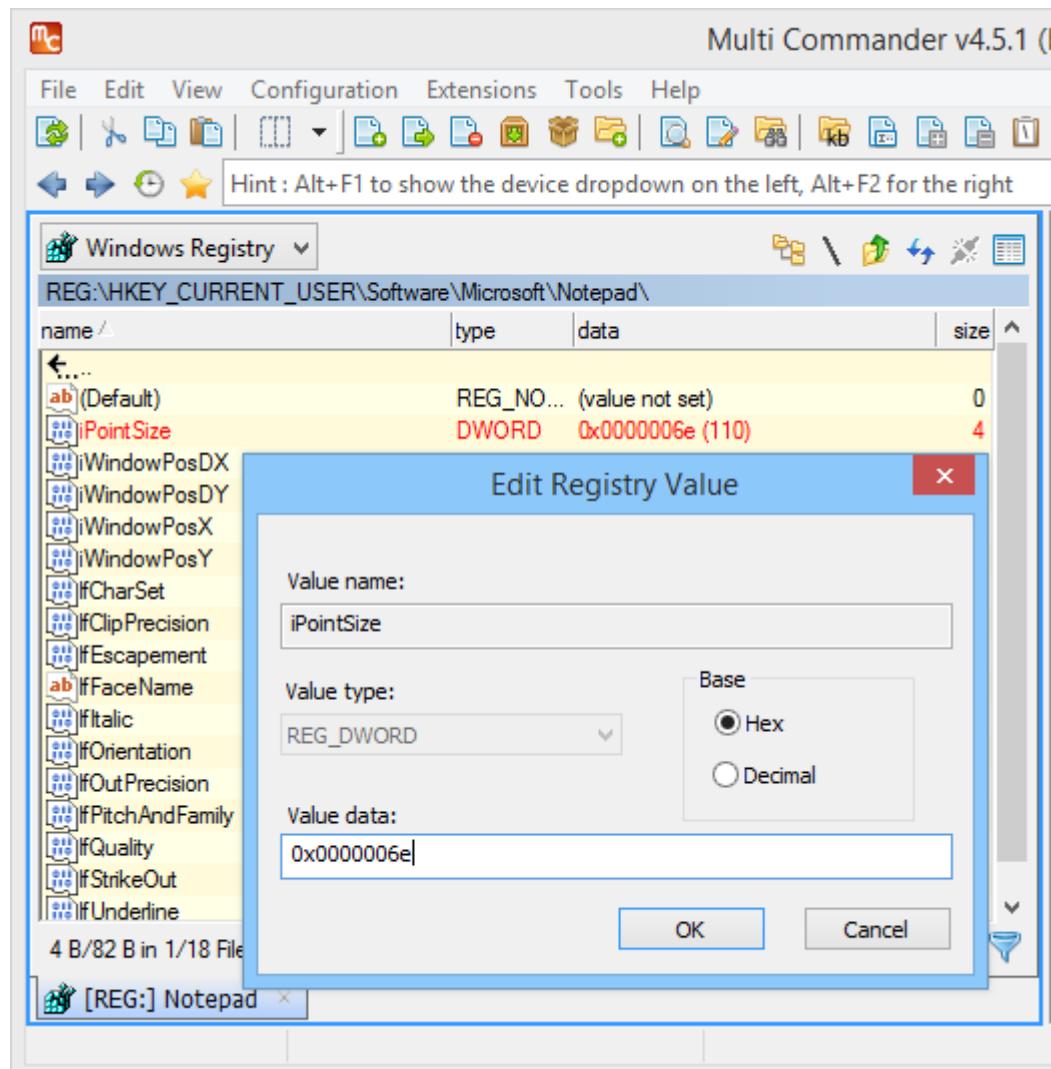
FileSystem - FS-Reg (REG:)

The virtual file system device extension **FS-Reg (REG:)** allows the Windows Registry to be accessed as a file system under the device path REG:

Since MultiCommander treats it as a file system most things you can do in a file system can be done in the Windows Registry.

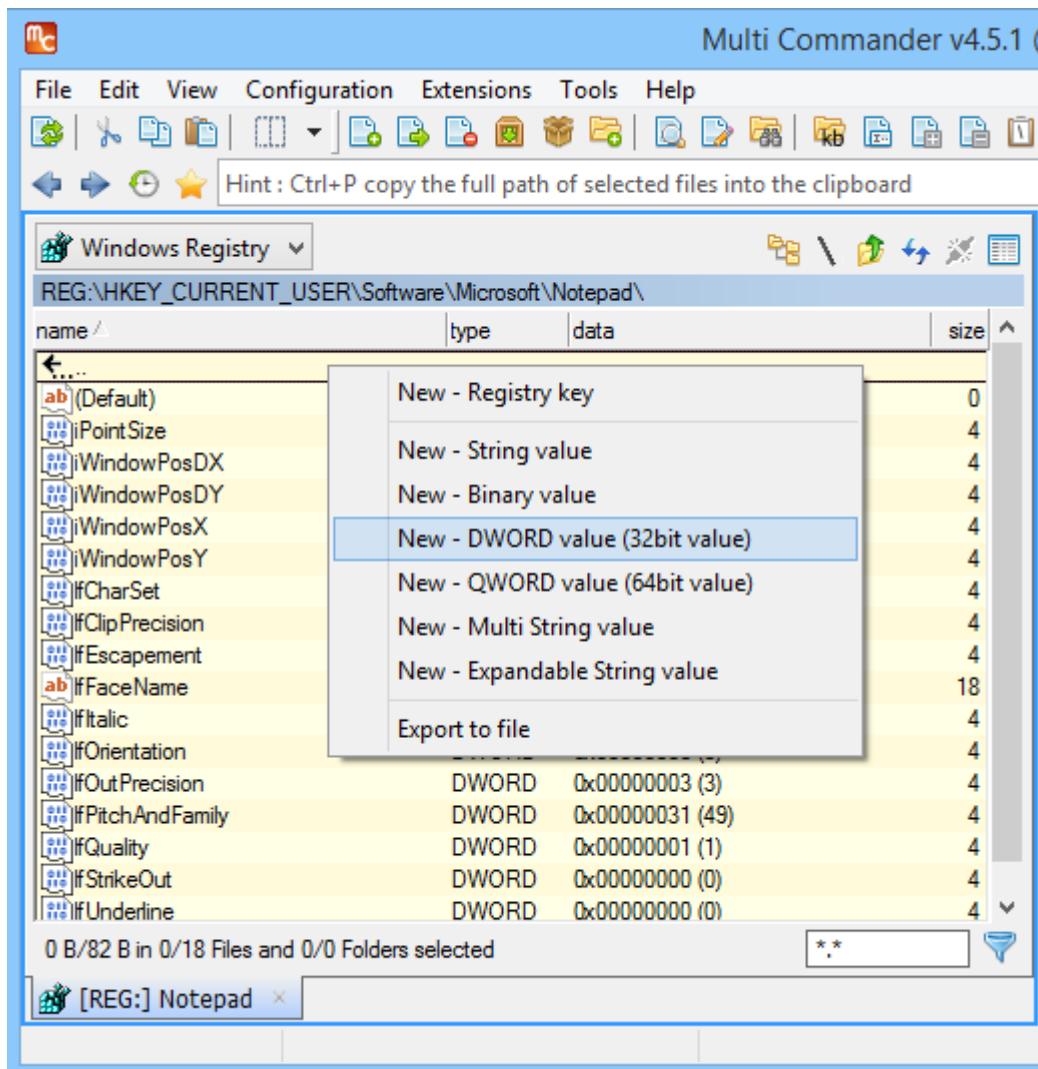
To browse the registry you select "Windows Registry" from the device list dropdown, or you can type in a registry path in the commandline field e.g. "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion" and it will go to the path you specified. You can also type REG: if you want to browse to the registry location yourself.

When you create a **new folder** in the registry you are actually creating a new registry key. You can also **Delete** keys and values or **Rename** them. You are able to **copy** keys and values from one place in the registry to another and you can also copy keys and values to disk where they will be represented as folders and files. You can also copy them back from the filesystem to the Registry.



Create New Value

Select what type of value you want to add from the context menu (*normally triggered by the right mouse button or the context menu key on your keyboard*).



Export Values

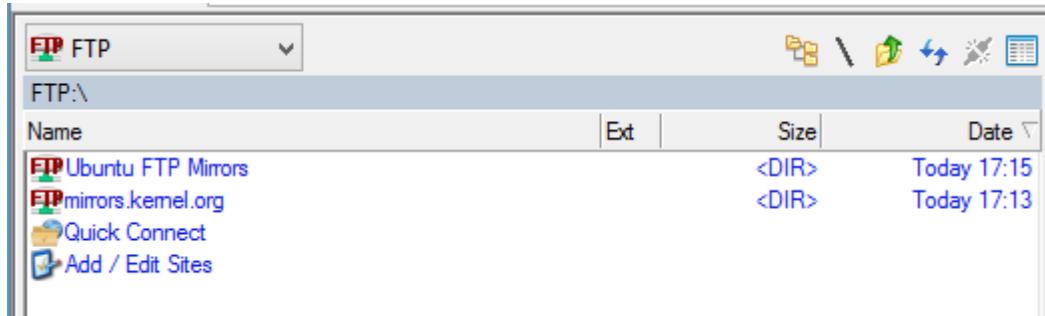
Selecting *export* from the context menu will allow you to export registry values to a standard *.reg* file. The difference between this and the export function in Windows' Registry editor is that here it will only export the values that you have selected. You do not need to export the entire key contents.

Filesystem - FTP (FTP:)

The virtual file system device extension **FTP (FTP:)** allows network connections to other machines using the FTP protocol.

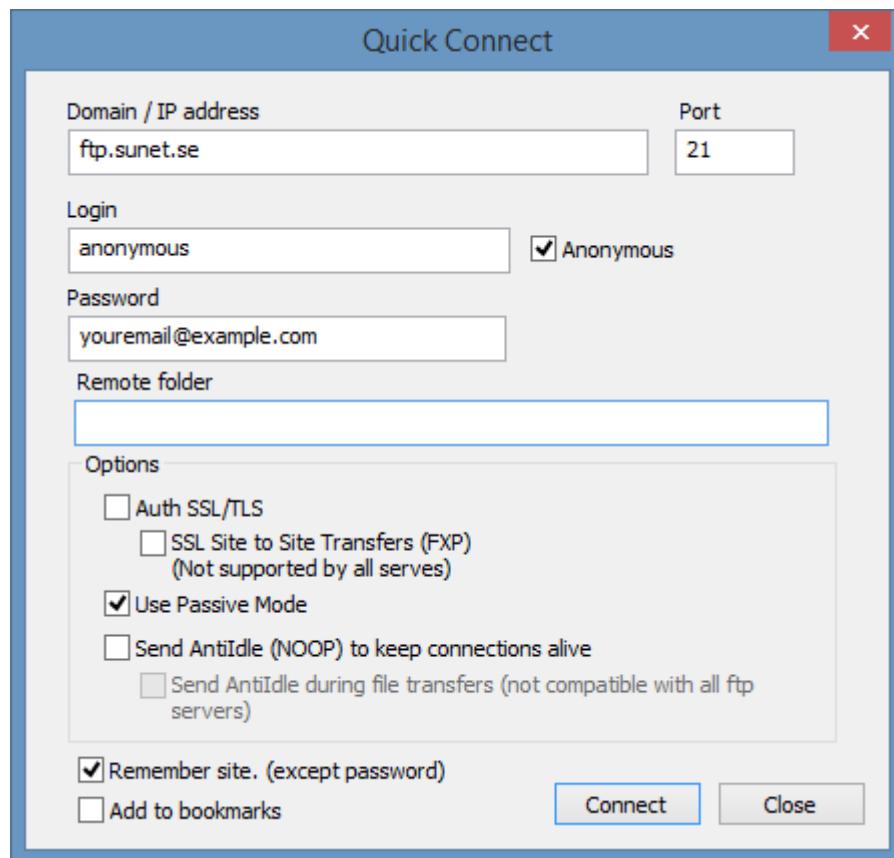
A FTP Connection will be integrated into Multi Commanders virtual file system and anything can then be access as a normal file system

The file system location **FTP:** is special, Special FTP commands and all FTP Bookmarks exists there. When a FTP connection is established the virtual file system will connect that location to a number device. From 0: to 9: , The first FTP connection will be 0: , Next will be 1: and so on.



Quick Connect

The Quick Connect command will the Quick Connect window allowing a connection to be created without adding a bookmark.



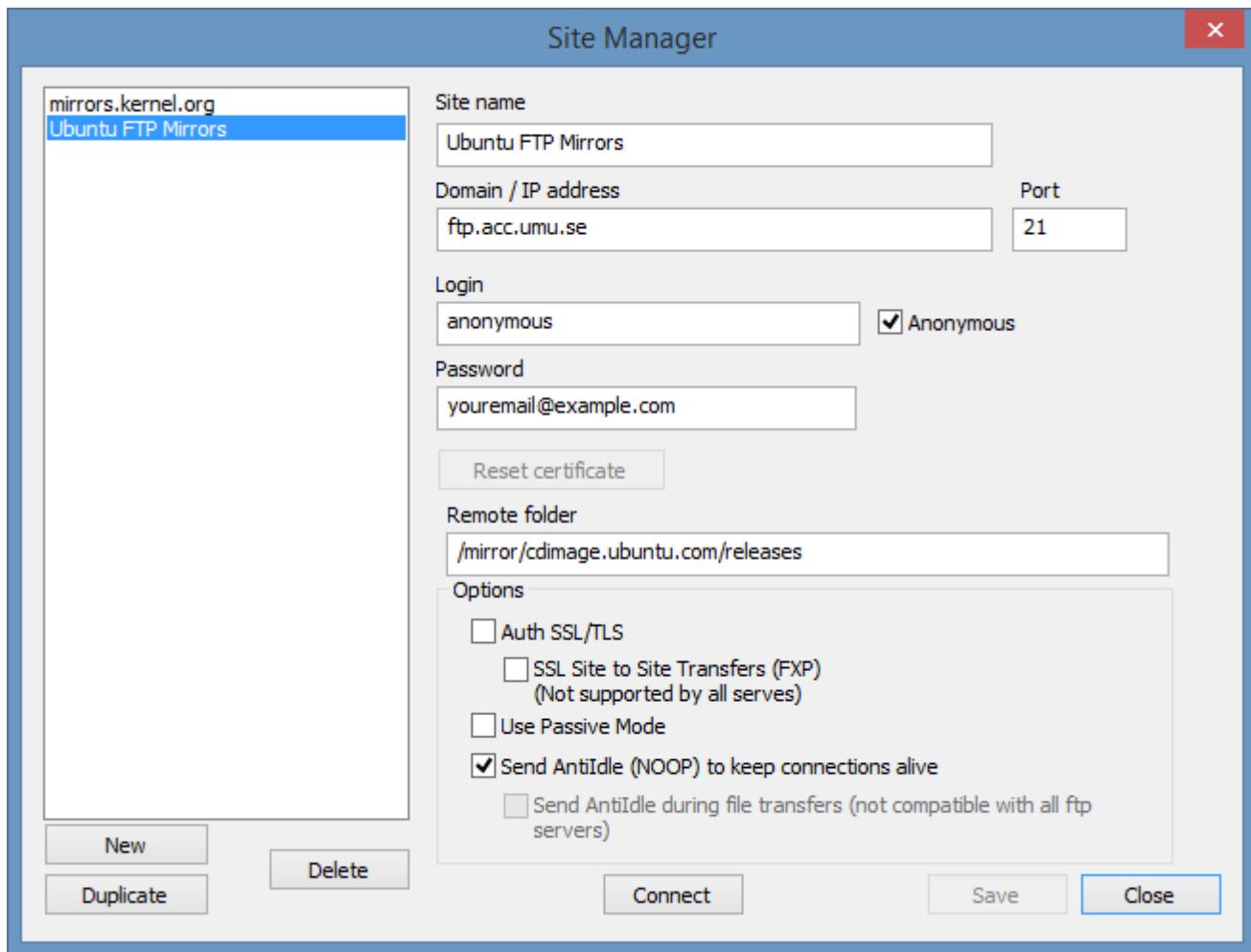
Enter the information needed to connect to a FTP site. Press then the Connect button connect to that site.

By checking the option "**Add to bookmark**" a bookmark for this site will also be created.

The quick connection information (But not the password) is remember for the next time. Uncheck the option "**Remember site**" and the Quick Connect window will not remember anything entered into it

Add/Edit Bookmark

This command will show the Site Manager window. In this window new bookmarks for sites can be added, removed or changed.



Create a new bookmark by pressing the "**New**" button. Then enter all the information needed for that bookmark and then press the "**Save**" button.

The "**Duplicate**" button will create a duplicate of the selected bookmark. Then change the Name and edit the new bookmark and then press the "**Save**" button.

To delete a bookmark. First selected the bookmark to delete and then press the "**Delete**" button.

When closing the Site Manager window the FTP: view should refresh and show the new sites. If this does not happen do a manual refresh by pressing the refresh button or the press F2/F5 (depending of default setup)

Connect

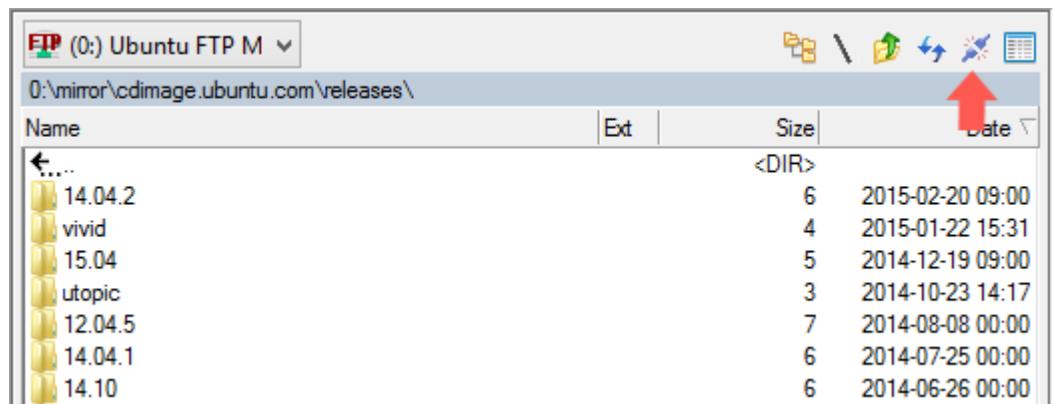
Connect to a FTP site can be done from the Site Manager by selecting the bookmark to connect to and pressing "**Connect**".

Executing a bookmark in the list by double clicking it or pressing return/enter key on it will also initiate a connection to the site.

Logging is very important in FTP. The FTP Logs is used for error message. But FTP server also use the log to show shows information about the site you connecting to. A lot of FTP server will send a lot of important connection information here. A log tab will be created for every site/bookmark that it tries to connect to.

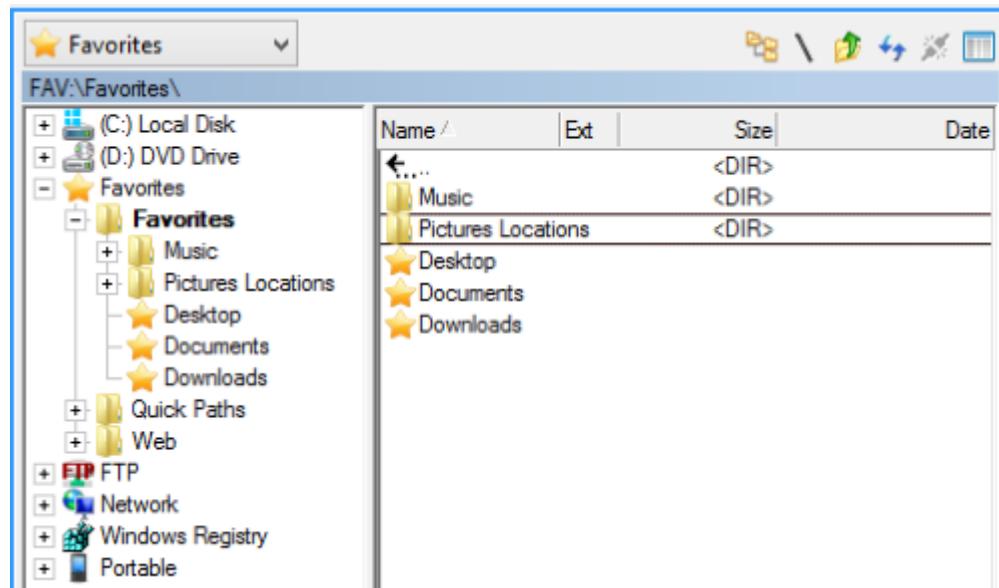
Disconnect

Press the disconnect button that is located in the Explorer Panel to disconnect from the FTP server



FileSystem - Favorites (FAV:)

The virtual file system device extension **Favorites (FAV:)** allows the favorites in Multi Commander to be browsed like a file system.



Accessing the favorites from the file system you can you drag and drop to add new favorites or to copy files to the target for a favorite.

Drag a file/folder and drop it on empty space or on a section to place a favorite there.

Drag a file/folder and drop it on a existing favorite and a copy operation of the dragged files to that favorites target location will be started

Right click an a favorites item will allow you to edit and remove favorites. Also normal file operations like rename/delete will also work in favorites

[Read more about Favorites here](#)

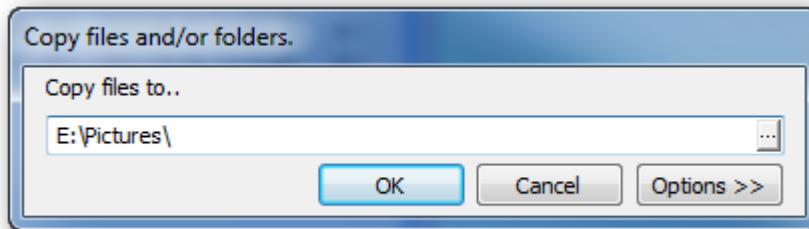
Copy/Move Operations

To **copy** or **move** files and folders you first have to select the desired items; then there are a number of ways to start the copy/move operation:

1. You can drag and drop the selected files to where you want them copied or moved.
2. You can select "Copy..." or "Move..." in the File menu.
3. You can press the Copy or Move Button in the button panel, or in the File Commands toolbar.
4. You can press the shortcut key for the Copy or Move command.

(The default for Commander-styled setup is F5/F6, for Explorer style it is F4/F6.)

When a Copy or Move operation has been requested a dialog asking where to copy/move the files to is shown.



The target for the copy/move operation will by default be set to the path of the target panel.

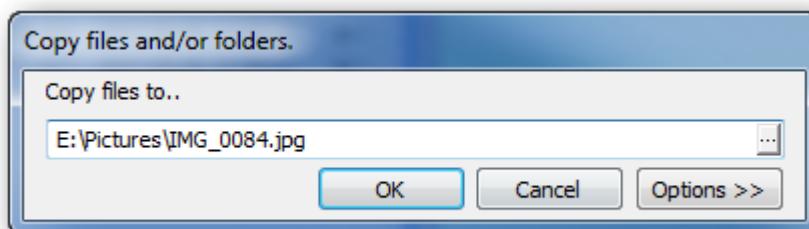
To start the Copy or Move operation you then press the OK button.

Running multiple copy/move operations concurrently on the same disks can affect performance.

Therefore If another Copy or Move operation is already in progress a dialog will be shown asking if the new Copy/Move operation should be added to the queue of operations that are already running.

[Read more about the Progress Window](#) that will be shown during the copy/move operation.

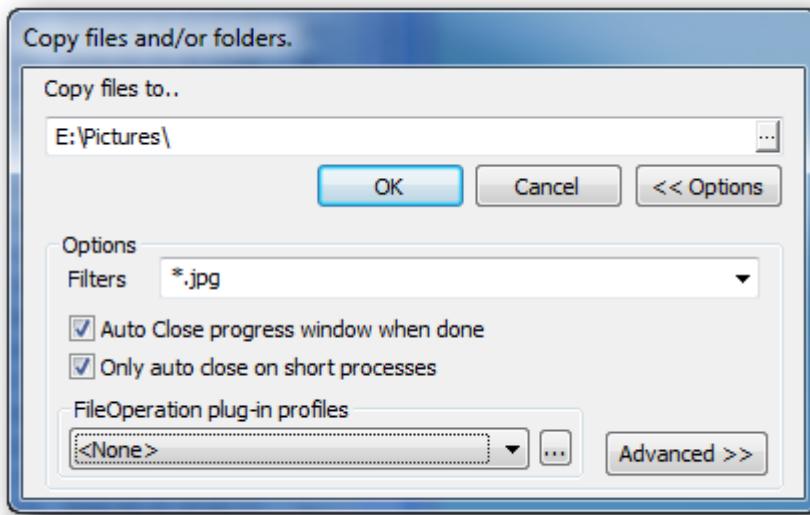
Exceptions for Single File



If you are copying/moving a single file then the file name will also be shown in the target path field. This allows you to change the target name of the file if you wish.

If the target path is removed and only a filename is entered, then the file is written to the same path that it is read from but it is written using its new name.

Options



Pressing the **Options** button causes the dialog to expand and show the basic file operation options.

Filters

Filters can be used to exclude files and folders from being copied/moved. Prefix the filter with "-" to exclude files and "/" to only target folders. Separate filters with spaces. and if a space is needed in the filter then put quotes ("") around the filter.

Examples..

*.jpg *.tiff	Only copy jpg and tiff files. Subfolders will be copied but only files matching the filter will be copied.
-.jpg	Do not copy/move any jpg files.
-/Img -*.txt	Do not copy/move any folder named "Img" (all files inside that folder will also be ignored). Also exclude all .txt files.

Auto Close

These 2 options specify when the copy/move progress bar should be closed, whether the progress dialog should be closed automatically when done, and if it should do that only for short file operations. The default values for these options are configured in Core Configuration.

(Menu -> Configuration -> Core Configuration -> (tab) File System).

When you only do short copy operations such as copying a few images that will only take a few seconds, then you might want the progress dialog to auto close when finished. However if the copy operation takes a very long time such as copying thousands of files, then you might want to keep it open after it is finished.

FileOperation Plug-in profiles

FileOperation Plug-ins can be used when copying/moving files. These plug-ins can interact with the copy/move operation in difference stages of the process: when files are placed in the queue, before they are read, during data transfer, and after a file is written. Multiple plug-ins can be run consecutively, therefore the plug-ins can be assigned to profiles, and then you select what profile you want to use.

The **FileOperation Plug-in Profiles** are modified and configured in the "**FileOperation plug-in Profile manager**" dialog that is opened by clicking on the "..." button next to the FileOperation Plug-in profiles drop-down list.

By default 3 plug-ins exist:

1. Copy Content

This plug-in takes the content of selected folders and places them in the queue as if you selected all the files in those folders rather than the folders themselves.

This plug-in is mainly used when merging multiple folders into a single target folder. For example, the folders **D:\Images1**, **D:\Images2** and **D:\Images3** are selected and copied using the **Copy Content** plug-in to **D:\All-Images**. Thus all the content from those three folders will be merged together into **D:\All-Images**.

Without this plug-in you would get 3 folders in **D:\All-Images** named **Images1**, **Images2** and **Images3**, each containing all its respective files. To merge all the folders together you would have had to go into **D:\Images1**, select all its files and copy them to **D:\AllImages**, then repeat that process for each of the other folders you wanted to merge.

2. AutoSort (A-Z)

This plug-in will effectively modify the target location of the selected files and folders and distribute them into an A-Z folder structure in the selected target location.

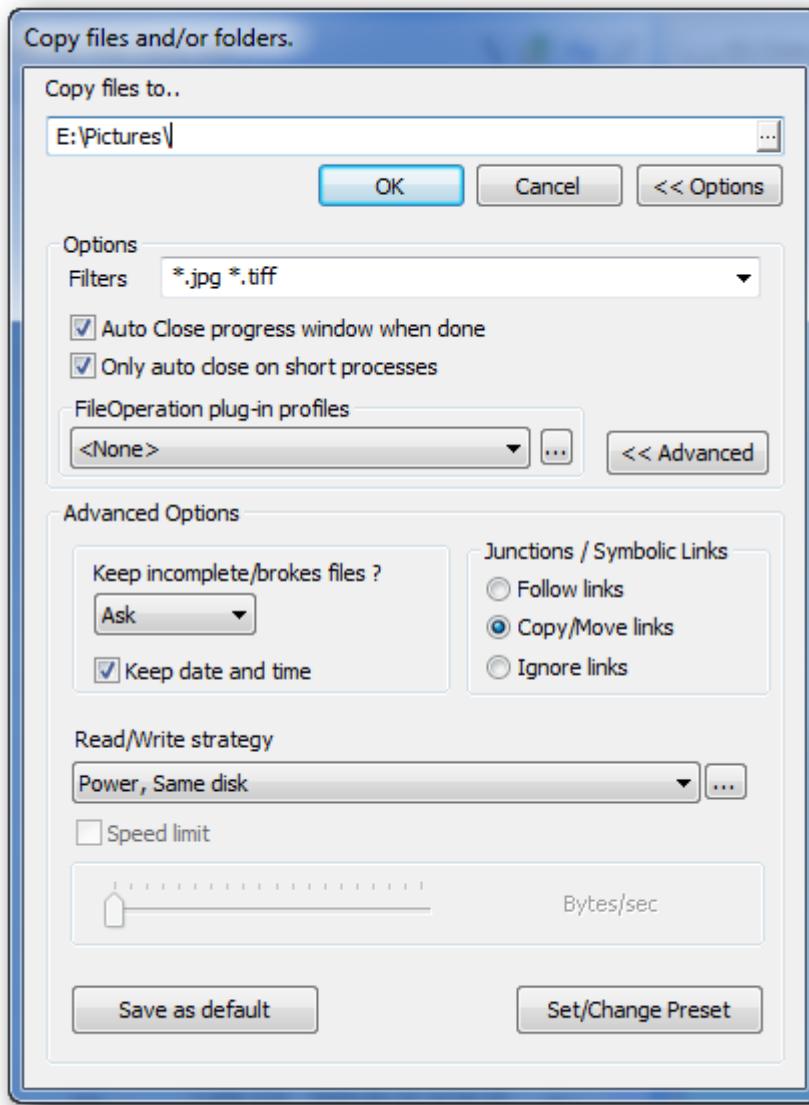
For example if the file "**D:\Images1\Australia_001.jpg**" is specified to be copied to **D:\Images**, it will be placed in **D:\Images\A** instead.

3. Unpacker

This plug-in will unpack all .zip/.rar files (depending on configuration) into the target folder instead of copying/moving them.

For example if it is told to copy the file "**D:\Images_Collections\TripAustralia.zip**" into **D:\Trips**, then it will take all the files inside the .zip archive and copy them into **D:\Trips** instead.

Advanced Options



Pressing the **Advanced** Button will expand the dialog and show the Advanced options. These are options that are rarely changed; some options here should only be changed if you really know what you are doing.

• Keep Incomplete/Broken Files

If a file operation is interrupted when copying a file and the file is incomplete, or if an archive filesystem plug-in for .zip or .rar files informs you that the unpacked file is broken (CRC check may have failed), you can select if you always want to keep it, delete it or be asked what to do each time it happens.

The default value for this option can be configured under the "Filesystem" tab in Core Configuration (found under Menu ->Configuration).

• Keep date and Time

Have this checked if you want the date and time of the files you copy/move to be retained.

• Junctions / Symbolic Links

If the copy/move operation comes to a Junction or Symbolic link in the file system, you can configure how it should be handled. The default for this option can be configured under the "Filesystem" tab in Core Configuration (found under Menu ->Configuration).

(Symbolic links only exist on NTFS file systems, but not on Windows XP/2003 and older.)

- **Follow Links**

This specifies that Multi Commander should copy the file/folder of the link target to the target folder.

- **Copy/Move Link**

This will create a new link on the target and this link will point to the same location that the source link is pointing to. This however requires you to have permissions to Symbolic links or Junctions.

- **Ignore Links**

This will ignore all types of links, and will not follow or copy/move them.

- **Read/Write Strategy**

This is a list of different read/write strategies that can be used when reading/writing data. The system tries to determine what read/write strategy is best for the current copy operation and it will automatically select that one. It is possible to change that and select another one and also configure it using the "..." button. That is however not recommended if you are not an expert.

- **Speed Limit**

This allows you to limit the max speed of the copy/move operation.

This feature is not yet enabled, and cannot be used at present

Save As Default

Pressing this button will save the settings you currently have chosen to be the default value for all future copy/move operations. The next time you start a copy operation and this dialog is shown the configuration will be set to what you have now.

Set/Change Presets

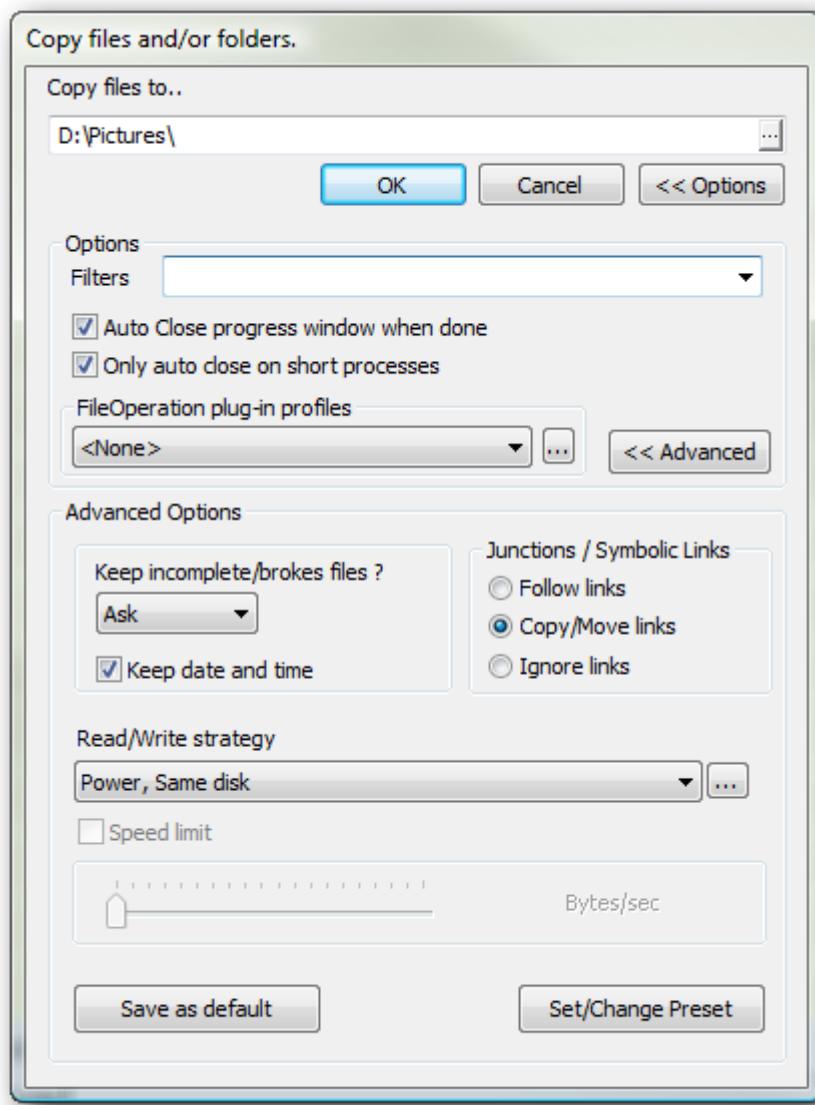
Presets are a special feature where you can save some or all configurations including filters and target locations. These configurations can then be loaded using the functions keys (F1 to F10). A preset can also be configured to be auto loaded when you start a copy/move operation from a specific folder.

[Read more about FileOperation Presets](#)

Copy/Move using Presets

Creating a **preset** can be helpful if you frequently copy or move files from/to the same locations and are using filters or some other settings that are not there by default. The preset will store those settings for you, and you can then load that preset with a simple press of a button (F1 to F10). If you always want to load the preset when you start a copy or move from a certain path, you can make the preset auto load and you do not even have to press a button to load it.

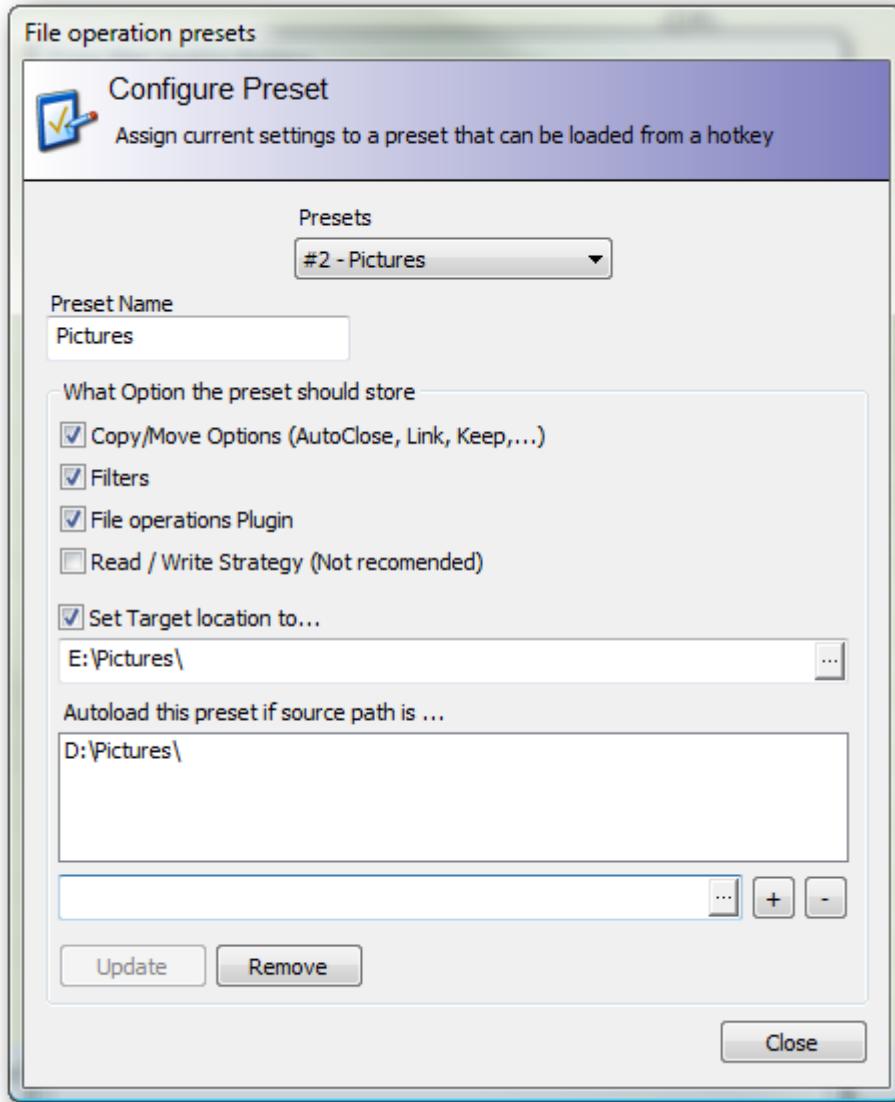
To create a preset begin your normal process of copying/moving the file. Set your **filter** and **FileOperations plug-in** and other **options** you want to use. Then expand the dialog to max and click on **Set/Change Preset**.



Select a free **Presets** slot and enter the name you want to use for the preset. Then check the option you want the preset to store.

You can also check the option to "**Set Target Location to...**" if you want the preset to store the target location and always set that location as target no matter what path you had as target when you started the copy/move process.

If you want to **Auto load** the preset, you need to add what source path it should be connected to. (*Wildcard for the path is supported.*)



In this example you will see that we have a preset that we named "**Pictures**", and we wish it to store all the **Copy/Move Options**, **Filters** and **FileOperation plugins** that we defined previously before going into this dialog. We also want the target path to always be set to "**E:\Pictures**" and the preset to be auto loaded when we start a copy/move operation from "**D:\Pictures**".

When you enter the Copy/Move dialog to start a copy/move operation and a preset is auto loaded, you can unload the preset with **F12** or press the "**Unload Preset**" button that appears next to the OK button. (*not shown in screenshots*)

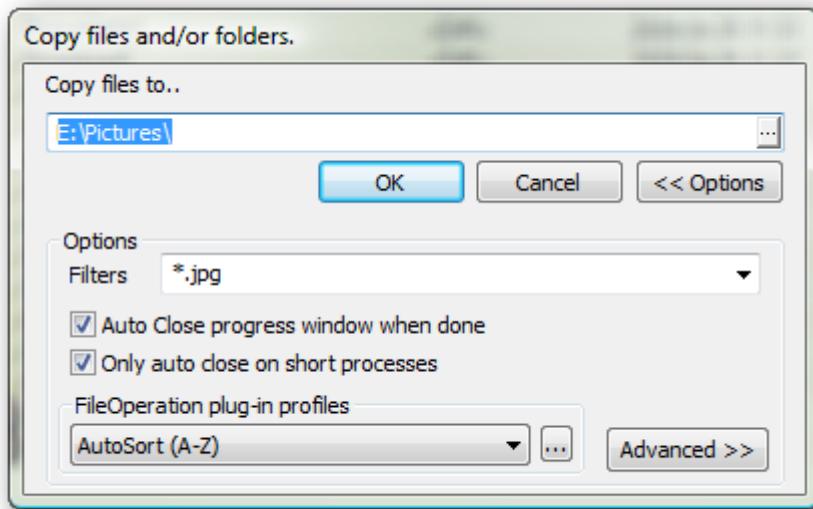
Copy files and folders using filters

When copying or moving files and folders you can use a **filter** to limit the files and folders that should be processed.

If a filter should match a folder then prefix the filter with a slash "/" and if you want to exclude any files prefix it with a dash "-". If a folder should be excluded use the dash first (eg "-/MyFolder")

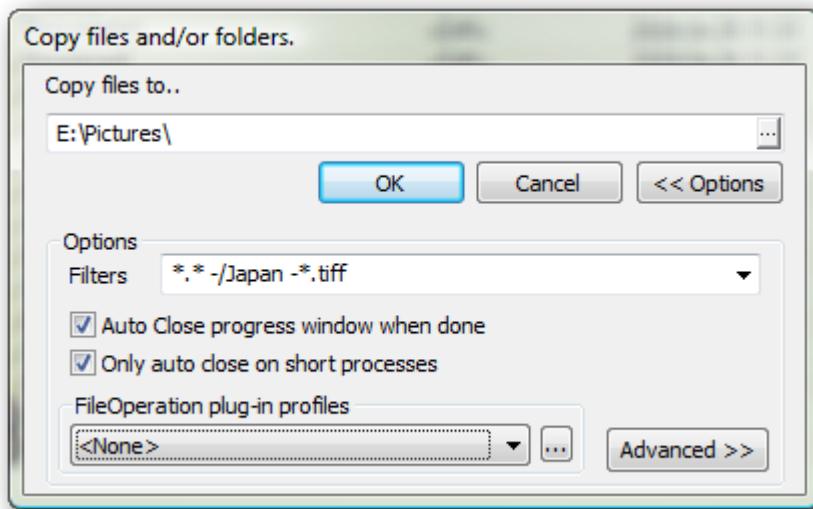
Example #1

*A number of folders were selected and we choose to copy them to E:\Pictures|. Using filters we also specify that we only want to copy the *.jpg files. We also select the AutoSort plugin so the files will be stored in A-Z folders.*

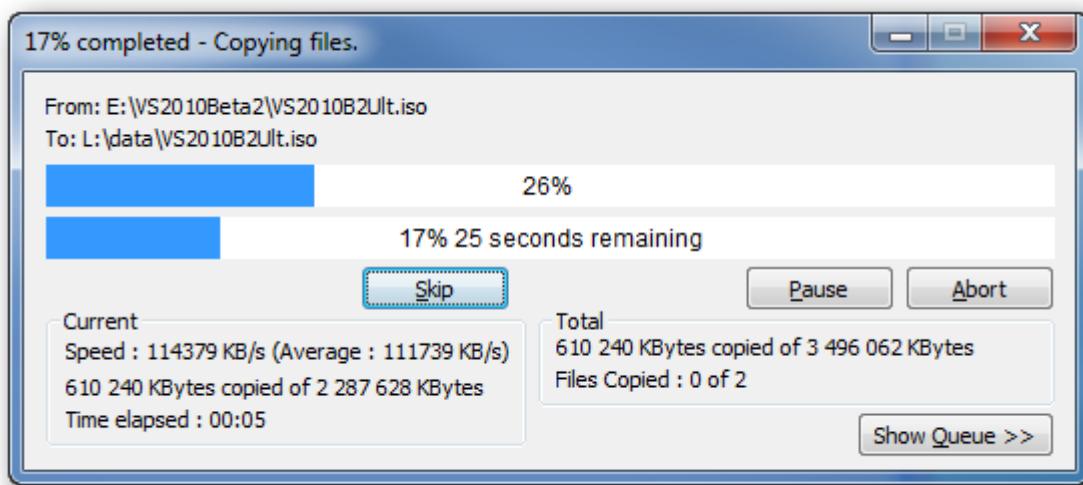


Example #2

Some folders have been selected and we choose to copy them to E:\Pictures|. In filters we also specify that we want to include all files (.*) but we want to exclude the folder named **Japan** and all *.tiff files.*



Copy/Move - Progress Window



When a copy/move operation has started a **progress window** is shown.

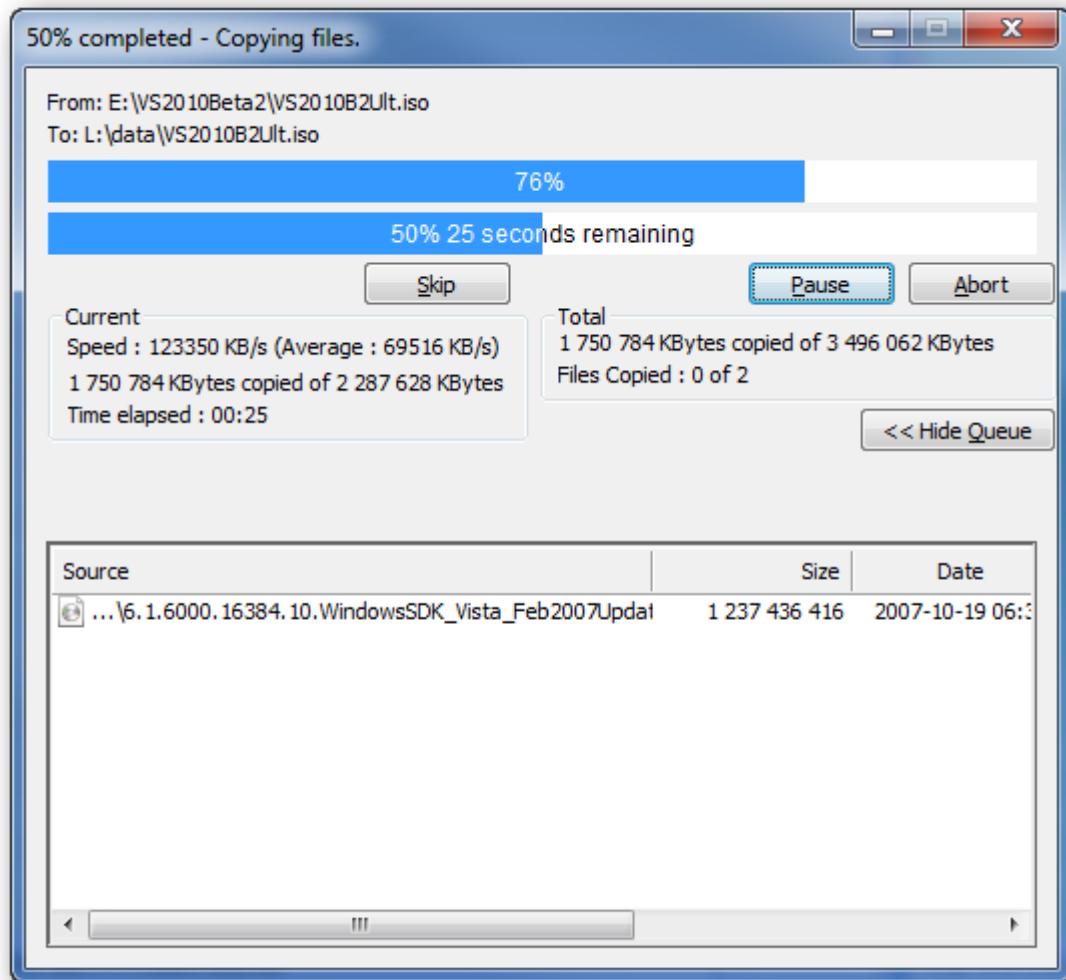
The progress window shows you the speed and progress of the current file and the total progress of the entire operation. You can pause the operation by pressing the "**Pause**" button. The button's name will then be changed to "**Continue**" and you can press it again to continue processing.

To stop the copy/move operation you can press the "**Abort**" Button or you can press the X button in the caption bar.

Pressing the "**Skip**" button will allow you to skip the current file. It will then continue on to the next file in the queue.

Show Queue

Pressing the "**Show Queue >>**" button will allow you to see the copy/move queue.

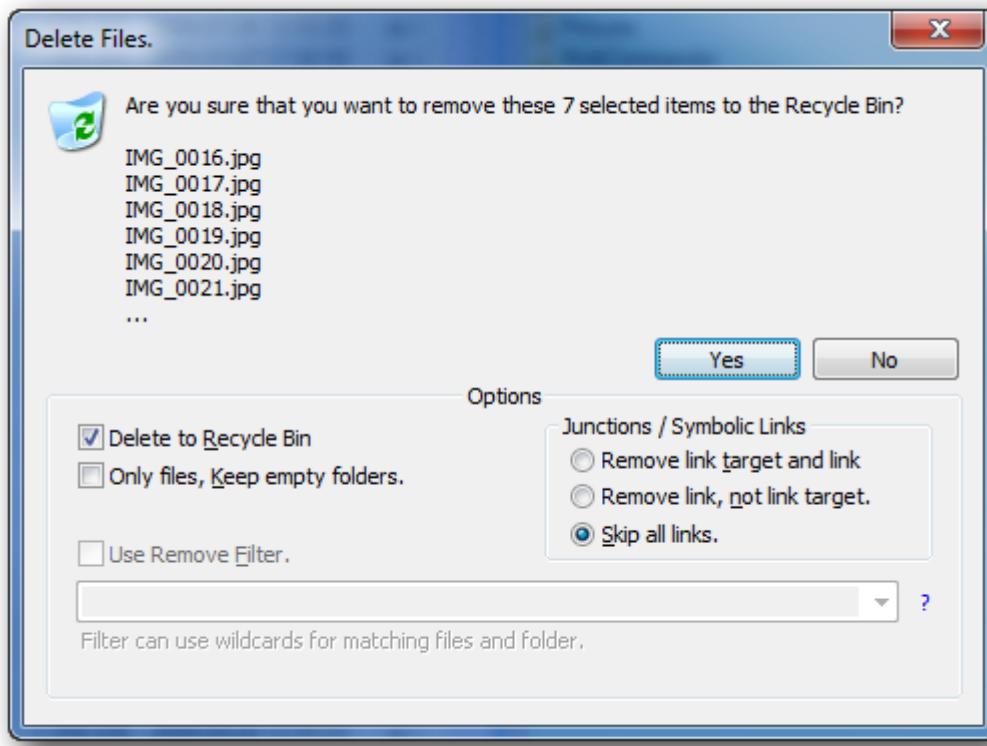


Delete Operation

You can delete files or folders using any of the following:

1. The Delete Button in the File Operations tool bar.
2. The Delete Button in the Button Panel.
3. The Function Key F8.

All these display a dialog that lets you modify options for the delete operation, as shown below:



The following options are available:

- **Delete to Recycle Bin**

When checked, the selected files and folders are moved to the Recycle Bin and can be recovered if needed. When deleting files inside an archive or on a network device, this option will be disabled and cannot be selected.

- **Only Files**

If checked, only files are deleted. Files in folders will be deleted but the empty folders will remain on the system.

- **Junctions and Symbolic Links**

This section controls how Multi Commander should react when it finds Junctions and Symbolic links. The recommendation is to ignore or remove the link (if permissions allow it).

It is **not** recommended to set **Follow links**, which should be used only if you really know what you are doing. A warning is shown if that option is selected.

- **Remove Filter**

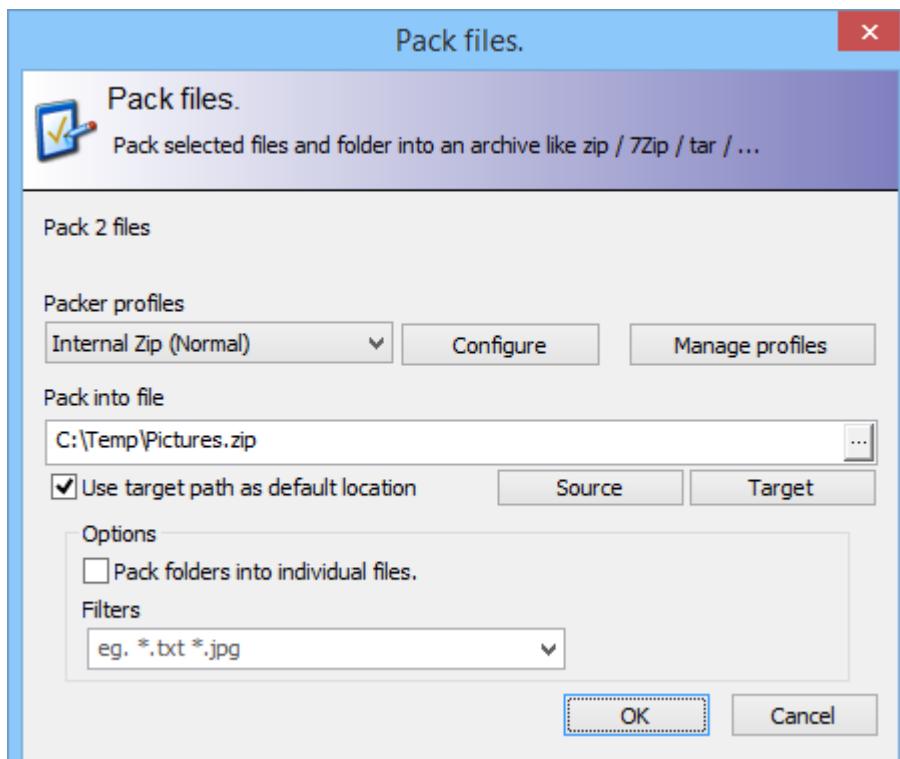
This is a filter to be used for files in selected folders. Only Files and Sub-Folders matching the given filter will be deleted.

This option is not completed yet, and cannot be used.

Pack Files

To pack files into an archive you first need to select the files and folders you want to pack.

To start a pack operation, either press the shortcut (default Alt+F5 if running with commander styled setup) or select "Pack files..." in the file menu.



From the Pack Files dialog you can select what packer profile you want to use. Different profiles can use a different packer with a different default configuration.

To reconfigure a profile, press the "**Configure**" button. You can then configure the selected Packer Profile, i.e. change compression levels and whatever else that packer supports.

Press "**Manage Profiles**" to Create or Delete Packer profiles.

Source (Button)

This button changes the target path for the packed file to the path of the source panel.

Target (Button)

This button changes the target path for the packed file to the path of the target panel.

Pack folders into individual files

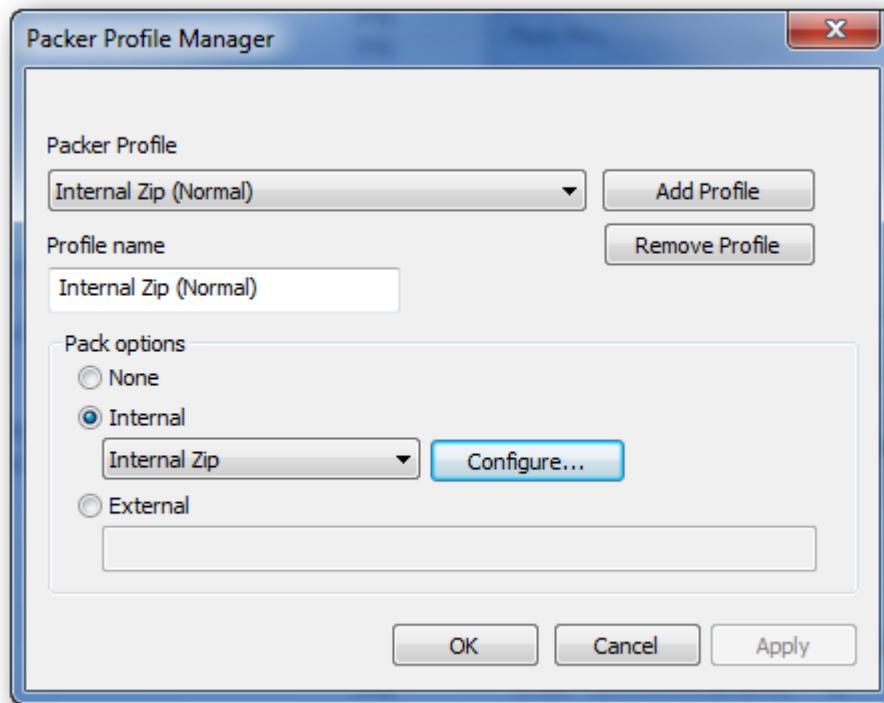
When this is checked, each of the selected folders will have its files packed into a separate archive file (using the source folder name as its filename). Otherwise, the selected folders will be combined into a single archive file. They will **not** be packed individually.

Filter

You can specify Filters to include and/or exclude files/folders when packing. These filters work the same as filters in the Copy/Move options [here](#).

Packer Profiles

The Pack Files dialog lets you manage **Packer Profiles** by pressing the **Manage Profiles** button. This displays a **Packer Profile Manager** dialog that lets you create your own Packer Profiles or change the default values of the existing ones.



Select the Packer Profile you want to change or press the "Add Profile" button to create a new one. Then select what packer you want to use (e.g. Zip or 7-Zip) and press the "Configure..." button to configure the default values for this profile. The Default values are the values the packer should use if the user does not change it in the Pack Files dialog.

External Packers

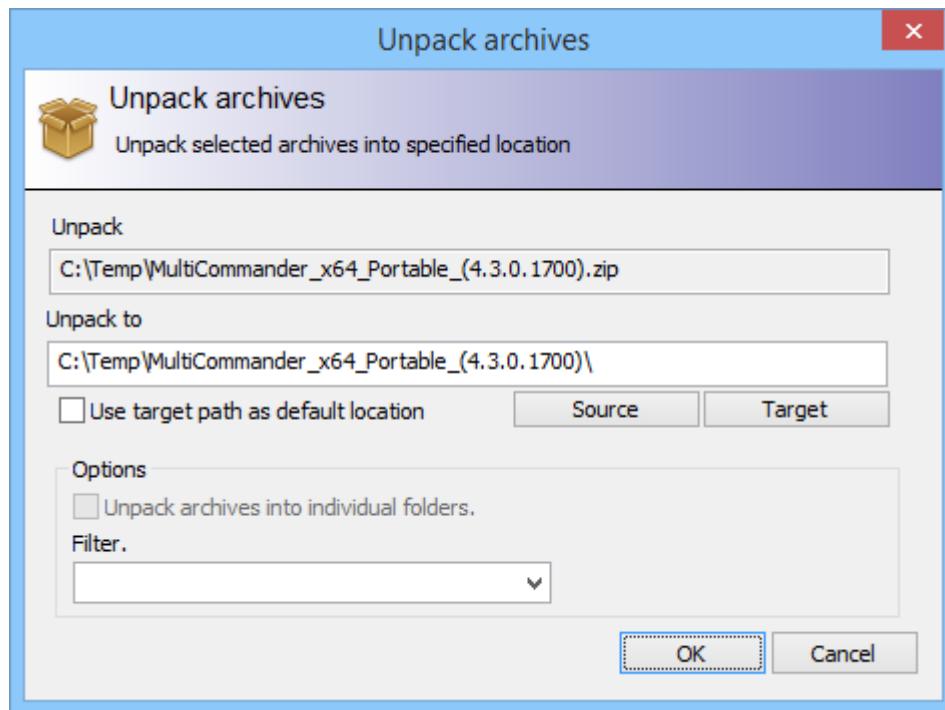
This is not activated yet.

Packers - External Packer Profiles

The Packer Profiles do not yet support external packers. But this feature is on the roadmap and will be included.

Unpack Files

Select the file or files that you want to unpack then select Unpack file from the File Menu or press the keyboard shortcut.



Shortcuts

Alt+F6 - Unpack selected files

A dialog will be shown where you can choose where the file should be unpacked. By default the path in the target view is selected.

You can easily toggle between the source and target path by pressing the source/target buttons.

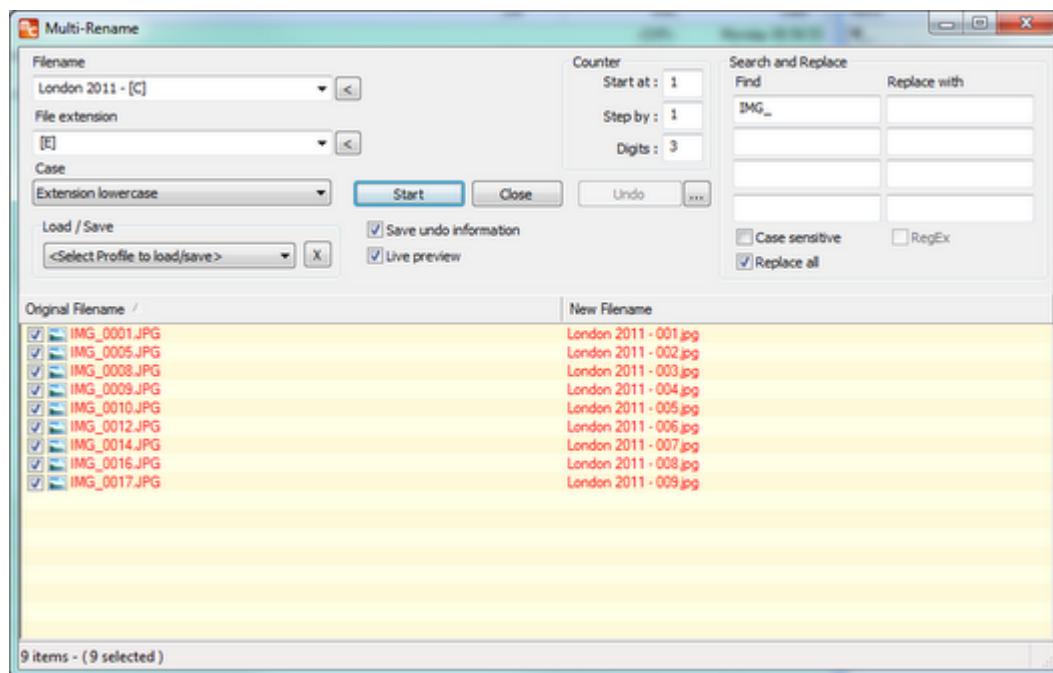
If the files are unpacked into the same folder they are located in (source folder) then they will be unpacked into a folder with the same name as the packed file, but with no file extensions.

For example if you unpack *c:\temp\myfile.zip* into *c:\temp* then it will be unpacked into *c:\temp\myfile*

Filter

It is also possible to specify a **filter**. The filter will exclude and/or include files/folders from the unpacking. The filter works the same as the filter for file copy.

Multi Rename



Overview

With Multi Rename you can rename multiple files or folders by creating a *rename rule*.

You can easily select part of the existing file name, add text, replace characters or words, insert file property data in the file name, (E.g. Date, Exif information for photos and many more)

Undo information is also saved so if you rename files incorrectly or by mistake renamed too many files, you can easily load a previous rename session and undo it.

If you are doing the same multiple renames many times you can save the rules and then load them when you need them again.

Rename Rules

By insert different tags you will insert different text parts into the name.

By pressing the [<] button that are next to the filename and file extension field will be see a list of the most common tags you can use and if you press "Plugin and more..." you will be get a list will all the extended tags you can add.

You also can specify parameters into the tag. The parameters tell if what part of the tag text it will insert.

[<tag>]	Insert the complete text from the tag
[<tag><num>]	Insert the first specified number of characters from the tag. Example [N4] will insert the first 4 letters of the filename (same as above)
[<tag>,<num>]	Insert the first specified number of characters from the tag. Example [N4] will insert the first 4 letters of the filename.
<tag>,<FromPos> ,<length>]	Insert the text from position "pos" of the tag data. But with a max length of <length> Example [N,3,4] - If filename is "MyFile" then it will insert "File"

Special tag - Counter [C]

There are some special tags that works a bit differently. The counter tag [C] is one of them. By default it is controlled by the counter option specified in Multi-Rename window. But you can override that by using tag parameters.

[C]	Insert a default counter tag
[C,<startnum>,<inc>]	Insert a default counter tag that will start counting at <startnum> and increment the value with <inc> for every file Example [C,10,2] - It will start at 10 and incremtn the value with 2 so you get 10,12,14,16
[C,<startnum>,<inc>,<digits>]	the two first parameters works as above, <digits> let yo uspecify how many digits you want to have as minimum, Example[C,10,2,3] - It will start with 10. but since digits are 3 you will get 010 , 012, 014

Special tag - Current Date/Time [Y],[M],...

Another special tag is the date/time tags. They will get you the date/time of the current time.

[Y]	Insert current year as 4 digits eg 2013
[Y,2]	Insert current year with 2 digits: eg 13
[M]	
[D]	Inserts Day, Month, hour, minutes, second of current time.
[h]	Month and days will be inserted as digits. If the month is April it will insert 4
[m]	
[s]	
[M,<digits>]	
[D,<digits>]	Inserts Day, Month, hour, minutes, second of current time.
[h,<digits>]	It will insert a minimum of specified digits. If month is 4 it and <digits> is set to 2 it will be inserted as 04
[m,<digits>]	
[s,<digits>]	

Special tag - Date/Time property from Plugin.

If you want the file date/time then you need to select a tag from the plugin list, Like the [?ExtendedProp.datecreate] tag.
(Btw all plugin tags starts with a ?)

All file properties from plugins that returns a date/time will return the date/time formated using system settings.
However this can be a problem because if time is returned as **12:22:22** the rename will fail since ":" is an invalid character for filenames.
You then need to replace the : with some other letter (or blank) in the **Search and Replace** section in the Multi-Rename Windows.

You can also specify a custom date and time formatting to the tag.

[?ExtendedProp.datecreate]	Insert current date/time from this date/time file propertyThe date/time is formated using current system setting. so it might return 2013-03-03 08:23:23 Make sure you replace invalid characters in "Search and Replace" or you might get rename problems.
[?ExtendedProp.datecreate-{dateformat timeformat}]	Insert current date/time from this date/time file property using a custom date/time formatting Example: [?ExtendedProp.datecreate-{yyyy-MM-dd HHmmss}]
[?ExtendedProp.datecreate-{dateformat}]	Insert only date part using custom date formatting Example: [?ExtendedProp.datecreate-{yyyy-MM-dd}]
[?ExtendedProp.datecreate-{ timeformat}]	Insert only time part using custom time formatting Example [?ExtendedProp.datecreate-{ HHmmss}] (Do not forget the before the time format.)
[?Ex...datecreate-{dateformat timeformat},<fromPos>,<Length>]	You can still use the from pos and length parameters Example [?ExtendedProp.datecreate-{yyyy-MM-dd HHmmss},1,5]

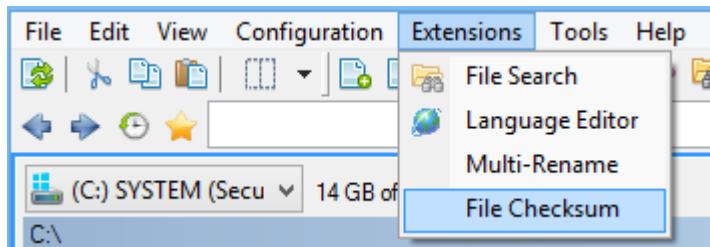
How the date and time format works [Date Formatting - Time Formatting](#)

Examples

Tags	Original filenames	New Filenames
File: Pre-[N,2,4]-[C,1,2,3]-X	Done.txt	Pre-one-001-X.txt
Ext: [E]	MyFile.txt	Pre-yFil-003-X.txt
	Todo.txt	Pre-odo-005-X.txt
File: [C] - [N] - [?MCAudioProp.genre]	Song A.mp3	01 - Song A - Trance.mp3
Ext: [E]	Song B.mp3	02 - Song B - Pop.mp3
	Song C.mp3	03 - Song C - House.mp3

File Checksum - Create or Verify file CheckSum (CRC/SFV/MD5/SHA1)

With the File Checksum extension you can create and verify checksum files.



Supported checksum file are SFV(CRC32) , MD5 and SHA1.

You can both create and verify .sfv .md5 checksum files and you can also compare verification values directly.

You can verify multiple .sfv / md5 files at the same time. If you send a folder to the FileVerifier it will scan that folder for all .sfv /.md5 files found.

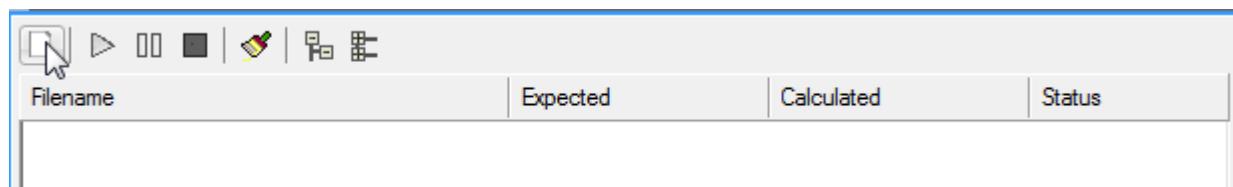
Read more about [Creating Checksum](#) files.

Read more about [Verifying Checksum](#) files.

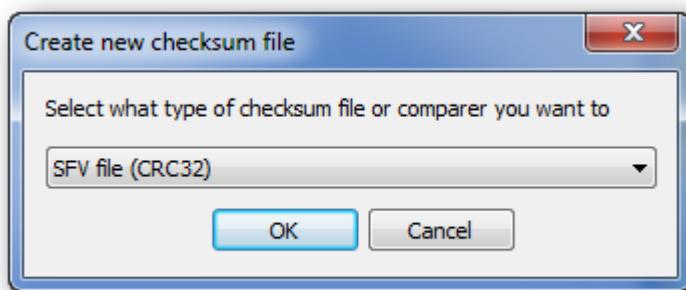
Under the tools menu you will find a couple of quick command for File Checksum. [Read more about them here](#)

File Verifier - Create

To create a new checksum file or to compare checksum values, first select "File Verifier" in the Extension menu, then click on the New button in the File Verifier toolbar that is shown.



In the Create New Checksum dialog, select what type of checksum file you want to create, SFV or MD5.



A new .sfv or .md5 item will be created, named "Untitled sfv file.sfv" for .sfv as default; to change the name you need to click to edit that name. If the name is not changed, a new name will automatically be generated from the name of the first file that is added to this checksum file

Add Files

Add files to the checksum file by using the "copy" command or drag and drop the files you want to add
(The default keyboard shortcut for copy is F5 if using CommanderStyle setup, or F2 for WindowsExplorer style.)

Filename	Expected	Calculated	Status
- <input checked="" type="checkbox"/> Untitled sfv file.sfv			
IMG_0072.jpg			Not verified
IMG_0071.jpg			Not verified
IMG_0070.jpg			Not verified
IMG_0069.jpg			Not verified
IMG_0068.jpg			Not verified
IMG_0067.jpg			Not verified
IMG_0066.jpg			Not verified
IMG_0065.jpg			Not verified

When all the files have been added, press the start button in the FileVerifier toolbar.

The checksum for all the files will now be calculated and saved into the checksum file you specified.

Filename	Expected	Calculated	Status
- <input checked="" type="checkbox"/> IMG_0072.sfv			Succeeded!
IMG_0072.jpg		9f214f4a	OK
IMG_0071.jpg		38da3065	OK
IMG_0070.jpg		87efe23d	OK
IMG_0069.jpg		c4e3be12	OK
IMG_0068.jpg		8c8d3ee4	OK
IMG_0067.jpg		bbc93242	OK
IMG_0066.jpg		322108f	OK
IMG_0065.jpg		4220ee56	OK

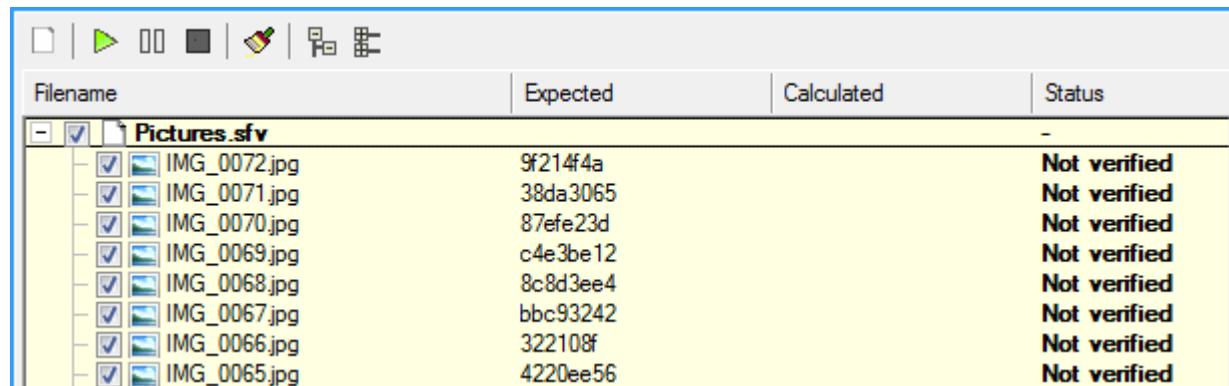
File Verifier - Verify

There are two ways to verify checksums, *Verify Checksum Files* or *Verify a Checksum Value* directly.

Verify Checksum Files (.SFV .MD5)

If you have a .sfv or .md5 file and want to verify the checksum for the files that this file references, simply launch the checksum file from within MultiCommander and the FileVerifier will automatically start a verify process.

You can also start the File Verifier extension you select and then copy or drag and drop the .sfv/.md5 files to the panel with the File Verifier open. If you copy/drag and drop folders it will scan those folders for .sfv or .md5 and automatically add them to the file verifier layout.



The screenshot shows the Multi Commander interface with the 'File Verifier' extension active. The toolbar at the top includes icons for Start, Stop, Pause, and other file operations. Below the toolbar is a table with four columns: 'Filename', 'Expected', 'Calculated', and 'Status'. The table lists files from a file named 'Pictures.sfv'. All files listed have a status of 'Not verified'.

Filename	Expected	Calculated	Status
-	-	-	-
<input checked="" type="checkbox"/> Pictures.sfv			
IMG_0072.jpg	9f214f4a		Not verified
IMG_0071.jpg	38da3065		Not verified
IMG_0070.jpg	87efe23d		Not verified
IMG_0069.jpg	c4e3be12		Not verified
IMG_0068.jpg	8c8d3ee4		Not verified
IMG_0067.jpg	bbc93242		Not verified
IMG_0066.jpg	322108f		Not verified
IMG_0065.jpg	4220ee56		Not verified

You then press the Start button in the file verifier toolbar to start the verify process

If Errors found

You will be presented with a dialog asking what to do the the bad files. You will have the option to rename or delete them, or to do nothing.

Context Menu

From the context menu (right mouse button menu) you can choose to rescan files that previously failed or to remove all the files that were successful from the list.

Verify Values

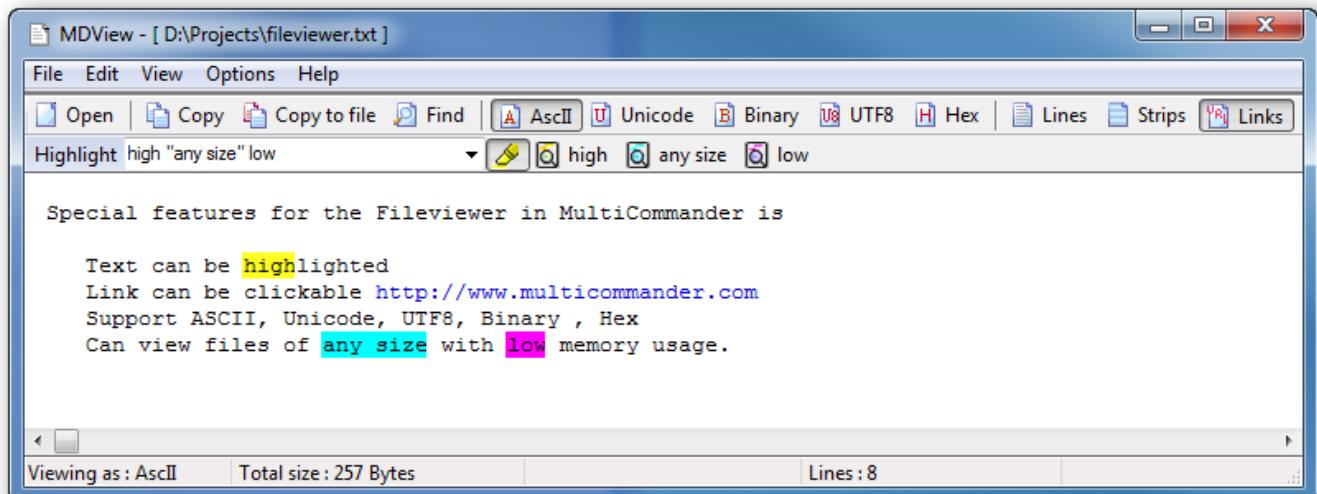
Sometimes you want to verify a file but you do not have a .sfv/.md5 file but you know the checksum values. In this case you can create a new file verification with the New button and select "compare value".

You then copy or drag over the file you want to verify, click in the "expected value" column and fill in the expected value. Finally, press start to begin verifying.

MultaDataViewer - The File Viewer

To view the content of a file, select "**View File**" from the File menu, or press the keyboard shortcut key **F3** (default shortcut key if using the Commander Styled Setup).

The Default FileViewer in MultiCommander is MultaDataViewer. It is a file viewer that has some special unique features.

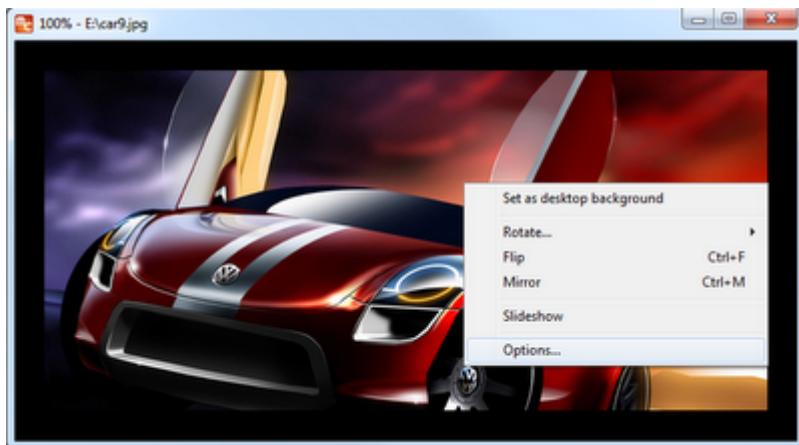


MultaDataViewer Features

MultaDataViewer can:

- **View files of any size**, even if the file is several **gigabytes**, using very little memory.
- View files as **ASCII / Unicode / UTF8 / Binary** and **Hex**.
- Be very **customizable**. **Font / Color / Background / Gridlines / Strip and more** can be changed to different profiles that are loaded for different files based on format or file type.
- Support **syntax coloring**.
- Support **links**. Links in the text are **clickable**, so you can click on the link and a browser will be opened for that location.
- **Highlight** all the words in different colors.
- **AutoSelect** an entire word when double clicking on it (It is possible to customize what special characters are allowed in a word).
- Copy Selections to **clipboard** or save selections into a file.
- **Support Drag&Drop**. Drop a file on it to view that file.
- Provide **MouseWheel Zoom**: Hold **CTRL** down when using the **mouse wheel** and it will **increase / decrease** font size.
- **Queue files to view**. and Go to next/previous file with **CTRL+ PAGEUP / PAGEDOWN**
- **Autoreload** files if changed. (Good when viewing a live log file.)

Multi Commander Internal Picture Viewer



The MC Picture Viewer supports most common picture formats such as *jpg*, *png*, *gif*, *bmp*, *tiff*, *psd* and many raw formats.

If the picture has an *exif* tag with rotation information it will autorotate the picture for you when it is viewed. You can also manually rotate and flip the picture if you wish.

You can also re-save the picture after it has been rotated. JPG pictures will be saved using lossless operations, so that the quality of the picture is not degraded.

It is also possible to set a picture as your *desktop background*, and if multiple files are viewed you can select *slideshow* mode and it will automatically load the next image every couple of seconds.

Fullscreen mode is supported by pressing F11, and you can also *zoom* the picture using the + / - keys or the mouse wheel.

Show Picture Information

EXIF and picture information can be shown on top of picture.

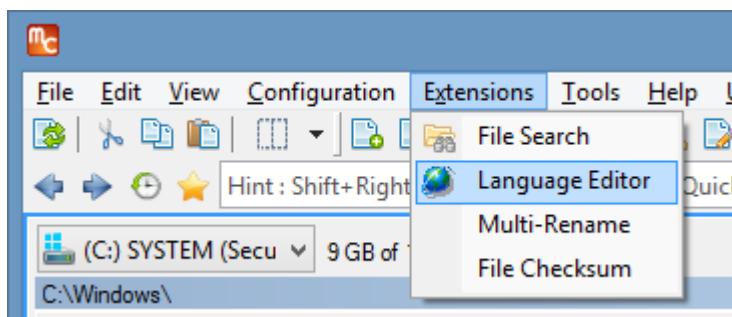


- Pressing I will show Picture Information (Left)
- Pressing E will show the EXIF information (Right)
- Pressing T will show the timestamp information from the EXIF data (Bottom)
- Pressing P will show Phot Information taken from EXIF data like Exposure, Bias, F/Num, ISO

Keys

- ESC - Close Picture Viewer
- Ctrl-R - Rotate 90 degree
- Ctrl-F - Flip image Up/Down
- Ctrl-M - Mirror image Left/Right
- Ctrl-S - Save changes to image
- Ctrl-C - Copy image to the clipboard
- Ctrl-0 - Auto zoom - Stretch to fit
- Ctrl-1 - Zoom to 50%
- Ctrl-2 - Zoom to 100%
- Ctrl-3 - Zoom to 150%
- Ctrl-4 - Zoom to 200%
- + - Zoom In
- - - Zoom Out
- Ctrl+Shift+R - Resize
- Ctrl+Shift+S - Save As...
- Ctrl+Shift+G - Make Greyscale
- Ctrl+Shift+I - Invert colors
- Arrow Keys - Move zoomed in image inside the window
- I - Show Picture Information
- E - Show Exif Information
- T - Show Timestamp
- P - Show Photo Information

Language Editor



The built in language editor will allow anyone to create new language packs or to correct language texts in existing one.

How Language Files Are Used

The language packs are stored in the Languages folder where MultiCommander.exe is located. The language packs is a zip archive per language. Each zip files contains all the language files for all the extension/plugins for that language.

Multi Commander always reads the English language pack first and store all of the language texts in memory. Then the language that was configured is read and it will overwrite all its translated texts. If some text item are not translated then the English text are shown.

The Language Pack only contains languages files for the built in Core and Extension/Plugin that are delivered with Multi Commander. So an Extension/Plugin that is created by somebody their language files will not be found here. If MultiCommander do not find a matching language file in the language packs it will search for the language files in the folder where the extension/plugin is located

Having the language files in a zip archive is not optimal when creating translations or modifying them. It makes it harder to test and verify. And the Built in language editor needs to have them unpacked.

Because of this MultiCommander will at started first look for the unpacked language packs in the "**Languages**" folder. And if the version is a match it will use them. If MultiCommander is newer then the unpacked language files they will not be used.

Tip

Type "**:gouserdata**" in the commandline bar to take you to the Multi Commander UserData folder
Eg "C:\Users\Mathias\AppData\Roaming\MultiCommander\UserData\"

When Starting the Language Editor it will ask if it should unpack the language pack to the userdata folder.

The next time that Multi Commander is started it will now read the unpacked language files from the userdata folder.
Allow modification and addition to be verified

When the unpacked language files in the "userdata\Languages" folder are no longer needed or wanted they can be deleted.
And Multi Commander will then automatically start to load them from the zipped language packs again.

Language Editor Panel

ID /	Text	Description, Tooltip, Caption, ...
- 215	Redigera sökväg	<i>Go to the path label and be able to edit it manually</i>
- 216	Återställ tidigare val	<i>Restore the file/folder selection from before the copy/mc</i>
- 217	Växla sökväg mellan Käll och Mål vyema	<i>Swap the explorer panel location of the source and target</i>
- 218	Lägg till filnamn till kommandoraden	<i>Send the filename the file or folder in focus to the command line</i>
- 219	Lägg till komplett sökväg till kommandoraden	<i>Send the full filename path of the file or folder in focus to the command line</i>
- 223	Show ViewFilter popup menu	
- 224	Show Select popup menu	
- 225	Show Unselect popup menu	
- 230	Ändra Snabb Sökväg	
- 231	Tryck på knappen (0-9) för den snabb sökväg du vill använda	
- 232	Ställ in Snabb Sökväg för CTRL+%	
- 233	Sätt Snabb Sökväg!	

The language editor panel allow text items to be modified.

It is also possible to add support for new language and to export an language pack.

Save Button

The save button will save all the changes for the currently selected language.

Language Properties Button

Show a dialog with some properties for the selected language. like what the language name is and language code it should use.

Type Dropdown

List what extension/plugin type there is in Multi Commander. Selecting one of them and the next dropdown will update with all the extension/plugins for that type.

Each extension/plugin has their own language file.

Extension/Plugin Dropdown

List all extensions/plugins for the selected type. Selecting a extension/plugin here and all text item for it will be shown. It will be shown for the language that is set in the Language Dropdown

Language Dropdown

List all available language and at the bottom there is an option to add a new language.

Changing language here and the list will reload with the text item for the selected extension/plugin for that language

Export/Send Button

The Export/Send button will allow a language pack to be exported or to be sent to MultiCommander staff for inclusion into next release.

Restore Button

The Restore button allows previous version of language files to be restored. Every time you press Save it will backup previous version and you can also restore to a previously exported language pack.

Text Item List

There are three (3) columns in the text items list. ID , Text and Description/Tooltip/Caption/...

The ID is the Text ID used to identify the text item. It is made up of a letter + a number, The letter is a way to categorize different text id. often **M** are used for text item for MenuItem, **D** for Dialog, **E** for error messages, **C** for Commands. The text item in the picture above is under the C, So 231 that are selected is C-231

The Text column is the main text that are shown for that text id.

The Last Column are text item that are alternative. Every text id can have two text items, one main and one secondary. The secondary is often used for the Tooltip (Hover text)

But can sometimes be used for the descriptions text for script command, or for caption text for error message.

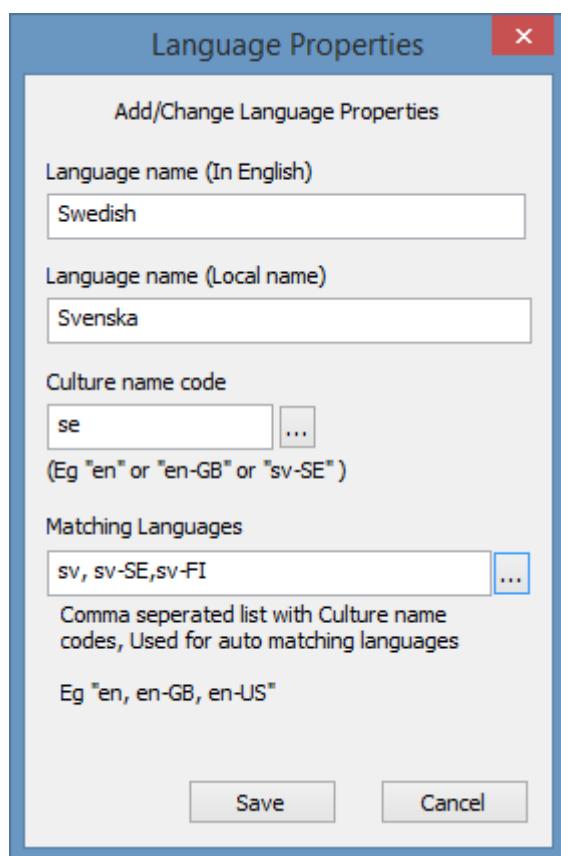
Text items that are in ***Red and Italic*** are text items that are**NOT** translated. It is the core English text that are shown. The red items are NOT saved to the localized language files. The localized language files should not contain any of the English texts. The English text for untranslated text are automatically picked up since the English language file are always read first.

Items that are shown as **BOLD** are text item that are modified but not yet saved.

Clicking on the right mouse button it is possible to select to show only not translated items. Then all the translated items goes way and only the red not translated ones are shown

Create New Language Pack

Start the language editor and from the language dropdown select the last options "-- Add New Language --"
The Language Properties window will be shown



There are two language name field. The first is the language name in English and the second is how the language is spelt in the language that are added.

Under "Culture name code" enter the two or five letter code for that language. If there exists many variations of the language you are adding use the five letter code instead. So that there will be no conflict.

In the "Matching Languages" enter all the language codes that match this language. This is used to auto selected the correct language. So if a user is running with Windows set any of the matching languages that language will be auto picked the first time Multi Commander is run for that users.

Example from the picture above, If a user have Windows set to "sv-FI" it will pick the language pack that are about to be created.

"Matching Languages" can be changed later by pressing the Language Properties button next to the save button

Then press Save to create support for that language. The file UserData\Langauge\SupportedLanguages.xml is now updated and Multi Commander will use this file next time it start.

It is not possible to start translating using the language editor.

Special Characters

Some text item can contain control characters and this must be kept.

The most common is **\n** and this means "new line" and are often used for text that are shown in text block or message boxes.

Make sure there is no space after the control character

Line1 \n Line2 : Wrong "Line2" text will contain now have an extra space character before text "Line2" text. And that can make the output look bad.

Line1\nLine2 : Correct No extra space characters before the "Line2" text

There are also **%s** and **%d** , This control characters mean that another texts or numbers will be inserted at those positions and this control characters must also be kept.

For example in Explorer Panel there is text id "**L-31**" that looks like

```
%s/%s in %d/%d Files and %d/%d Folders selected
```

All the %s and %d control characters will be replaced with texts and numbers when the text item is used.

Another control character is the & character. This is a control character text item that are shown in the menubar or popup menus and they are for Automatic Shortcuts.

Automatic Shortcuts

Menu items and Popup menu can be shown using keyboard, For example if you press the ALT key you see that the "File" on the menu bar is highlighter.

Also note that the file menu now has an underscore under the F and the Edit menu has it under the E and so on.

The menu bar now reads : File Edit View and so on

The underscore under the V in the View menu means that you can now press V to go to that menu. And the menu that then is shown can have similar underscores for many of the menu items making it possible to access them fast using keyboard.

The decision on what letter to use as a shortcut is decided by the menu item text. Any & characters in menu item text will make the next character have a underscore under it and it will be used as a shortcut automatically. For Example "&View" will be shown as "View", and "Vie&w" will be shown as "View"

This is nothing that is unique for Multi Commander, this is standard windows behaviour. For every menu that is shown there should not be duplicated shortcuts. There must not be two or more items that has an underscore under the same letter. Then the shortcut will only work for the first one.

Because of that the & character might need to be moved to be in front of another character in the translated text.

Edit Language Files With External Editor

Some people find it easier to translating with there favourite text editor instead. To do that you first need to do the same step as above when creating a new language.

But since the language files do not contain untranslated text you need copy the English files to the new languages you are creating. This require a bit of manual work.

- Unpacking the english language pack file manually MC_LangPack_en.zip
- rename all of them files to match you new language.
If you new language culture code is "se" as in the example picture above you should rename then *_en.xml part of all the language files to *_se.xml
- Then zip them together again to a file called MC_LangPack_se.zip and place it in the standard language pack folder
- Go to UserData\Languages\ folder and delete everything EXCEPT "**SupportedLanguages.xml**", Do not delete that file since this file is updated with your new language
- Start the Internal Language Editor again and it will unpack all language pack again to UserData\Languages\
- Start editing the new language files located in UserData\Languages\ from you favourite editor

- When done. Start Internal Language Editor and select Export/Send to export or send the language pack

Editing the new language files in the UserData\Languages location is going to make the export function work without any problems. So then you can create a language pack for sending to be included into the next version of MultiCommander

Be aware...

When editing the language files with an external editor be aware that the files are in a format called xml and they are encoded using UTF8. Be careful not to remove xml tags. If xml tags are destroyed the entire xml file will be unreadable. Also special characters might need special encoding, Like & and "

If a language files is not readable by MultiCommander try to open the xml file in a webbrowser like IE. If some xml tags have been lost the browser often can say that it found a xml error on line xx and you can find the error

Submit Language Pack

After modifying the an existing language or created a new a language a new language pack for that language may be wanted. So that it can be moved to another machine easy or sent to the developer of Multi Commander so it get included into the next release. To create an updated language pack first make sure that the Language dropdown is set to the language you want to create a language pack for. Then press then "Export/Send" button.

A window that has a dropdown with three options will be shown. This is for what type of language pack you want to create.

- **Internal format (Submit to developer using e-mail)**

This will create a Language pack in the internal format. This is used if you are sending the Language pack to the developers of Multi Commander via email

- **Internal format (Submit to developer using HTTP) (*recommended way*)**

This will create a Language pack in the internal format and it will then send the language pack to the MultiCommander web service and there is not need for email or anything else.

However using this way you need a Submit key that you get by contacting MultiCommander staff.

A key is needed so that nobody just send in incorrect changes on purpose that will destroys a language packs.

This process is entire automatic so language pack received this way will automatically be included into the next version. If you got a submit key you can submit language changes as often at you like.

- **Export to Public Language Pack Format**

This will export the language pack in the public format. Same format as the language files found in the install folder. This is used if the language pack created should be giving to somebody else or if you want to try it on another machine

Incomplete Translation Work

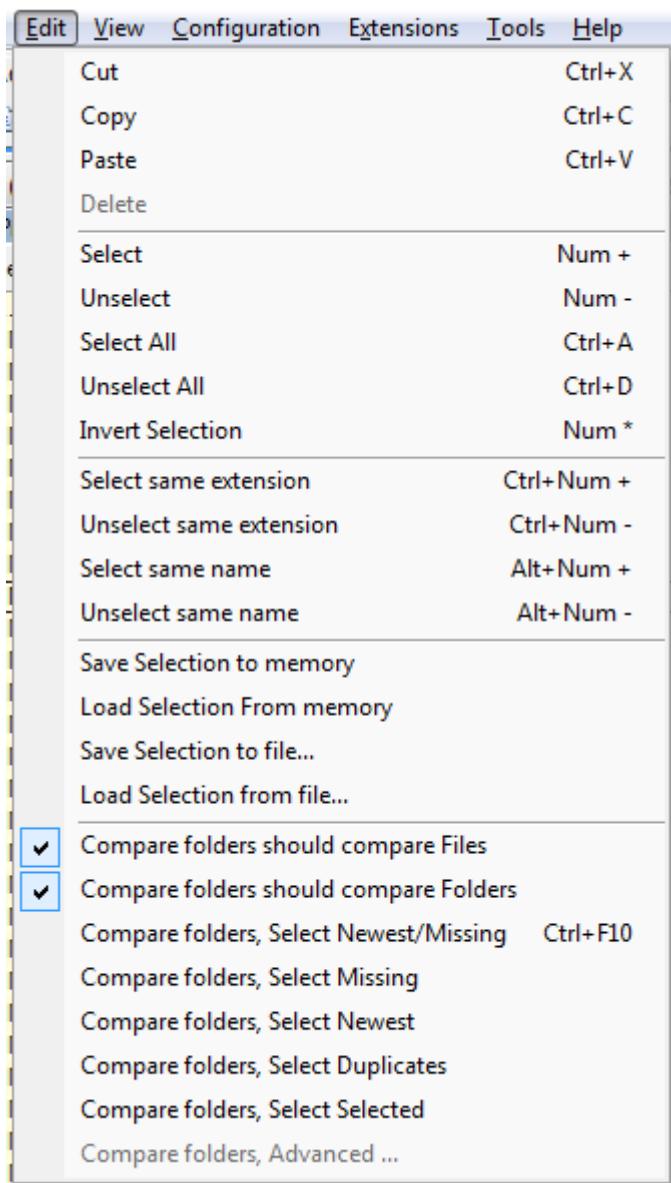
Multi Commander contains a lot of text items and translating them can be a very long process.

So if you start a translation but gives up because it is to much work. Then submit what you have. Somebody else might continue with where you left off.

A partial translations is better then none at all.

Select/Unselect Files and Folders

Under the **Edit** menu there are a number of special commands for selecting files and folders.



Select using filters

The **Select** and **Unselect** options show a select/unselect dialog where you select files using filters. You can enter a **wildcard** filter to select/unselect files, e.g. `*.jpg *.tiff` for all jpg and tiff files.

Save/Load Selection to/from File or Memory

Save selection to file will save all the selected files to a filename you specify. You can then load that file and get back all your selections at any time.

Load selection from file will load a file and treat every line in that file as a selection filter. Each line can be an exact match for a file/folder that you want to select or it can be a selection filter containing wildcards.

Save/Load to/from memory options work the same way but will not survive a restart of MultiCommander.

Compare Folders

Compare Folders will compare the paths in the source and target panel and select differences depending on which compare command is called

It is possible to enable or disable whether the compare command should work on only files or only folders.

Compare Folders, Select Newest/Missing

Will select the files that are missing in the other panel, or if folders/files are found in both the newest file/folder is selected

Compare Folders, Select Missing

Will select the files that are missing from the other panel.

Compare Folders, Select Newest

Will find files/folder that exist in both panels and select only the newest of them.

Compare Folders, Select Duplicates

Will select only files/folders that exist in both panels.

Compare Folders, Select Selected

The files that are currently selected in the source panel will also be selected in the target panel if they exist.

Calculate Folder Size

To determine the size of folders, you can do any of the following:

1. Select "Calculate folder size" under the Tool menu.
2. Press the "Calculate folder size" button in the File Commands toolbar.
3. Press SHIFT + ALT + RETURN.

Name	Ext	Size	Date	Attrib
←...			2010-02-27 15:49:06	---d
WTLEditor		64 319	2010-02-27 15:49:06	---d
Wizard97Test		288 744	2010-02-27 15:49:06	---d
TabBrowser		117 329	2010-02-27 15:49:06	---d
SPControls		66 009	2010-02-27 15:49:06	---d
MTPad		133 071	2010-02-27 15:49:06	---d
MiniPie		64 043	2010-02-27 15:49:06	---d
MDIDoc\Vw		55 935	2010-02-27 15:49:06	---d
ImageView		162 036	2010-02-27 15:49:06	---d
GuidGen		90 257	2010-02-27 15:49:06	---d
BmpView		197 633	2010-02-27 15:49:06	---d
Alpha		42 931	2010-02-27 15:49:06	---d
Aero		40 536	2010-02-27 15:49:05	---d

Size calculation will be performed on all the selected folders, if no folders are selected then the size of all folders will be calculated.

Selecting folders with the **space-key** will automatically calculate the size of those folders.

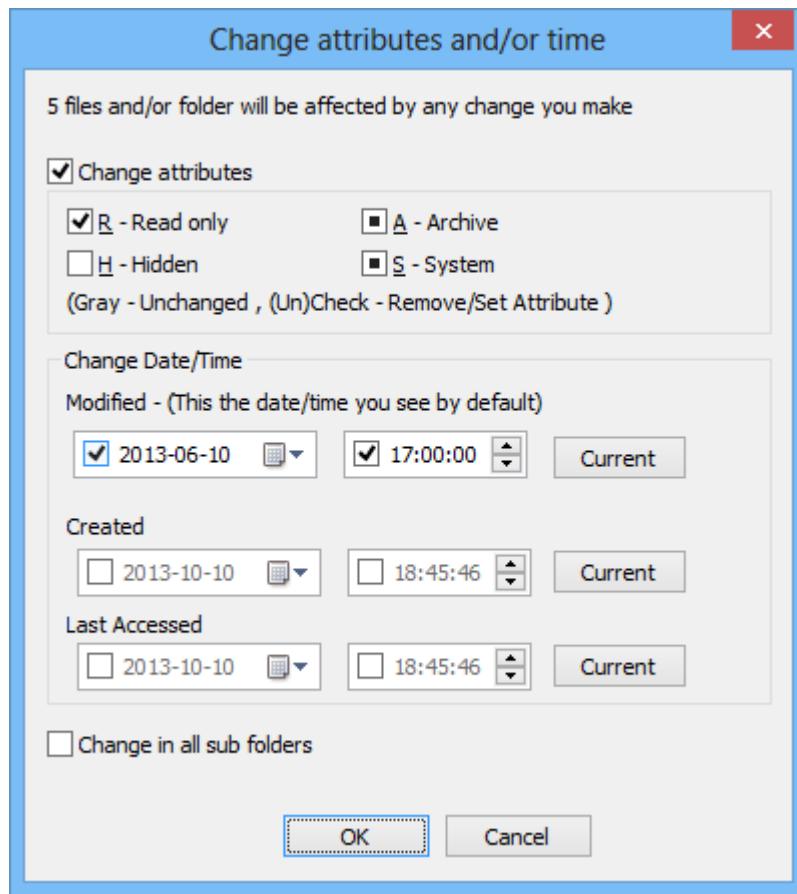
The size calculation is done in the background and can be aborted by pressing the **ESC** key.

If a folder's size has been calculated, and files are **deleted** from that folder from within MultiCommander, the folder size will automatically be updated.

The default shortcut key for calculating folder sizes is **Shift+Alt+Return**.

Change File Properties

Attributes and Date/Times of files and folders can be changed from the Change Properties dialog.



Select the files or folders whose properties you want to change, then select "Change Properties" under the Tools menu.

Multiple files and folders can be changed at once. It is also possible to change all the files in folders and sub-folders.

The attribute checkboxes can have one of 3 states:

If **Unchecked**, the attribute will be turned off.

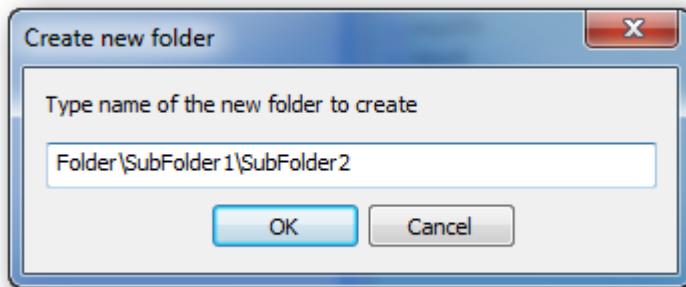
If **Checked**, the attribute will be turned on.

If **Blocked** (as seen in image above), the attribute will not be changed, the current attribute will be kept.

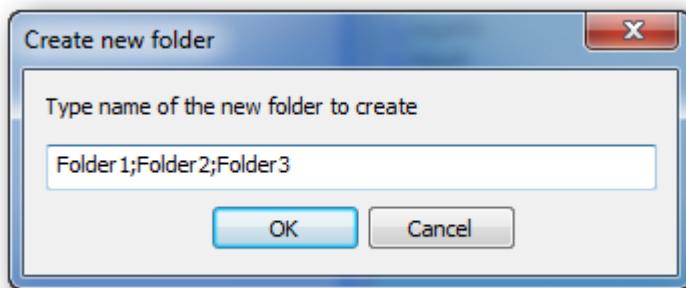
If you hold down **CTRL** and **Double Click** in the date or attribute column for a file/folder, the Change Properties window will be shown. But only for that file/folder, even if multiple files are selected.

Create Folders

Using the **Create Folder** command found in the Tools menu, you can create new folders at the current path in the Explorer Panel.



The folder specified can be a complete folder path. Intermediate sub-folders will be created as needed.



You can also create multiple folders at once as long as you separate the folder names with ";".

Shortcuts

The default shortcut key for Create Folder is F7.

Shift+F7 will insert the text from the clipboard into the folder name field when the dialog is shown.

When the Create Folder Dialog is shown there are a couple of actions that can be done

F5 - Inserts the text from the clipboard into the folder name field.

F8 - Removes all dots in the text name field.

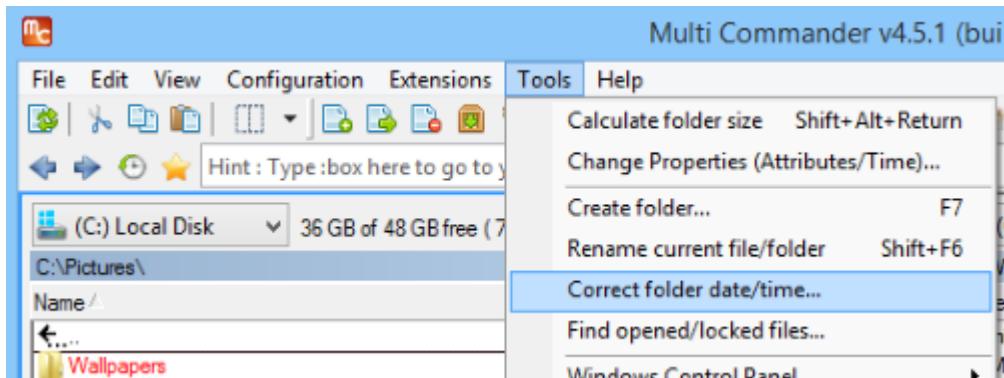
F9 - Insert todays date as YYYY-MM-DD in the folder name field.

Shift+F9 - Append todays date as YYYY-MM-DD in the folder name field

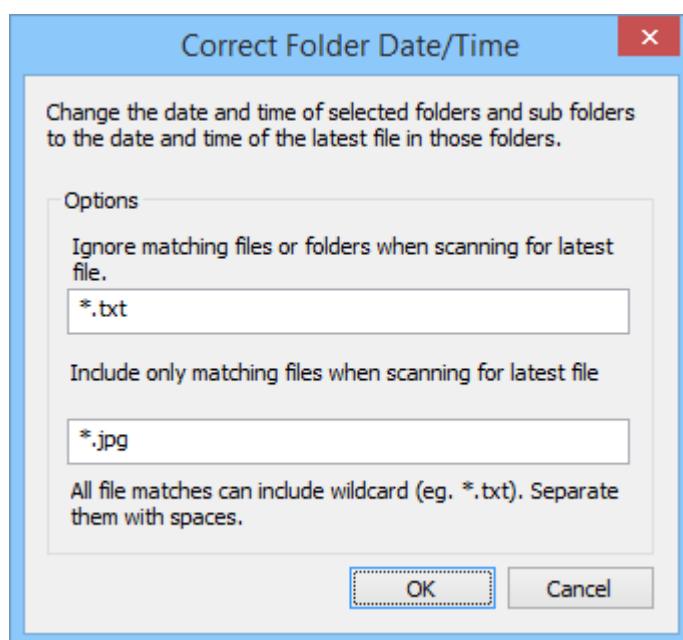
Correct Folder Date/Time

The date/time of a folder is changed whenever you copy a file into the folder, delete a file, or change a file. In some situations this is not what you want, because you might list and sort those folder by date, and you want the date/time of the folder to show the date/time of the most recent file in the folder rather than when something was last changed in that folder.

This tool will scan the selected folder and subfolders and set the date/time of the folders to that of the most recent file it finds.



Select the folder you want to change then select "Correct folder date/time" from the menu.



You can specify two kind of filters for the scanning. Ignore and Include filters:

Ignore Filter

The first filter is the *ignore* filter. It specifies the files and folders that you want to ignore when finding the most recent date/time.

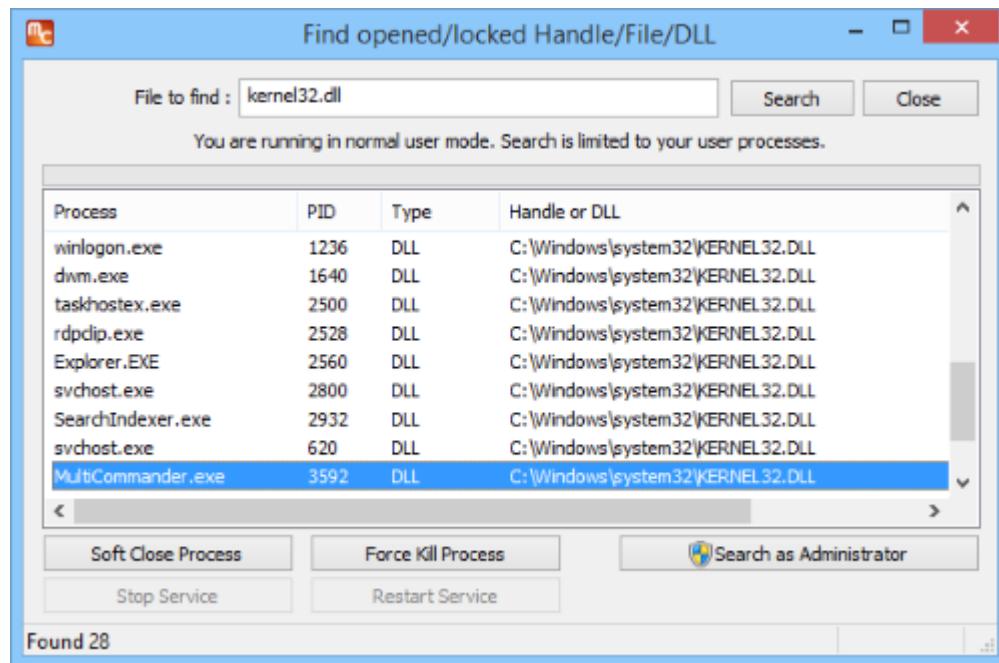
If you enter *.txt here. It will ignore all *.txt files, even if they have a date/time that is more recent then other files.

Include Filter

If you specify an *include* filter then only files matching this filter will be checked. E.g. if you enter *.jpg here it will only check the date/time on *.jpg files.

Find Opened/Locked Files

If you are unable to rename or delete a file or folder it might be because it is already in use by another program. A program that uses something in a sub-folder can also lock up the parent folder so that cannot be changed. Which program has a particular file open is often not obvious, and it can be hard to figure out what program you need to close to be able to rename or delete the file or folder.

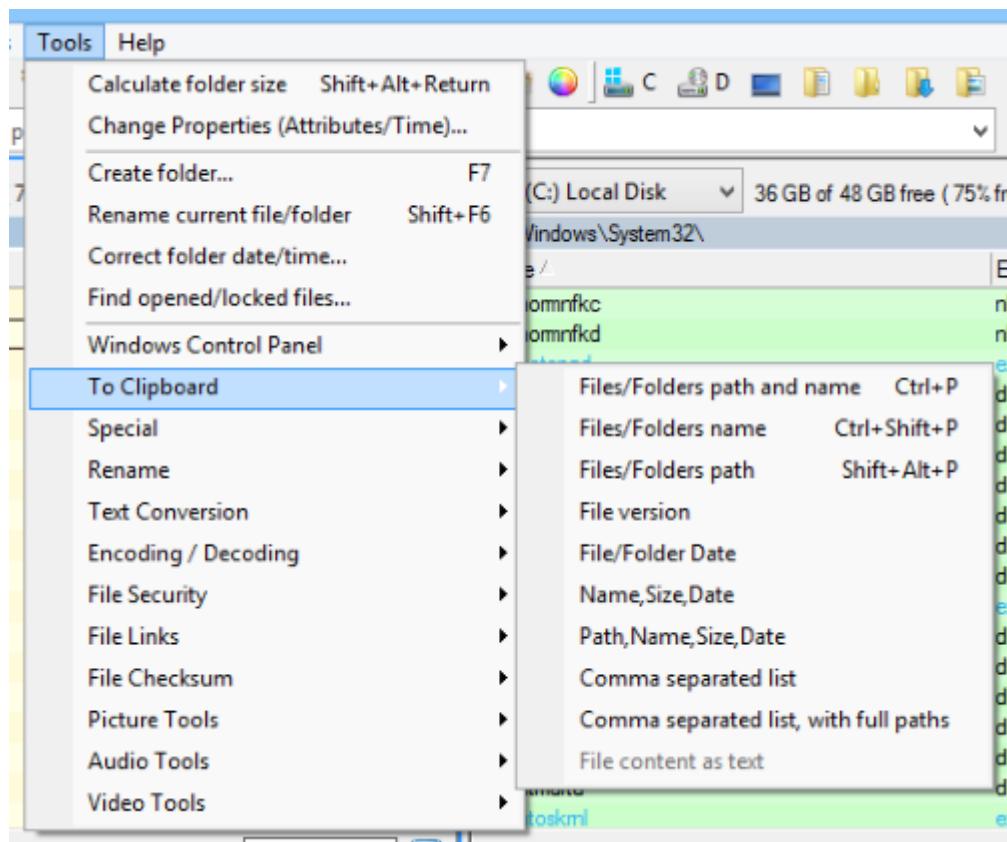


Under the Tools menu you can select "**Find Opened/Locked Files**" to find what program has a file/folder locked so that you can close or kill it. If you are running with standard user permission you will only be able to scan the system for processes that are running under your account; system & service processes will not be found.

Running Multi Commander as administrator will allow you to also find services and system processes that have files open or locked.

Send to Clipboard

Selecting **To Clipboard** under the *Tools* Menu brings up a submenu that allows you to copy various file properties directly to the clipboard so that you then can paste that information wherever you wish.



Properties for all the selected files/folders will be sent to the clipboard. If no files/folders are selected the file/folder in focus will be used.

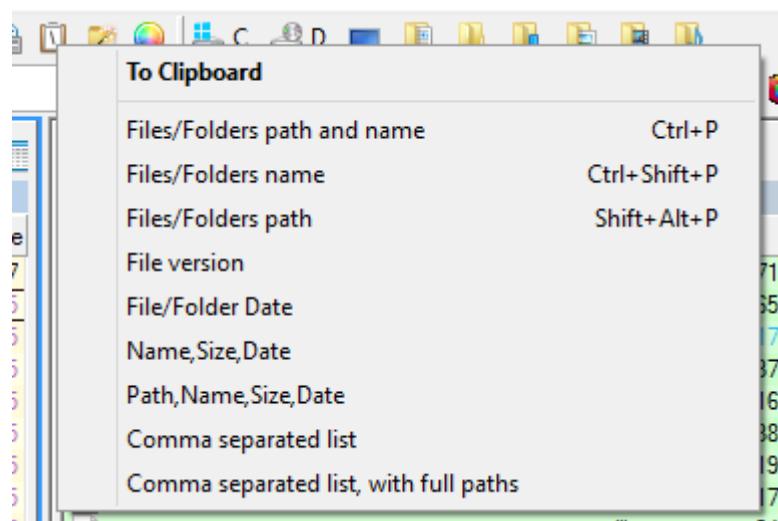
What properties to send depends on what command you select. The most commonly used are the full path including filename or just the filename of the selected files or folders.

Shortcuts

The default keyboard shortcut for copying the complete path is **Ctrl+P**
for copying only the file/folder name it is **Shift+Ctrl+P**

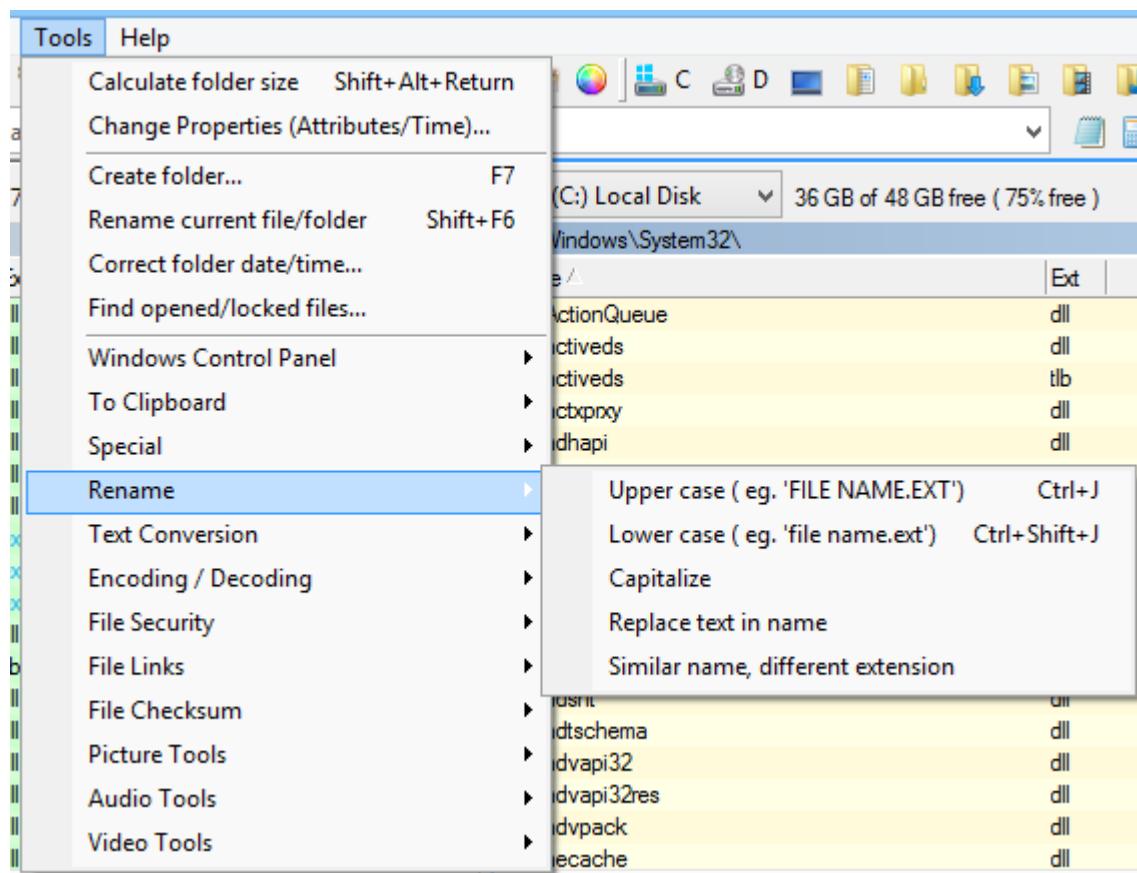
Send to Clipboard is also available on the toolbar. And if it is click the complete path of selected files are copied to the clipboard.

If the right mouse button is pressed on the toolbar button a popup menu with the other options is shown



Rename Tools

Under "Rename" in the Tools menu there are a number of special rename operations that can be performed.



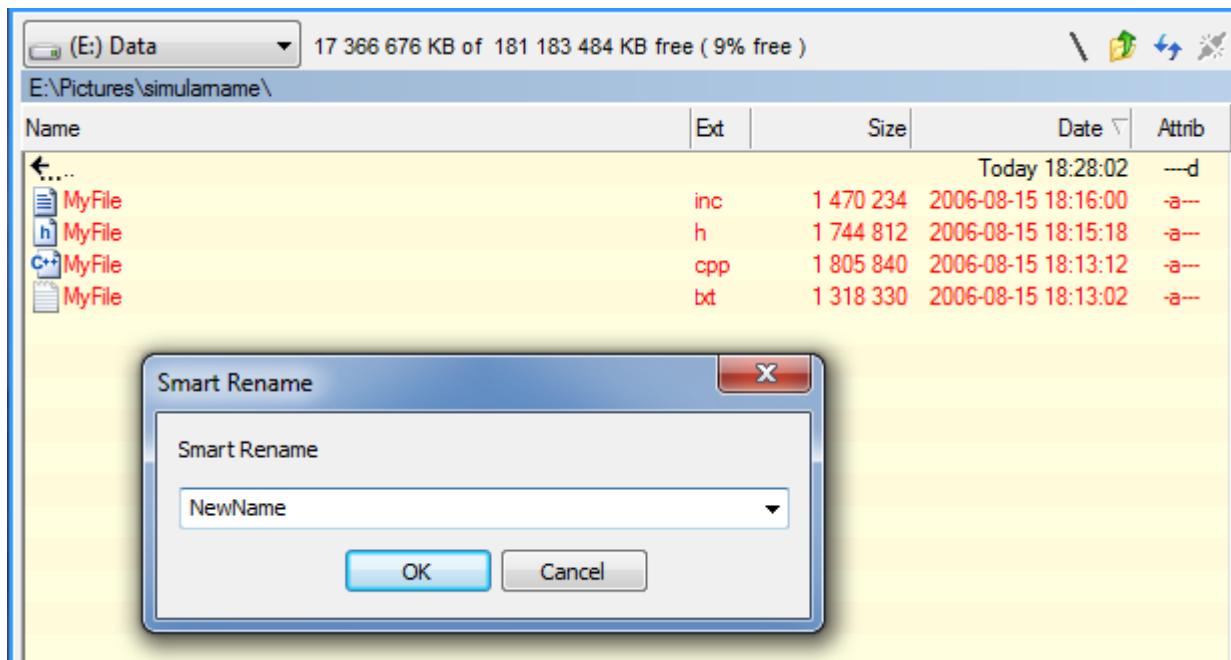
The Rename commands are performed on all selected files and folders.

Uppercase / Lowercase renames the selected files/folders to the same name but with all uppercase or lowercase letters.

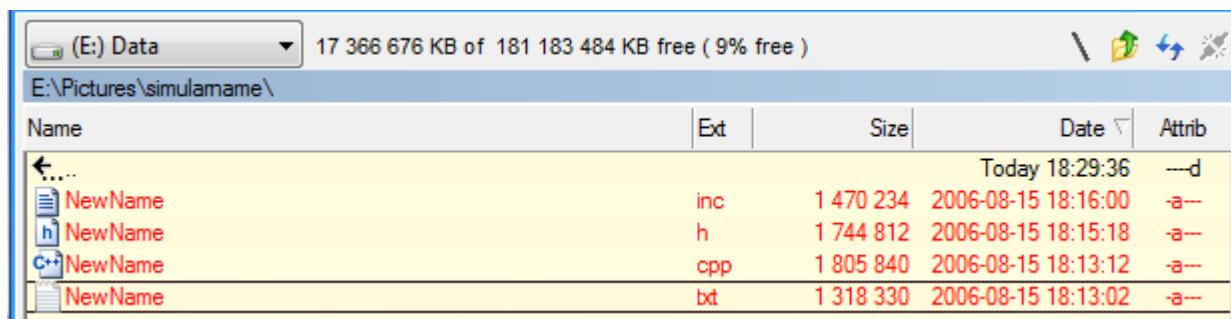
Capitalize renames all selected files and folders to lowercase letters with the first letter in each word uppercase. New words are defined as having letters that are preceded by space, dot or underscore.

Replace text in name will replace a substring in all the selected file/folder names with other text.

Similar name, different extension is a special rename operation that will only rename selected files (*not* folders).

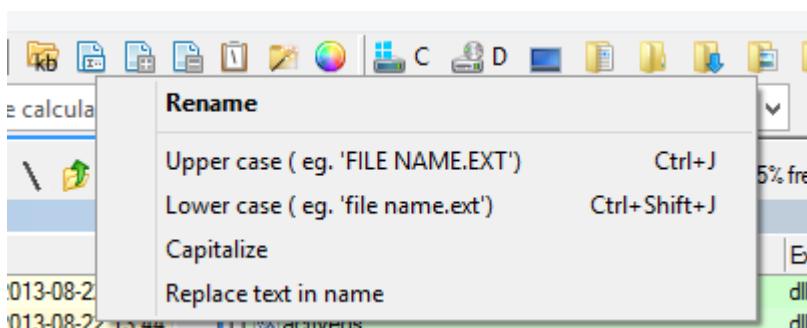


All the selected files should have the same name but with different file extensions.



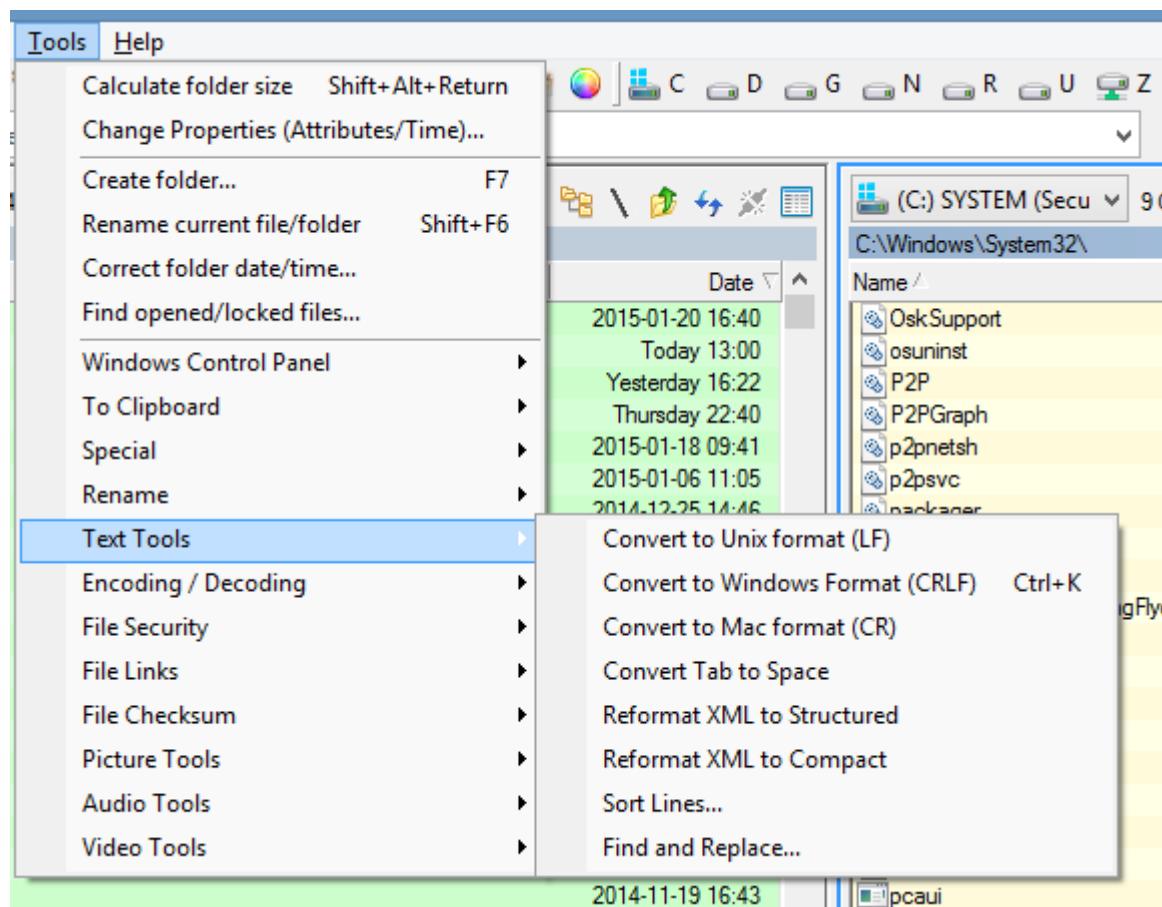
The selected files will be renamed with the new name, but the file extensions are left unchanged.

Some of the tools are also available when right clicking on the Rename button on the toolbar



Text Tools

Under the Tools menu you can find some special **Text Tools**.



You find tools for converting the line endings of text files to different platforms formats.

There is also a tool for converting tab character to spaces.

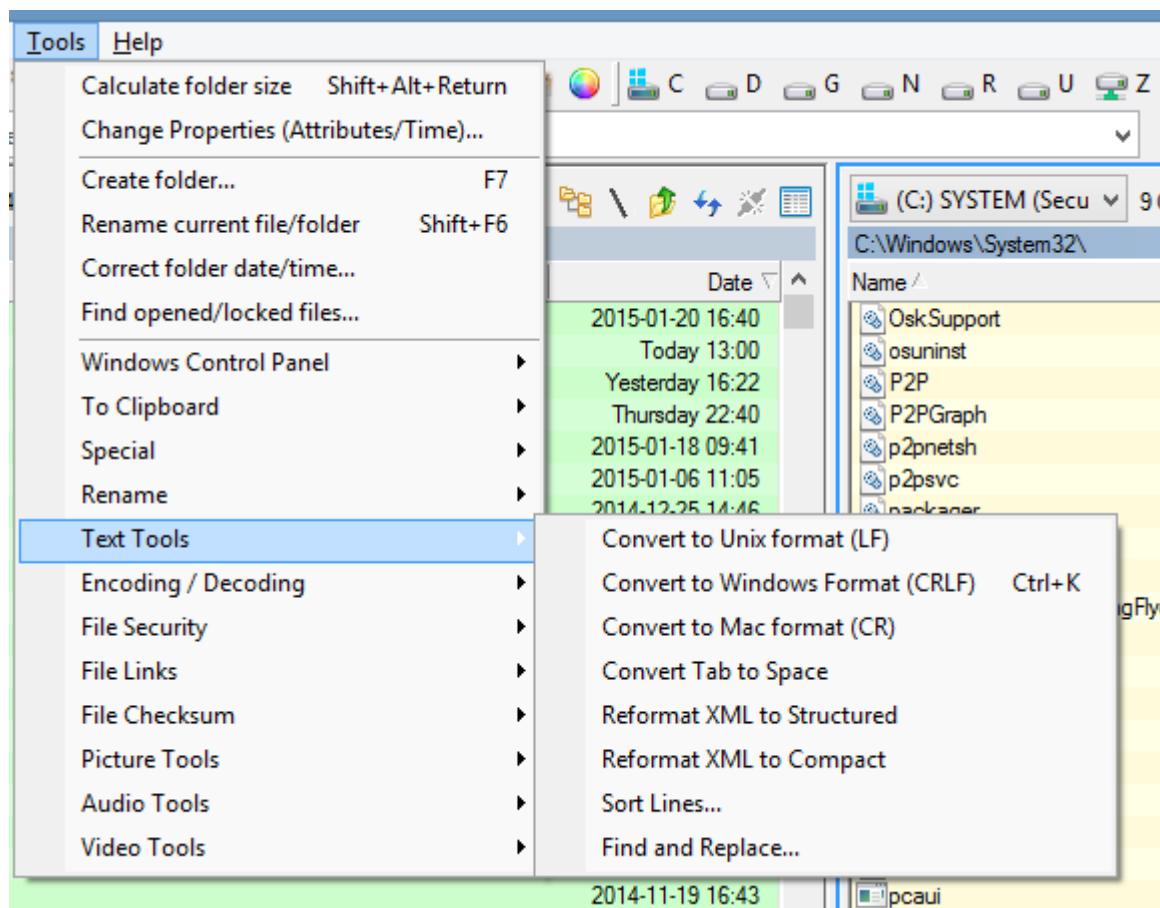
And you can also reformat XML files to structured layout or compact layout.

You find more information about them under [Text Conversion Tools](#)

[SortLines](#) will allow sort the lines in text files. It have many advanced options.

[Find and Replace](#) will allow you find and replace text parts in text files. It support MultiLine find/replace matches and many other options.

Convert Tools



Convert Text Files

The contents of all selected files will be converted. The following options are available in the submenu:

Convert To Unix Format (LF)

Convert text files to Unix format, making sure that newline is only a Linefeed (LF) character.

Convert To Windows Format (CRLF)

Convert text files to Windows format, making sure that newline is both Carriage Return(CR) and Linefeed (LF) characters.

Convert To MAC OS Format (CR)

Convert text files to MAC OS format, making sure that newline is only a Carriage Return (CR) character.

Convert Tab to Space

Replace all tab characters with a specified number of space characters.

Reformat XML to Structured

Reformat xml files so they are more readable, by including indentation after the xml items.

Reformat XML to Compact

Reformat xml files so they are compact, by removing all unnecessary newlines and indentations.

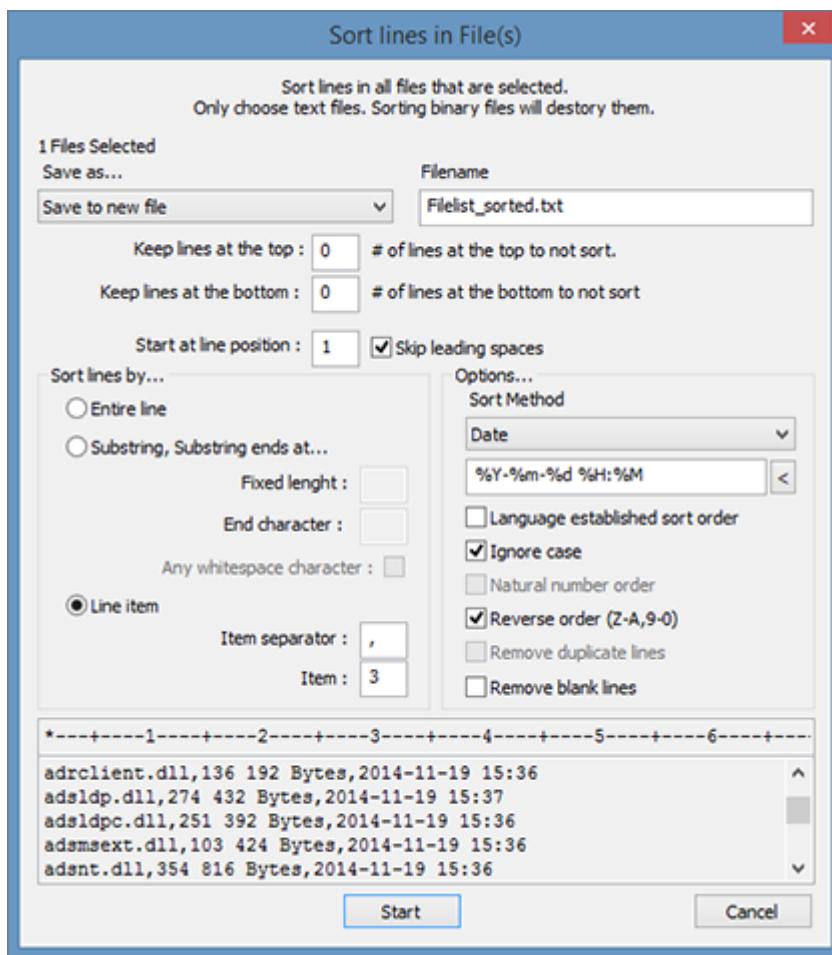
Sort Lines..

Sort lines in file. Read more about this at [Sort Lines](#)

Find and Replace...

Find and Replace text in files. Rad more about that at [Find and Replace](#)

Sort Lines in Files



Sort lines will sort all the line in a text file. There are many options to set that will control how the sorting should be done.

SortLine will sort the line of all the selected files. The number of files it will try to sort is shown in the dialog.

Sort lines will automatically identify file formats. Files encoded in ASCII/Unicode/UTF8 is supported. If a file is identified as binary it will be ignored.

The new line character that the file is using is kept, So if a file has CR (carriage return) as new line character the new file will also have CR.

In the file name field you will see what the name of the new file will be. Files will always be saved in the same folder. Only filename can be set

If multiple files are select you will see a "*" in the file name field. the * represent the original file name and file extension. If you have two files select named "MyFile1.txt" and "MyFile2.txt" and the file name field say "*_sorted.*" the new file names will be "MyFile1_sorted.txt" and "MyFile2_sorted.txt".

In the save as dropdown list you can select if it should save the new file as a different name, Of if it should keep the original name but keep a the original files as a backup by renaming by adding .bak to do. Or if it should delete the original file.

Lines and Position limits

Keep lines at the top

This define how many line at the top that should not be touch. If you got a header in the text that is 2 lines you can set this to "2" then the new sorted file will still have that header on top.

Keep lines at the bottom

This define how many lines at the bottom that should not be moved. If you got a footer text that you always want to keep at the bottom you can use this option to make sure that the lines at the bottom is not moved.

Start at line position

Where on the line the sorting should start at. If the lines should not be sorted by the beginning of the line. you can here control at what position it should sort all the lines after. By changing this settings you will also see that the "*" character in the ruler in the preview below will move. So you can there see where in the preview text the sorting will start at.

Skip Leading spaces

By checking this option all leading white spaces will be ignored when comparing lines during search.

Sort lines by

There are 3 different way to sort lines. Entire line, By Substring, or by Item (record)

Entire line

This will compare the entire line from the defined start position to the end.

Substring, Substring ends at

This can be used if sorting should be based on a specific substring in a text.

Use the option above "Start at line position" to set the begining of the substring.

Then use any of the following settings to set where the end of the string is.

Fixed length

If the substring that the sorting should be based on has a fixed length you can defined it here.

End character

If the substring that the sorting should be based on should end when it comes to a set character

Any white space character

If the substring that the sorting should be based on ends when any whitespace character is found.

Line Item

If every line has items or records that are separate by a special character you can defined what item/record it should do the sorting by

Item separator

What character will separate all the items/records from each other. (Eg for a comma separated list, that would be comma ",")

Item

What item on the line the sorting should be based one. First item is 1, second is 2. and so on.

Options

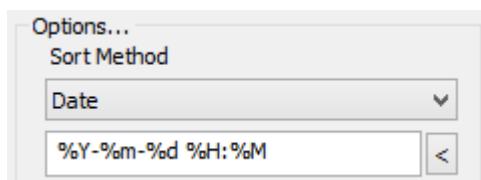
Sort Method

There are three methods to sort by. Text, Number and Date

Text

The text that is compared for sorting is normal text.

Date



The text that are going to be compared for sorting are actually date or time written as text.

To be able to compare date/time correctly the correct date/time formatting of the date/time need to be provided.

If the date/time matching format is not correct the sorting will not be correct.

Press the "<" button under the dropdown list to get sample of date/time matching formats to use

[Date Format](#) - [Time Format](#)

Number

The text that are being compare are actually numbers. And the number will be compared as a digital value and not as text. So leading zeros (0) are ignored. Only digits are used.

Language established sort order

Use the rule for sorting that the current language has. Some language sort some characters in different order. by checking this option the sorting will be done using the current language settings. This sorting is slower and can affect performance for very large files.

This options will disable the "Ignore case" option. Language established sort order always ignore case.

Ignore case

Ignore case when sorting. "A" and "a" will be treated as the same character.

Natural number order

Sort numbers in a natural order. This option also require the option "Language established sort order" So instead of being sorted in the order "Text1", "Text10", "Text2" it will be "Text1", "Text2", "Text10" (This option is is not used if sort method is numbers)

Reverse order

Reverse the sort order. By default sorting will be A to Z and 0 to 9. Enabling this and it will be Z to A, and 9 to 0

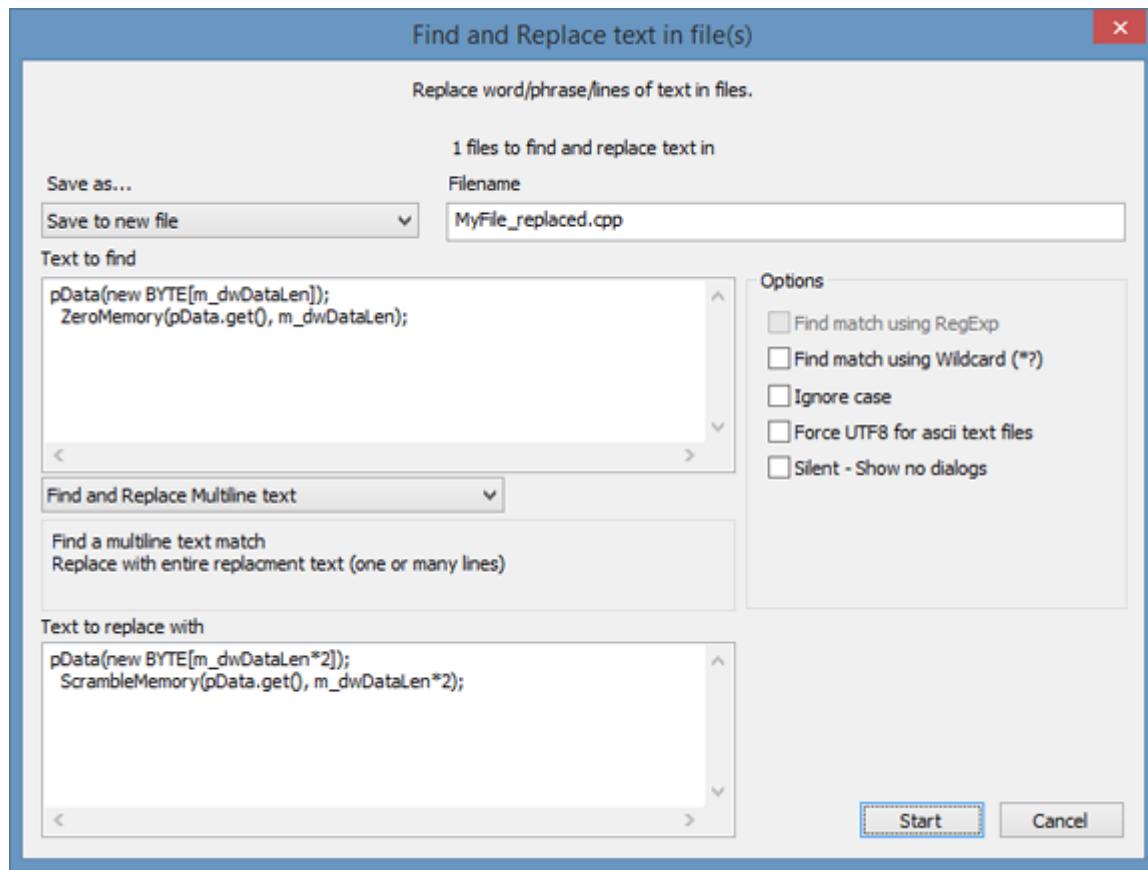
Remove duplicate lines

This option is not yet enabled

Remove blank lines

After sorting the line remove all blank lines.

Find and Replace in files



Find and Replace will find one or many text parts and replace them with one or many new text parts.

Find and Replace will search and replace in all the selected files. The number of files it will work on is shown in the dialog.

It will automatically identify file formats. Files encoded in ASCII/Unicode/UTF8 is supported. If a file is identified as binary it will be ignored.

But the identification of Binary files can never be 100% correct so be careful and do not try to do find and replace of binary files

The newline character that the file is using will be kept. And if doing Multiline find/replace the new line inserted will use the same newline type as the file already has.

In the file name field you will see what the name of the new file will be. Files will always be saved in the same folder. Only filename can be set

If multiple files are select you will see a "*" in the file name field. the * represent the original file name and file extension. If you have two files select named "MyFile1.txt" and "MyFile2.txt" and the file name field say "*_replaced.*" the new file names will be "MyFile1_replaced.txt" and "MyFile2_replaced.txt".

In the save as dropdown list you can select if it should save the new file as a different name, Of if it should keep the original name but keep a the original files as a backup by renaming by adding .bak to do. Or if it should delete the original file.

Text to find

In the field the text part(s) that should be found is entered.

Exactly how the find will work depends on what find and replace mode is selected.

Text to replace with

This fields contains the text(s) that the found text parts will be replaced with.

Exactly how the replace will work depends on what find and replace mode is selected

Find and Replace Modes

There are four (4) modes to select from

Find one, Replace with one

Only first line from the Find and Replace fields are used.

In will replace all instances found that matches the first line in the find field and replaces that with the first line in the replace field.

Find many , Replace with many

Every line in the find field will use the matching line in the replace field when doing Find and Replace.

Example : Line 3 in the find field with use line 3 in the replace field when doing find and replace.

Find Many, Replace with same

There can be many lines set in the find field. All of this matches will be replaced with the first line from the replace field.

Find and Replace Multiline text

Only one (1) find and replace match can be entered. But the find and replace texts entered support Multiline.

You can enter a find match that stretches over multiple lines and if a match is found the entire multi line text entered in the replace field is inserted.

The Newline character the file has when entering a find match does not matter. And the Replace text will be inserted with the same newline characters that the file already have.

Options

Find match using RegExp

Finding text using regular expression is not yet enabled.

Find match using Wildcard

Enable wildcard matching in the find field. * and ? can be used to match text.

Using * at the beginning and end of the find expression can give match that you did not expect so make sure you test so it is correct before overwriting original file

Ignore case

Ignore case when finding matches

Force UTF8 for ascii text files

Enable this if you want to use UTF8 encoding for all files identified as ASCII.

File encoding is auto detected. But sometimes it is not possible detect if a file is ASCII or UTF8. This option will force the usage of UTF8

Silent - Show no dialogs

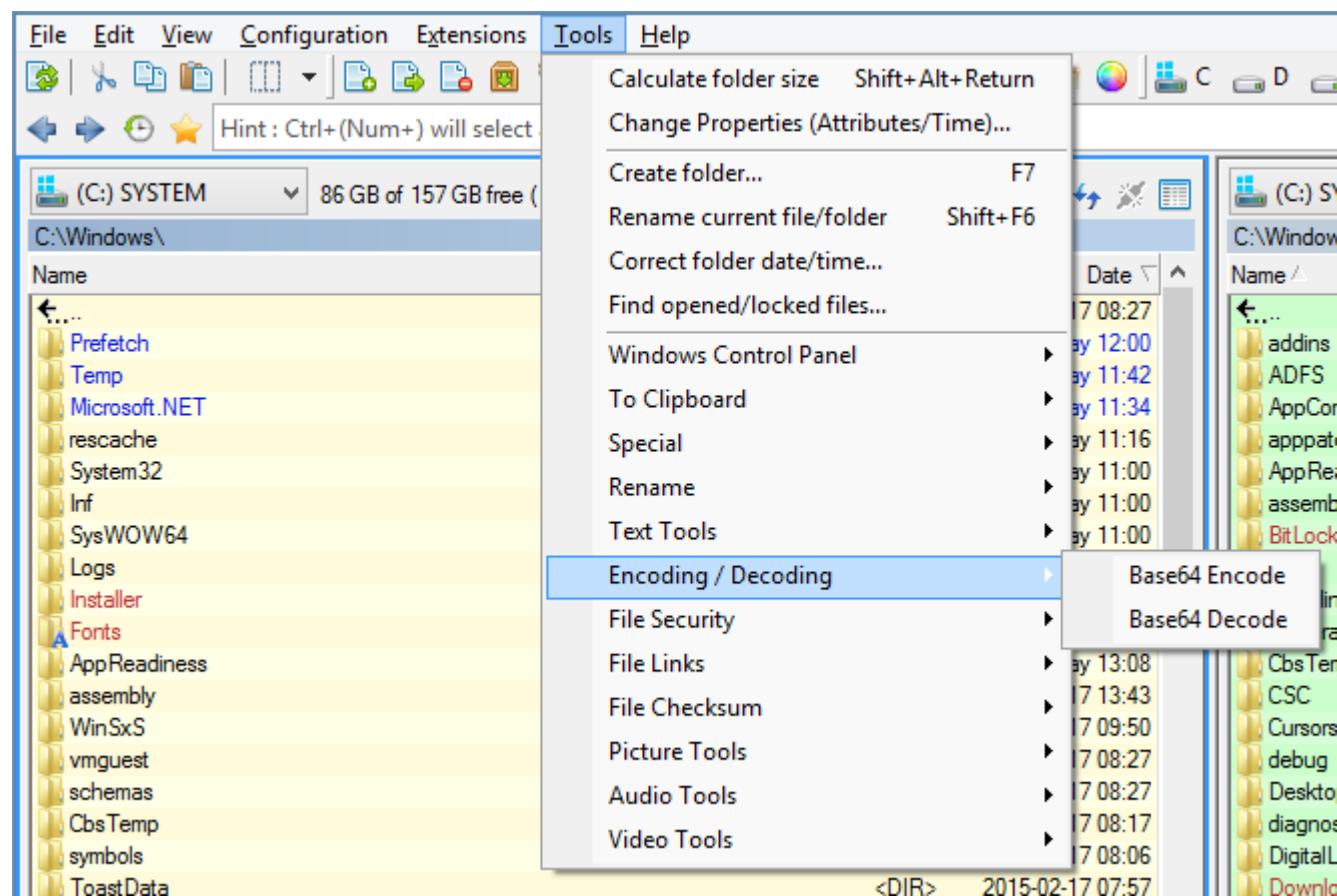
When enabled there will be no confirmation dialogs shown for every match it finds. "Replace All" is assumed. And no error dialog will be shown.

Errors will be logged to the application log.

Base 64 - Encoding and Decoding

The Encoding and Decoding commands are found under the Tools menu.

There are commands to encode or decode all the selected files into/from BASE64 format.

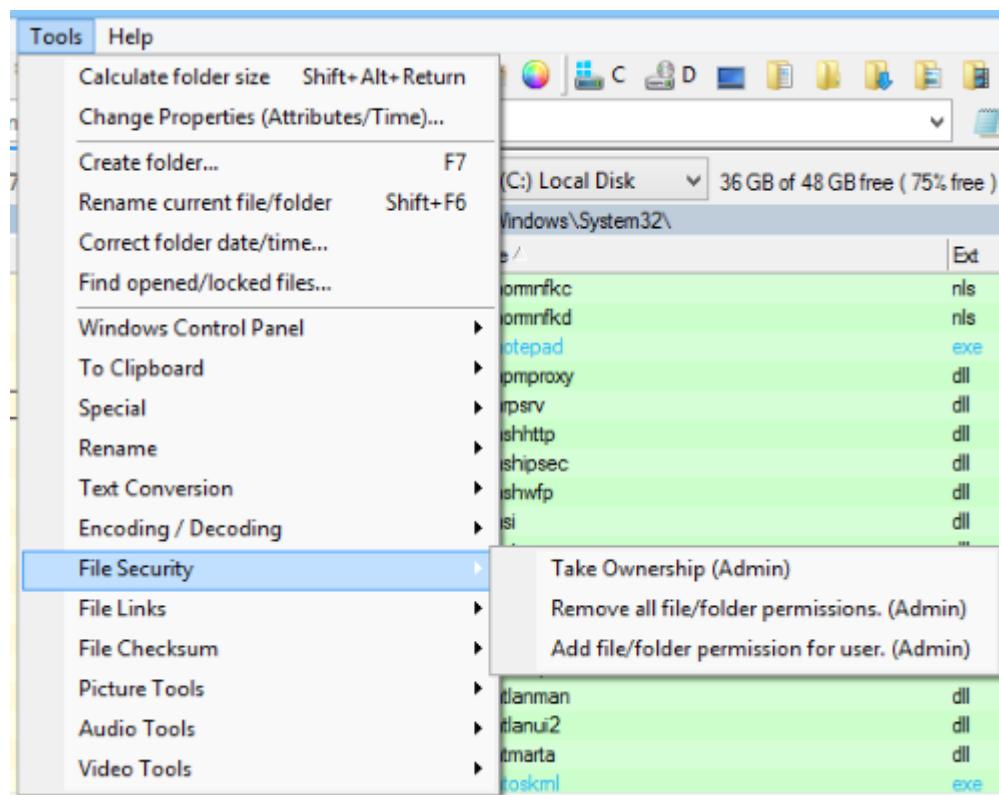


Encoding a file into base64 format is done if you want to convert a binary file into a format that is text friendly. Base64 encoded files grow about 25-50% in size from their binary format.

Decoding a base64 file will turn the file back into its original form.

Read more about base64 at wikipedia [here](#).

File Security



Under the *Tools* menu is a **File Security** entry that lets you change permissions for files and folders. This function requires administrator permission to operate, so you might need to run MultiCommander as administrator.

The File Security submenu provides the following options:

Take Ownership

This lets you become the owner of the selected files and folders.

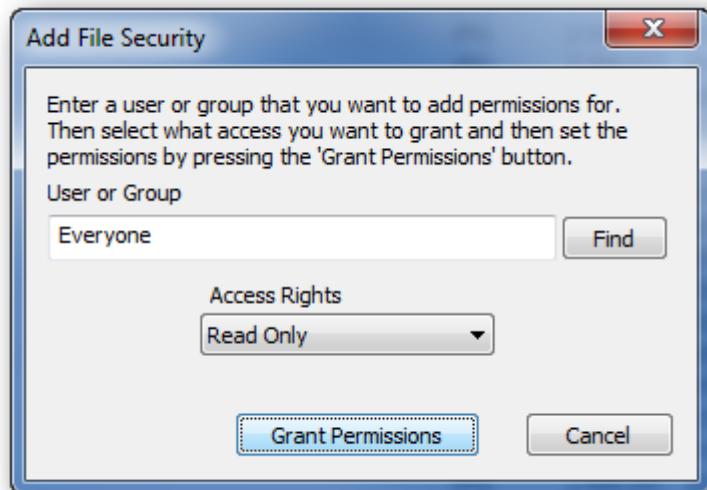
Remove all permissions

This removes permissions for all users and groups to the selected files and folders. No one will have access to the files afterwards. Use the *Add Permissions* command to add permission to yourself and/or others if needed.

Add Permissions

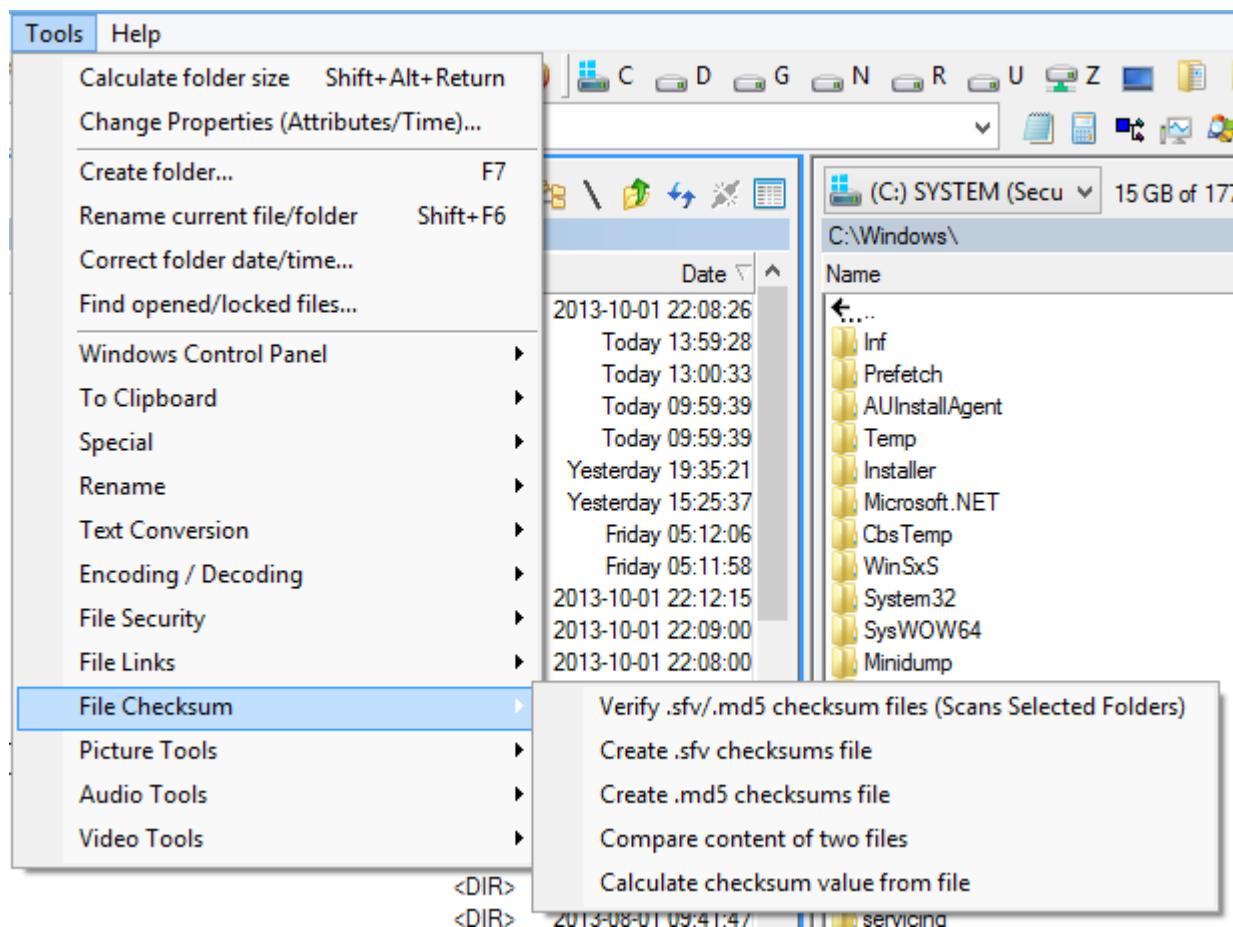
This adds permissions for users to the selected files and folders. You can easily add either read-only access or read/write or full-control.

If you want to give everyone access to a file/folder you can add read/write or full-control access to the group **Everyone** for the selected files/folders.



File Checksum Tools

The FileChecksum tools are quick commands for the [File CheckSum extension](#)



Verify .sfv/.md5 Checksum files (Scans selected folders)

This will scan all selected folder recursively for .sfv/.md5 check sum files. It will then start the FileChecksum Extension in the opposite panel and add all the found .sfv/.md5 files to that. And then it will automatically start a verify process of the added checksum files.

Create .SFV checksum file

This will start the FileChecksum extension in the opposite panel and automatically add all the selected files to a new file checksum of .SFV (CRC32) type.

Create .MD5 checksum file

This will start the FileChecksum extension in the opposite panel and automatically add all the selected files to a new file checksum of MD5 type.

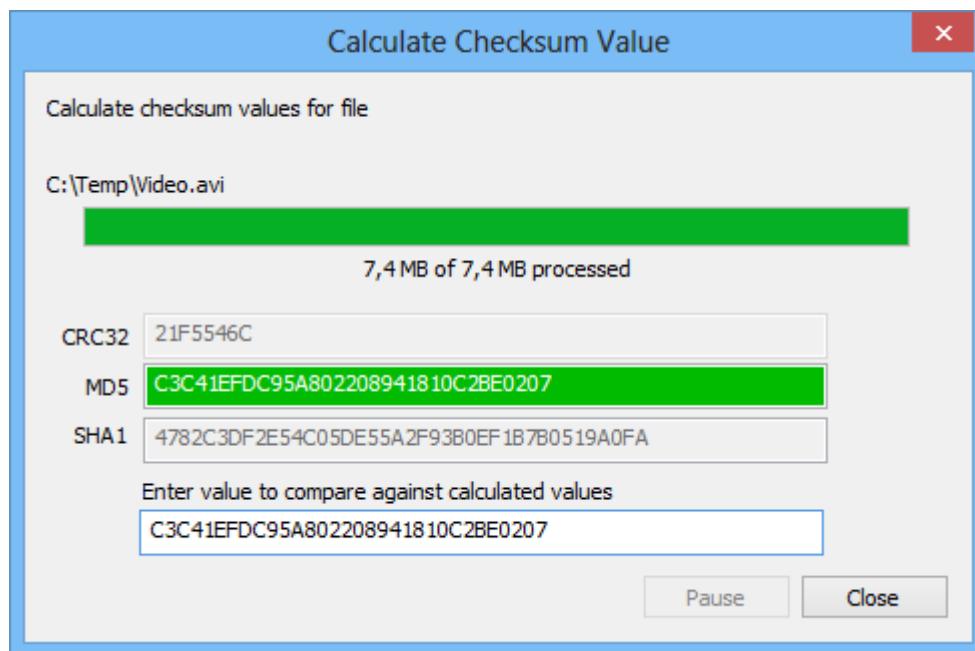
Compare content of two files

This will compare the content of two by calculating checksums values. It will take the first selected from the source panel and the first selected file from the target panel and compare them

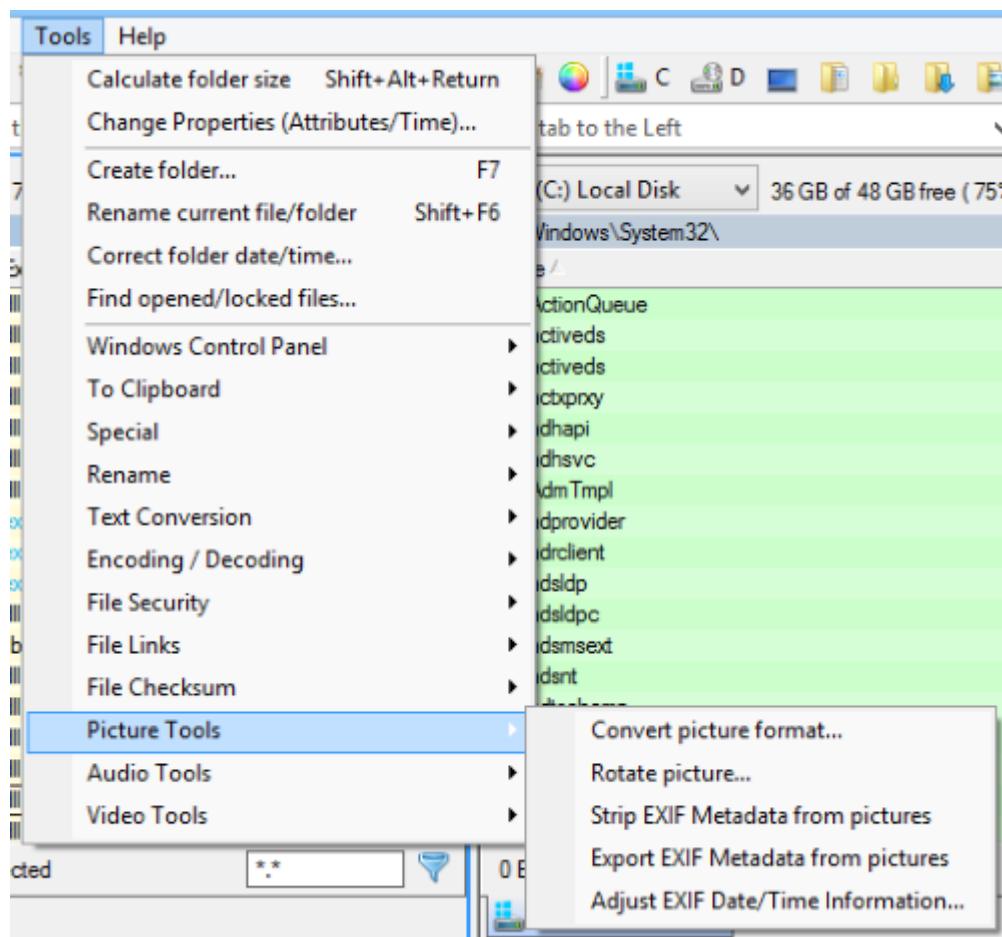
Calculate checksum value from file

This will calculate the checksum values for the selected file.

It will calculate CRC32,MD5 and SHA1 values. and if you enter a checksum value in the input field it will compare that value against the calcuated values and if a match is found it will turn green.



Picture Tools



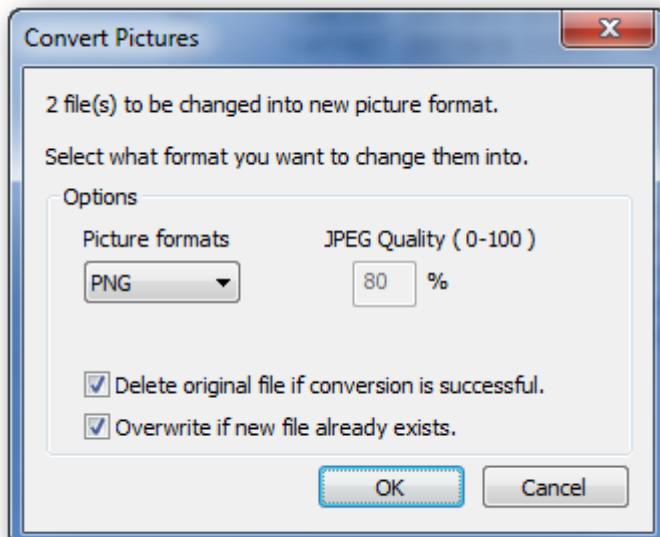
Under the **Tools** menu is a **Picture Tools** option that opens a submenu containing the following choices:

Convert Picture Format...

This converts the selected pictures into a specified format.

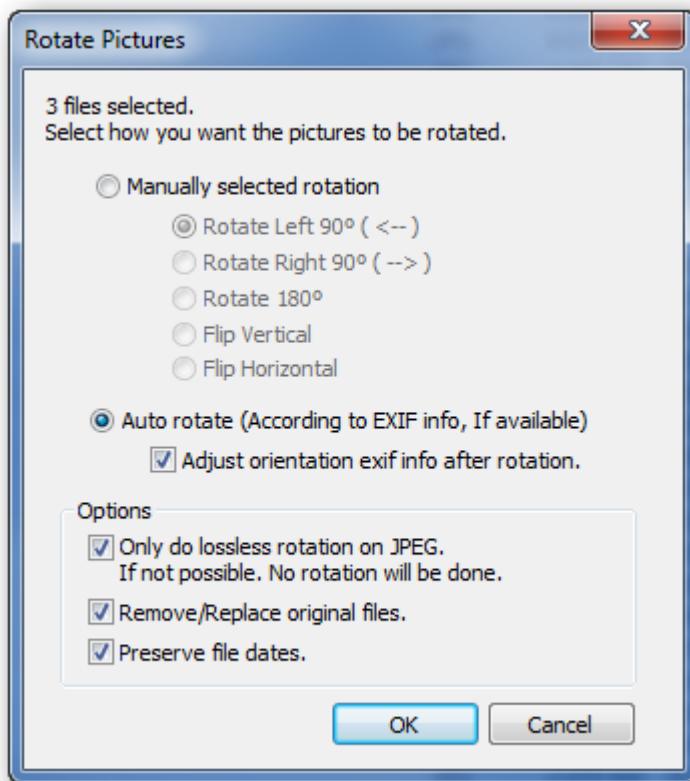
Target picture formats that are supported are *JPEG*, *PNG* and *Bitmap (BMP)*.

Options that may be selected can delete the original file if successful, and/or overwrite if the target file already exists.



Rotate Pictures

This rotates the selected picture(s) according to the selected rotation or flip settings.



Auto Rotate

Auto Rotate reads the *EXIF* information from the picture file; if orientation information is present the image is rotated according to that information.

Lossless JPEG Rotation

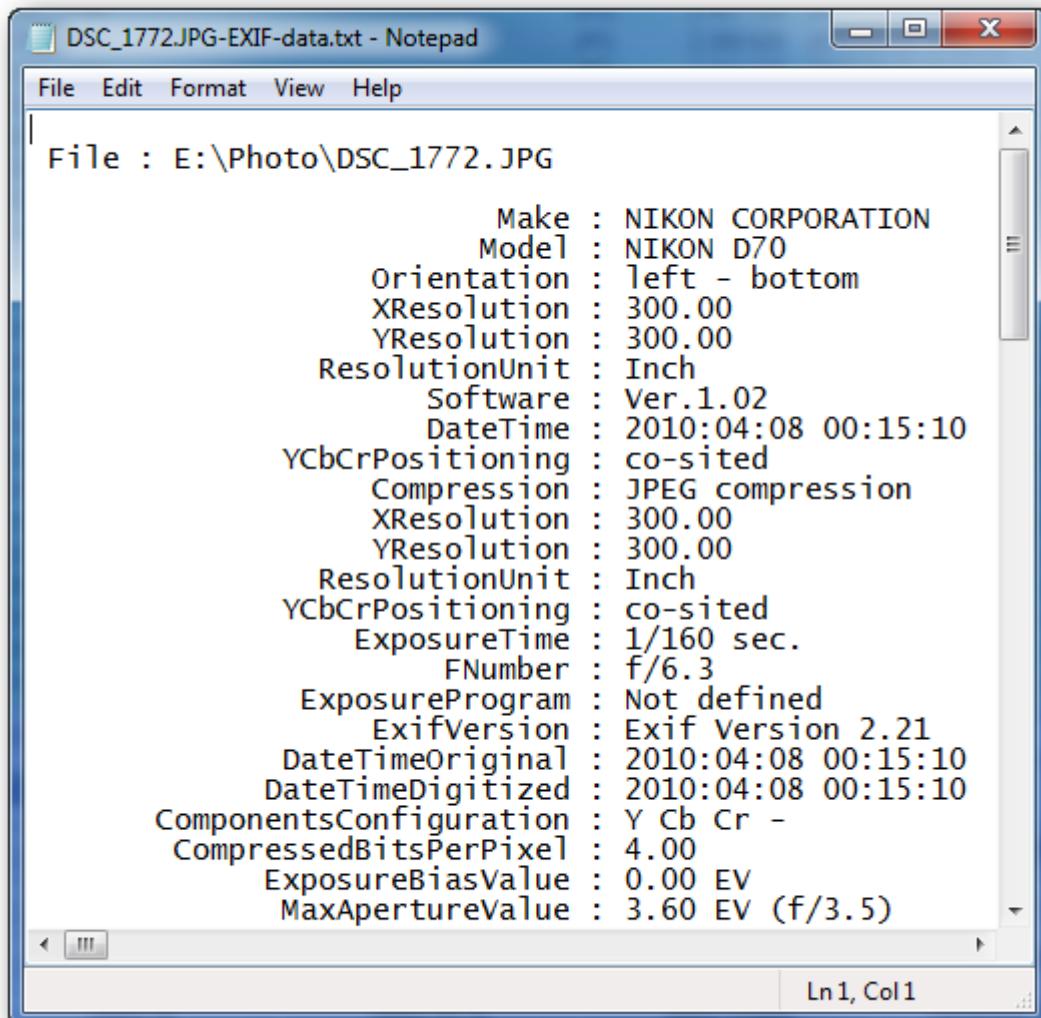
If a picture's height and width are dividable by 16 then it is possible to do a rotation without reencoding the pictures and without loss of quality. If the option **Only do lossless rotation on JPG** is enabled it will rotate pictures only if it is possible to do so without quality loss; otherwise it will skip rotating that picture.

Strip EXIF Information

Removes all *EXIF* information from the selected picture files.

Export EXIF Information

Exports all *EXIF* information from the selected pictures into text files. The text files will be named the same as the picture files but with EXIF-data text appended to the end of them.

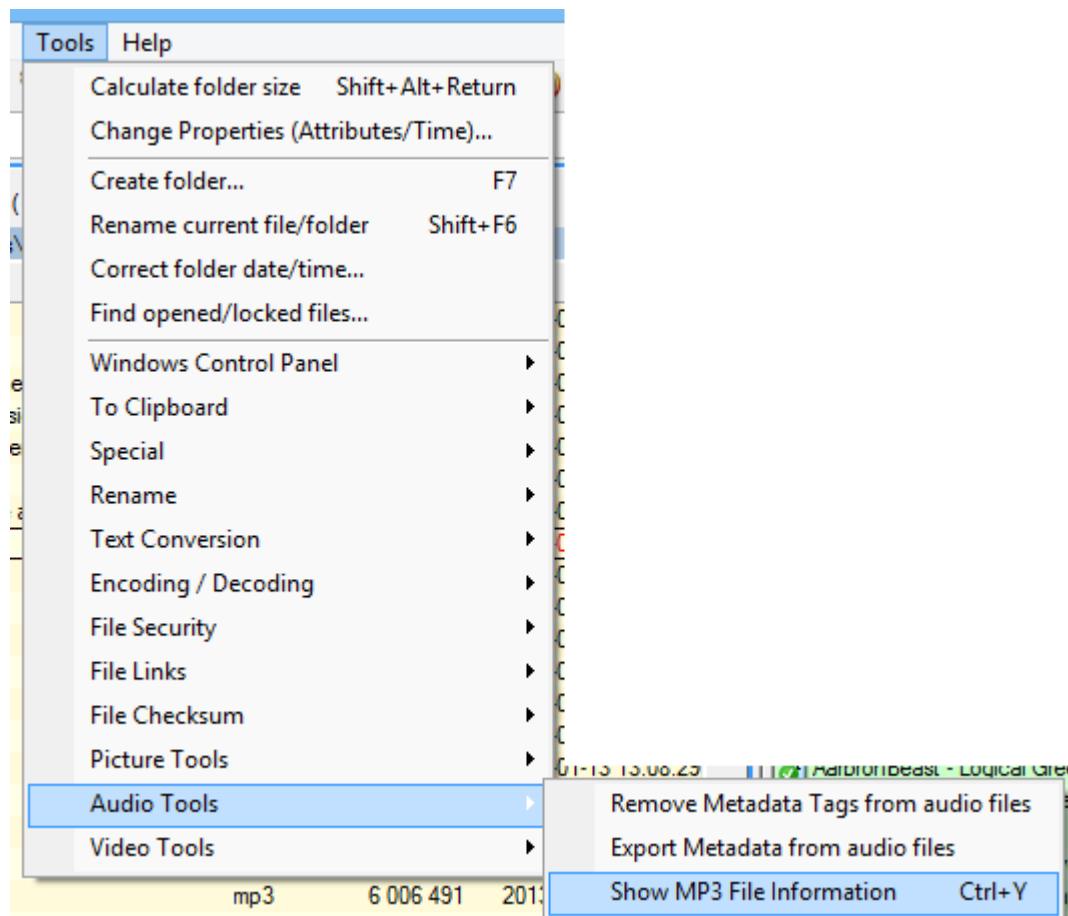


Audio Tools

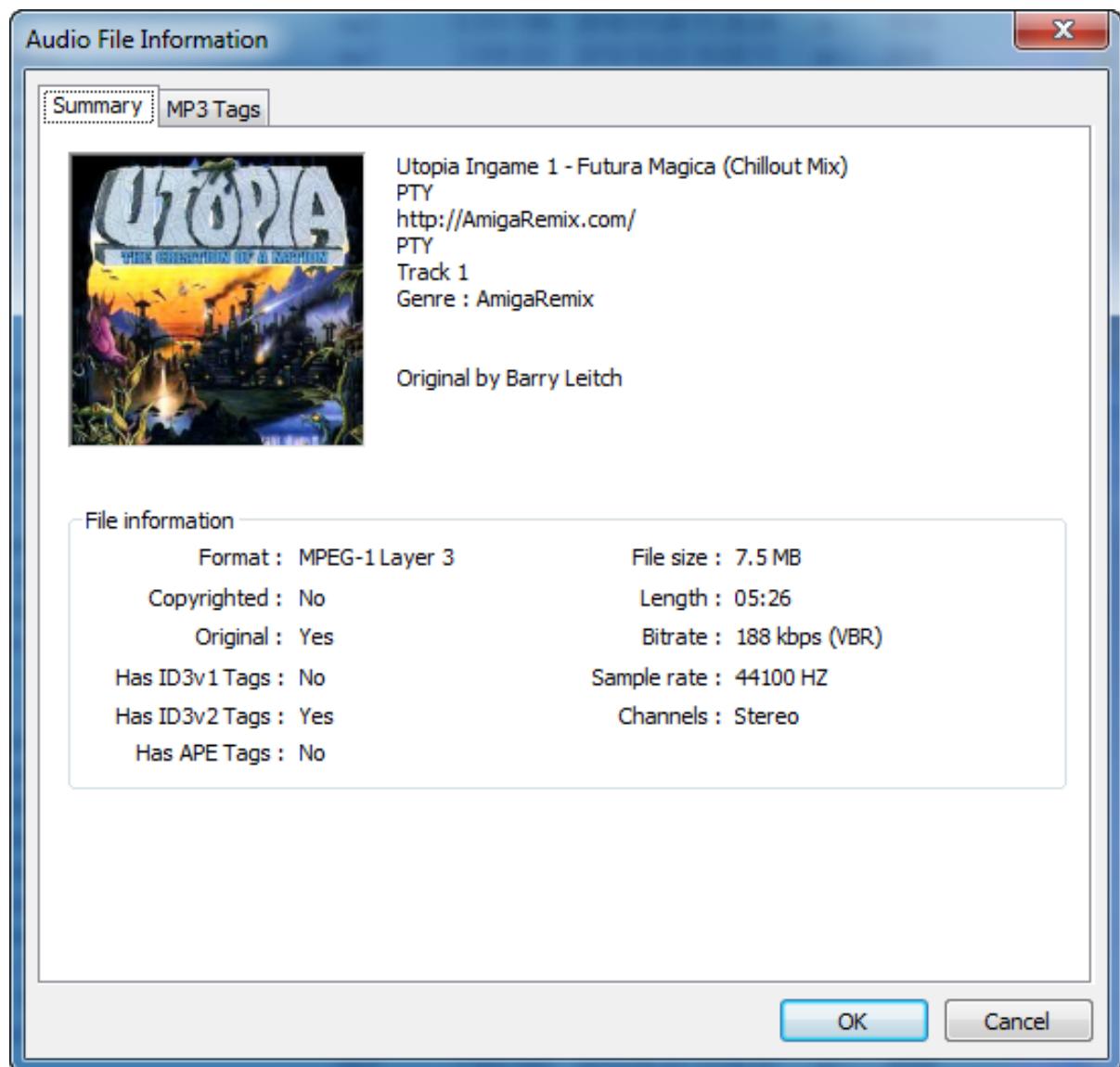
Multi Commander Audio Tools is a default extension that is included with Multi Commander. This extension will add audio properties like ID3 tags as columns for file browsing.

Namn /	Andelse	Storlek	Genre	Bit Rate (kbps)	Sample Rate (Hz)
BuzzarD - Body Blows	mp3	10 141 865	AmigaRemix	256	44100
BuzzarD - Jesus On E's (2010 Remix)	mp3	84 546 830	AmigaRemix	256	44100
Buzzer - Complications (Moar Cowbell Mix)	mp3	5 157 323	AmigaRemix	261	44100
Carsten 'Mr.Soundwave' Herbst - Doctor Who - Dalek...	mp3	7 736 492	AmigaRemix	320	44100
Carsten 'Mr.Soundwave' Herbst - Hybris Returns	mp3	8 879 124	AmigaRemix	320	44100
Charlie - Jim Power - Sacred Forest	mp3	8 406 957	AmigaRemix	192	44100
Charlie - Pinball Dreams - Digital World	mp3	5 631 499	AmigaRemix	192	44100

It will also provide a few audio related tools like removing all ID3/APE tags from an mp3 file, and exporting all meta data like ID3 or APE tags and audio properties from an audio file into a text file.



Showing MP3 file Information will show an Window where the ID3 tags can be viewed and edited.



Video Tools

Movie information in columns

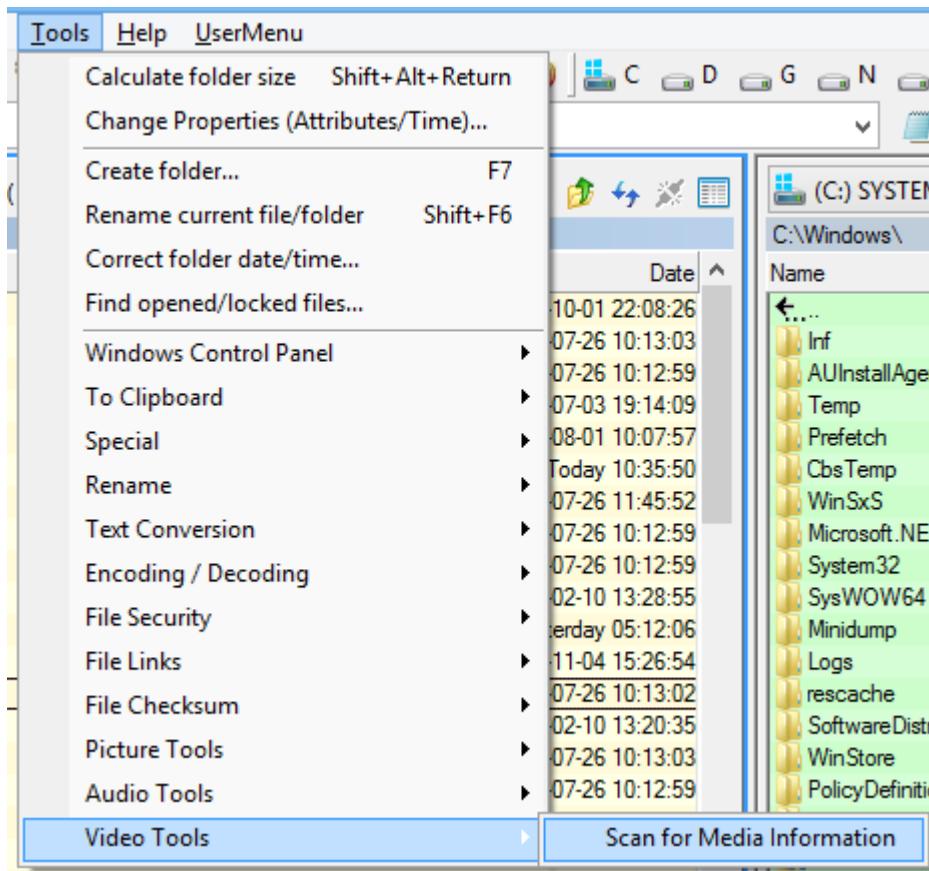
For a while several users have asked for a way to show columns with movie information on movie files they have on their drives. With the help of some of you there is now a first version of this feature working.

The screenshot shows the Multi Commander interface with the path E:\Movies\ selected. The status bar indicates 23 GB of 172 GB free (13% free). The main window displays a table with columns: Name, Ext, Size, Date, Attrib, Rating, IMDB Link, and Genre. The data includes:

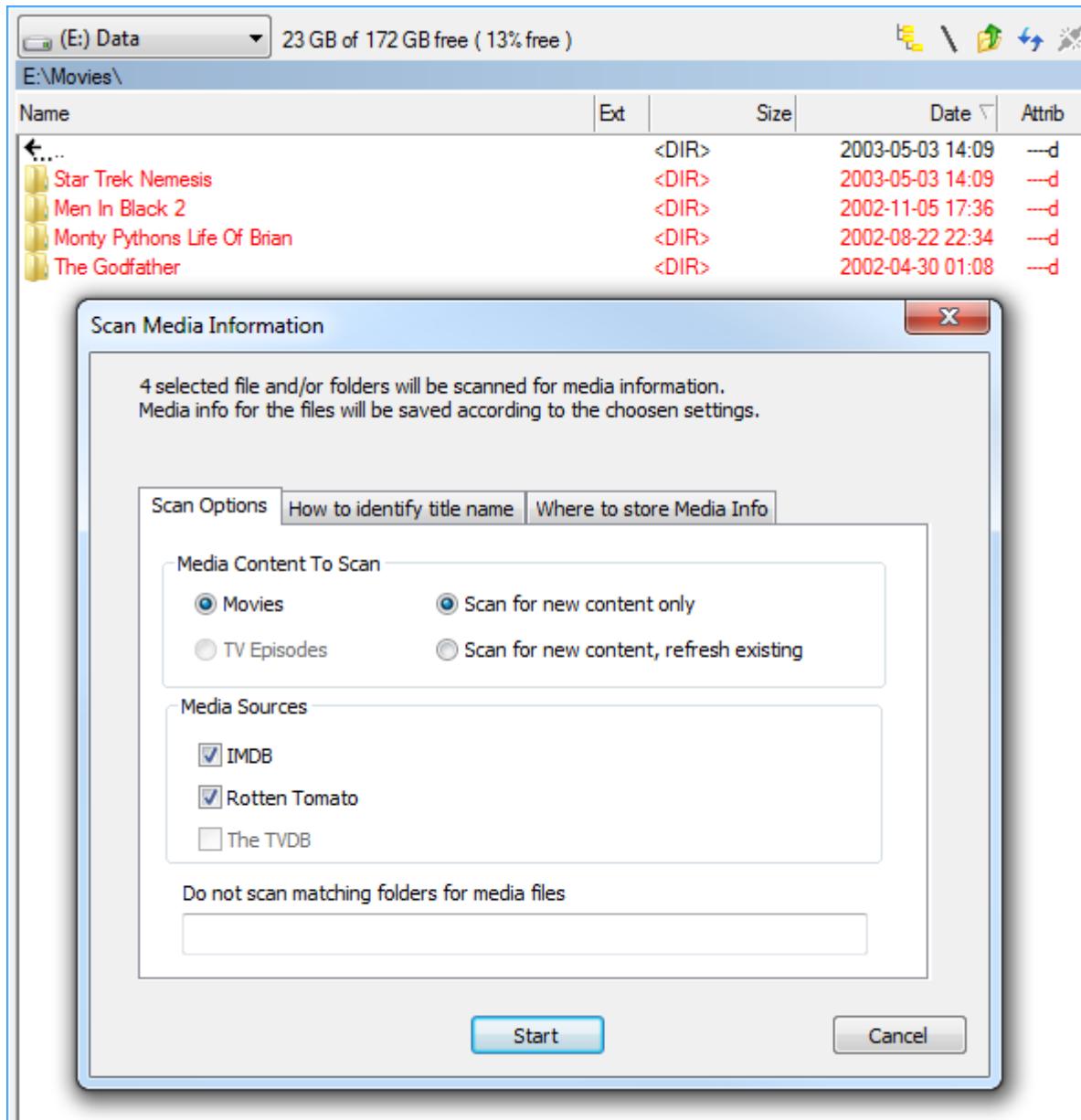
Name	Ext	Size	Date	Attrib	Rating	IMDB Link	Genre
←....	<DIR>		2003-05-03 14:09	---d			
Star Trek Nemesis	<DIR>		2003-05-03 14:09	---d	6.4	Web	Action, Sci-Fi, Thriller
Men In Black 2	<DIR>		2002-11-05 17:36	---d	5.6	Web	Action, Comedy, Sci-Fi
Monty Pythons Life Of Brian	<DIR>		2002-08-22 22:34	---d	8.2	Web	Comedy
The Godfather	<DIR>		2002-04-30 01:08	---d	9.2	Web	Crime, Drama

(If you ctrl+double click on "Web", the browser will open to the page for that movie on imdb.com.)

Before you can show the movie information in columns, it must be retrieved and cached locally. This is done by selecting **Menu->Tools->Video Tools->Scan for Media Information**.



Then you need to configure how it should identify the movies, how to store the media information and more.



For it to be able to identify the movie you have, you must configure how it should identify the movie title. Should it get the movie title from the file or folder name? (To make it easier for the system to figure out the Movie Title it is recommended to store the movie file in a folder that is named the same as the movie title.)

Two sources are available for getting movie information, *IMDB* and *Rotten Tomatoes*.

(Getting information for TV Episodes is not supported yet.)

If you only want to scan for new movies that you have not already indexed, select **Scan for new content only**. If a movie already has an *.mediainfo file or is listed in the local MediaDB, then the movie is skipped.

If you want to re-scan movies you have previously scanned, select **Scan for new content, refresh existing**.

Where and How to store the Media Information

You can store the Media information in two ways: as a *.MediaInfo file located next to the movie file with the same name as the movie files but with the file extension ".mediainfo", or in a MediaDB, or both.

Storing them in a MediaDB makes the access faster when showing the columns. (Especially if you want to show movie information on the folder that contains the movie.) But if you have your movie files on a NAS or access them over the network from another computer and you access these movies from multiple computers, then you need to rescan your movie archive on every computer you access it from, then store them as *.MediaInfo files that are stored next to the movie files.

To get the best of both situations, i.e. the speed of accessing the movie information and the need to not re-index the movies on multiple computers, you can elect to store them in both locations. If you scan for new content and both *.MediaInfo storage and MediaDB storage are selected and it finds an existing *.mediainfo file, it will then copy it to the MediaDB on

the current computer. (Just make sure that the option "Scan for new content only" is selected, or it will go out and try to update it.)

Tips and Trick

If you always want to view some columns with movie information when you go to where you have your movies, you can create a column set, and this column set can then automatically be set when going to that location

[Read about Column Sets](#)

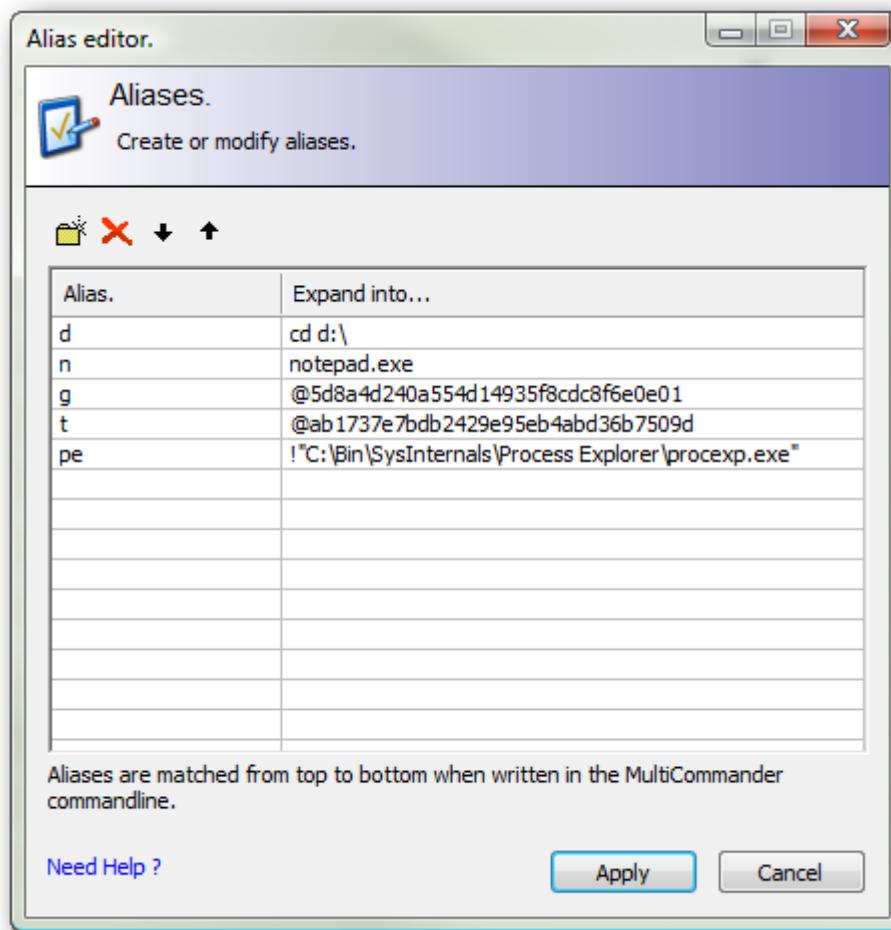
Managing Aliases

Aliases are abbreviations for commands that can be used on the MultiCommander CommandLine

Instead of writing a long command or path you can use the Alias Manager to create an *alias* for that command. An *alias* is a one word abbreviation that can be automatically expanded into the complete command. Everything that is written after the alias is appended as a parameter to the expanded command.

For example, if you have created an alias for **notepad.exe** and called it "n", then you can in the MultiCommander CommandLine write "**n mytext.txt**" and it will be expanded into "**notepad.exe mytext.txt**" before it is executed.

The Alias Manager is found under **Configuration** in the Menu Bar.



Run a User Defined Command

If the 'expand into' text is prefixed with "@" followed with the unique id for a **User Defined Command**, it will run that command. (E.g. @8c83a15a69394bd3a0ad4ba3a58515a0).

Force File To Be Executed

Expand-into text that is prefixed with "!" is forced to be launched by the Windows Shell. In the example image above, the alias "**pe**" would expand into a full path to a file, and if that text were not prefixed with "!", then instead of launching procexp.exe it would go to that path and set the focus to the procexp.exe file.

Quick command

From the commandline field it is possible to use the built-in quick command "alias" or "alias2" to create an alias command.

Usage Syntax

```
alias <alias>=<expand into>
alias2 <alias>
```

The command **alias2** takes the current path and assigns it to the specified alias.

Example

```
alias pic=E:\MyPictures
```

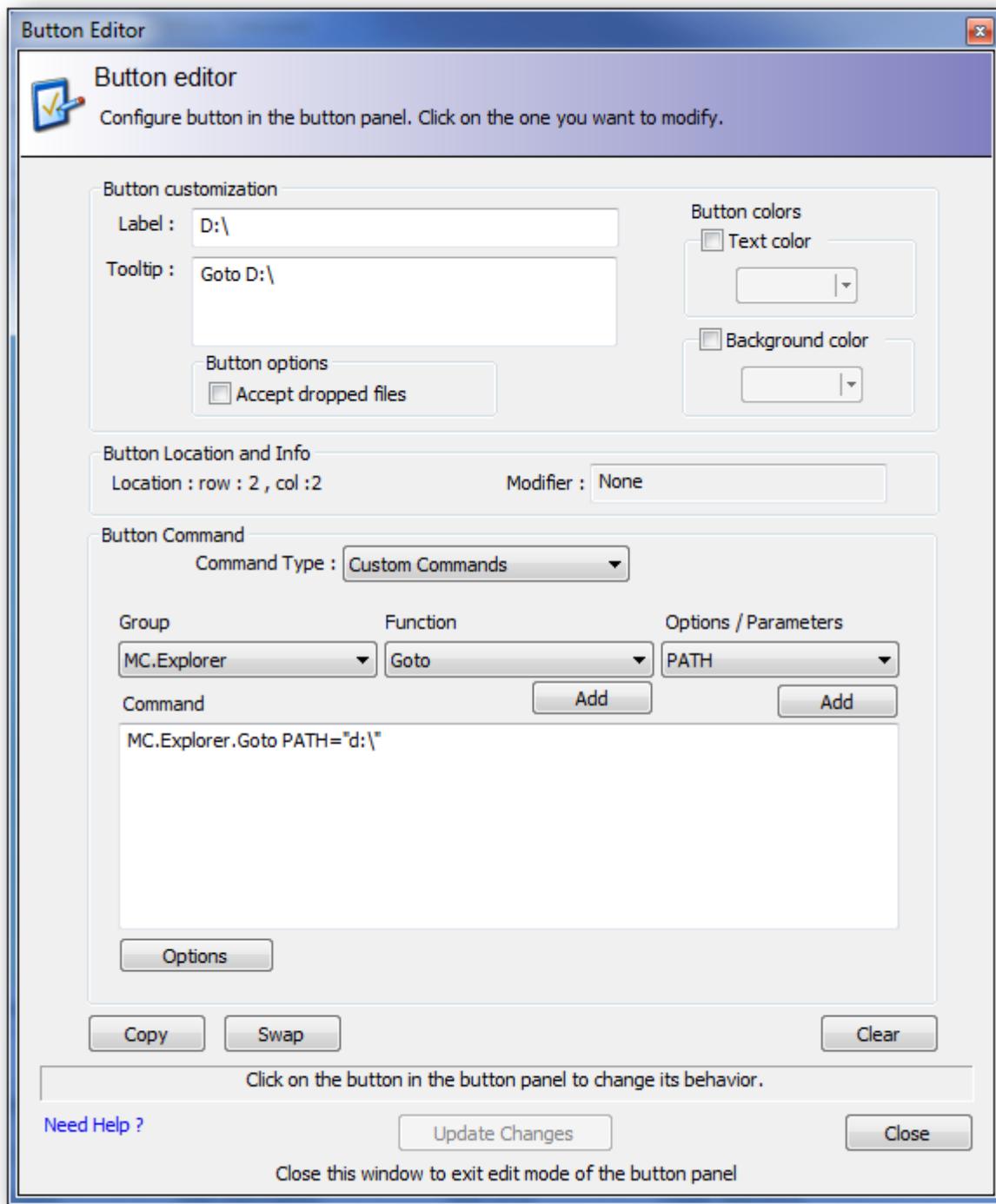
If the current folder is E:\MyPictures then it is enough to write

```
alias2 pic
```

Button Editor

The Button Editor is used to change/customize the buttons on the Button Panel.

To change how the buttons in the button panel work you need to open the **Button Editor**. This is done by clicking on **Button Editor** in the **Configuration** menu.



When the button editor is shown the button panel goes into edit mode. **Click on the button** you want to edit. To edit a button in one of the alternative modes hold down a mouse or key modifier to toggle the button panel state to see the button you want to change. (*Ctrl / Shift / Ctrl+Shift or Right/Middle mouse button*.)

There are six kind of commands you can create for a button. (These are the same types of command that you can create in the "user defined command" window.)

- [User Defined Commands](#)
- [Internal Commands](#)

- [External](#)
- [Custom Commands](#)
- [Batch Script \(.BAT\)](#)
- [Multi-Script](#)

User Defined Commands

The commands listed are User Defined Commands that you have already created with the User Defined Commands dialog.

Select what command you want to have run when the button is clicked.

Internal Commands

These are Commands that are either internal to MultiCommand or are extensions (mostly commands that you find in the menu bar).

Two dropdowns will be shown; in the first you select what module contains the command; in the second you select the command within that module.

External Commands

These are external programs to be executed.

Custom Commands

Custom commands are simple commands that are exposed by **MultiCommander** or some **Extension** or **Plug-in**.

You can also use the helper function list boxes to create the command.

To use them, you first select what **group** of functions you want to add a command from. All Extensions and Plug-ins are located in separate groups. When a group is selected the function list is updated with all functions that are available from that group. Select the **function** that you want to use and press the '**Add**' button; this will add the function to the "**Command**" field. Then select what **option** or **parameter** you want to add. If the parameter ends with an "=" sign, then you are required to add additional parameter information.

Examples of some of the most common quick command:

- [How to change the view filter](#)
- [How to select files and folder](#)
- [Launch external program](#)
- [Launch External program with file in focus as parameter to it](#)

Batch Script (.bat)

This is a .bat script that will be generated and run when the button is clicked.

You can insert dynamic tags into the script that will be expanded when the script is generated.

Multi-Script

This is the most advanced type of command you can create. It uses Scripts that run under the MultiCommander Script Engine.

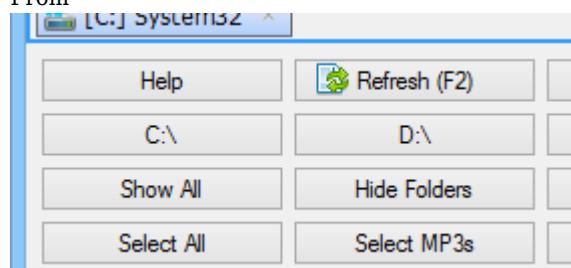
Button Panel - Wizard Mode

To more easily create commands for the button panel, you can turn on the Button panel *wizard mode*.

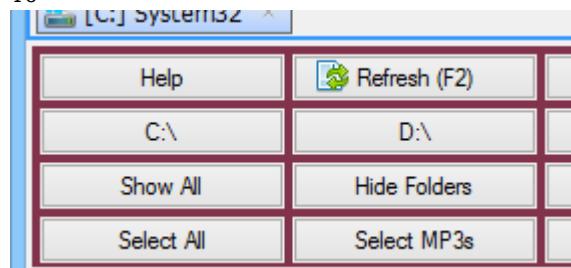
When the wizard mode is on, a file or folder can be dropped on a button and a Wizard window will be shown where you can specify how the dropped file should be connected to the button.

When the Wizard Mode is active you will see that the button panel will change its look

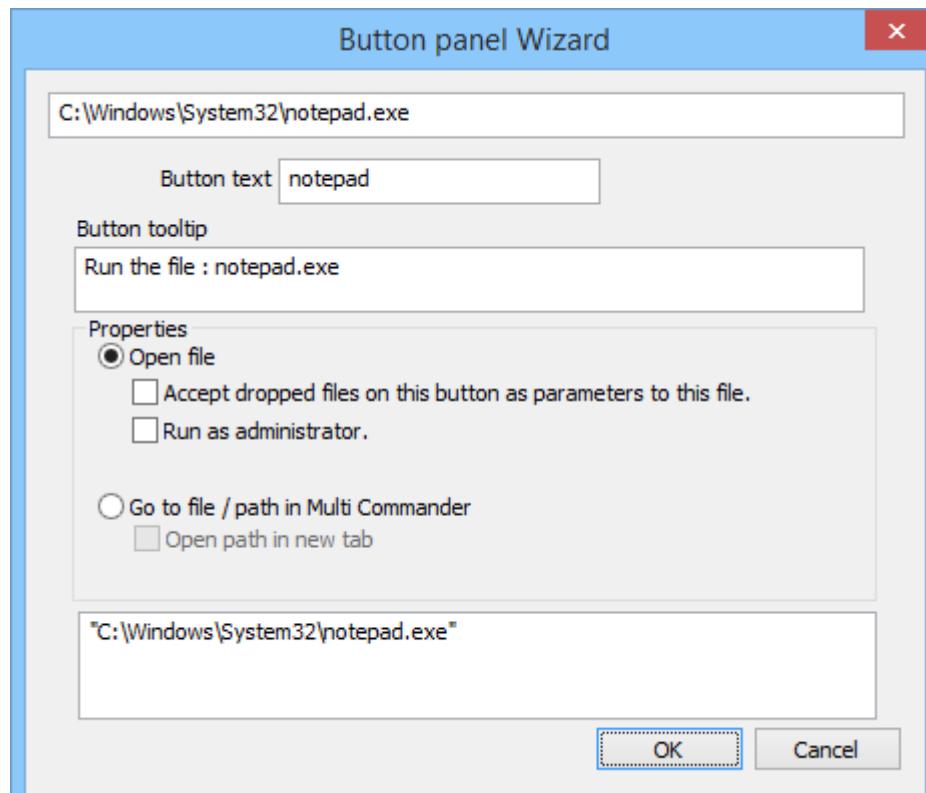
From



To



When a File or Folder now is dropped on a button the Wizard Window will be shown:

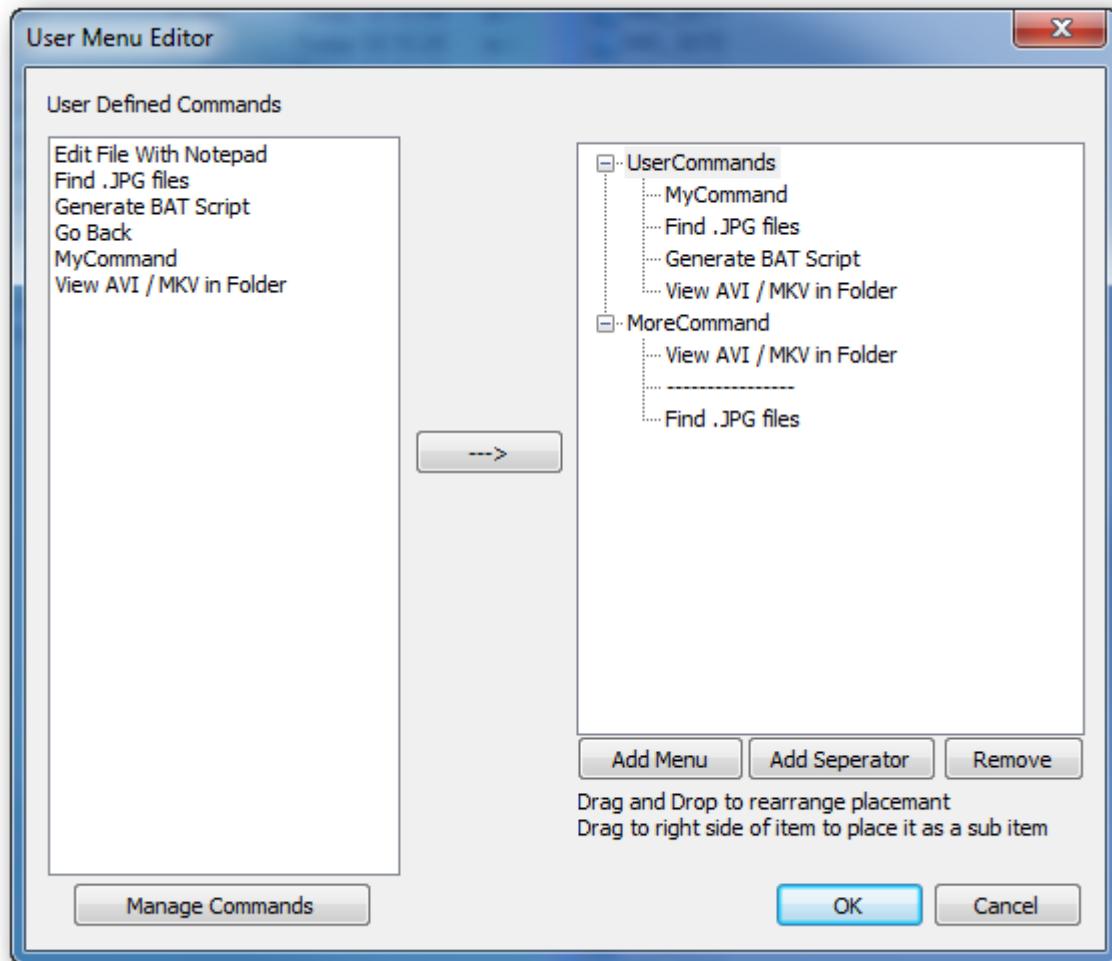


The name and tooltip are generated automatically but you can change them if you wish. Then you need to specify whether it should generate an Open File command or a Go to File/Path command for the button, so that when the button is pressed the file you dropped on the button should be launched or if you want the Explorer Panel to go that file or path.

Pressing the Wizard Mode button again will turn off the Wizard Mode for the button panel.

User-Menu Editor

To add your own **User Defined Commands** to the menu you need to start the Menu Customization Editor that you find under the Configuration menu.



This will list all of your User Defined Commands on the left and on the right it will show your user menus.

The menu is shown as a tree list with the top level being sub-menus on the menu bar.

You can drag and drop menu items in the tree list to rearrange how you want to display items.
The Add Menu button adds a new top-level Menu or Sub-Menu.

Select the User Defined Command you want to Add on the left, and Select under what menu you want that command added on the right. Then press the "-->" to add that command to the menu.

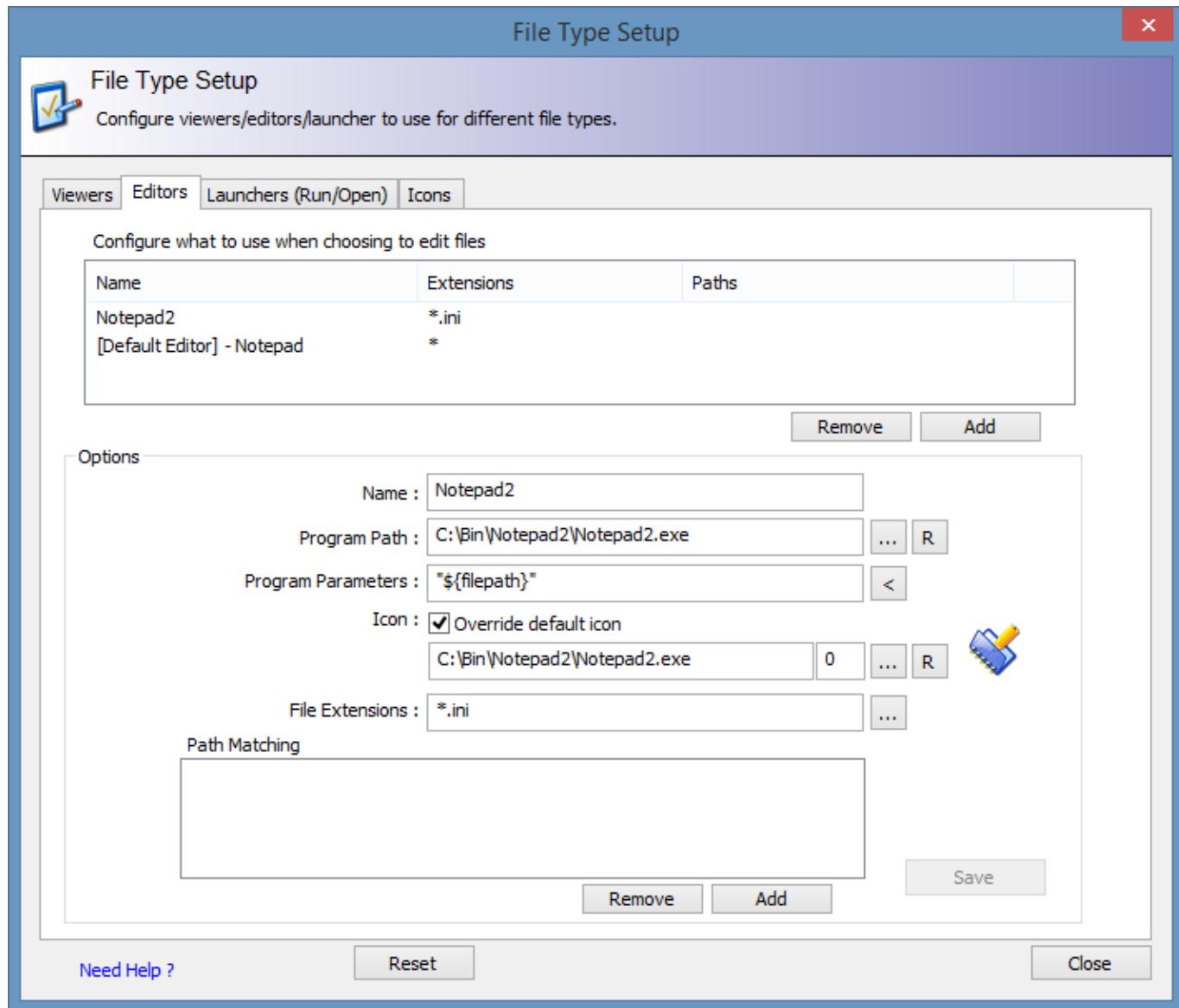
File Type Setup

File Type Setup is used to configure how Multi Commander should handle different file types. The Configuration here will override the file associations settings set in Windows.

The setup is separated into different actions that can be configured. View, Edit, Run/Open and Icon.

Info

Configure of which File System Extension should be used for different file extension is NOT configured here.
(Eg what file extensions the Zip FileSystem extensions should use)
It is configured under each File Extension in [Extensions and Plugins](#).



Normally when a file is run/opened, Windows will decide what program should be started for that file.
The configuration you do here is used before Windows gets a chance, and if running Multi Commander as portable you can configure what other portable programs it should use here.

When you decide to **View/Edit** or **Run/Open** a file from Multi Commander it will go through the list of configurations and try to find a matching one. It will match configurations based on the "Extensions" field. You can also specify that a entire should only be valid for a certain paths.

For example if two different viewers are used for viewing jpg files, but for files located in "D:\Photos" perhaps you want to use a special one. For all other locations the other viewer is used.

If it does not find any matching configuration entry the **[Default Editor]** / **[Default Viewer]** entire is used.

To change the default viewer/editors that are used change the **[Default Editor]** / **[Default Viewer]** entries under Viewer/Editors.

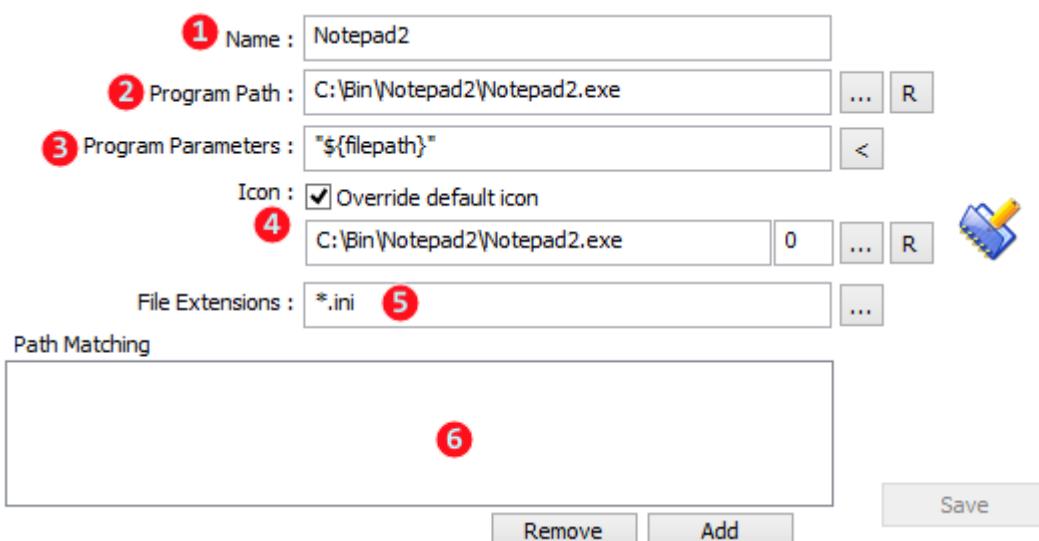
Icon setup is used if the Icon should be overridden. Viewers/Editors can also override the icon if they override the icon it will be listed under the icon page. Running Multi Commander in portable mode you can with icon set make sure that the icons always look the same.

Press the **Add** button under the list of setup entries to add a new. Select existing entry and press the **Remove** the button to remove an entire. **[Default ...]** Entries can not be deleted.

Pressing the **Reset** button at the button will reset all configurations.

Viewer / Editor / Run/Open

The following fields can be modified when configuring a Viewer/Editors/Run entry



1. Name

Name of this setup item.

1. Program Path / Internal Extension / Windows Action

This field is changed depending on what type is selected. Press "..." then select program

- **External Program**

With external program any external program can be run.

- **Internal Extension**

Internal Extensions can register themselves as viewers/editors/launchers. They will then be listed as extensions that you can select from.

For Viewer there exists two default extension, "MC Picture Viewer" and "MultiDataViewer"

- **Windows Action**

Choose this if Windows should decide what to do with the file.

Windows has three (3) actions default, open , edit to choose from

If a external program was set it is possible to press the **[R]** button to convert the path to a relative path. This is needed if you are running in portable mode. The selected program must be located on the same drive as MultiCommander.exe

3. Program Parameters / Options

If internal extension was set above this field will be "**Options**". Then extension specific options may be entered here.

If external program was set above this field will be "**Program Parameters**". Then you can add any program specific parameters that should be sent here. Pressing the "[<]" button and a list with special tags that can be added are shown. this tags will be expanded when the external program is run.

4. Icon

By checking the option "**Override default icon**" a icon can be configured. This icon will then be used for all matching file extensions and overriding the default icons that would be used

However if you have Multiple FileType Setup entire that are matching the same file extensions and all of them have the override default icon enabled. Then only the first of them will be used.

5. File Extensions

Space separated list with all file extensions this setup should affect

Pressing the "..." button and an Windows will be shown that will allows for easier editing of the file extension list.

6. Path Matching

A file type setup entire can be set to only be valid for a special path(s)

Press the [Add] button under the "**Path Matching**" box and add that path there. If you see an "*" in front of the path it means that the Path Matching rules are valid for that path and all its sub paths.

To remove a Path Matching select it and press [**Remove**]

Icon

By default Multi Commander will get the icon that Windows has defined for a file type. On this page it is possible to override that.

This list will show all the icons that Multi Commander is overriding

Extensions	Path to file containing icon	Icon Index	Information
.ini	C:\Bin\Notepad++\notepad++.exe	0	Controlled by Viewer or Editor file type setup
.pdf	\${mcpath}\MCIcons.dll	103	
.zip	\${mcpath}\MCIcons.dll	114	

If an icon is overridden by a viewer/editor file type entry then it will be listed here. But it is not possible to remove the icon here. It must be done on the Viewer/Editor file type page.

To add an entry press the "**[Add]**" button. Then enter the file extension that the icon should be shown for.
Enter the path to a .exe/dll/.ico and select the icon idx from that file that should be used.

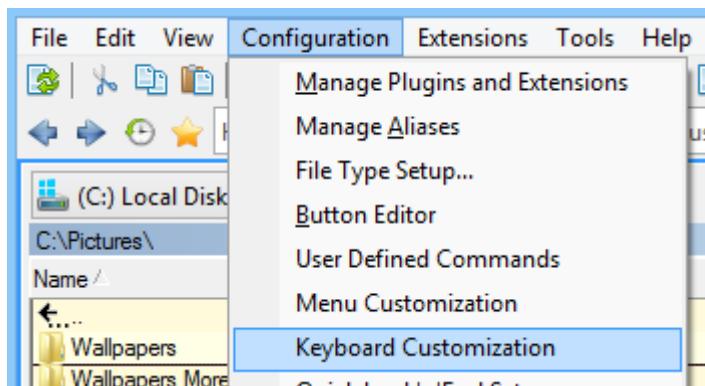
Pressing "**[...]**" button a Icon Picker window is shown. that Windows allows the .exe/.dll/.ico file to be browsed and viewed.
Making it a lot easier to find the wanted icon.

Press the **[R]** button to convert the path a relative path. This is useful if you are running in portable mode and want to have relative paths.

All internal icons used by Multi Commander are stored in MCIcons.dll, This files also contains a couple of extra icons that can be useful. This file is found at the same location that MultiCommander.exe is at.

Customize Keyboard Shortcuts

To customize the keyboard shortcuts in MultiCommander, open **Keyboard Customization** under the **Configuration** menu.

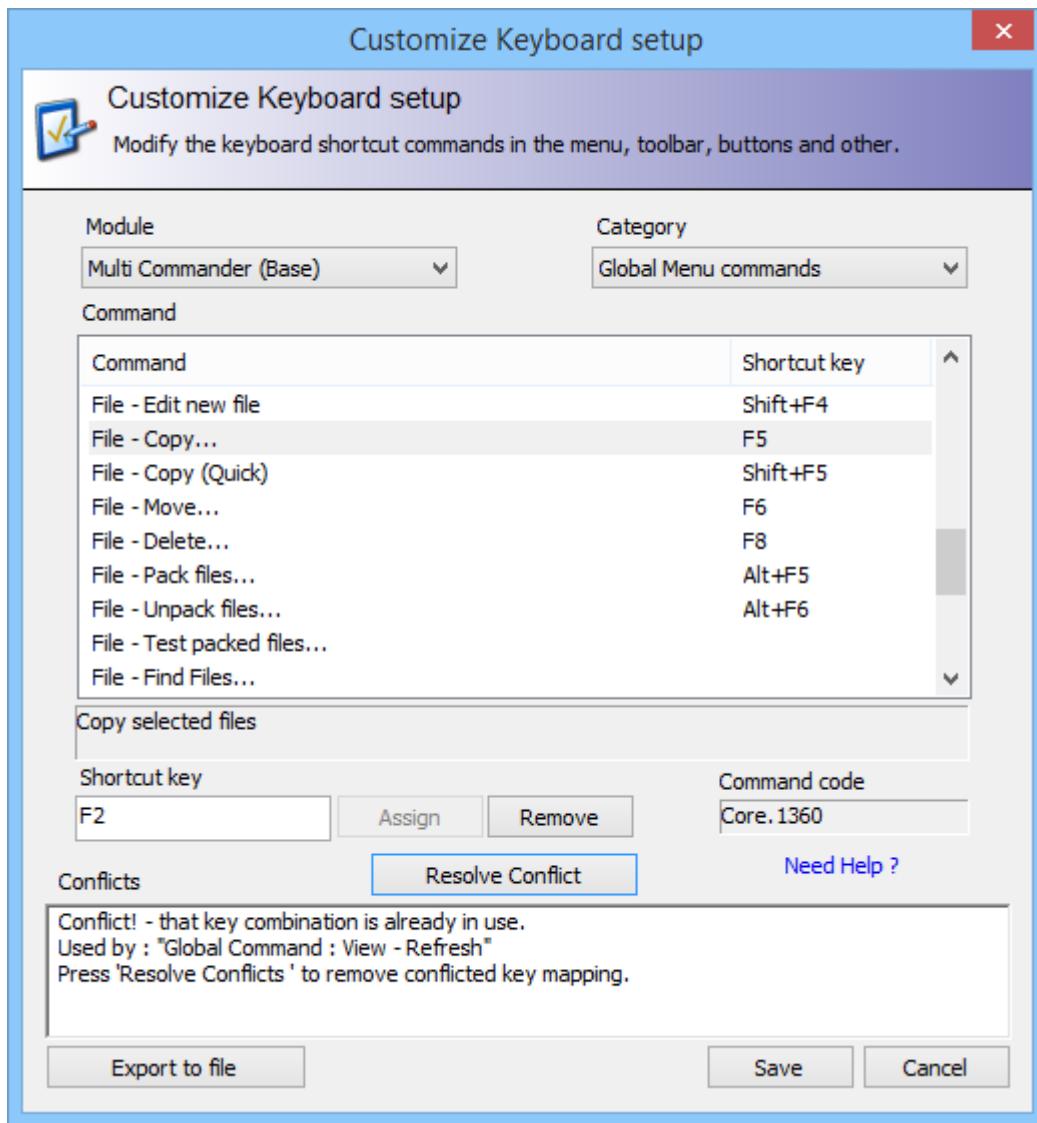


Most of the shortcut keys in MultiCommander can be changed. The shortcuts are split up into different sections: each module/extension has its own list of commands, split into global and module-specific commands.

Global commands can be called from any module. These are generally menu commands, and are available in all modules and extensions that support them. Since the commands are global within MultiCommander, shortcut keys must be unique across extensions.

Module Specific commands are only available when that extension is currently in focus. Multiple extensions can have the same shortcut assigned.

Change Shortcut key



To find the command you want to change, first select the module it belongs to, and then choose the category (global or module specific).

When you have found the command you wish to change, click on it to select it and then click on the shortcut box below the list. There you just press the keys that you want to have assigned to the command, then press "**Assign**" to set the shortcut. If the command you have chosen is already taken it informs you about the conflict and what command already has that shortcut assigned. You need to remove the conflicting shortcut before you can assign the new one.

To remove a shortcut, select the command and then press the "**Remove**" button.

"**Command code**" is the internal name of the command. It is used if a command should be execute from the custom command [MC.RunCmd](#)

Tips

To get a list of all the shortcuts, you can export them to a text file by pressing the "Export to file" button. If you hold **SHIFT** while clicking on the Export button the exported list will also contain the internal command code

References

- [Default keyboard shortcut \(Commander styled setup\)](#)
- [Default keyboard shortcut,\(Windows Explorer styled setup\)](#)

Default Keyboard Shortcuts (Commander Style)

MultiCommander Core shortcuts

These are global shortcuts that work the same no matter what extension you are in.

Command	Shortcut
Go Back	Alt + Left
Go Forward	Alt + Right
Show History	Alt + Down
Tab - Close Tab	Ctrl + W
Tab - Duplicate Tab	Ctrl + T
File - View File	F3
File - Edit File	F4
File - Copy	F5
File - Copy Quick	Shift + F5
File - Move	F6
File - Delete	F8 , Delete
File - Pack Files	Alt + F5
Edit - Cut	Ctrl + X
Edit - Copy	Ctrl + C
Edit - Paste	Ctrl + V
View - Full width	Ctrl + F11
View - Log Panel	Ctrl + L
View - Button Panel	Ctrl + B
View - Refresh	F2

File Manager

Global file manager hotkeys are also valid when focus is in another extension if the extension supports it.

Command	Shortcut
File Tools - Calculate folder size	Shift + Alt + Return
File Tools - Create Folder	F7
File Tools - Rename	Shift + F6
Send to clipboard - File/Folder path and name	Ctrl + P
Send to clipboard - File/Folder name	Ctrl + Shift + P
Send to clipboard - File/Folder path	Shift + Alt + P
Select using filter	Num +
Unselect using filter	Num -
Select All	Ctrl + A
Unselect All	Ctrl + D
Invert Selection	Num *
Select all with same file extension	Ctrl + Num+

Unselect all with same file extnesion	Ctrl + Num-
Select all with same name	Alt + Num+
Unselect all with same name	Alt + Num-
Compare Source and Target, Select Missing/Newest	Ctrl + F10

FileManager specific hotkeys are only available when focus is in the file manager view.

Command	Shortcut
Show drive selection box (Left side)	Alt + F1
Show drive selection box (Right side)	Alt + F2
Set View filter	Ctrl + F
Start a Search from current path	Alt + F7
Edit path	Ctrl + E
Go to Quick Path #1	Ctrl + 1
Go to Quick Path #2	Ctrl + 2
Go to Quick Path #3	Ctrl + 3
Go to Quick Path #4	Ctrl + 4
Go to Quick Path #5	Ctrl + 5
Go to Quick Path #6	Ctrl + 6
Go to Quick Path #7	Ctrl + 7
Go to Quick Path #8	Ctrl + 8
Go to Quick Path #9	Ctrl + 9
Go to Quick Path #10	Ctrl + 10
Sync path to the Left	Ctrl + Left
Sync path to the Right	Ctrl + Right
Force Refresh (Flush cache)	Shift + F2
Go to parent folder (Go up)	Back
Go to the root of the device	Ctrl + Back
Go to same path as target view	Ctrl + .
Show file properties	Alt + Return

File Search

Command	Shortcut
Show / Hide Search window	Ctrl + F
Save a search	Ctrl + S
Load Search	Ctrl + O
Start Search	Ctrl + Return
Pause Search	Ctrl + P

Customize your own keys

You can easily customize your own hotkeys by going to the Keyboard Customization entry in the Configuration menu.
There you can also export all your hotkeys into a text file.

- [How Do I ... Customize keyboard shortcuts](#)

Default Keyboard Shortcuts (Windows Explorer Styled)

MultiCommander Core shortcuts

These are global shortcuts that work the same no matter what extension you are in.

Command	Shortcut
Go Back	Alt + Left
Go Forward	Alt + Right
Show History	Alt + Down
Tab - Close Tab	Ctrl + W
Tab - Duplicate Tab	Ctrl + T
File - View File	F1
File - Edit File	F9
File - Copy	F4
File - Copy Quick	Shift + F5
File - Move	F6
File - Delete	F8, Delete
File - Pack Files	Alt + F5
File - Pack Files	Alt + F6
Edit - Cut	Ctrl + X
Edit - Copy	Ctrl + C
Edit - Paste	Ctrl + V
View - Full width	Ctrl + F11
View - Log Panel	Ctrl + L
View - Button Panel	Ctrl + B
View - Refresh	F2

File Manager

Global file manager hotkeys are also valid when focus is in an another extension if the extension supports them.

Command	Shortcut
File Tools - Calculate folder size	Shift + Alt + Return
File Tools - Create Folder	F7
File Tools - Rename	Shift + F6
Send to clipboard - File/Folder path and name	Ctrl + P
Send to clipboard - File/Folder name	Ctrl + Shift + P
Send to clipboard - File/Folder path	Shift + Alt + P
Select using filter	Num +
Unselect unsing filter	Num -
Select All	Ctrl + A
Unselect All	Ctrl + D
Invert Selection	Num *

Select all with same file extension	Ctrl + Num+
Unselect all with same file extnesion	Ctrl + Num-
Select all with same name	Alt + Num+
Unselect all with same name	Alt + Num-
Compare Source and Target, Select Missing/Newest	Ctrl + F10

FileManager specific hotkeys are only available when focus is in the file manager view.

Command	Shortcut
Show drive selection box (Left side)	Alt + F1
Show drive selection box (Right side)	Alt + F2
Set View filter	Ctrl + F
Start a Search from current path	Alt + F7
Edit path	Ctrl + E
Go to Quick Path #1	Ctrl + 1
Go to Quick Path #2	Ctrl + 2
Go to Quick Path #3	Ctrl + 3
Go to Quick Path #4	Ctrl + 4
Go to Quick Path #5	Ctrl + 5
Go to Quick Path #6	Ctrl + 6
Go to Quick Path #7	Ctrl + 7
Go to Quick Path #8	Ctrl + 8
Go to Quick Path #9	Ctrl + 9
Go to Quick Path #10	Ctrl + 10
Sync path to the Left	Ctrl + Left
Sync path to the Right	Ctrl + Right
Force Refresh (Flush cache)	Shift + F2
Go to parent folder (Go up)	Back
Go to the root of the device	Ctrl + Back
Go to same path as target view	Ctrl + .
Show file properties	Alt + Return

File Search

Command	Shortcut
Show / Hide Search window	Ctrl + F
Save a search	Ctrl + S
Load Search	Ctrl + O
Start Search	Ctrl + Return
Pause Search	Ctrl + P

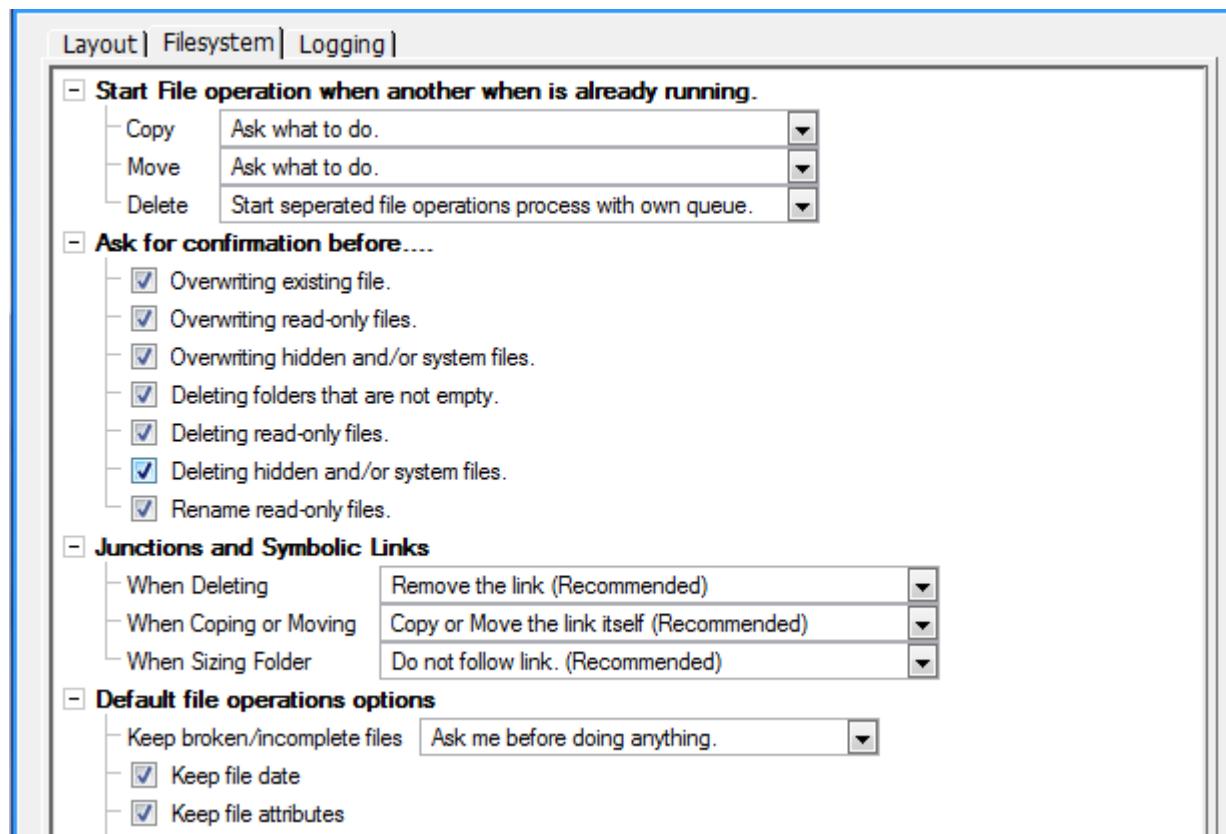
Customize your own keys

You can easily customize your own hotkeys by going to the Keyboard Customization selection in the Configuration menu. There you can also export all your hotkeys into a text file.

- [How Do I ... Customize keyboard shortcuts](#)

Core Settings

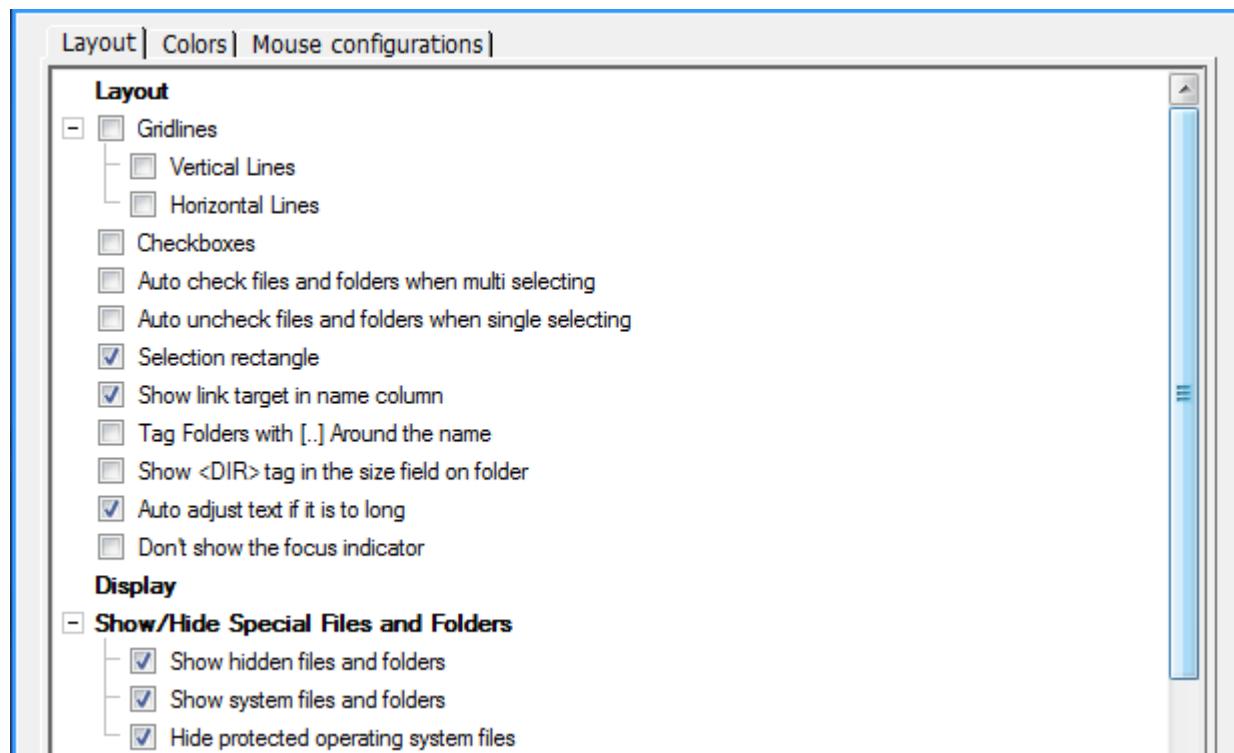
The Core Settings are settings in Multi Commander that affect how the core part works. To configure these, select "Core Settings" under the Configuration menu.



Settings like Language, Tab Location, File System Operations and Logging can be found here. These are settings that are not affected by extensions and plug-ins.

Explorer Panel Settings

The settings for the Explorer Panel let you configure the look and feel of the Explorer panel.



You can change the Layout, the colors , how the mouse should work, etc.

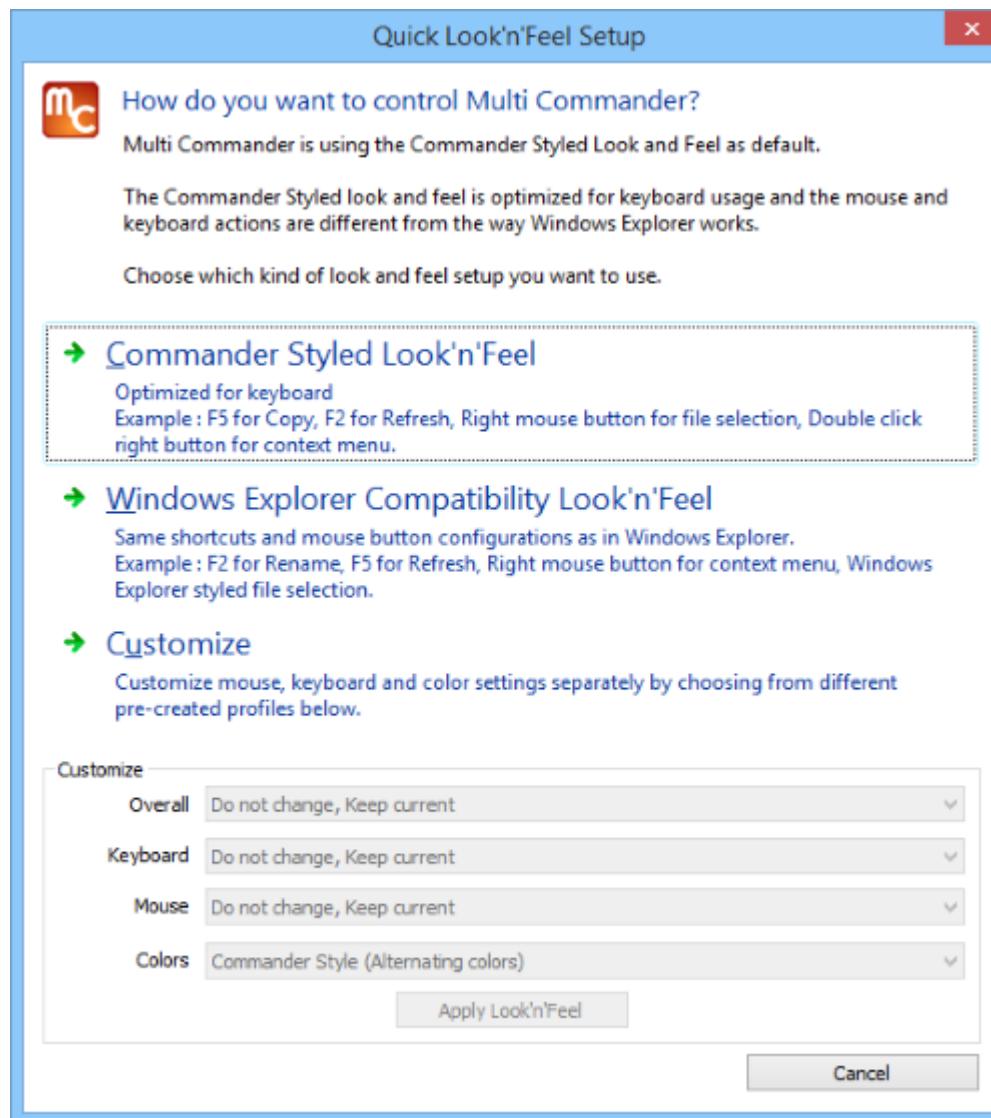
For examples of different layouts and color setups, see screenshots [here](#).

Change To Windows Explorer Look and Feel

MultiCommander by default uses the popular Commander-Styled mouse and keyboard layout.

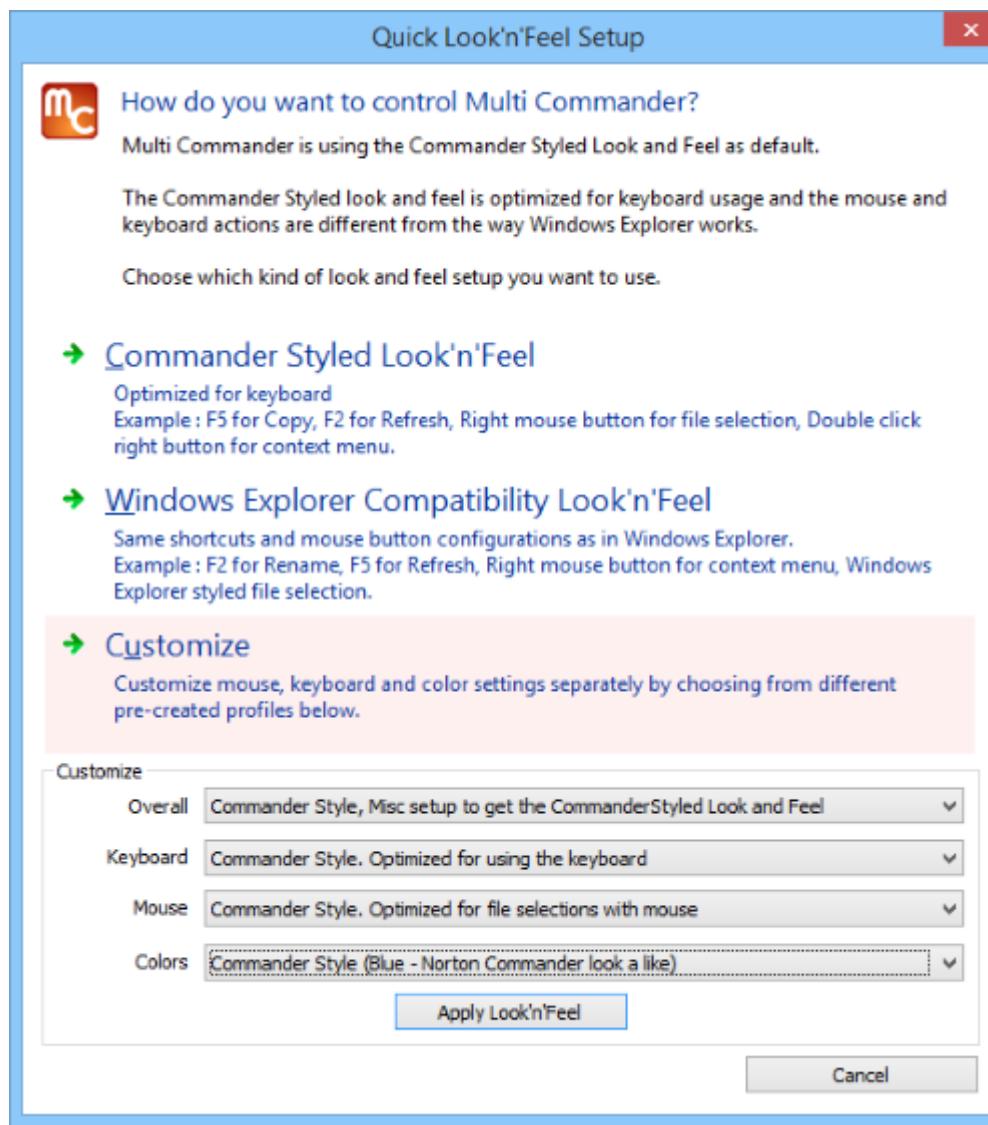
Users who have never used this style before might feel more comfortable with the way **Windows Explorer** works. Since MultiCommander is very configurable it can be changed to look and feel like Windows Explorer, but that is tedious work because there are a lot of configurations that need to be changed. However, the **Quick Look'n'Feel Setup** makes it possible to change this quickly and easily.

To find this, select "**Quick Look'n'Feel Setup**" in the **configuration menu**.



To Change everything to the Windows Explorer look and feel, select **Windows Explorer Compatibility Look'n'Feel**.

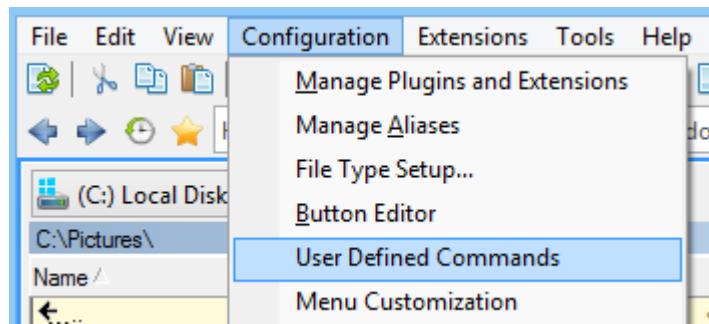
If you only want to Change some part of the configuration Like the mouse or colors settings. Choose the **Customize** option.



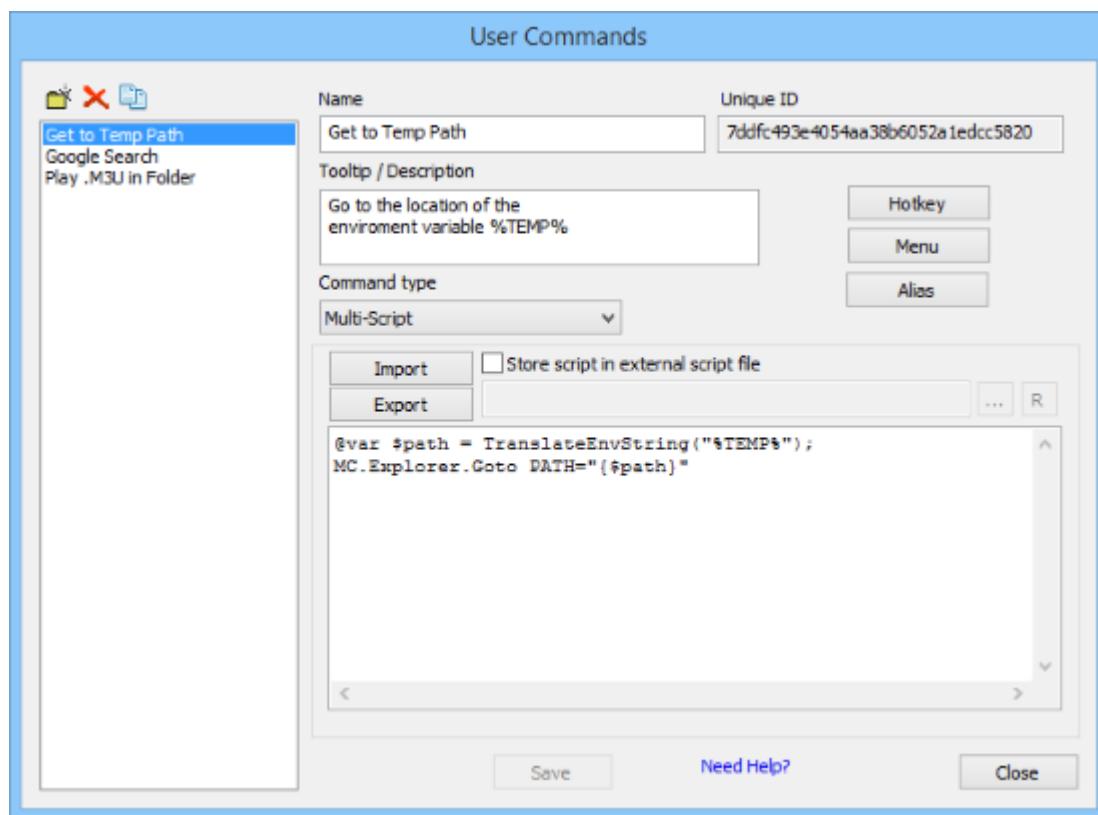
There you can select between different configurations for keyboard, mouse and color setup

Create User Defined Commands

Multi Commander allows you to create your own **commands**, which can then be run from the *menubar*, *button panel*, a *hotkey* or the *commandline*.



Open the User Defined Commands dialog by selecting "**User Defined Commands**" from the Configuration menu.



Create Command

Follow these instructions to create a command:

1. Click on the new button and enter the name you want to give your command. (The name can be changed later by editing the name property and re-saving the command.)
2. Fill in the tooltip / description for the command. This text will be shown as a tooltip when you hover over a command (if the command is assigned to a button on the button panel) or as a description when needed.
3. Next select what type of command you want to create. There are five types of command to choose from:
 1. [Internal Commands](#)
 2. [External](#)
 3. [Custom Commands](#)
 4. [Batch Script \(.BAT\)](#)
 5. [Multi-Script](#)
4. When you are done, press the **Save** button to save the command. Your command is created but there is no way to access it yet.

5. You now need to assign the command to a button, hotkey, menu item or alias so you will have a way to access the command.

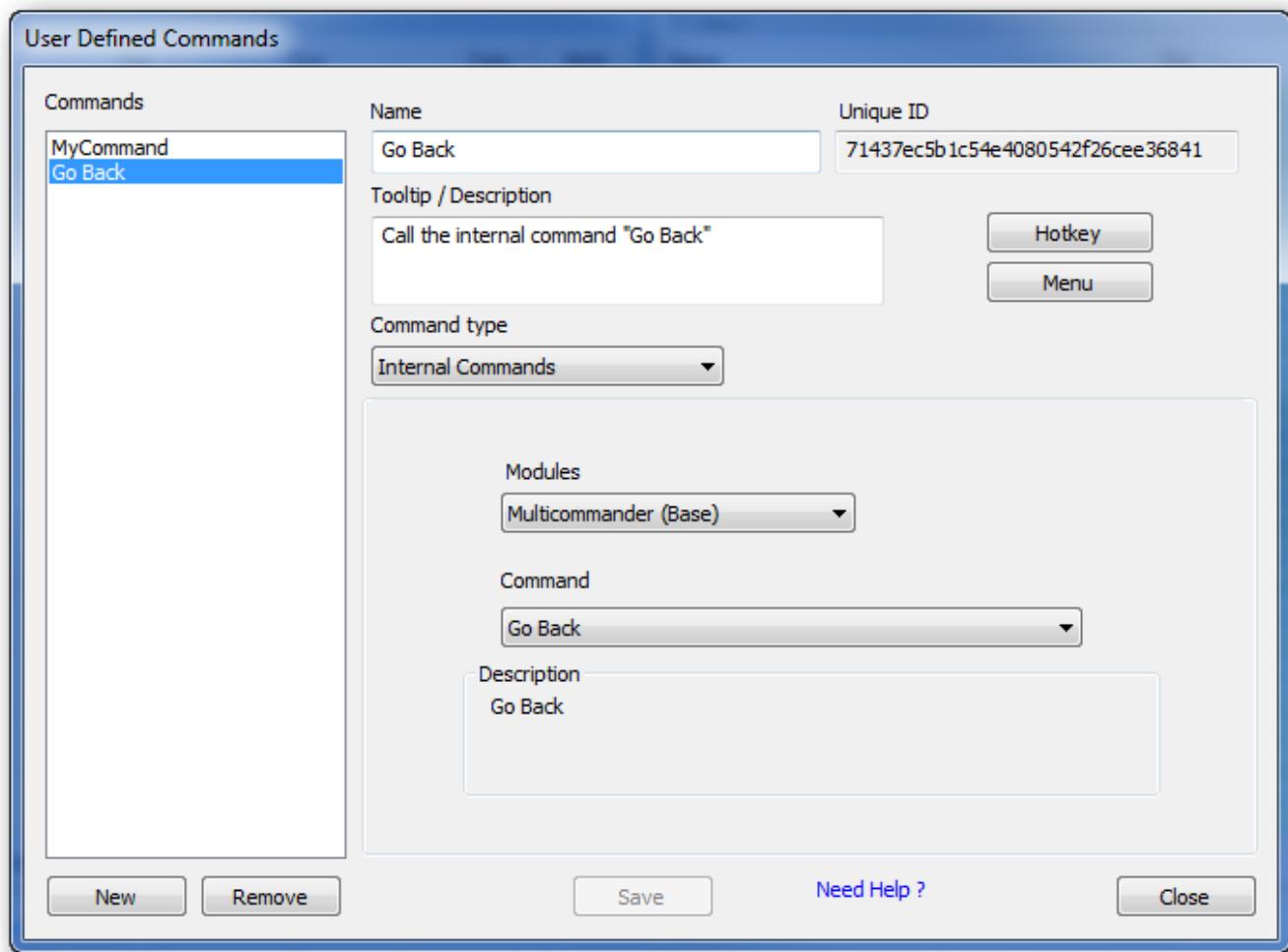
From the [ButtonEditor](#) you can set a button to use "User Defined Command" and then choose which of your commands you want to assign to that button.

From the Menu button you can add your command to a *menu item* that will be shown on the menu bar. The HotKey button will bring up the [Keyboard customization](#) window with your command preselected so that you can assign a hotkey for it.

Using the unique ID that is shown in the Unique ID field (generated when the command was saved) you can use this command from the *commandline* bar. The easiest way to do that is to create an alias for that id with a name you can more easily remember, using the [Alias Editor](#) (found under the configuration menu).

User Defined Commands - Internal Commands

Internal commands are commands that MultiCommander or an Extension/Plug-in has created. For example the menu and tool bars use these commands. They are simple commands that just tell MultiCommander or the Extension that the command has been called, and no parameter or customization for that command can be done.

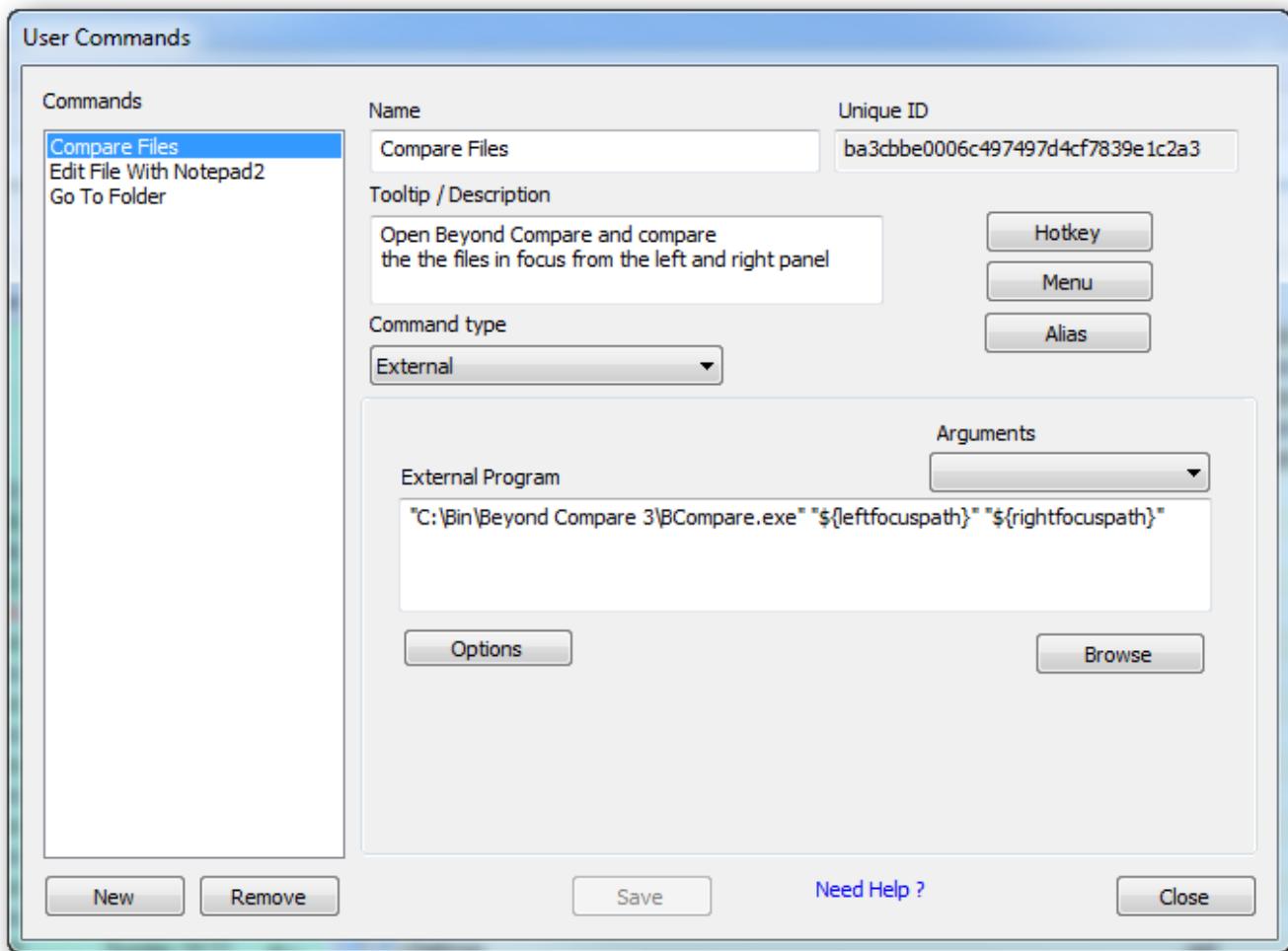


One reason to create a User Defined Command that uses *internal commands* is to create an *alias* for that command so you can call that command from the commandline bar. Another reason would be if you want to assign that command to a button on the button panel.

The command might also be something that is not normally visible in the menu bar, but you would like to put it there. In this case, you can create a User Defined Command that calls the internal command, then add it to the menu bar using the *User Menu Configuration*.

User Defined Commands - External Commands

External commands lets you choose an external program to be launched when the command is called.



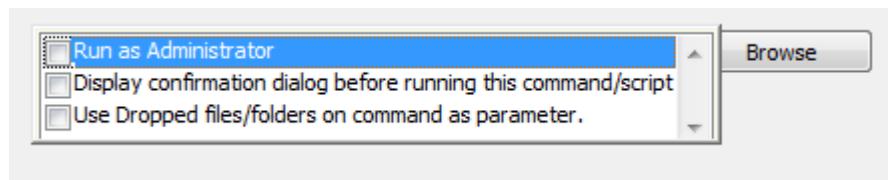
Multi-tags

The command is run through a process that will translate **Multi-Tags** before it is executed. Multi-Tags are specially formatted texts that will be replaced with dynamic information. For example, the MultiTag \${focusfilepath} will be replaced by the full file path to the file currently in focus.

Paths are not quoted by default, so if you need quotes around the path you have to put them around the multi-tag, e.g. "\${focuspath}"

[Read more about Multi-Tags](#)

Options



When you press the **Options** button you will get a list of options you can assign to the command, as shown above.

Run as Administrator

When this option is checked the program will be run as administrator. If you are running as a normal user you will be presented with a UAC dialog asking for permission to run the program as Administrator. You might also need to enter an Administrator login and password, depending on your user account permissions.

If you run with UAC disabled then the system will launch the program as administrator without showing you a UAC dialog.

(This option is not valid on Windows XP and Windows 2003.)

Display Confirmation

This option will display a dialog asking the user to confirm that the command really should be run. This is recommended to be used if the program to be started will change anything on the computer, such as removing or modifying files. (If the command is for example shown in the menu bar, it is very easy to accidentally call the command, and you might make changes to your system when you do not want to.)

Use Dropped Files as Parameters

This is a special option in case the command is assigned to a button on the button panel that accepts files dropped on it. The dropped files will be transformed into parameters for the command, and are accessed with the multi-tag **`${param:<num>}`** .

Command Arguments

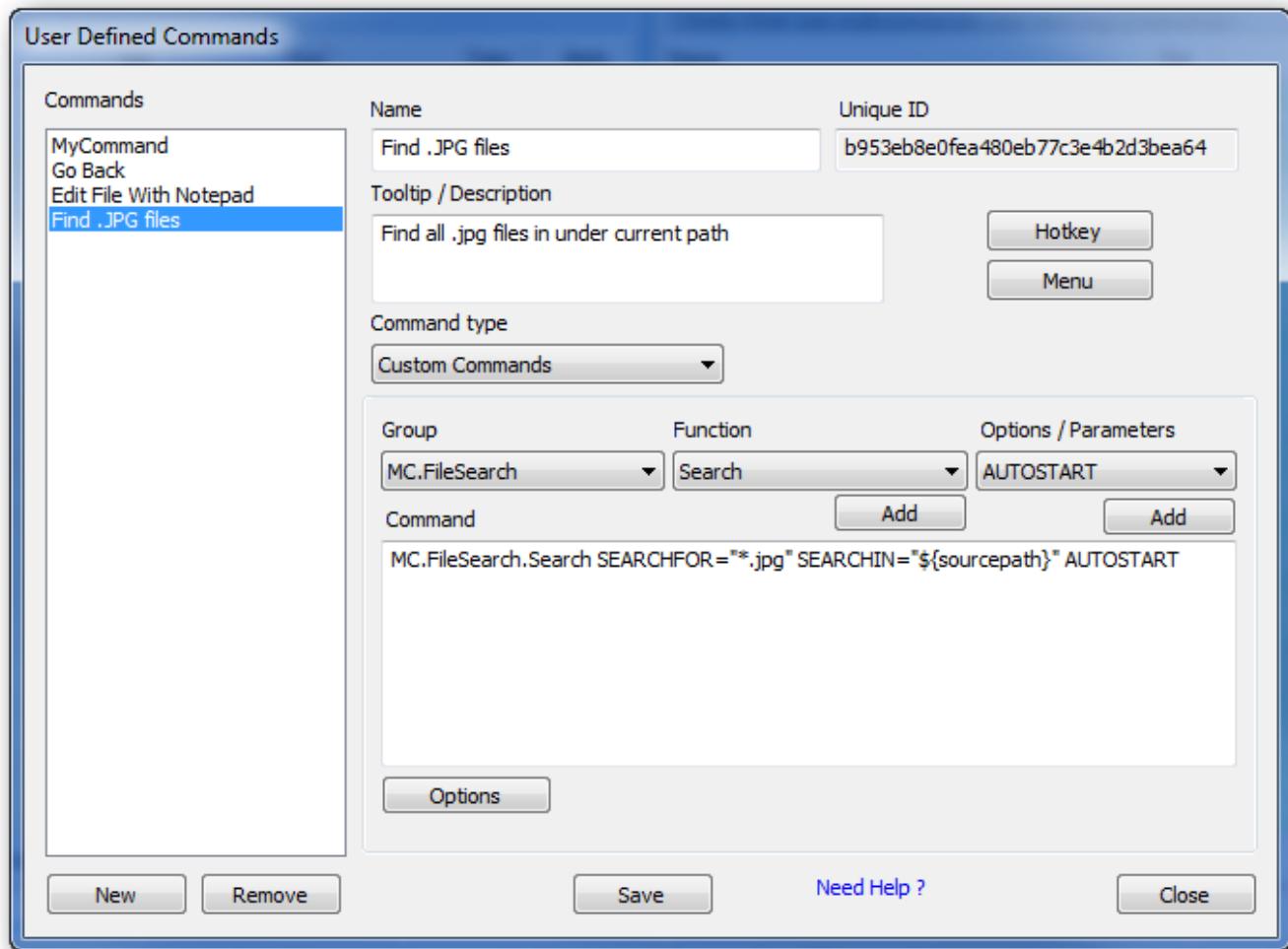
The dialog supports arguments to be sent to the command. These are sent to it if the command is called from the Commandline bar or if files are dropped on a button and the command has enabled the option "Use Dropped files as parameter", or if this command is called from a CustomCommand or MultiScript command.

The parameters are accessed by using the Multi-Tags **`${param:<num>}`** where <num> identifies which argument to use. (**`${param:0}`** will be the first parameter, **`${param:1}`** the second, **`${param:2}`** the third and so on.)

User Defined Commands - Custom Commands

Custom commands are commands that MultiCommander or its Extensions / Plug-ins create. These commands often expose features you already have, but you can now call them with special settings and options available to them.

Custom Commands can also be called from Multi-Script and with that combination you can create *very* advanced commands.



(The example in the image above will start a filesearch for *.jpg file in the path of the current source panel. Furthermore, the search will autostart.)

Create a command

Custom commands are created by choosing *Custom Commands* in the Command Type dropdown of the *User Commands* dialog. To help you create the custom command there are three dropdown boxes where you can choose *Group*, *Function* and *Options/Parameters*. From these boxes you can find all the functions and options that are provided for each group.

The *Group* drop down list shows all the modules that have registered custom commands. The commands are separated to show which module, extension or plug-in they belong to.

When a module is selected the *Function* drop down list is updated to show what functions that module has. When a function is selected the Options/Parameters drop down list is updated with the options and parameters you can send to that function.

If you know the command you want you can write it directly into the editor. You do not have to use the drop down lists; they are just there to help you.

[Custom Commands Examples](#)

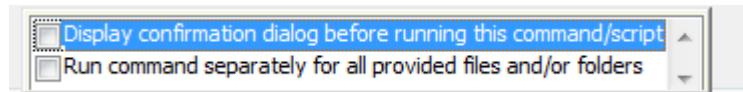
[List And Description of Custom Commands](#)

Multi-tags

The command is run through a process that will translate **MultiTags** before it is executed. Multi-Tags are specially formatted texts that will be replaced with dynamic information. For example, the MultiTag **`${sourcepath}`** will be replaced with the full path of the source panel (the currently active panel).

[Read more about Multi-Tags](#)

Options



When you press the **Options** button you get a list of options that can be assigned to the command, as shown above.

1. Display Confirmation

This option displays a dialog asking the user to confirm that the command really should be run. This is recommended to be used if the program to be started will change anything on the computer, such as removing or modifying files. (If the command is for example shown in the menu bar, it is very easy to accidentally call the command, and you might make changes to your system when you do not want to.)

2. Run Separately

When this option is enabled the command is run separately for each file that is selected, with the Multi-Tag **`${currentfilepath}`** changed to match that file path.

Command Parameters

The script supports parameters to be sent to the command if it is called from the Commandline bar.

The parameters are accessed by using the Multi-Tags **`${param:<num>}`** where **<num>** is the index of the parameter to use. (I.e. **`${param:0}`** will be the first parameter, **`${param:1}`** the second, **`${param:2}`** the third and so forth.

List of Custom Command

MC.Run	Run a external program
MC.RunUserCmd	Run a User Defined Command
MC.RunCmd	Run a Internal or User Defined Command
MC.CmdLineSet	Set a text in the command line field
MC.CmdLineRun	Run a command via the command line field.
MC.View	View file. Viewer is selected by finding a match in the file type setup
MC.Edit	Edit file. Editor is selected by finding a match in the file type setup
MC.SetActivePanel	Set what panel side should be active
MC.SetActiveTab	Set what tab should be active
MC.CloseAllTabs	Close all tabs
MC.XChangeSettings	Change a MultiCommander UI settings. (See online doc for supported settings to change)
MC.BindKey	Bind a keyboard key combination to a command (Eg BindKey CMD=Core.1360 KEY=Ctrl+N)
MC.UnBindKey	Unbind a keyboard key combination to a command (Eg UnbindKey CMD=Core.1360)
MC.ShowFavWindow	Show Favorites Window
MC.ShowFavPopup	Show Favorites Popup menu
MC.Filesystem.Rename	Rename a file or folder. Current file will be used if FILE parameter is not used.
MC.Filesystem.Delete	Delete a file(s) or folder(s). if no PATH parameter is giving the current selected files/folders are used.
MC.Filesystem.Makedir	Create a new folder.
MC.Filesystem.PackFiles	Pack files into an archive (eg .zip file), If no FILES parameter is giving the current selected files/folders are used
MC.Filesystem.Unpack	Unpack archvies into a folder
MC.Explorer.NewBrowser	Open a new explorer panel tab.
MC.Explorer.CloseAll	Close all Explorer Panel tabs.
MC.Explorer.Goto	Show path in the explorer panel.
MC.Explorer.Select	Select files and folders in the Explorer Panel.
MC.Explorer.Deselect	Deselect files and folders in the Explorer Panel.
MC.Explorer.Refresh	Refresh the explorer panel.
MC.Explorer.RefreshTree	Refresh the explorer panel.
MC.Explorer.Copy	Start a Copy operation.
MC.Explorer.Delete	Start a Delete operation.
MC.Explorer.Move	Start a Move operation.
MC.Explorer.Makedir	Create folder.
MC.Explorer.SetViewFilter	View filter for the explorer panel.
MC.Explorer.GetViewFilter	View filter for the explorer panel.

MC.Explorer.SetColumns	Change what columns to show, Change to a defined column set
MC.Explorer.SetViewMode	Change view mode of Explorer Panel that are in focus.
MC.Explorer.SetTabProp	Change tab properties of active explorer panel tab
MC.Explorer.Sort	Change what column data is sorted by
MC.Explorer.SizeFolder	Size current selected folder(s)
MC.Explorer.SetItemFocus	Change focus to specified a file/folder in the current path
MC.Explorer.SetColoringRules	Set Color Rules profile to use. Do not specific any name if you want to use the default color profile
MC.Explorer.RefreshColoringRules	Refresh file coloring for all files and folders
MC.Explorer.ChangeSetting	Change a Explorer Panel settings for the current active tab. Settings changed will NOT be saved
MC.Explorer.Selection.Select	Select items by filter (Support all views)
MC.Explorer.Selection.Unselect	Unselect items by filter (Support all views)
MC.Explorer.Selection.SelectAll	Select all items
MC.Explorer.Selection.UnselectAll	Unselect all items
MC.Explorer.Selection.InvertSelection	Invert selections of items
MC.Explorer.Selection.SaveSelectionToMemory	Save selection to memory
MC.Explorer.Selection.LoadSelectionFromMemory	Load selection from memory
MC.Explorer.Selection.SaveSelectionToFile	Save selection to file
MC.Explorer.Selection.LoadSelectionFromFile	Load selection from file
MC.Explorer.Selection.LoadSelectionFromClipboard	Load selection from clipboard
MC.Explorer.Selection.CompareFoldersForDuplicates	Compare target and source views and select duplicates
MC.Explorer.Selection.CompareFoldersForSelected	Compare target and source views and select selected on target.
MC.Explorer.Selection.CompareFoldersForMissingAndNewer	Compare target and source views and select missing and newest.
MC.Explorer.Selection.CompareFoldersForMissing	Compare target and source views and select missing.
MC.Explorer.Selection.CompareFoldersForNewest	Compare target and source views and select newest item.
MC.FileSearch.Search	Search for files and folders
MC.Special.SelectOldestByName	Select oldest based on file/folder name.
MC_Utils.CreateLink	Create a File System Link, Symbolic, Hardlink, Junction or ShortCut
MC_Utils.SortLines	Sort Lines in text file
MC_Utils.FindAndReplace	Find and Replace in file
MC_CheckSum.Verify	Start CheckSum Tab and verify file(s)
MC_DataViewer.View	Open a file for viewing
MC_PictureTools.Convert	Convert one or many pictures into another picture format
MC_PictureTools.Resize	Resize one or many pictures

MC.Run

Run a external program

Options

ADMIN	Run program as administrator
ARG	Arguments to sent to program
CMD	Path to program to run
SHELL	Let determine how the file should be started.
STARTIN	Path to folder where the program should be started in
WAIT	Wait for program to close before continuing

MC.RunUserCmd

Run a User Defined Command

Options

ID	Guid ID of the command to run. (eg "ID=fg3432f1ab4aBD422DAB3")
----	--

MC.RunCmd

Run a Internal or User Defined Command

Options

ASYNC	Run the command asynchronous.
ID	ID name of the command to run. (eg "ID=Core.1234")

MC.CmdLineSet

Set a text in the command line field

Options

SETOCUS	Change focus to the command line field.
TEXT	Text to set to the command line field.

MC.CmdLineRun

Run a command via the command line field.

Options

CMD	Command to be run as if it was written in the command line field.
-----	---

MC.View

View file. Viewer is selected by finding a match in the file type setup

Options

FILE	Path to file to view, If not specified the file in focus will be used.
VIEWAS	When looking for viewer to use, Select as if file is this file type (eg VIEWAS=".txt")

MC.Edit

Edit file. Editor is selected by finding a match in the file type setup

Options

EDITAS	When looking for editor to use, Select as if file is this file type (eg EDITAS=".jpg")
FILE	Path to file to view, If not specified the file in focus will be used.

MC.SetActivePanel

Set what panel side should be active

Options

PANEL	What panel should be active. LEFT,RIGHT,TOP,BOTTOM,TOGGLE
-------	---

MC.SetActiveTab

Set what tab should be active

Options

PANEL	What panel side the tabs are located in. LEFT,RIGHT,TOP,BOTTOM,ACTIVE,INACTIVE
TAB	Tab to activate. Starting with 1 and going up. (Eg TAB=2) or NEXT,PREV

MC.CloseAllTabs

Close all tabs

Options

LEFTONLY	Close all tabs on the left/top side only
RIGHTONLY	Close all tabs on the right/bottom side only

MC.XChangeSettings

Change a MultiCommander UI settings. (See online doc for supported settings to change)

Options

NAME	Name of setting.
VALUE	Value of the new settings.

MC.BindKey

Bind a keyboard key combination to a command (Eg BindKey CMD=Core.1360 KEY=Ctrl+N)

Options

CMD	Command to bind to key (eg Core.1360)
KEY	Key to bind to (eg Ctrl+N)
SAVE	(Optional) if the keybinding should be saved between restarts

MC.UnBindKey

Unbind a keyboard key combination to a command (Eg UnbindKey CMD=Core.1360)

Options

CMD	Command to bind to key (eg Core.1360)
SAVE	(Optional) if the keybinding should be saved between restarts

MC.ShowFavWindow

Show Favorites Window

Options

SECTION	Favorite section to show (Name or Number)
SHOWPATH	Show target path instead of favorite name

MC.ShowFavPopup

Show Favorites Popup menu

Options

POS	Where to show the popup (Mouse, CenterScreen, CenterMC, "X,Y" pos)
SHOWPATH	Show target path instead of favorite name

MC.Filesystem.Rename

Rename a file or folder. Current file will be used if FILE parameter is not used.

Options

ASKNAME	Show a dialog asking for the new name, If 'NEWNAME' is also specified. It will be used as a suggested name
FILE	Specify file you want to rename
NEWNAME	The new name you want the file/folder renamed into

MC.Filesystem.Delete

Delete a file(s) or folder(s). if no PATH parameter is giving the current selected files/folders are used.

Options

FILTER	Filter what to delete. separate with ';' eg "*.jpg;*.bmp;*.tif"
PATH	The file or folder to delete

MC.Filesystem.Makedir

Create a new folder.

Options

PATH	Full path to the folder you want to create
------	--

MC.Filesystem.PackFiles

Pack files into an archive (eg .zip file), If no FILES parameter is giving the current selected files/folders are used

Options

FILES	What files or folder to pack. Separate multiple files and folder with ;
FILTER	Filter to use. separate filters with space. (eg. *.txt -readme.txt *.info')
METHOD	Pack method to use Zip,7Zip,Tar
SEPARATE	Pack the selected files/folders into separate archives. (TARGET must have wildcard in name. eg "e:\file_*.zip")
TARGET	Target name of the archive. (set to auto to auto-generate name)

MC.Filesystem.Unpack

Unpack archives into a folder

Options

FILE	What archives unpack. Separate multiple archives with ;
FILTER	Filter to use. separate filters with space. (eg. *.txt -readme.txt *.info')
PASSWORD	Password to use if the archive is password protected. (If not set, A dialog will be shown if password is needed during unpacking)
TARGET	Folder where the archives should be unpack into.

MC.Explorer.NewBrowser

Open a new explorer panel tab.

Options

PATH	The path you want to show
SIDE	Open the new explorer panel tab on the LEFT,RIGHT or AUTO
TABCOLORS	Tab Colors to use. Comma separated list TABCOLORS=<background>,<textcolor>,<Gradient (0/1)>,<HotWhenActive (0/1)>
TABLOCK	Lock Tab
TABLOCKALLOWSUBCHANGE	Lock Tab - Allow subpath change
TABNAME	Force the tab to be named using provided text

MC.Explorer.CloseAll

Close all Explorer Panel tabs.

MC.Explorer.Goto

Show path in the explorer panel.

Options

LEFT	Left Path
PATH	The path you want to show
RIGHT	Right Path
SOURCE	Source Path
TARGET	Target Path

MC.Explorer.Select

Select files and folders in the Explorer Panel.

Options

FILTER	Filter of the files and folder to select.
ONLYFILES	Only filter files
ONLYFOLDERS	Only filter folders

MC.Explorer.Deselect

Deselect files and folders in the Explorer Panel.

Options

FILTER	Filter of the files and folder to select.
ONLYFILES	Only filter files
ONLYFOLDERS	Only filter folders

MC.Explorer.Refresh

Refresh the explorer panel.

Options

CACHEUPDATE	Update the cache.
FORCESCAN	Rescan the file system and update filesystem cache.
GUI	Redraw GUI.

MC.Explorer.RefreshTree

Refresh the explorer panel.

MC.Explorer.Copy

Start a Copy operation.

Options

NEWQUEUE	Always start a new file operations queue
NODIALOG	Do not show the 'Copy To' dialog. Use default settings.
USEEXISTINGQUEUE	Use existing file operations queue if one exists

MC.Explorer.Delete

Start a Delete operation.

Options

NEWQUEUE	Always start a new file operations queue
USEEXISTINGQUEUE	Use existing file operations queue if one exists

MC.Explorer.Move

Start a Move operation.

Options

NEWQUEUE	Always start a new file operations queue
NODIALOG	Do not show the 'Move To' dialog. Use default settings.
USEEXISTINGQUEUE	Use existing file operations queue if one exists

MC.Explorer.Makedir

Create folder.

Options

FOLDERNAME	Folder name.
------------	--------------

MC.Explorer.SetViewFilter

View filter for the explorer panel.

Options

FILTER	The ViewFilter you want to set.
--------	---------------------------------

MC.Explorer.GetViewFilter

View filter for the explorer panel.

MC.Explorer.setColumns

Change what columns to show, Change to a defined column set

Options

NAME	ColumnSet name
------	----------------

MC.Explorer.SetViewMode

Change view mode of Explorer Panel that are in focus.

Options

MODE	(1 = Detail, 2 = List, 3 = Thumbnail List, 4 = Thumbnail Detail)
------	--

MC.Explorer.SetTabProp

Change tab properties of active explorer panel tab

Options

RESET	Reset all tab properties. No other parameter used if this is set
TABCOLORS	Tab Colors to use. Comma separated list TABCOLORS=<background>,<textcolor>,<Gradient (0/1)>,<HotWhenActive (0/1)>
TABLOCK	Lock Tab
TABLOCKALLOWSUBCHANGE	Lock Tab - Allow subpath change
TABNAME	Force the tab to be named using provided text

MC.Explorer.Sort

Change what column data is sorted by

Options

COL	Number of the column to sort by. (1-)
COLNAME	Name of the column to sort by(Eg. COLNAME=name)
ORDER	Sort order ASC for ascending, DES for descending

MC.Explorer.SizeFolder

Size current selected folder(s)

Options

ALL	Force sizing of all folders in current location
ONLYFOCUS	Only size folder that are in focus

MC.Explorer.SetItemFocus

Change focus to specified a file/folder in the current path

Options

ITEM	name of the file/folder
------	-------------------------

MC.Explorer.SetColoringRules

Set Color Rules profile to use. Do not specific any name if you want to use the default color profile

Options

NAME	Name of the color profile
------	---------------------------

MC.Explorer.RefreshColoringRules

Refresh file coloring for all files and folders

MC.Explorer.ChangeSetting

Change a Explorer Panel settings for the current active tab. Settings changed will NOT be saved

Options

NAME	Setting name (See Doc for supported names and values)
TOGGLE	Will toggle the named setting (if possible)
VALUE	New Settigns Value (See Doc for supported names and values)

MC.Explorer.Selection.Select

Select items by filter (Support all views)

Options

FILTER	Filter of what to match against
ONLYFILES	Only filter files
ONLYFOLDERS	Only filter folders

MC.Explorer.Selection.Unselect

Unselect items by filter (Support all views)

Options

FILTER	Filter of what to match against
ONLYFILES	Only filter files
ONLYFOLDERS	Only filter folders

MC.Explorer.Selection.SelectAll

Select all items

MC.Explorer.Selection.UnselectAll

Unselect all items

MC.Explorer.Selection.InvertSelection

Invert selections of items

MC.Explorer.Selection.SaveSelectionToMemory

Save selection to memory

Options

NAME Name that you want to save the selection as (Optional)

MC.Explorer.Selection.LoadSelectionFromMemory

Load selection from memory

Options

NAME Name that you want to load the selection from (Optional)

MC.Explorer.Selection.SaveSelectionToFile

Save selection to file

Options

FILE File to save selection to

MC.Explorer.Selection.LoadSelectionFromFile

Load selection from file

Options

FILE	File to load selection from
------	-----------------------------

MC.Explorer.Selection.LoadSelectionFromClipboard

Load selection from clipboard

Options

WILDCARD	Do wildcard selection, (Will insert * before and after every part before doing the selection)
----------	---

MC.Explorer.Selection.CompareFoldersForDuplicates

Compare target and source views and select duplicates

Options

FILES	Compare files
FOLDERS	Compare folders

MC.Explorer.Selection.CompareFoldersForSelected

Compare target and source views and select selected on target.

Options

FILES	Compare files
FOLDERS	Compare folders

MC.Explorer.Selection.CompareFoldersForMissingAndNewer

Compare target and source views and select missing and newest.

Options

FILES	Compare files
FOLDERS	Compare folders

MC.Explorer.Selection.CompareFoldersForMissing

Compare target and source views and select missing.

Options

FILES	Compare files
FOLDERS	Compare folders

MC.Explorer.Selection.CompareFoldersForNewest

Compare target and source views and select newest item.

Options

FILES	Compare files
FOLDERS	Compare folders

MC.FileSearch.Search

Search for files and folders

Options

ATTRIBUTE	Match file attributes (Any of the following characters are valid : ARHSDCE)
AUTOSTART	Autostart the search when the command is run
CASE	Match content case sensitively.
CONTENT	Search for this content in matching files.
EXCLUDE	Path(s) to exclude. if folder with matching name is found that folder is not searched.
HIDE	Hide the 'Find Files' Dialog
NOTTHERE	File is a match if content is missing from the file.
PANEL	What panel to open the search in. (LEFT, RIGHT, ACTIVE, INACTIVE)
SEARCHFOR	Match files or folder that has this substring in the name, (or use wildcard)
SEARCHFOR_REGEXP	Match files or folder against this regular expression.
SEARCHIN	Path(s) to where do the search. separate by space or ;
SIZE	Match files with a file size. Use > or < for more or less then. Eg : ">2000000"
SUBLEVELS	Search only # sub folder down.
TIME	Set a time range from now. (Valid values : TODAY, YESTERDAY, THISWEEK , THISMONTH)
TIMEFROM	Match files with a date from this time. (Format date/time as YYYY-MM-DD HH:MM:SS)
TIMETO	Match files with a date to this time. (Format date/time as YYYY-MM-DD HH:MM:SS)
WORDS	Match content against whole words.

MC.Special.SelectOldestByName

Select oldest based on file/folder name.

Options

SOURCE	Select in source view
TARGET	Select in target view

MC_Utils.CreateLink

Create a File System Link, Symbolic, Hardlink, Junction or ShortCut

Options

ASADMIN	Run as Administrator
LNKSRC	Link Source
LNKTRG	Link Target
LNKTYPE	Link type. 1 = Hardlink, 2 = Junction, 3 = Symlink, 4 = Shortcut

MC_Utils.SortLines

Sort Lines in text file

Options

DATEFORMAT	Date format
DELETE	Overwrite original file.
ENDATWHITE SPACE	End at any whitespace character
ENDCHAR	End character
FILE	File to sort lines in
IGNORECASE	Ignore case
ITEM	Item to sort after
ITEMSEP	Item separator
KEEPBACKUP	Keep a backup of original file (*.bak)
KEEPBOTTOM	Lines at the bottom to keep.
KEEPTOP	Line to start sorting from
LANGUAGEORDER	Use language established sort order
LINEOFFSET	Where on the line it should start sorting from (1 based, First position is 1 not 0)
NATRUALNUMORDER	Natural number order
OVERWRITE	Overwrite if target file already exists
REVERSE	Reverse sort order
SILENT	Silent. No error/done dialogs are shown.
SKIPLEADINGSPACE	Skip leading spaces
SORTAS	Sort as Text,Number or Date (Default is : Text)
SORTBY	Sort lines by Line, SubString or Item. (Default is line)
SUBSTRLEN	Substring length
TARGET	Name of the new file. (Name filter like "*_New.*" , Don't include path)

MC_Utils.FindAndReplace

Find and Replace in file

Options

DELETE	Overwrite original file.
FILE	File to replace text in
FIND	Text to find
FORCEUTF8	Force file format to UTF8 if detected as ascii
IGNORECASE	Ignore case
KEEPBACKUP	Keep a backup of original file (*.bak)
MODE	Find and Replace Mode. One, Many, ManySame, Multiline (Default : One)
OVERWRITE	Overwrite if target file already exists
REPLACEALL	Replace All
REPLACEWITH	Text to replace with
SILENT	Silent. No error/done dialogs are shown.
TARGET	Name of the new file. (Name filter like "*_New.*" , Don't include path)
WILDCARD	

MC_CheckSum.Verify

Start CheckSum Tab and verify file(s)

Options

ALLSELECTED	Verify all selected files, (of files inside selected folders)
FILE	Path to file to verify. If folder then all checksum files in that folder will be used.

MC.DataViewer.View

Open a file for viewing

Options

ASCII	View file as ASCII
BGCOLOR	Background Color
BINARY	View file as BINARY
COLOR	Text Color
FILE	Path to file to view
FONT	Font you want to use.
FONTSIZE	Font size you want to use
FORMAT_1	Add formating. Add FORMAT_2 and so on for more formats.
HEX	View file as HEX
HIGHLIGHT	Highlight words
UNICODE	View file as UNICODE

MC.PictureTools.Convert

Convert one or many pictures into another picture format

Options

DELETE	Delete the original file if it was converted
FILE	File to convert
FILES	List of files to convert, Separate with Semicolon (";")
FOCUS	Convert file in focus only
FORMAT	Target format for conversion , Can be one of the following : PNG,JPG,GIF,BMP,TIF
JPGQUALITY	JPG Quality to use. (0-100) (Only valid if FORMAT is JPG)
OVERWRITE	Overwrite target file if it already exists
SAVEAS	Save new file as. (Only valid if you also used FILE) By default the file name is kept but with the new file extension.
SELECTED	Convert all selected files

MC.PictureTools.Resize

Resize one or many pictures

Options

FILE	File to resize
FILES	List of files to resize, Separate with Semicolon (";")
FOCUS	Resize file in focus only
HEIGHT	Height of target image in pixels or percent Use * to auto calcuate for keeping aspect ratio
OVERWRITE	Overwrite target file if it already exists
REPLACE	Replace original image file.
SAVEAS	Save new file as. (Only valid if you also used FILE) By default the file name is kept but with the new file extension.
SELECTED	Resize all selected files
WIDTH	Width of target image in pixels or percent Use * to auto calcuate when keeping aspect ratio

Custom Commands - Examples

Examples of Custom commands

- [Set View filter for the Explorer panel](#)
- [Select files and folder in the explorer panel](#)
- [Run an external program](#)
- [Pack Files](#)
- [Search For Files](#)

Set view filter

Set a filter to choose what files and/or folders should be visible in the explorer panel:

```
MC.Explorer.SetViewFilter FILTER="*.jpg"
```

```
MC.Explorer.SetViewFilter FILTER="*.jpg *.gif *.png *.tiff"
```

Exclude folders named ".svn":

```
MC.Explorer.SetViewFilter FILTER="-/.svn"
```

Select files and folders

Select files and/or folders in the explorer panel:

```
MC.Explorer.Selection.Select FILTER="*.jpg *.png"
```

Run Application

Run an external application:

```
MC.Run CMD="Notepad.exe"
```

Run an application with parameters:

```
MC.Run CMD="Notepad.exe" ARG="c:\myFile.txt"
```

Run an external application as Administrator:

```
MC.Run CMD="Notepad.exe" ADMIN
```

Let Windows decide how the file should be run (In the example below, Windows will open it in your default browser):

```
MC.Run CMD="http://www.google.com" SHELL
```

Run external application with the file in focus as argument:

```
MC.Run CMD="MyApp.exe" ARG="${focusfilepath}"
```

```
MC.Run CMD="MyApp.exe" ARG="-a -f -i:${focusfilepath}"
```

Pack Files

Pack all files in "D:\MyFiles" with zip into a file named Backup.zip in the target location:

```
MC.Filesystem.PackFiles FILES="D:\MyFiles\*.*" METHOD="Zip"  
TARGET="${targetpath}Backup.zip"
```

Pack all files in "D:\MyFiles" with zip into a file named "Backup-2010-03-10-12:22:22.zip". (The date and time should be the date/time at the moment the command is run) and store the file in the current target location:

```
MC.Filesystem.PackFiles FILES="D:\MyFiles\*.*" METHOD="Zip"  
TARGET="${targetpath}${date:yyyy-MM-dd}-${time:hh:mm:ss}.zip"
```

Search for Files

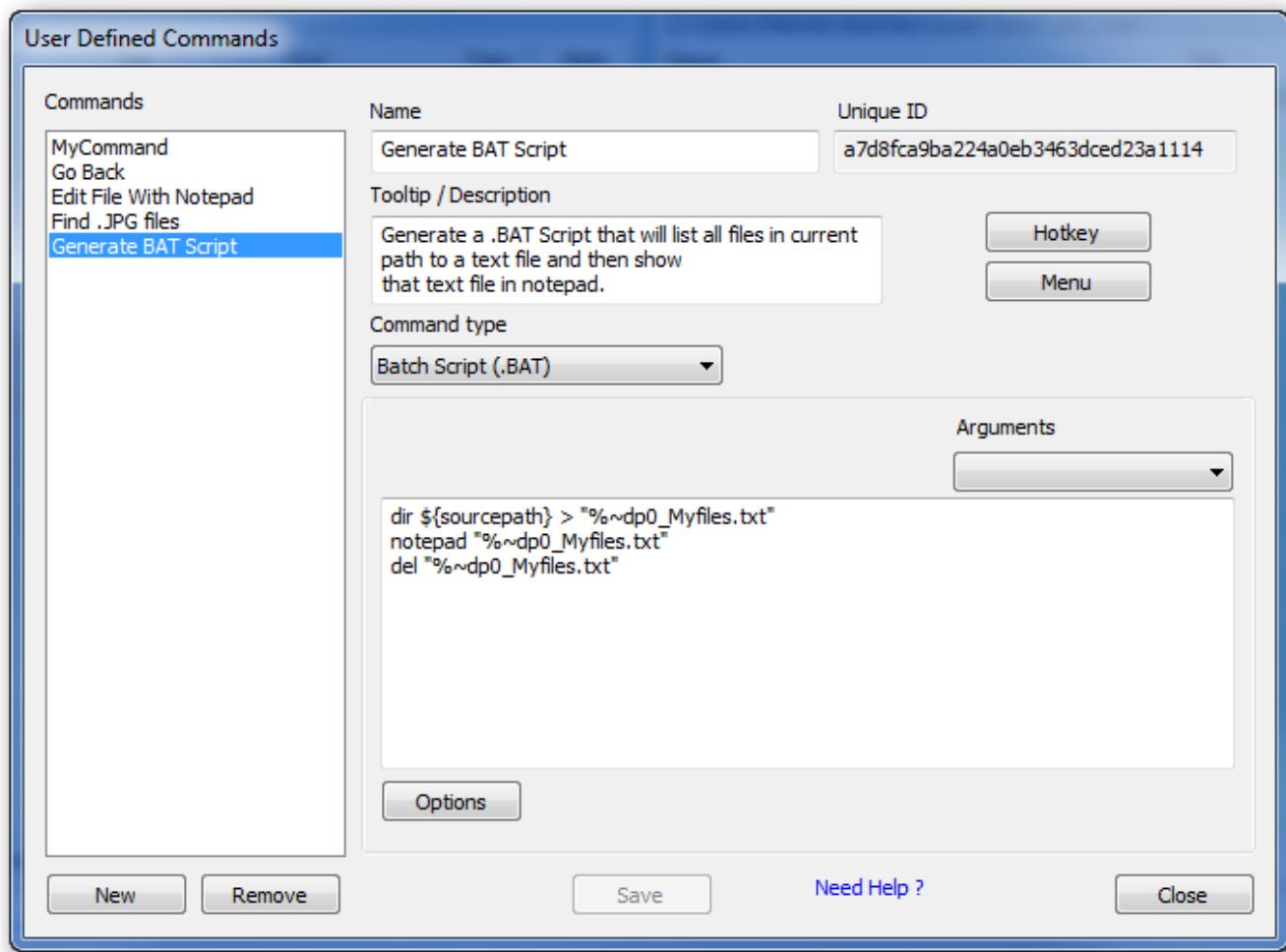
Find all *.jpg and *.tiff files in the current source path. Do not search in the ".backup" folder. Auto start the search when the command is run;

```
MC.FileSearch.Search SEARCHIN="${sourcepath}" SEARCHFOR="*.jpg *.tiff"  
EXCLUDE=".backup" AUTOSTART
```

User Defined Commands - Batch Script (.BAT)

This command will generate a batch script (.bat) and run it.

Batch Scripts are based on the old script system in Windows that goes back to the old DOS days. It is still used and is quite useful, especially if you want to run a lot of external programs one after the other. In addition, you can send the output of one program to be used as input to another.



When a Batch Script command is run a .bat file is generated from this command; the script is then run with its working folder set to the current source path.

Multi-tags

The command is run through a process that will translate *MultiTags* before it is executed. *Multi-Tags* are specially formatted texts that will be replaced with dynamic information. For example, the MultiTag **`\${focusfilepath}`** will be replaced by the full file path to the file currently in focus.

Paths are not quoted by default, so if you need quotes around the path you have to put them around the multi-tag, e.g. **"`\${focuspath}`"**.

[Read more about Multi-Tags](#)

Options

When you press the Option button you will get a list of options you can assign to the command.

(This list of options for Batch Script is not finalized yet)

Command Arguments

The dialog supports arguments to be sent to the command. These are sent to it if the command is called from the Commandline bar or if files are dropped on a button and the command has enabled the option "Use Dropped files as parameter", or if this command is called from a CustomCommand or MultiScript command.

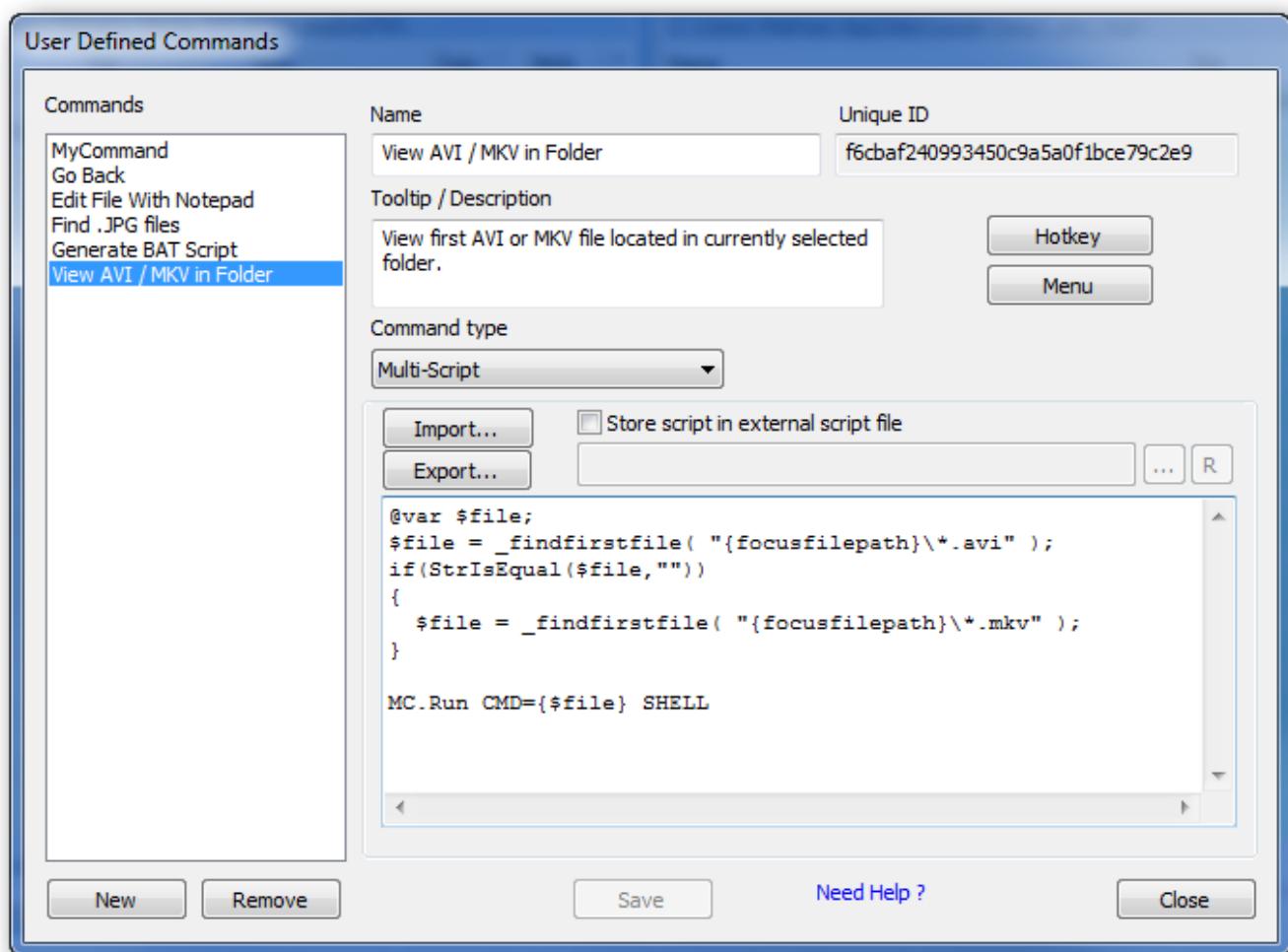
The parameters are accessed by using the Multi-Tags **`${param:<num>}`** where <num> identifies which argument to use. (**`${param:0}`** will be the first parameter, **`${param:1}`** the second, **`${param:2}`** the third and so on.)

User Defined Commands - MultiScript

MultiScript Commands are the most advanced type of User Defined Commands you can create. MultiScript uses the built-in **MultiScript Engine**, which provides a wide variety of functions to use. Furthermore, it can be extended with more functions through Extensions and Plug-ins.

MultiScript functions are low level extensions of functionality that you can find in MultiCommander and its extensions. If you use Custom Commands to customize an existing command, you almost always see a GUI, but from its script you cannot act on errors from within the script. However with MultiScript you have full access to these commands in a more low level way. You can execute different commands depending on conditions, act on errors, work with variables and inline functions and a lot more.

MultiScript functions can also be pure script based functions that extend the script language and make it possible to write even more advanced scripts.



(The example in the image above will find the first *.avi or *.mkv it finds in the folder in focus, and then send that file to the Windows Shell to be opened by the default video player)

Script Engine

The **MultiScript engine** is a simple script engine but is also quite powerful. It has support for all the normal things you want in a script engine such as variables, script defined functions, arrays and more. Furthermore, Extensions and Plug-ins can easily extend the script language with more functions.

There are however some restriction when writing scripts. The Script Engine is a line by line engine, meaning it executes the script one line at the time. Therefore, commands cannot be split into multiple lines. Also it does not prioritize operators, therefore "2 + 4 * 8" will result in 48 and not 34. However it does support parentheses so you can use them to prioritize operations to get the correct behavior. The correct way to write it is then "2 + (4 * 8)".

[Read more on how to write MultiScripts](#)

[List of Built-In Multi-Script Functions](#)

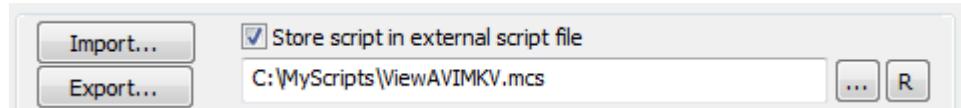
[List of Extensions And Plugin Defined Functions](#)

[MultiScript Examples](#)

Mixing MultiScript and Custom Commands

MultiScripts can call Custom Commands directly as long as there is no other MultiScript syntax before the Custom Command. Therefore it is not possible to directly use a return value from a Custom Command. To use MultiScript expressions or variables inside a Custom Command parameter you need to enclose them in brackets { }, as in the image above {\$file} .

Store Multi-Script in External File



By Enabling this option you can store the script in an external file.

Import / Export

Import will let you open a file and load the content of that file into the editor.

Export will let you save your script into an external file.

Multi-Script Language Syntax

Supported Language Features

- [Language Syntax](#)
- [Reserved Keywords](#)
- [Data Types](#)
- [Variables](#)
- [Constants](#)
- [Escape Sequence](#)
- [Arrays](#)
- [Operators](#) (+, /, *, -, ...)
- [Index Operator](#)
- [Assignment Operator](#) (=, +=, -=)
- [Compare Operators](#) (==, >, <, >=, <=)
- [Negation Operator](#) (!)
- [Special Path Operators](#) (^)
- [IF - Condition](#)
- [FOR - Loop](#)
- [WHILE - Loop](#)
- [Functions](#)
- [Class](#)

Language Syntax

MultiScript is a line-by-line script engine, meaning that every script command must be on one line. A command cannot be split across multiple lines.

Every script command should end with ";". (This is not required yet, but it will be in the future; when that occurs, it will then be possible to split commands over multiple lines.)

New scope is declared with { } and is used for conditions, loops, functions and classes.

Example:

```
$len = StrLen($str);
if( $len > 10 )
{
    $str = "Too long";
}
```

Lines starting with // or # are handled as comment lines and are ignored.

Comments are **not** allowed to be on the same line as valid code.

```
$len = StrLen($str); // Get string length (Comments and code on same line are not
allowed)
```

The script engine evaluates expressions from left to right, but parentheses are evaluated first. Because of this, parentheses **must** be used if multiple expressions are used within a line, to guarantee that they are evaluated in the correct order.

```
@var $result;

// INCORRECT
$result = 10 + 5 * MyFunc_GetMultiplyFactor();
```

```
// CORRECT
$result = 10 + (5 * MyFunc_GetMultiplyFactor());
```

Function parameters can be variables, constants or expressions. Function parameters are evaluated *before* they are sent to the function. Variables are always passed *by value*, meaning a copy of the variable is actually passed.

```
@var $result;

$result = CalculateResult( 4 + 5, "Parameter2");

$result = CalculateResult( MyFunc_GetValue("height") + 2);
```

Reserved Keywords

The following words are *reserved* and cannot be used for variables or function names.

if	for	while	break
continue	return	function	class
new	@var		

Data Types

Variables and Constants can be of different *data types*. Valid data types are *String*, *Number*, *Array*, *Object* and *Handle*.

Conversion from string to number and number to string are often done automatically, making it possible to concatenate a number variable or constant to a string.

Variables

Variables are declared with the special keyword [@var](#). The variable name must start with a \$, and can only have the characters a-z , A-Z and 0-9 in the name. Special characters are not allowed.

```
@var $v1;
```

Multiple variables can be declared on the same line.

```
@var $v1 , $v2 , $v3;
```

Variables can also be assigned a value when declaring them, either from a constant value or from a function call.

```
@var $v1 = 0;
@var $v2 = MyFunc();
```

Declaring a variable with a name that already exists is allowed as long as the variable exists in a different scope.

```
@var $v1 = 0;
if( $v1 == 0 )
{
@var $v1;

// $v1 outside the if statement is not changed
$v1 = 20;
}
```

Arrays must be declared with a size, although you can manipulate the size later.

```
@var $arr[2];
```

Arrays can also be declared with values. (In this case the size is implicit.)

```
@var $arrStr[] = {"String1", "String2", "String3" };
@var $arrNum[] = {1, 2, 3};
```

Constants

Variables can be declared using number and string constants. Functions can also return constant arrays.

```
@var $v1 = 2;
@var $v2 = "string";
if($v2 == "string")
{
...
}
```

Escape Sequence

To be able to type constant strings containing special characters such as the *newline* character, *double quotation mark*, tab, etc., they need to be written in a special way, using an *escape sequence*.

Escape sequence consists of a backslash (\) followed by a letter. Each Escape sequence is regarded as a single character.

Escape Sequence	Represents
\n	New line
\r	Carriage return
\t	Tab
\"	Double quote mark ("")
\\\	Backslash (\)

If you want an actual backslash in your constant string you need to write "\\\". This will translate into "\\".

```
$var = "C:\\Temp\\MyFolder\\File.txt";
```

Escape sequences are only translated inside string constants enclosed in double quotes. Using single quoted strings will not translate escape sequences. Using single quoted strings may make the code more readable, but some characters like newline and single (or double) quotes cannot be used inside those strings.

```
$var = 'C:\Temp\MyFolder\File.txt';
```

Arrays

Arrays can hold multiple values. The values can be Strings, Numbers, Handles or other arrays.

The *index* operator ([]) is used to access array items. It can be used to get or set values.

If using it to set a value you must be sure that you are not trying to set an item outside the bounds of the array.

When accessing array items the index value is 0-based. Thus the first item is index 0, the second is index 1. and so forth.

```
@var $arr[3];
$arr[0] = 2;
$arr[1] = 6;
$arr[2] = 8;

//This will fail (out-of-bounds).
$arr[3] = 10;
```

When assigning one array to another, a copy of each of the items is created.

```
@var $arr[3];
$arr[0] = 2;
$arr[1] = 6;
$arr[2] = 8;

@var $arr2;
$arr2 = $arr;
```

Array Functions

There are several built-in array functions.

[Read more about array functions](#)

```
@var $arr = $arr[2] = {"item0", "item1"};
@var $count = arrayCount($arr);
// $count is 2

// Replace item 0 and 1
$arr[0] = "item1";
$arr[1] = "item2";

@var $str = $arr[1];
// $str is now "item2";

// Add and increase size of array
arrayAdd($arr, "item3");
$count = arrayCount($arr);
// $count is now 3

@var $idx = arrayFind($arr, "item3");
// $idx is 2

arrayRemove($arr, 0);
$count = arrayCount($arr);
// $count is now 2 and item "item1" has been removed
```

Operators

The usual arithmetic operators (+, -, *, /) are supported.

```
@var $v1, $v2;
$v1 = 5 + 2;
$v1 = $v1 - 1;
$v2 = $v1 * 2;
$v1 = $v2 / $v1;
```

Since the script engine does not do any operator prioritization, parentheses **must** be used if multiple operators are used on one line to guarantee that they are evaluated correctly. On number values, mathematical operations are used. The **+** operator for two strings will concatenate the right string with the left.

```
@var $result;

$result = 5 + 4 * 2;
// $result is now 18

$result = 5 + (4 * 2);
// $result is now 13

$result = "StringA" + " " + "String B";
// $result "StringA StringB";
```

Index Operators

The *index operator* (`[]`) is supported for both Arrays and strings. On Arrays it is used to get/set an array item. On strings it is used to get/set a character in the string.

```
@var $str = "MyString";
$str[0] = "Z";
// $str is now "ZyString";

$str = "abcdef";
@var $str2 = "12345";
$str[1] = $str2[3];
// $str is now "a4cdef";
```

Assignment Operators

The *assignment operators* (`=`, `+=`, `-=`) are used to set the left side from the result of the expression on the right side.

```
@var $val = 1;
$val = 5;

// += is increment
$val += 2;
// $val is now 7

// -= is decrement
$val -= 2;
// $val is now 5
```

Compare Operators

Compare operators are used to compare two values. The result of a compare operator is 0 for false or 1 for true. The following compare operators are supported: == (equal), > (greater than), < (less than) , >= (equal or greater than), <= (less than or equal)

```
@var $val = 3;
$val = $val > 2;
// $val is now 1
$val = 4;
if( $val >= 4 )
{
...
}
```

Negation Operators

The *negation* operator (!) is used to negate a true/false value. The result of a negation is 0 or 1.

```
@var $val = 1;
if( !$val )
{
// if $val is false
...
}
```

Special Path Operator

"^" is used as a special path operator. It will assume that both the left and right side are path items, and will insert a path separator (\) if needed.

```
@var $path;
$path = "C:\\Path";
$path = $path + "MyFile.txt";
// $Path is now "C:\\Path\\MyFile.txt"

$path = "C:\\Path";
$path = $path ^ "MyFile.txt";
// $Path is now "C:\\Path\\MyFile.txt"

$path = "C:\\Path\\";
$path = $path ^ "\\MyFile.txt";
// $Path is now "C:\\Path\\MyFile.txt"
```

IF - Condition

The *IF condition* will evaluate its test expression. If the expression is greater than or equal to 1 (true) then it will execute the scope after the if statement. If it is less then 1 (false) it it will jump over the scope or go to the else statement if present.

New variables created inside the IF scope will only exist within that scope. New scope variables can be named the same as variables in the outer scope.

```
@var $v = 2;
@var $s = 1;
if($v == 2)
{
    // Local scope variable
    @var $s = 10;
    $v = 1;
}
else if( $v >= 3)
{
    $v = 3;
}
else
{
    $v = 4;
}

if( MyFync($v) )
{
    $v = 6;
}
```

FOR - Loops

A **FOR** loop statement has three (3) expressions that are evaluated.

```
for( <init expression> ; <condition expression> ; <loop expression>)
{
    <statement>
}
```

The **Init Expression** is evaluated once, before any other element of the **for** statement.

The **condition expression** is evaluated and tested each time it is about to enter the **for** scope.

The **loop expression** is evaluated each time it has ended a scope execution of the **for** loop.

The Initialization variable in the init-expression must be declared outside of the **for** statement.

```
// Valid for expression
@var $n;
for( $n = 0; $n < 5; $n++)
{
    ...
}

// Invalid for expression
for(@var $n = 0; $n < 5; $n++)
{
    ...
}
```

A **for** loop terminates when the condition expression evaluates to 0 (false), or a **break** or **return** statement is executed within the **for** statement. A **continue** statement in a **for** loop terminates only the current iteration.

```
@var $n;
for( $n = 0; $n < 5; $n++ )
{
    if($n == 2)
    {
        continue;
    }
    if($n == 4)
    {
        // break the loop
        break;
    }
}
```

WHILE - Loops

A **While** loop has one (1) expression that is evaluated every time through the loop until it is 0 (false).

```
while( <expression> )
{
    <statement>
}
```

A **while** loop can terminate when a **break**, or **return** within the statement body is executed. Use **continue** to terminate the current iteration without exiting the **while** loop. **Continue** passes control to the next iteration of the **while** loop.

```
@var $n = 10;
while( $n > 0 )
{
    $n--;
}

@var $success = 1;
while( $success )
{
    $success = MyFunc_DoWork();
}
```

Functions

With the **function** keyword it is possible to define a function that is called by the script.

The function must be defined before it is called. Therefore it is recommended to define all the script functions first in the script.

Functions support arguments. Parameters sent to a function will be passed *by value*, meaning they are passed as copies. If a variable passed as a parameter to a function is changed within the function, the original variable is *not* changed.

The function will return when it reaches the end of the function definition. It will by default always return 1 in this case. If some other return value from the function is wanted then a **return** statement must be used.

```
function MyFunc( $a1 , $a2 )
{
    @var $result = $a1 + $a2;
    $result = $result * 2;
    return $result;
}

@var $n = 0;
$n = MyFunc( 2 , 6 );
```

The **return** statement can also be used to exit from the function *before* it reaches the end.

```
function MyFunc( $a1 , $a2 )
{
    @var $result = $a1 + $a2;
    if( $result > 10 )
    {
        return $result;
    }
    $result = $result * 2

    return $result;
}

@var $n = 0;
$n = MyFunc( 4 , 8 );
```

Classes

With the **class** keyword it is possible to define a *class*. A class is a group of variables and functions that are collected together within an *object*. Objects in MultiScript are very simple variations of the "object" you find in languages like C++, but without the support for public/private. Everything is public here.

A class is created from a definition using the **new** keyword. This will create a new instance of a class.

It is not possible to delete objects while running the script.

To access a function or variable in an object a dot (".") is used to as separator between the object variable and the class variable name or function name.

When referring to a variable in an object variable the \$ on the class variable is not written out.

```
// INCORRECT
@var $n = $obj.$a;

// CORRECT
@var $n = $obj.a;
```

```
class MyClass
{
    @var $a = 0;
    @var $b = 1;

    function( $v1 , $v2 )
    {
        $b = $v1 + $v2;
        return $b;
    }
}

@var $obj = new MyClass;
@var $result = $obj.MyFunc( 2 , 5 );
@var $n = $obj.a;
```

MultiScript - Core built in functions

Data Type Description

Type	Description
<num>	A Variable or constant of type <i>number</i>
<str>	A variable or constant of type <i>string</i>
<array>	A variable or constant of type <i>array</i>
<handle>	A Variable of type <i>handle</i>
<var>	Any variable

Script Language Functions

Return Value	Function Definition	Description
<num>	IsTypeString(<var>)	Returns 1 if <var> is of type string.
<num>	IsTypeNum(<var>)	Returns 1 if <var> is of type number.
<num>	IsTypeArray(<var>)	Returns 1 if <var> is of type array.
<num>	IsTypeHandle(<var>)	Returns 1 if <var> is of type handle.
<num>	strtonum(<str>)	Converts a number of type string to number
<str>	numtostr(<num>)	Converts a number of type num to string

See Also

[Array Functions](#)
[String Functions](#)

Multiscript - Array functions

Array is an internal data type in Multiscript.

An array can contain multiple values of type **string** or **number**.

The following is a list of array functions that are supported:

array initialization	Initialize an array with values when declared.
arrayCount	Count the items in an array.
arrayAdd	Add an item to an array.
arrayRemove	Remove an item from an array.
arrayFind	Find a string item in an array.
arrayIFind	Find a string item in an array, Ignoring case differences.
ArraySort	Sort items in an array.
StrLines2Array	Transform a string with multiple lines into an array of lines.
StrLinesArray2String	Transform an array of strings into a multi-line string.
StrTokenize2Array	Tokenize a string into an array.
operator []	Get or Set an array item at a specified index position.

See also

[SaveArray](#)
[LoadArray](#)

Initialize Array

You can **initialize** an array when it is declared.

```
@var $arr[] = {<str>|<num>, <str>|<num> , ... , .... };
```

Parameters

str|num

String or Number values to insert into the array

Return value

An array with values inserted into it.

Example

```
@var $arr[] = {"String1", "String2", "String3"};
@var $len = arrayCount($arr);
// $len is 3.
```

arrayCount

Count items in an array.

```
<num> arrayCount( <array> input );
```

Parameters

input

array to count items in.

Return value

The number of items there are currently in the array.

Example

see 'Initialize Array'

arrayAdd

Add an item to an array

```
<none> arrayAdd(<array> input, <str>/<num> value);
```

Parameters

input

Array to add value into

value

String or Num value that should be added to the array

Return value

Nothing

arrayRemove

Remove an item from the array

```
<none> arrayRemove(<array> input, <num> index);
```

Parameters

input

Array to remove item from.

index

Remove item from this index position in the array.

Return value

nothing

arrayFind

Search an array for a string value.

```
<num> arrayFind(<array> input, <str> string);
```

Parameters

input

Array to be searched for the string

string

The string item to look for

Return value

Index position of the found string, or -1 if the string is not found.

arrayIFind

Find a string value in array, ignoring case differences

```
<num> arrayIFind(<array> input, <str> string );
```

Parameters

input

Array to be searched for the string

string

The string item to look for

Return value

Index position of the found string, or -1 if string is not found.

ArraySort

Sort items in an array

```
<arr> ArraySort(<arr> arr);
```

Parameters

arr

array that should be sorted.

Return value

a new array that are sorted

Example

```
@var $arr[] = {3, 2, 5};
@var $arrNew = ArraySort($arr);
// arrNew = 2,3,5

@var $arr[] = {"DD", "CC", "AA"};
@var $arrNew = ArraySort($arr);
// arrNew = "AA", "CC", "DD"
```

StrLines2Array

Transform a string with multiple lines into an array.

```
<array> StrLines2Array(<str> input);
```

Parameters

input

Multi-line string. (All forms of CR/LF are supported.)

Return value

An array where the strings in *input* have been split at the line breaks.

StrLinesArray2String

Transform an *array* of strings into a multi-line array.

```
<str> StrLinesArray2String(<array> input, [<num> eol]);
```

Parameters

input

Array of strings

eol (Optional)

what kind of linebreak to use:

0 = CRLF (Windows) (This is used if not specified)

1 = LF (Linux)

2 = CR (Mac)

Return value

A multi-line string where the strings in the array *input* have been appended together with line breaks inserted between the strings. The line break type used is determined by the *eol* value.

Example

```
@var $arr[] = {"String1", "String2", "String3"};
@var $strLines = StrLinesArray2String($arr);
@var $strLinesLF = StrLinesArray2String($arr, 1);
```

StrTokenize2Array

Tokenize a string into an array, splitting it at specified delimiters.

```
<array> StrTokenize2Array(<str> input, <str> delimiter);
```

Parameters

input

String to tokenize.

delimiter

characters to split the string

Return value

An array of strings based on the string *input* after it is split. It is split wherever there are characters that match the characters in *delimiter*

operator []

Index operator for setting or getting a value from an array

```
<any> operator [<num>] <any>;
```

Parameters

[<num>]

at what index position to set or get a value.

Return value

The item at the selected index position.

Example

```
@var $arr[] = {"Str1", "Str2", "Str3"};
$arr[1] = "Frog";
// string now contain the strings "Str1","Frog","Str3";
@var $str = $var[2];
// $str now contain the value "Str3"
```

Multiscript - String Functions

The following string functions are supported in Multiscript:

StrLen	Return the length of a string
StrSub	Return a substring
StrFind	Return the position of a substring
StrRFind	Return the position of a substring using reverse find. (Back to front search)
StrReplace	Replace a substring in a string
StrToUpper	Return a string in uppercase
StrToLower	Return a string in lowercase
StrTrim	Return a string that has been trimmed both at the beginning and end.
StrTrimLeft	Return a string that has been trimmed on the left (beginning).
StrTrimRight	Return a string that has been trimmed on the right (end).
StrSplit	Split a string and return the separated strings as an array.
StrCompareNoCase	Return the difference of two strings ignoring case differences.
operator ==	Return the difference of two strings. (Does a case sensitive comparison.)
StrIsEqual	Return 1 if two strings are equal, otherwise 0. (Case sensitive.)
StrIsEqualNoCase	Return 1 if two strings are equal, ignoring case differences.
StrIsWildMatch	Return 1 if two strings match using a case sensitive wildcard algorithm.
StrIsWildMatchNoCase	Return 1 if two strings match using a case insensitive wildcard algorithm.
StrIsRegExpMatch	Return 1 if two strings match using regular expressions. (Case sensitive.)
StrIsRegExpMatchNoCase	Return 1 if two strings match using regular expressions, ignoring case difference.
StrRegExpFind	Find a substring using a regular expression.

See also

[SaveStringToFile](#)
[LoadStringFromFile](#)

StrLen

Return the length of a string.

```
<num> StrLen( <str> string );
```

Parameters

string

string to get the length of.

Return value

Returns the number of characters in *string*.

Example

```
@var $str = "MyString";
@var $len = StrLen( $str );
// $len is 8
```

StrSub

Return a substring beginning at a specified location and having a specified length

```
<str> SubStr(<str> input, <num> offset, <num> count);
```

Parameters

input

A string from which the substring is extracted.

offset

The starting position of the desired substring, The index of the first character in the string is zero.

length

The number of characters to include in the returned substring, or -1 to include all characters through the end of the string

Return Value

A string that is the substring of *input* string at the specified location.

StrFind

Find the position of a substring in a string.

```
<num> StrFind(<str> input, <str> find, <num> offset);
```

Parameters

input

The string to be searched

find

The string to be found

offset

Initial position to begin the search

Return Value

The position in the *input* string in which the search string *find* is found, or -1 if *find* string is not found. (The first character position is 0.)

StrRFind

Find the position of a substring in a string search the string in reverse. (Back to front).

```
<num> StrRFind(<str> input, <str> find);
```

Parameters

input

The string to be searched

find

The string to be found

Return Value

The position in the *input* string in which the search string *find* is found, or -1 if *find* string is not found. (The first character position is 0.)

StrReplace

Replace a substring within a string.

```
<str> StrReplace(<str> input, <str> find, <str> with );
```

Parameters

input

A string from which to replace the substring in.

find

The substring to be replaced.

with

The string that the substring should to be replaced with.

Return Value

The string *input* with the substring *find* replaced by the string *with*.

Example

```
@var $str = "MyString";
@var $str2 = StrReplace( $str, "My" , "New" );
// $str2 is "NewString"
```

StrToUpper

Transform a string into all uppercase characters.

```
<str> StrToUpper(<str> input);
```

Parameters

input

String that should be transformed.

Return Value

An uppercase version of the string *input*.

StrToLower

Transform a string into all lowercase characters.

```
<str> StrToLower(<str> input);
```

Parameters

input

String that should be transformed..

Return Value

A lowercase version of the string *input*

StrTrim

Removed unwanted characters from both the left and right side of a string.

```
<str> StrTrim(<str> input, <str> trimchars);
```

Parameters

input

String that should be trimmed

trimchars

String specifying the characters to be removed.

Return Value

A string where all the characters in *trimchars* have been removed from both the left (beginning) and the right (end) side of the string *input*.

StrTrimLeft

Remove unwanted characters on the left side of the string.

```
<str> StrTrimLeft(<str> input, <str> trimchars);
```

Parameters

input

String that should be trimmed.

trimchars

String specifying the characters to be removed.

Return Value

A string where all the characters in *trimchars* have been removed from the left (beginning) side.

StrTrimRight

Remove unwanted characters on the right side in the string.

```
<str> StrTrimRight(<str> input, <str> trimchars);
```

Parameters

input

String that should be trimmed.

trimchars

String with all the characters that should be removed.

Return Value

A string where all the characters in *trimchars* have been removed from the right (end) side.

StrSplit

Split a string into an array of strings.

```
<array> StrSplit(<str> input, <str> delimiter);
```

Parameters

input

A string that should be split.

delimiter

String containing a character that will be used as a delimiter.

Return Value

The string *input* split into multiple strings in an array of type string. The string is split wherever the characters in the string *delimiter* are located.

StrCompareNoCase

Compare strings, Ignoring case difference

```
<num> StrCompareNoCase(<str> string1, <str> string2);
```

Parameters

string1

String to compare.

string2

String to compare.

Return Value

A number showing the differences between the strings *string1* and *string2*. The return value indicates the lexicographic relation of *string1* to *string2*.

< 0 if *string1* is less than *string2*.

0 if *string1* is equal to *string2*.

> 0 if *string1* is greater than *string2*.

Operator ==

Compare string (case sensitive).

```
<num> <str> == <str>;
```

Parameters

string1

String to compare.

string2

String to compare.

Return Value

Return the differences of the string *string1* and *string2*. The return value indicates the lexicographic relation of *string1* to *string2*.

< 0 if *string1* is less than *string2*.

0 if *string1* is equal to *string2*.

> 0 if *string1* is greater than *string2*.

Example

```
@var $str1 = "ABCDEF";
@var $str2 = "ABCDEF";
@var $val = $Str1 == $Str2;
// $val == 0
```

StrIsEqual

Check whether two strings are Equal

```
<num> StrIsEqual(<str> string1, <str> string2);
```

Parameters

string1

String to compare.

string2

String to compare.

Return Value

A value of 1 if *string1* is equal to *string2*, 0 if the strings are not identical. (Case sensitive.)

StrIsEqualNoCase

Check whether two strings are Equal, ignoring case differences.

```
<num> StrIsEqualNoCase(<str> string1, <str> string2);
```

Parameters

string1

String to compare.

string2

String to compare.

Return Value

A value of 1 if *string1* is equal to *string2*, 0 if the strings are not identical. Ignore case differences when comparing strings.

StrIsWildMatch

Check if a string matches a wildcard criteria.

```
<num> StrIsWildMatch(<str> input, <str> wildcard);
```

Parameters

input

String to check

wildcard

Wildcard string to check against.

Return Value

1 if the *wildcard* string matches the string *input*, 0 If no match is found.
wildcard is a string containing "*" (for anything) and "?" (for any character).

Example

```
@var $str = "123def567";
// If the character sequence "def" is in the string $val1 will be 1
@var $val1 = StrIsWildMatch( $str , "*def*");

// The two first characters can be anything,
// But the third character must be a 3.
// And anything can be after it. then $val1 will be 1
@var $val1 = StrIsWildMatch( $str , "??3*");
```

StrIsWildMatchNoCase

Check if a string matches wildcard criteria, ignoring case difference

```
<num> StrIsWildMatchNoCase(<str> input, <str> wildcard);
```

Parameters

input

String to check

wildcard

Wildcard string to check against.

Return Value

1 if *wildcard* matches the string *input*, 0 If no match is found.

Wildcard is a string containing "*" (for anything) and "?" (for any character). Case differences between the wildcard string and the *input* string will be ignored.

StrIsRegExpMatch

Check if a string matches a regular expression.

```
<num> StrIsRegExpMatch(<str> input, <str> regexp);
```

Parameters

input

String to check

regexp

Regular expression to check against.

Return Value

1 if the regular expression in *regexp* matches the string *input*, 0 if no match is found.

StrIsRegExpMatchNoCase

Check if a string matches a regular expression. Ignore case difference

```
<num> StrIsRegExpMatchNoCase(<str> input, <str> regexp);
```

Parameters

input

String to check

regexp

Regular expression to check against.

Return Value

1 if the regular expression in *regexp* matches the string *input*, 0 if no match is found. Case differences are ignored.

StrRegExpFind

Find substring using regular expression.

```
<num> StrRegExpFind(<str> input, <str> regexp);
```

Parameters

input

String to find a substring in

regexp

Regular expression to find the substring.

Return Value

Position of where inside the string *input* a match was found, or -1 if no match was found

Example

```
@var $str = "Program v3.10";
@var $len = StrRegExpFind( $str , ".v[0-9].[0-9][0-9].");
// $len is 9
```

MultiScript - Core Functions - RowList functions

The *RowList functions* are special functions to make it easier to work with text lists.

Example

Suppose the content of the file "d:\\temp\\list.txt" looks like the following:

Person Z	0202-1111111	Fake Street 1
Person A	0202-2222222	Fake Street 2
Person B	0303-3333333	Fake Street 3
Person C	0202-4444444	Fake Street 4
Person D	0202-5555555	Fake Street 5

The columns are separated with a tab character "\\t".

This script will load that file into a rowlist and then get every column item one by one.

```
@var $list = RowListLoad("d:\\temp\\list.txt", "\\t");
@var $rowcnt = RowListCount($list);
@var $idx = 0;
for( $idx = 0; $idx < $rowcnt; $idx++)
{
    @var $Name    = RowListItem($list, $idx, 0));
    @var $Phone   = RowListItem($list, $idx, 1));
    @var $Street  = RowListItem($list, $idx, 2));
}
```

Row List Functions

RowListLoad	Load a row list
RowListClose	Close a row list
RowListCount	Get the number of rows in the row list
RowListColumnCount	Get the number of columns
RowListArray	Return a specified row as an array
RowListItem	Get a row item.
ReplaceTagsInString	Replace the tags in a string with values from a row list.

RowListLoad

Load a file as a rowlist.

```
<handle> RowListLoad(<str> filename, <str> delimiter);
```

Parameters

filename

File to load as a rowlist

delimiter

Columns are split using this character.

Return value

A handle to a rowlist, or 0 if it failed to load.

RowListClose

Close a rowlist

```
<num> RowListClose(<handle>);
```

Parameters

handle

Handle to a rowlist

Return value

1 if successful, 0 if failed.

RowListCount

Count how many rows there are in the list

```
<num> RowListCount(<handle>);
```

Parameters

handle

Handle to a rowlist

Return value

number of rows in the list.

RowListColumnCount

Get column count

```
<num> RowListColumnCount(<handle>);
```

Parameters

handle

Handle to a rowlist

Return value

How many columns there are in the list.

RowListArray

Get a row as an array

```
<array> RowListArray(<handle> <num> row);
```

Parameters

handle

Handle to a rowlist

row

Index of the row to return as an array.

Return value

An array with all the columns of the specified *row*.

RowListItem

Get a list item

```
<string> RowListItem(<handle> <num> row, <num> col);
```

Parameters

handle

Handle to a rowlist

row

Row index

col

Column index

Return value

The value of the specified *column* in the specified *row*.

ReplaceTagsInString

Replace tags in a string with values from a rowlist

```
<string> RowListItem(<str> tagString, <str> tag, <handle> rowlist, <handle> rowIdx);
```

Parameters

tagString

A string with tag that should be replace with values from the rowlist

tag

The tags used in *tagString*

rowlist

Handle to a rowlist

rowIdx

Index of what row to use when replacing values in string

Return value

a String where the tags are replaced with values from the row list

Example

```
@var $list = RowListLoad("d:\\temp\\list.txt", "\\t");

@var $strTemplate = "Name : %%TAG0%% - Address : %%TAG2%% - Phone : %%TAG1%%";
@var $rowIdx = 0;

@var $strOutput = ReplaceTagsInString($str, "%%TAG(#)%%", $list, $rowIdx);

// $strOutput == "Name : Person Z - Address : Fake Street 1 - Phone :
0202-1111111"
```

MultiScript - Date/Time Functions

GetTime	Get current date and time
FormatDate	Format date
FormatTime	Format time
TimeLocal2UTC	Convert a time value from local to UTC
TimeUTC2Local	Convert a UTC time value to local time
ParseDateTime	Parse a datetime string and return as value

Related

[GetFileTime / SetFileTime](#)

GetTime

Get current date and time.

```
<num> GetTime([<num> getUTCTime]);
```

Parameters

getUTCTime

(Optional : Default : 0)

If 1, the time will be UTC else it will be local time. If parameter not given local time is used

Return value

Return the number of seconds since 1-Jan-1970 00:00:00

Example

```
@var $nowLocal = GetTime();
@var $nowUTC = GetTime(1);
```

FormatDate

Format a number value to its date value.

```
<str> FormatDate(<str> format, <num> timevalue);
```

Parameters

format

Date formatting string

timevalue

value that is the number of seconds since 1-Jan-1970 00:00:00.

Notes

See [Date Formatting](#) to see how to format dates

Return value

Date formatted string.

Example

```
@var $now = GetTime();
@var $date = FormatDate( "yyyy-MM-dd" , $now );
```

FormatTime

Format a number value to its time value.

```
<str> FormatTime(<str> format, <num> timevalue);
```

Parameters

format

Time formatting string

timevalue

value that is the number of seconds since 1-Jan-1970 00:00:00.

Notes

See [Time Formatting](#) to see how to format times

Return value

Time formatted string.

Example

```
@var $now = GetTime();
@var $date = FormatTime( "HH:MM:SS" , $now );
```

TimeLocal2UTC

Convert a time value from local to UTC.

```
<num> FormatTime(<num> timevalue);
```

Parameters

timevalue

value that is the number of seconds since 1-Jan-1970 00:00:00.

Return value

Time value converted to UTC.

Example

```
@var $now = GetTime();
@var $utc = TimeLocal2UTC($now);
@var $date = FormatTime( "HH:MM:SS" , $utc );
```

TimeUTC2Local

Convert a UTC time value to local time

```
<num> TimeUTC2Local(<num> timevalue);
```

Parameters

timevalue

value that is the number of seconds since 1-Jan-1970 00:00:00.

Return value

Time value converted from UTC to local.

Example

```
@var $now = GetTime();
@var $utc = TimeLocal2UTC($now);
@var $local = TimeUTC2Local($utc);
// $local and $now should be the same
@var $date = FormatTime( "HH:MM:SS" , $local );
```

ParseDateTime

Convert a Date/Time string to time value

```
<num> ParseDateTime(<str> strDateTime, <str> format);
```

Parameters

strDateTime

Date/Time as a string.

format

How the date/time is formated.

Return value

Time value converted to numeric value.

Formating information <http://pubs.opengroup.org/onlinepubs/009695399/functions/strftime.html>

Example

```
@var $str = "2012-01-01 12:32:11";
@var $val = ParseDateTime($str, "%Y-%m-%d %H:%M:%S");
```

Multiscript - File system functions

File and Folder

GetFileTime	Get the date/time of a file or folder.
SetFileTime	Set the date/time of a file or folder.
IsFolder	Check if a file item is a folder.
FileExists	Check if a file exists
GetFileSize	Get the size of a file.
GetFileProp	Get file property.

Filesystem

FindFirstFile	Return the first matching file
FindFiles	Return an array with all matching files.

File Attributes

GetFileAttributes	Return the attributes of a file or folder
AddFileAttributes	Add a attribute to a file or folder
RemoveFileAttributes	Remove attribute from a file or folder.
SetFileAttributes	Set attribute of a file or folder. (Replace existing).
HasFileAttributes	Return 1 if a file or folder has a specified attribute.

Filesystem Operations

CopyFile	Copy file.
MoveFile	Move file.
RenameFile	Rename a file.
UnpackFile	Unpack a archive (zip,7z,rar,tar,tgz)
PackFile	Pack files into an archive (zip,7z,tar,gz)
DeleteFile	Delete a file or folder.
DeleteFiles	Delete multiple files or folders.
MakeDir	Create a new folder.

FilePath String Operations

PathGetPathPart	Get the path part of a full filename path.
PathGetNamePart	Get the name path of a filename path.
PathGetFileExtPart	Get the file extension from a filename path.
PathGetParts	Get an array with all the filepath parts

FilePath Conversion Operations

PathMakeRelativeMC	Convert path to be relative to where MultiCommander is started from
PathMakeAbsoluteMC	Convert a relative path to absolute.
PathTranslatePath	Convert a relative path to absolute and also expand environment string.

GetFileTime

Get the date/time of a file or folder

```
<num> GetFileTime(<str> path, [<num> utctime], [<num> filetimepart]);
```

Parameters

path

Path to the file/folder of interest

utctime

(Optional, Default : 0)

0 = Return time in local time 1 = Return time in UTC.

filetimepart

(Optional, Default : 0)

0 = Last Write time

1 = Create time

2 = Last Access time

Return value

0 for failure else it returns the number of seconds since 1970-01-01 00:00:00

Example

```
// Change the time of the file item in focus to 1h previous. ( 3600 sec )
@var $time = 0;
@var $file = GetSourceFocusPath();
if( StrLen($file) > 0 )
{
    $time = GetFileTime($file);
    $time = $time - 3600;
    SetFileTime($file, $time);
}
```

SetFileTime

Set the date/time of a file or folder

```
<num> SetFileTime(<str> path, <num> time, [<num> utctime], [<num> filetimepart]);
```

Parameters

path

Path to the file/folder of interest

time

Time in seconds from 1970-01-01 00:00:00

utctime

(Optional, Default : 0)

0 = Return time in local time

1 = Return time in UTC.

filetimepart

(Optional, Default : 0)

0 = Last Write time

1 = Create time

2 = Last Access time

4 = Last Write,Create and Last Access

Return value

0 for failure else 1

Example

```
// See GetFileTime
```

IsFolder

Return 1 if the specified path is a folder

```
<num> IsFolder(<str> path);
```

Parameters

path

Path to a file or folder to be checked

Return value

1 if path is a folder, else 0

Example

```
@var $res = IsFolder("C:\\temp");
```

FileExists

Return 1 if the specified file exists

```
<num> FileExists(<str> path);
```

Parameters

path

Path to a file be checked

Return value

1 if path exists as a file, 2 if it is a folder and 0 if it does not exist.

Example

```
@var $res = FileExists("C:\\temp\\MyFile.txt");
```

GetFileSize

Return the size of the file

```
<num> GetFileSize(<str> path);
```

Parameters

path

Path to a file or folder to be checked

Return value

File size in number of bytes

Example

```
@var $res = GetFileSize("C:\\temp\\MyFile.txt");
```

GetFileProp

Return a file property from a file.

```
<num> GetFileProp(<str> path, <str> fileProp);
```

Parameters

path

Path to the file or folder of interest

fileProp

File property name (eg "MCAudioProp.genre")

Return value

A string with the file property

Example

```
@var $res = GetFileProp("C:\\temp\\MyPopSong.mp3", "MCAudioProp.genre");
// res == "pop"
```

Notes

See the [File Property List](#) for a list of all the available file properties.
(These are the same file properties that are used by the Multi-Rename tool.)

FindFirstFile

Return the first matching file name (also exists as "_findfirstfile")

```
<str> FindFirstFile(<str> path, [<num> onlyFiles], [<num> dbgInfo]);
```

Parameters

path

Path with filter. eg "c:\\myfolder*.txt"

onlyFiles (Optional)

If 1 then only return first matching file.

dbgInfo (Optional)

If 1 then output extra debugging information to the logwindow.

Return value

Full file path to the first matching file

Example

```
@var $file = FindFirstFile("C:\\temp\\*.txt");
```

FindFiles

Return array of matching files (also exists as "_filefiles")

```
<arr> FindFiles(<str> path, [<num> onlyFiles], [<num> dbgInfo]);
```

Parameters

path

Path with filter. eg "c:\\myfolder*.txt"

onlyFiles (Optional)

If 1 then only return first matching file.

dbgInfo (Optional)

If 1 then output extra debugging information to the logwindow.

Return value

array of strings with the full file paths to matching files

Example

```
@var $files = FindFiles("C:\\temp\\*.txt");
```

GetFileAttribute

GetFileAttribute

```
<str> GetFileAttribute(<str> path);
```

Parameters

path

File or Folder to get attribute for

Return value

str with the character ARSH for any valid attribute.

A - Archive

R - Readonly

S - System

H - Hidden

Example

```
$attributes = GetFileAttribute( $path );
```

AddFileAttribute

Add attribute to file or folder

```
<str> AddFileAttribute(<str> path, <str> attributes);
```

Parameters

path

File or Folder to get attribute for

attributes

string with the attributes to add.

Valid attributes are "rsh"

Return value

1 if attributes was set else 0.

Example

```
// Add Read Only attribute
AddFileAttribute( $path, "r" );
```

RemoveFileAttribute

Remove attributes to file or folder

```
<num> RemoveFileAttribute(<str> path, <str> attributes);
```

Parameters

path

File or Folder to get attribute for

attributes

string with the attributes to remove.

Valid attributes are "rsh"

Return value

1 if attributes was set else 0.

Example

```
// Remove the Read Only attribute
RemoveFileAttribute( $path, "r" );
```

SetFileAttribute

Set attributes to file or folder. Replace existing

```
<num> SetFileAttribute(<str> path, <str> attributes);
```

Parameters

path

File or Folder to get attribute for

attributes

string with the attributes to set.

Valid attributes are "arsh"

Return value

1 if attributes was set else 0.

Example

```
// Set the Read Only and Hidden attribute
SetFileAttribute( $path, "rh" );
```

HasFileAttribute

Set attributes to file or folder. Replace existing

```
<num> SetFileAttribute(<str> path, <str> attributes);
```

Parameters

path

File or Folder to get attribute for

attributes

string with the attributes to check for.

Valid attributes are "arsh"

Return value

1 if the file or folder has the requested attributes else 0. -1 for error

Example

```
// Check if a file has Read Only attribute
$isReadonly = HasFileAttribute( $path, "r" );
```

CopyFile

Copy one file or folder.

```
<num> CopyFile(<str> targetPath, <str> sourceFile, <string> options);
```

Parameters

targetPath

Target Path to where the file/folder should be copied

sourceFile

Source Path of the file/folder to copy

options

Optional parameter; if multiple options are wanted separate them with commas.

"NOWAIT" - Script will continue directly. Will not wait until copy is completed.

"NOPROGRESS" - Show no progress window. No error dialog will be shown. It will always skip files on error.

"NODIALOG" - Show no confirmation dialog

"OVERWRITE_ALL" - If a file already exists on the target, overwrite it without asking.

"SILENT" - Silent. Do not ask what to do on error. Skip on errors and continue.

"NEWQUEUE" - If a file operation already exists. Start a new queue. (override default setting)

"USEEXISTINGQUEUE" - If a file operation already exists then use it. (override default setting)

Return value

0 for failure, 1 for success

Example

```
CopyFile( $targetPath, $sourceFile, "NODIALOG, OVERWRITE_ALL");
CopyFile( 'D:\Temp\', 'R:\MyFile\File.txt', "NODIALOG, OVERWRITE_ALL");
```

MoveFile

Move one file or folder

```
<num> MoveFile(<str> targetPath, <str> sourceFile, <string> options);
```

Parameters

targetPath

Target Path to where the file/folder should be moved

sourceFile

Source Path of the file/folder to move

options

Optional parameter; if multiple options are wanted separate them with commas.

"NOWAIT" - Script will continue directly. Will not wait until copy is completed.

"NOPROGRESS" - Show no progress window. No error dialog will be shown. It will always skip files on error.

"NODIALOG" - Show no confirmation dialog

"OVERWRITE_ALL" - If a file already exists on the target, overwrite it without asking.

"SILENT" - Silent. Do not ask what to do on error. Skip on errors and continue.

"NEWQUEUE" - If a file operation already exists. Start a new queue. (override default setting)

"USEEXISTINGQUEUE" - If a file operation already exists then use it. (override default setting)

Return value

0 for failure, 1 for success

Example

```
MoveFile( $targetPath, $sourceFile, "NODIALOG, OVERWRITE_ALL");
MoveFile( 'D:\Temp\', 'R:\MyFile\File.txt', "NODIALOG, OVERWRITE_ALL");
```

RenameFile

Rename a file or folder using the MultiCommander virtual filesystem.

```
<num> RenameFile(<str> filepath, <str> NewName);
```

Parameters

filepath

Full path of the file to rename (eg. "C:\temp\myfile.txt")

NewName

The new file name. **ONLY THE NAME**. No path parts may be included

Options

Option (Optional.)

"SILENT" - Show no error dialog. If error then abort and return error code

"RENAME_RO" - Allow rename file with readonly attribute. (As long as the file system allows it)

"RENAME_OVR" - If a file with the new name already exists, then overwrite it.

Return value

1 if okay, else error code

Example

```
@var $filename = "C:\\temp\\MyFile.txt";
@var $newname = "MyNewName.txt";

RenameFile( $filename, $newname, "SILENT, RENAME_OVR" );
```

UnpackFile

Unpack a archive using the file operations in MultiCommander

```
<num> UnpackFile(<str> fileToUnpack, <str> targetFolder, <str> filter, <str>
options, <str> password);
```

Parameters

fileToUnpack

Full path to the file you want to unpack (eg. "C:\temp\myfile.zip")

targetFolder

Path to where you want the files to be unpack into. (eg. "C:\temp\myfile\")

filter

Wildcard filter used to match files to unpack. Empty or *.* to unpack all.

Options

Option (Optional.)

"SILENT" - Show no error dialog. If error then abort and return error code

"NOWAIT" -

"NOPROGRESS" -

"OVERWRITEALL" -

"NOQUEUE" -

"USEEXISTINGQUEUE" -

Password

(optional) Password needed to unpack the archive. If archive is password protected.

Return value

1 if okay, else error code

Example

```
@var $filename = "C:\\temp\\MyFile.zip";
@var $target = "C:\\temp\\Up\\";

UnpackFile( $filename, $target, "*.*" , "NOWAIT, OVERWRITEALL", "password123" );
```

PackFile

Pack files and/or folders into a archive. (zip,7z..) using the MultiCommander file operations

```
<num> PackFile(<t;str> targetFile, <str> sourceFile, <str> filter, <str> method,
<str> Options, <str> password );
```

Parameters

targetFile

Full path to the file to create (eg. "C:\temp\myfile.zip")

sourceFile

Full path to file or folder to pack

filter

Wildcard filter to use when adding files to archive.

method

compression profile to use or default method for that type. (eg ""zip"")

Options

Option (Optional.)

"SILENT" - Show no error dialog. If error then abort and return error code

"NOWAIT" -

"NOPROGRESS" -

"OVERWRITEALL" -

"NOQUEUE" -

"USEEXISTINGQUEUE" -

Password

(optional) Password you want to set on the archive. If you want to password protected it.

Return value

1 if okay, else error code

Example

```
@var $filename = "C:\\\\temp\\\\MyFile.zip";
@var $folderToPack = "C:\\\\temp\\\\MyFiles\\\\";

PackFile( $filename, $folderToPack, "*.*", "zip", "NOWAIT");
// or
PackFile( $filename, $folderToPack, "*.txt", "Internal Zip (Max)", "NOWAIT",
"MyPassword");
```

DeleteFile

Delete one file using the MultiCommander virtual filesystem

```
<num> DeleteFile(<str> filename, <array> options);
```

Parameters

filename

file or folder to delete

options

array with string for options

"RECYCLE" - Delete to recycle bin

"NOPROGRESS" - Show no progress window

"NODIALOG" - Show no confirmation dialog

"SILENT" - Show no error dialog if delete fails.

Return value

0 for failure, 1 for sucess

Example

```
@var $options[] = {"NODIALOG", "SILENT", "RECYCLE"};
DeleteFile( $filename, $options);
```

DeleteFiles

Delete multiple files using the MultiCommander virtual filesystem

```
<num> DeleteFiles(<array> filenames, <array> options);
```

Parameters

filenames

array of files or folders to delete

options

array with string for options

"RECYCLE" - Delete to recycle bin

"NOPROGRESS" - Show no progress window

"NODIALOG" - Show no confirmation dialog

"SILENT" - Show no error dialog if delete fails.

Return value

0 for failure, 1 for sucess

Example

```
@var $filenames[] = {'e:\file1.txt', 'e:\file2.txt', 'e:\Folder1'};  
@var $options[] = {"NODIALOG", "SILENT", "RECYCLE"};  
DeleteFile( $filenames, $options);
```

DeleteFiles

Delete multiple files using the MultiCommander virtual filesystem

```
<num> DeleteFiles(<array> filenames, <array> options);
```

Parameters

filenames

array of files or folders to delete

options

array with string for options

"RECYCLE" - Delete to recycle bin

"NOPROGRESS" - Show no progress window

"NODIALOG" - Show no confirmation dialog

"SILENT" - Show no error dialog if delete fails.

Return value

0 for failure, 1 for sucess

Example

```
@var $filenames[] = {'e:\file1.txt', 'e:\file2.txt', 'e:\Folder1'};  
@var $options[] = {"NODIALOG", "SILENT", "RECYCLE"};  
DeleteFile( $filenames, $options);
```

MakeDir

Create a new folder using the MultiCommanders virtual filesystem

```
<num> MakeDir(<str> path, <str> options);
```

Parameters

filename

Full path to folder to create. Can also be any virtual path like WPD:\ REG:\ FTP:\ or inside zips or 7z

options

"LOCAL" - Hint to the virtual filesystem that the path is a normal local path.

"RECURSIVE" - Create folders recursively. Only work if LOCAL also is specified.

Return value

0 for success, else error code

Example

```
MakeDir( "C:\\Temp\\NewFolder\\SubFolder\\", "LOCAL,RECURSIVE");
MakeDir( "WPD:\\MyPhone\\SubFolder\\", "" );
MakeDir( "C:\\Temp\\MyFiles.zip\\Folder", "" );
```

PathGetPathPart

Get the path part of a full filename path.

```
<str> PathGetPathPart(<str> path, [<num> stripEndingSlash]);
```

Parameters

path

Full file/folder path.

stripEndingSlash (OPTIONAL)

If 1, Then the ending slash of the path will be removed.

Return value

The path part of a full file/folder path

Example

```
@var $path = PathGetPartPath("C:\\temp\\MyPopSong.mp3");
// res == "C:\\temp\\"
```

PathGetNamePart

Get the name path of a filename path..

```
<str> PathGetNamePart(<str> path, [<num> stripExtension]);
```

Parameters

path

Full file/folder path.

stripExtension (OPTIONAL)

If 1, Then the file extension will be removed.

Return value

The name part of a full file/folder path

Example

```
@var $nameA = PathGetPartPath("C:\\temp\\MyPopSong.mp3");
// nameA == "MyPopSong.mp3"
@var $nameB = PathGetPartPath("C:\\temp\\MyPopSong.mp3", 1);
// nameB == "MyPopSong"
```

PathGetFileExtPart

Get the file extension from a filename path.

```
<str> PathGetFileExtPart(<str> path, [<num> stripDot]);
```

Parameters

path

Full file/folder path.

stripDot (OPTIONAL)

If 1, then the dot in the fileextension will be removed.

Return value

The file extension of a filename path

Example

```
@var $nameA = PathGetFileExtPart("C:\\temp\\MyPopSong.mp3");
// nameA == ".mp3"
@var $nameB = PathGetFileExtPart("C:\\temp\\MyPopSong.mp3", 1);
// nameB == "mp3"
```

PathGetParts

Get an array with all the filepath parts.

```
<arr> PathGetParts(<str> path);
```

Parameters

path

Full file/folder path.

Return value

Return an array with all the filename part.

Array item 0 = Device

Array item 1 = Path

Array item 2 = Filename

Example

```
@var $arr = PathGetParts("C:\\temp\\MyPopSong.mp3");
// $arr[0] == C:\
// $arr[1] == temp\\
// $arr[2] == MyPopSong.mp3
```

PathMakeRelativeMC

Convert path to be relative to where MultiCommander is started from.

```
<str> PathMakeRelativeMC(<str> path);
```

Parameters

path

Path to convert

Return value

The converted path

Example

```
// MC is installed in "C:\\Program\\MultiCommander"
@var $path = PathMakeRelativeMC("C:\\temp\\MyFile.txt");
// $path == "...\\Temp\\MyFile.txt"
```

PathMakeAbsoluteMC

Convert a relative path to absolute. (Relative to where MultiCommander is started from)

```
<str> PathMakeAbsoluteMC(<str> path);
```

Parameters

path

Path to convert

Return value

The converted path

Example

```
// MC is installed in "C:\Program\MultiCommander"
@var $path = PathMakeAbsoluteMC("../temp\\MyFile.txt");
// $path == "C:\Temp\MyFile.txt"
```

PathTranslatePath

Convert a relative path to absolute and also convert environmental tags

```
<str> PathTranslatePath(<str> path);
```

Parameters

path

Path to convert

Return value

The converted path

Example

```
// MC is installed in "C:\Program\MultiCommander" and ENVVAR is MyFile.txt
@var $path = PathTranslatePath("../temp\\%ENVVAR%");
// $path == "C:\Temp\MyFile.txt"

@var $path = PathTranslatePath("%WINDIR%\calc.exe");
// $path == "C:\Windows\calc.exe"
```

Multiscript - Registry Functions

RegKeyExists	Check if a key exists.
RegKeyAdd	Create new registry key.
RegKeyDel	Delete registry key.
RegKeyFind	Find registry key.
RegKeyFindRegExp	Find registry key with a regular expression.
RegValueExists	Check if a registry value exists.
RegValueAdd	Create new registry value.
RegValueDel	Delete a registry value.
RegValueFind	Find registry value.
RegValueFindRegExp	Find registry value using regular expression.
RegValueGetSZ	Get string registry value.
RegValueGetDWORD	Get DWORD registry value.

Where **regRoot** is required as a parameter use any of the specified short paths

HKLM	HKEY_LOCAL_MACHINE
HKCU	HKEY_CURRENT_USER
HKCR	HKEY_CLASS_ROOT
HKUR	HKEY_USER

RegKeyExists

Check if a registry key exists.

```
<num> RegKeyExists(<str> regRoot, <str> regPath);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path of the key you want check.

Return value

1 if exists. 0 if key does not exist.

Example

```
$exists = RegKeyExists("HKLM", "Software\\Microsoft");
```

RegKeyAdd

Add a new registry key.

```
<num> RegKeyAdd(<str> regRoot, <str> regPath);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path of the key you want to add.

Return value

1 if key was added, 0 if failed to add key.

Example

```
<num> RegKeyDel(<str> regRoot, <str> regPath);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path of the key you want to delete.

Return value

1 if key was deleted, 0 if failed to delete key.

Example

RegKeyFind

Get all matching subkeys on a specified registry path. Match keys using wildcards.

```
<arr> RegKeyFind(<str> regRoot, <str> regPath, <str> match);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path of where you want to find a key.

match

Wildcard for matching key, e.g. "Version_5.*"

Return value

Array with all the matching key names.

Example

RegKeyFindRegExp

Get all matching subkeys on a specified registry path. Match key using regular expression.

```
<arr> RegKeyFindRegExp(<str> regRoot, <str> regPath, <str> regExp);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path where you want to find a key.

regExp

Regular expression used for matching keys.

Return value

Array with all the matching key names.

Example

RegValueExists

Check if a registry value exists

```
<num> RegValueExists(<str> regRoot, <str> regPath, <str> valuename);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

The registry path to the key that should have the value

valueName

Name of the value to check for.

Return value

1 if value exists. 0 if it does not.

Example

```
$exists = RegValueExists("HLKM", "Software\\Microsoft", "installed");
```

RegValueAdd

Add new registry value

```
<num> RegValueAdd(<str> regRoot, <str> regPath, <str> valueName, <str> value,
<str> type);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path to the key that the value should be created under

valueName

Name of the registry value to create

value

Value to set the registry value to.

type

Type of registry value to create. REG_SZ or REG_DWORD supported.

Return value

NAME

Example

RegValueDel

Delete registry value

```
<num> RegValueDel(<str> regRoot, <str> regPath, <str> valuename);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

The registry path to the key that should have the value

valueName

Name of the value to delete.

Return value

1 if value is deleted. 0 if it is not.

Example

RegValueFind

Find all matching registry value under a registry key

```
<arr> RegValueFind(<str> regRoot, <str> regPath, <str> valueMatch);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path to key under where you want to search for a value.

valueMatch

Wildcard for matching value. e.g. "Version_5.*"

Return value

Returns an array of string with all the matching registry values.

Example

RegValueFindRegExp

Find all matching registry value under a registry key, Using regular expression for matching

```
<arr> RegValueFindRegExp(<str> regRoot, <str> regPath, <str> valueRegExpMatch);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path to key under which you want to search for a value.

valueRegExpMatch

Regular expression for matching values.

Return value

Returns an array of string with all the matching registry values.

Example

RegValueGetSZ

Get the value of a REG_SZ registry value

```
<str> RegValueGetSZ(<str> regRoot, <str> regPath, <str> valuename);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path to key under which the value exists.

valuename

Name of the registry key value to return. (Must be a REG_SZ value)

Return value

Return the string value of a registry value. Returns empty string if value does not exist.

Example

RegValueGetDWORD

Get the value of a REG_DWORD registry value

```
<num> RegValueGetDWORD(<str> regRoot, <str> regPath, <str> valuename);
```

Parameters

regRoot

Registry root, e.g. "HKLM"

regPath

Registry path to key under where the value exists.

valuename

Name of the registry key value to return. (Must be a REG_DWORD value)

Return value

Return the value of a registry value. Returns 0 if value does not exist.

Example

```
[REDACTED]
```

MultiScript - Get Source/Target file Functions

Functions for getting focus/selected files and folder from source/target views.

GetSelectFileNames	Get an array of all the currently selected files.
GetSelectedPaths	Get an array with full paths to all the currently selected files.
GetSourceSelectedFileNames	Get an array with the names of selected items in the source view.
GetTargetSelectedFileNames	Get an array with the names of selected items in the target view.
GetSourceSelectedPaths	Get an array with the full paths to the selected items in the source view.
GetTargetSelectedPaths	Get an array with the full paths to the selected items in the target view.
GetSourcePath	Get the current source path.
GetTargetPath	Get the current target path.
GetTargetFocusPath	Get the full path to the item in focus in the target view.
GetSourceFocusPath	Get the full path to the item in focus in the source view.
GetTargetFocusName	Get the name of the item in focus in the target view.
GetSourceFocusName	Get the name of the item in focus in the source view.

GetSelectedFileNames

Get the filenames of all the selected files/folders in the source view.

```
<arr> GetSelectedFileNames();
```

Return value

An array with all the names of the selected files and folders in the source view.

Example

```
@var $arr = GetSelectedFileNames();
LogDumpArray($arr);
```

GetSelectedPaths

Get the complete filepaths of the selected files/folders in the source view.

```
<arr> GetSelectedPaths();
```

Return value

An array with the complete paths for all the selected files and folders in the source view.

Example

```
@var $arr = GetSelectedPaths();
LogDumpArray($arr);
```

GetSourceSelectedFileNames

Get an array with the filenames of the selected items in the source view.

```
<arr> GetSourceSelectedFileNames();
```

Return value

An array of file names (only the filenames, not complete paths).

Example

```
@var $arr = GetSourceSelectedFileNames();
LogDumpArray($arr);
```

GetTargetSelectedFileNames

Get an array with all the filenames of the selected items in the target view.

```
<arr> GetTargetSelectedFileNames();
```

Return value

An array of file names (only the filenames, not the complete paths).

Example

```
@var $arr = GetTargetSelectedFileNames();
LogDumpArray($arr);
```

GetSourceSelectedPaths

Get an array with the full paths to all of the selected items in the source view.

```
<arr> GetSourceSelectedPaths();
```

Return value

An array of full paths to files or folders

Example

```
@var $arr = GetSourceSelectedPaths();
LogDumpArray($arr);
```

GetTargetSelectedPaths

Get an array with the full paths to all of the selected items in the target view.

```
<arr> GetTargetSelectedPaths();
```

Return value

An array of full paths to files or folders

Example

```
@var $arr = GetTargetSelectedPaths();
LogDumpArray($arr);
```

GetSourcePath

Get the complete path to the path currently shown in the source view.

```
<str> GetSourcePath();
```

Return value

A string with the complete path to the path shown in the source view.

Example

```
@var $str = GetSourcePath();
// $str == "C:\MyFolder\";
```

GetTargetPath

Get the complete path to the path currently shown in the target view.

```
<str> GetTargetPath();
```

Return value

A string with the complete path to the path shown in the target view.

Example

```
@var $str = GetTargetPath();
// $str == "C:\MyFolder\";
```

GetSourceFocusPath

Get the full path to the item in focus in the source view.

```
<str> GetSourceFocusPath();
```

Return value

A string with the full path to the item in focus

Example

```
@var $str = GetSourceFocusPath();
// $str == "C:\MyFolder\MyFile.txt";
```

GetTargetFocusPath

Get the full path to the item in focus in the target view.

```
<str> GetTargetFocusPath();
```

Return value

A string with the full path to the item in focus

Example

```
@var $str = GetTargetFocusPath();
// $str == "C:\MyFolder\MyFile.txt";
```

GetSourceFocusName

Get the name only of the item in focus in the source view.

```
<str> GetSourceFocusName();
```

Return value

A string with the name of the item in focus

Example

```
@var $str = GetSourceFocusName();
// $str == "MyFile.txt";
```

GetTargetFocusName

Get the name only of the item in focus in the target view.

```
<str> GetTargetFocusName();
```

Return value

A string with the name of the item in focus

Example

```
@var $str = GetTargetFocusName();
// $str == "MyFile.txt";
```

Multiscript - Misc Core Functions

Sleep	Pause the script for a specified number if milliseconds.
Log	Log text to a log window.
LogDump	Dump the contents of a variable to the log.
LogDumpArray	Dump the contents of an array to the log.
LogAppInfo	Log text to the application log window.
MessageBox	Show a messagebox.
AskText	Ask the user for text via a dialog.
AskOption	Ask the user to pick one of many items from a dropdown list.
SaveArray	Save an array of strings to a file.
LoadArray	Load a file into an array.
SaveStringToFile	Save a string to a file.
LoadStringFromFile	Load a file into a string.
GetClipboardText	Get the text stored in the clipboard.
SetClipboardText	Set text in the clipboard.
GetTagValue	Expand a Multi-Tag value.
TranslateEnvString	Translate a environmental string.

Sleep

Pause the script for a specified number of milliseconds.

```
<none> Sleep(<num> mmsec);
```

Parameters

mmsec

Milliseconds to wait before continuing.

Return value

Nothing.

Examples

```
Sleep( 10 );
```

Log

Log text to a log window.

```
<num> Log(<num> logID, <num> logLevel, <str> logText);
```

Parameters

logID

ID of the logger that should receive this log text.

0 = Standard Application log

1 = FileOperation Log

x = ID of user created log

logLevel

Log level for this log text. Log can be set to only show warning, or error.

10 = Info

20 = Warning

30 = Error

logText

Text that should sent to the logger.

Return value

1 if message is logged, 0 if log is not found.

Examples

```
Log(0, 30, "Error - File Not found" );
```

LogDump

Dump the content of a variable to the log. (Useful for debugging.)

```
<num> LogDump(<var> var);
```

Parameters

var

Variable to dump to the log file.

Return value

always 1

Example

```
@var $str = "String1";
LogDump($arr);
// Log window output
// ==[ Name : $str ]==[ Type : variable:string ]==[ Value : "String1" ]==
```

LogDumpArray

Dump the content of an array to the log. (Useful for debugging.)

```
<num> LogDumpArray(<arr> array);
```

Parameters

array

Array to dump to the log file.

Return value

always 1

Example

```
@var $arr[] = {"String1", "String2", "String3"};
LogDumpArray($arr);
// Log window output
// ==[ Variable : $arr ]==
// [0] : "String1"
// [1] : "String2"
// [2] : "String3"
// =====
```

LogAppInfo

Send a log text directly to the application log, with loglevel INFO.

```
<num> LogAppInfo(<str> logText);
```

Parameters

logText

Text that should be sent to the logger.

Return value

always 1

MessageBox

Show a messagebox

```
<num> MessageBox(<str> caption, <str> text, <num> options);
```

Parameters

caption

The caption text for the messagebox

text

The message text that should be shown

option

Standard win32 numbers for the options. See MSDN MessageBox for options.

Return value

Same as standard Win32 MessageBox. See MSDN for values.

AskText

Show a dialog where the user can input text.

```
<num> AskText(<str> label, <str> inputText, <num> option );
```

Parameters

label

The Label text that will be shown.

inputText

Text that will be shown in the input field when dialog is showing.

options

not used yet.

Return value

0 if user canceled the dialog, else a string with the text from the dialog's input field.

AskOption

Show a dialog where the user can pick one of many items from a dropdown list.

```
<num> AskOption(<str> label, <arr> values, <num> selected);
```

Parameters

label

The Label text that will be shown.

values

Arrays of strings or nums that will be shown in the drop down list.

selected

Index of what item in the array that should be selected by default, -1 for no item selected.

Return value

Index of the item selected, or -1 if user pressed cancel.

Examples

```
@var $options[] = {"Option1", "Option2" , "Option3" , "Option4" };
@var $o = AskOption( "Select Option" , $options, 2 );
if( $o > -1 )
{
    @var $sel = $options[$o];
    @var $msg = "You selected option : \\" + $sel + "\";
    MessageBox("Selected Item", $msg, 0);
}
```

SaveArray

Save an array of strings to a file

```
<num> SaveArray(<str> filename, <arr> array, <num> overwrite, <num> format);
```

Parameters

filename

Complete file path where the file should be saved.

array

The array to save

overwrite

0 = Do not overwrite existing file, 1 = overwrite existing file.

format

0 = ASCII

1 = UTF8

2 = Unicode

Return value

1 if array was saved successfully else 0.

LoadArray

Load a text file into an array

```
<arr> LoadArray(<str> filename);
```

Parameters

filename

Complete path to the file to load.

Return value

An array where every line in the file *filename* is an item in an array.

SaveStringToFile

DESC

```
<num> SaveStringToFile(<str> filename, <str> string, <num> format);
```

Parameters

filename

Complete file path to where the file should be saved.

string

Content to save.

format

0 = ASCII

1 = UTF8

2 = Unicode

Return value

1 if *string* was saved successfully else 0.

LoadStringFromFile

Load the content of a text file into a string.

```
<str> LoadStringFromFile(<str> filename);
```

Parameters

filename

Complete file path of the file to load.

Return value

The content of the file as a string.

GetClipboardText

Get the text stored in the clipboard. If the content in the clipboard is not text then an empty string will be returned.

```
<str> GetClipboardText();
```

Return value

A string with the content from the clipboard

SetClipboardText

Store text in the global clipboard.

```
<num> SetClipboardText(<str>);
```

Return value

nothing

Example

```
@var $textClip = GetClipboardText();
// Replace all '-' with '_'
$textClip = StrReplace($textClip, "-", "_");
// set the clipboard text with the new modified text.
SetClipboardText($textClip);
```

GetTagValue

Expand a Multi-Tag.

```
<str> GetTagValue(<str>);
```

Return value

Expanded Multi-Tag value

Example

```
@var $tag = GetTagValue("${mcinstallpath}");
```

TranslateEnvString

Translate environmental string.

```
<str> TranslateEnvString(<str>);
```

Return value

Expanded environmental string

Example

```
@var $tag = TranslateEnvString("%WINDIR%");
```

Extension Defined MultiScript Functions

Before using **extension defined MultiScript functions** the functions must be imported into the script engine. This is done with the **import** function. The import function must be called before any extension defined functions are called or the script engine will fail to find the function.

Example

```
// Import the Extension defined function for FTP
import("FSFTPScript");

// Connect to the ftp site "sitel" in the bookmarks
@var $ftp = FTPConnect("sitel");

// change to the remote folder /pub
FTPChdir($ftp, "/pub");

// delete the remove file test.zip
FTPRemove($ftp, "test.zip");

// upload the file d:\temp\test.zip to current remote folder with the name test.zip
FTPPutFile($ftp, "d:\\temp\\test.zip", "test.zip");
```

Extension Defined MultiScript Functions - Examples

[FS-FTP](#)

MultiScript - Extension Defined Functions - FS-FTP

Import Directive

To be able to use any of the FS-FTP defined MultiScript functions the functions must first be imported into the MultiScript engine. This is done with *Import*:

```
import("FSFTPScript");
```

FS-FTP Script Defined Functions.

FTPConnect	Connect to a FTP site.
FTPChdir	Change remote directory.
FTPMkdir	Create remote folder.
FTPRmdir	Remove remote folder.
FTPRemove	Remove remote file.
FTPRename	Rename remote file or folder.
FTPList	List current directory
FTPListCount	Number of items obtained from previous FTPList command.
FTPListGet	Get list item
FTPExists	Check if remote file exists.
FTPPutFile	Upload a file.
FTPGetFile	Download a file.
FTPCommandRaw	Send raw FTP command.
FTPQuit	Close FTP Connection.

FTPConnect

Connect to an FTP site that is defined in bookmarks.

```
<handle> FTPConnect(<str> sitename);
```

Parameters

sitename

name of a site that is defined in bookmarks

Return value

A handle to an ftp site, or 0 if it failed to connect or if the sitename is not found in bookmarks.

FTPChdir

Change remote directory.

```
<num> FTPChdir(<handle> conn, <str> path );
```

Parameters

conn

Handle to an ftp connection.

path

path to a location on the remote connection.

Return value

1 if successful or 0 if it failed.

FTPMkdir

Create directory in current remote location.

```
<num> FTPMkdir(<handle> conn, <str> directory);
```

Parameters

conn

Handle to an ftp connection.

directory

name of the directory to create.

Return value

1 if successful, 0 if it failed.

FTPRmdir

Remove a directory

```
<num> FTPRmdir( <handle> conn, <str> directory);
```

Parameters

conn

Handle to an ftp connection.

directory

name of the directory to remove.

Return value

1 if successful, 0 if it failed.

FTPRemove

Remove a remote file

```
<num> FTPRemove(<handle> conn <str> file);
```

Parameters

conn

Handle to an ftp connection.

file

Name of the remote file to remove.

Return value

1 if successful, 0 if it failed.

FTPRename

Rename a remote file or folder

```
<num> FTPRename(<handle> conn , <str> oldFilename, <str> newFilename);
```

Parameters

conn

Handle to an ftp connection.

oldFilename

Name of the remote file to rename.

newFilename

Filename the file should be renamed to.

Return value

1 if successful, 0 if it failed.

FTPList

Get the current directory content.

```
<num> FTPList(<handle> conn);
```

Parameters

conn

Handle to an ftp connection.

Return value

1 if successful, 0 if it failed.

Remarks

The result of a List command is cached in the FTP connection. Use FTPListGet to get the content if the list items.

FTPListCount

The number of list items that are stored in the connection.

```
<num> FTPListCount(<handle> conn);
```

Parameters

conn

Handle to an ftp connection.

Return value

The number of items obtained in the directory listing from a previous FTPList command.

FTPListGet

Get a list item.

```
<array> FTPListGet(<handle> conn, <num> index);
```

Parameters

conn

Handle to an ftp connection.

index

Index of the list item to get.

Return value

An array of 2 items where the first is the Name and the second is the size, or 0 if no item was found.

FTPExists

Check if a file or directory exists

```
<num> FTPExists(<handle> conn, <str> file);
```

Parameters

conn

Handle to an ftp connection.

file

Filename to check if it exists.

Return value

1 if the file exists, 0 if it does not.

Remarks

An FTPList command must have been run before running this command.

FTPPutFile

Upload a file.

```
<num> FTPPutFile(<handle> conn, <str> filepath, <str> remotefile);
```

Parameters

conn

Handle to an ftp connection.

filepath

filepath to the file to upload.

remotefile

The name to store the file as on the ftp server.

Return value

1 if successful, 0 if it failed.

FTPGetFile

Download file.

```
<num> FTPGetFile(<handle> conn, <str> remoteFile, <str> localfile);
```

Parameters

conn

Handle to an ftp connection.

remoteFile

Path to a remote file to download.

localfile

Path to a local filename to save the download file as.

Return value

1 if successful, 0 if it failed.

FTPCmdRaw

Send a RAW FTP command.

```
<handle> FTPCommandRaw(<handle> conn, <str> command );
```

Parameters

conn

Handle to an ftp connection.

command

RAW FTP command to send.

Return value

1 if successful, 0 if it failed.

Example

```
FTPCmdRaw( $ftp , "site who");
```

FTPQuit

Close a connection to an FTP server.

```
<handle> FTPQuit(<handle> conn);
```

Parameters

conn

Handle to an ftp connection.

Return value

1 if successful, 0 if it failed.

MultiScript - Extension Defined Functions - File Check Sum

Import Directive

To be able to use any of the File CheckSum defined MultiScript functions the functions must first be imported into the MultiScript engine. This is done with *Import*:

```
import("MFTools-sfv");
```

File CheckSum Script Defined Functions.

[ChkSum_Calculate](#)

Calculate CheckSum of a file.

ChkSum_Calculate

Calculate CheckSum of a file.

```
<str> ChkSum_Calculate(<str> filename, <num> type);
```

Parameters

filename

name of the file you want to calculate the checksum of

type

0 = CRC32, 1 = MD5, 2 = SHA1

Return value

Return a string with the hex hash value

Multi-Script Debugger

The screenshot shows the 'Multi-Script Debugger - BETA' window. The title bar indicates it is 'Unfinished, Under development'. The menu bar has 'File' and 'Debug' options. Below the menu is a toolbar with icons for file operations. The main area contains a text editor with the following script code:

```
1 @var $SourceFocusItemName = GetSourceFocusName();
2 @var $SourceFocusItemPath = GetSourceFocusPath();
3 @var $TargetFocusItemName = GetTargetFocusName();
4 @var $TargetFocusItemPath = GetTargetFocusPath();
5
6 @var $SourceSelectedFilenames = GetSourceSelectedFileNames();
7 @var $TargetSelectedFilenames = GetTargetSelectedFileNames();
8 @var $SourceSelectedPaths = GetSourceSelectedPaths();
9 @var $TargetSelectedPaths = GetTargetSelectedPaths();
```

Below the text editor is a table showing variable values:

Name	Type	Value
Global		
\$SourceFocusItemName	string	temp
\$SourceFocusItemPath	string	C:\temp
\$SourceSelectedFilenames	array	array (size : 1)
\$TargetFocusItemName	string	IMG_0074.jpg
\$TargetFocusItemPath	string	D:\Pictures\pic\IMG_0074.jpg
\$TargetSelectedFilenames	array	array (size : 3)

Multi-Script Examples

Examples of Multi-Scripts

- [Search on google from the commandline bar](#)
- [View first *.txt file found in the folder currently in focus](#)
- [Play first *.avi or *.mkv in the folder currently in focus](#)

Search on google from the commandline bar

Take the parameters to this command and build a search url for google and then go to that url.

Create the command. then go to the Alias Manager and assign 'g' to "@<Unique command id>"

```
function CreateGoolgeSearchQuery()
{
    @var $query = "";
    @var $n = 0;
    for( $n = 0; $n < $argcount; $n = $n + 1 )
    {
        if( $n > 0 )
        {
            $query = $query + "+";
        }
        $query = $query + $arg($n);
    }
    return $query;
}
@var $s = "";
$s = "http://www.google.com/search?g=" + CreateGoolgeSearchQuery();
MC.Run CMD="{{$s}}" SHELL
```

View first *.txt file found in the folder currently in focus

Connect this command to a hotkey, for example ALT+V. When you then in the explorer panel have a folder in focus, if you press ALT+V and if a *.txt file exists under that folder, that file will now be shown.

```
MC.DataViewer.View FILE={_findfirstfile( "{focusfilepath}\\.\\*.txt" )}
```

Play First *.avi or *.mkv found in folder currently in focus

Connect this command to a hotkey, for example ALT+P. When you then in the explorer panel have a folder in focus, if you

press ALT+P, the first *.avi or *.mkv found will be played in your default movie player.

```
@var $file;
$file = _findfirstfile( "{focusfilepath}\\.\\*.avi" );
if(StrIsEqual($file,""))
{
    $file = _findfirstfile( "{focusfilepath}\\.\\*.mkv" );
}
MC.Run CMD={$file} SHELL
```

Multi Tags

Multi-Tags are text tags that can be used with some User Commands.

These tags are processed and translated before the command is executed.

Multi-Tags are supported by External Commands, Custom Commands and Batch Scripts.

Multi-Tags can **NOT be used in Multi-Script** directly. To use them from Multi-Script use the [GetTagValue](#) function

Multi-Tag	Description	Example Output
<code> \${focusfilepath}</code>	Full path to the file/folder in focus	C:\Programs File\MultiCommander\MyFile.exe
<code> \${date:<date-format>}</code>	Current local date	<code> \${date:yyyy-MM-dd} -> 2010-02-28</code>
<code> \${time:<time-format>}</code>	Current local time	<code> \${date:hh:mm:ss} -> "15:23:01"</code>
<code> \${targetpath}</code>	Current path of the target panel	D:\MyBackup\
<code> \${sourcepath}</code>	Current path of the source panel	C:\Programs File\MultiCommander\
<code> \${targetdevice}</code>	Current device of the target panel	D:
<code> \${sourcedevice}</code>	Current device of the source panel	C:
<code> \${sourcefocuspath}</code>	Full path to item in focus, in the source view. This MultiTag is the same as <code> \${focusfilepath}</code>	C:\my folder\filename.txt
<code> \${targetfocuspath}</code>	Full path to item in focus, in the target view.	D:\Temp\Folder 2\report.txt
<code> \${sourcefocusname}</code>	Name of item in focus, in the source view.	filename.txt
<code> \${targetfocusname}</code>	Name of item in focus, in the target view.	report.txt
<code> \${leftpath}, \${leftfocuspath}, \${leftfocusname} \${rightpath}, \${rightfocuspath}, \${rightfocusname}</code>	Same as the <code> \${sourcefocuspath}</code> tags above, but referring to the left/right panel, independent of which panel is source or target.	
<code> \${param:<num>}</code>	Script parameter. where <num> is the index of the parameter	<code> \${param:0} -> "D:\MyBackup\"</code>

All the paths below depend on how Multi Commander is installed, i.e. for single user, Multi User, or portable.

<code> \${mcinstallpath}</code>	The path to where Multi Commander is run from.	C:\Program Files\MultiCommander
<code> \${mcappdatapath}</code>	Path to Multi Commander's application data folder.	C:\Users\<username>\AppData\Roaming\MultiCommander\
<code> \${mclogpath}</code>	Path to Multi Commander's log folder	C:\Users\<username>\AppData\Roaming\MultiCommander\Logs\
<code> \${mcconfigpath}</code>	Path to Multi Commander's main config folder.	C:\Users\<username>\AppData\Roaming\MultiCommander\Config\
<code> \${mcuserappdata}</code>	Path to Multi Commander's user data storage. Used by plugins and extensions to store user data.	C:\Users\<username>\AppData\Roaming\MultiCommander\UserData\

Paths that are expanded from Multi-Tags are **NOT quoted**, so you might need to put quotes around the Multi-Tag. You might also need to put single quotes around the entire command. (See the MC.Run example below.)

Examples

Custom Commands

```
MC.Explorer.Goto PATH="${focusfilepath}"
MC.Run CMD="C:\bin\Compare.exe" ARG=' -s "${sourcepath}" -d "${targetpath}"'
```

As you see in the **MC.Run** example. The **ARG** command has spaces and quote ("") characters in it, so the entire ARG must then be encapsulated using single quote (') characters.

External Commands

```
"C:\Program Files\Beyond Compare 3\BCompare.exe" -s "${leftfocuspath}" -d "${rightfocuspath}"
```

MultiScript Commands

Multi-Tags are only supported on the line that is a **CustomCommand**.

```
@var $cmd = "C:\\Bin\\Compare.exe";
// INCORRECT
@var $path1 = "${sourcepath}"; // Not supported
@var $path2 = "${targetpath}"; // Not supported
MC.Run CMD="{{$cmd}} ARG=' -s {$path1} -d {$path2}'"

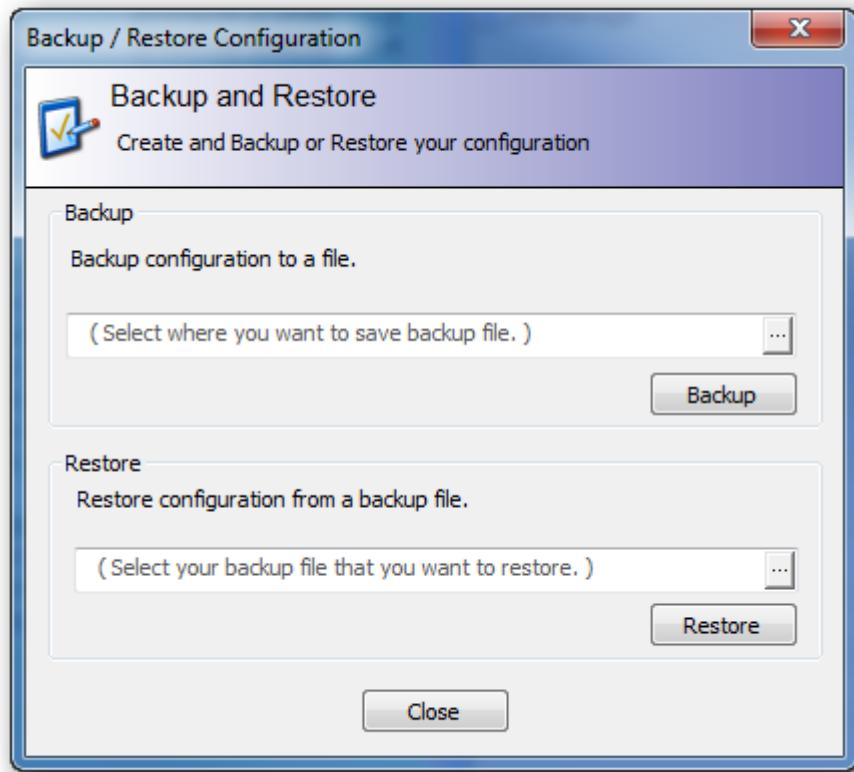
// CORRECT
@var $path1 = GetSourcePath();
@var $path2 = GetTargetPath();
MC.Run CMD="{{$cmd}} ARG=' -s {$path1} -d {$path2}'"

// CORRECT
MC.Run CMD="{{$cmd}} ARG=' -s ${sourcepath} -d ${targetpath}'"
```

MultiScript calling a CustomCommand with Script variables. Custom Commands that are run from MultiScript are run by the Custom Command engine, so script parts in the custom command line must be encapsulated with { } (as the variables are above). They will be evaluated and expanded before that line is sent to the Custom Command engine.

Backup and Restore Configuration

The configuration for MultiCommander and all the plug-ins can be backed up and restored. Select **Backup/Restore** from the *Help* menu to open the Backup and Restore dialog.



Backup

Enter a file path to a .zip file or click "..." to browse to the location where to you wish to save the backup; then click the **Backup** button.

All configuration files for MultiCommander and all its plug-ins will be backed up into a .zip file at that location.

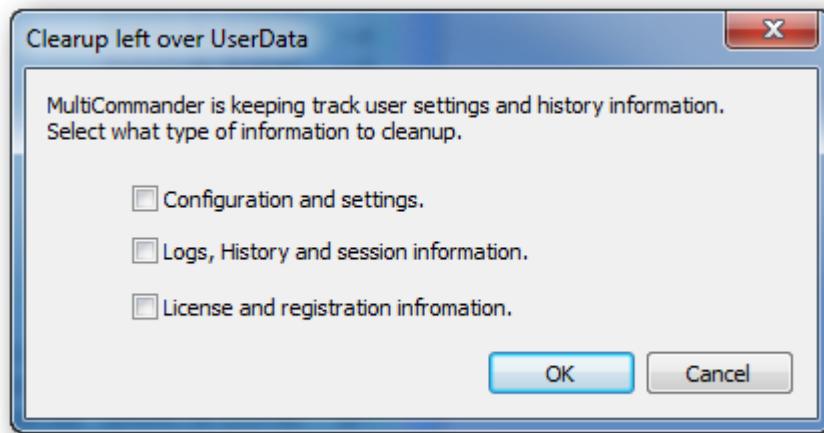
Restore

Enter a file path to a .zip file or click the "..." button to browse for the file that contains a backup of the configuration. Then press the **Restore** button to replace the current configuration with the configuration that was backed up in that file.

You need to restart MultiCommander after this for the restored configuration to fully take effect.

Cleanup User Data

MultiCommander stores user-specific data in different places. To remove all this data you can run **Cleanup UserData**. Select "Cleanup userdata..." from the Help menu to start it.



Select the type of user data you want to remove then click **OK**. After the user data is removed it is recommended to exit MultiCommander or new information might be stored.

Configuration and settings

MultiCommander stores configuration and settings files in an application data folder for the user account. Checking this option will remove all these files.

Logs, History and Session Information

MultiCommander stores logs of file operations and failures in the user's application data folder. These logs are used to identify why problems happen if MultiCommander fails. History and session information is stored in the Windows Registry. It is information that is used to create dropdown lists with selections you have had before. Previously run commands from the commandline bar are listed when you show the dropdown list of the commandline bar.

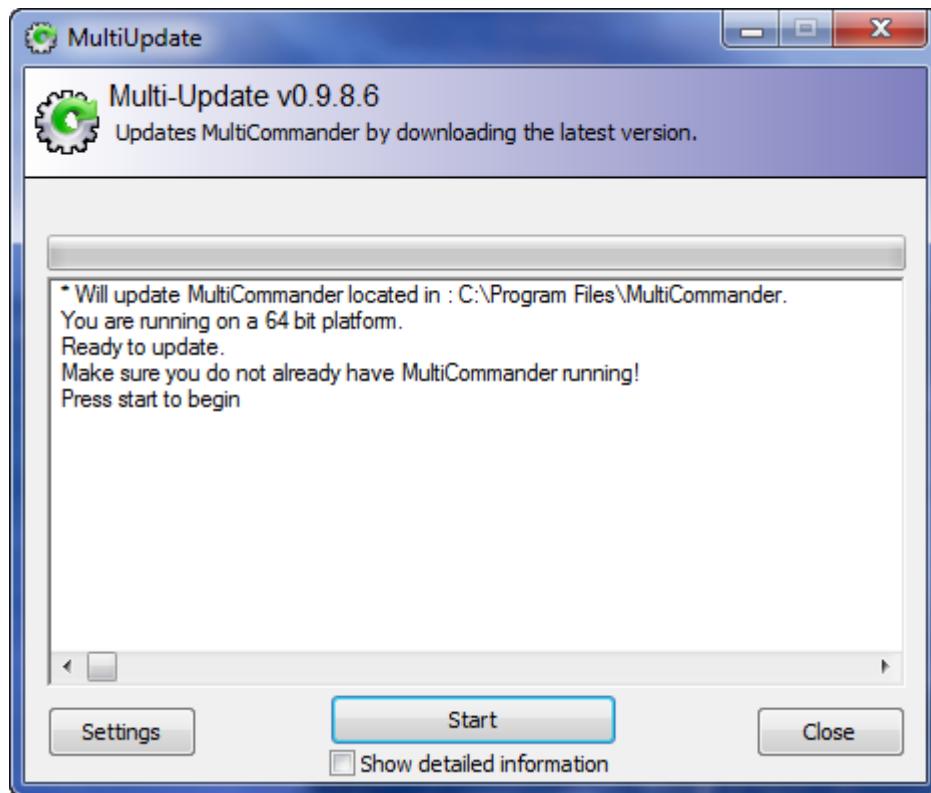
License and Registration information

Checking this option will remove the license and registration information for MultiCommander.

MultiUpdate

MultiUpdate is a separate tool that will check if an updated version of MultiCommander is available, if it is, it is downloaded and MultiCommander is updated.

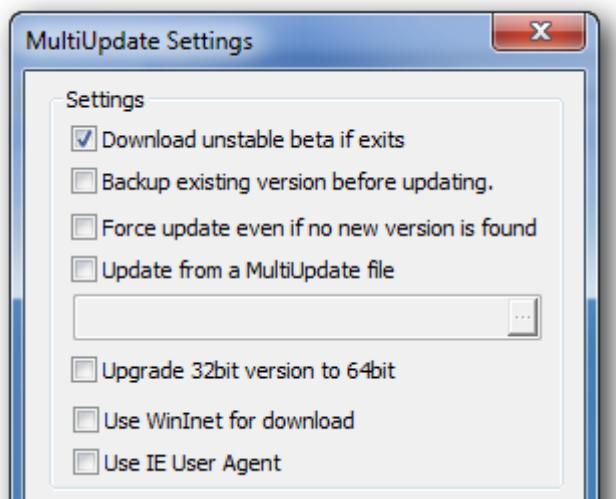
MultiUpdate can be launched separately by launching it from the start menu or from the folder where it is located or by selecting "Launch MultiUpdate..." from the *Help* menu in MultiCommander.



By default MultiCommander will check if an update exists before starting. If an update exists it will then launch MultiUpdate.

MultiCommander **must** not be running when MultiUpdate tries to update MultiCommander. If it is, the update process will fail since it will fail to replace files.

Settings



Download unstable beta if exists

This option checks for an unstable beta version and updates MultiCommander with it if available.

Unstable beta versions are often not yet fully tested and can contain experimental unfinished features.

Checking this is normally NOT recommended.

Backup existing version before updating

This option creates a backup of the current installation before updating, so that you can revert to a previous version if there are problems with the new version.

Force Update even if no new version is found

This will force the download of the latest version found and update with that. This can be used if the current version has been corrupted, or if you are running an Unstable Beta version and want to revert back to the latest stable version.

Update from MultiUpdate File.

If you already have a file containing MultiCommander updates, you can update from this file instead of downloading another.

Upgrade 32bit version to 64bit

If you are running a 32bit version of MultiCommander, you can check this option to force MultiUpdate to upgrade you to the 64bit version. (This of course requires you to be running on a 64bit version of Windows.)

Use WinInet for Download

If you have problems downloading an update, try activating this option.

Use IE User Agent

Some corporate networks have restrictions in firewalls that do not allow anything other than Internet Explorer to communicate with the internet. Checking this will make MultiUpdate download updates by pretending to be Internet Explorer.

Command Line Parameters

You can start MultiCommander.exe with some parameters at startup

```
-L=<path> - Path to show in the left panel
-R=<path> - Path to show in the right panel
-F=<startup.ini> - file that contains startup properties for path to show.
-T=<Any Text> - A Text that will be shown in the caption bar of the Multi Commander window.
-AutoRun=<User Defined Command to auto run when started>
-NOUPDATECHECK - Disable check for new version
```

Both - or / is supported for parameters. (eg /L=C:\Bin or -L=C:\Bin)

If you want an already running instance of Multi Commander to go to a path you can use the /OPEN parameter. This is used when you have an external program that should tell Multi Commander to go to a special path. or when Multi Commander is running in **Single Instance** mode

```
/OPEN "<path>"
```

Startup definition file (-F=<filename.ini>)

The startup file specifies multiple folders to be shown. Each folder will be shown in a separate tab.

```
[PANEL_1]
Path=d:\
Side=L

[PANEL_2]
Path=d:\bin
Side=L

[PANEL_3]
Path=d:\Projects
Side=L

[PANEL_4]
Path=d:\Projects\Jupiter
Side=R

[PANEL_5]
Path=d:\Projects\Pluto
Side=R
```

Path and **Side** parameters **must** be specified in the file.

The following parameters are optional

```
Locked=1|0
AllowSubPathChange=1|0
```

```
ShowTree=1|0
SortAsc=1|0
SortCol=[colnum]
TabColorText=#000000
TabColorBk=#000000
TabColorHotWhenActive=1|0
TabColorGradient=1|0
ViewMode=1|2|3|4
ViewModeListWidth=000
ViewModeDetailWidth=000
Side=L|R
Path=[Full Path]
```

1|0 means 1 or 0. 1|2|3 means 1 or 2 or 3. and so on

Examples

```
MultiCommander.exe "C:\Bin" "D:\\"
```

If no "-L" or "-R" parameter is specified, the first parameter is assumed to be a path to show in the left panel, and the second to be a path to show in the right panel.

```
MultiCommander.exe -L="C:\Bin" -R="D:\\"
```

```
MultiCommander.exe -F="AutoStart.ini"
```

```
MultiCommander.exe -AutoRun=CmdRunStartup
```

```
MultiCommander.exe /OPEN "D:\My Folder\SubFolder"
```


Multi Commander Windows Message API

Multi Commander support the **WM_COPYDATA** Windows message for setting/getting information. Other programs can use this message to tell Multi Commander to go to a path, change focus to an file item, or to query what the current path is, or what the current file in focus is.

Actions for getting informations from Multi Commander will get the answer sent back to them using WM_COPYDATA message to the calling program.

WM_COPYDATA is a special windows message. One of the parameters to send is a pointer to a COPYDATASTRUCT structure. This struct is used to stores all the data that should be send to MultiCommander

```
typedef struct tagCOPYDATASTRUCT {
    ULONG_PTR dwData;
    DWORD      cbData;
    PVOID      lpData;
} COPYDATASTRUCT, *PCOPYDATASTRUCT;
```

dwData is used to set some action flags for MultiCommander

lpData is a pointer to the data that should be sent over to MultiCommander

cbData is the size in bytes of the data that should be sent

The **WM_COPYDATA** message must be sent to the main MultiCommander Window.

This can be found by first finding the process id of the MultiCommander.exe instance and then find the toplevel window of that instance.

There are example on how to do that in the **MCCopyData.h** file in the sample.

When the HWND for the window have been found use the Windows API SendMessage to send the **WM_COPYDATA** message to the HWND of the windows of MultiCommander.

If requesting information that require MultiCommander to answer back, the HWND of the windows that will receive the answer is also needed.

Example

```
COPYDATASTRUCT cds;
cds.dwData = CPF_GOTOPATH | CPF_SOURCE | CPF_NEWTAB;
cds.cbData = wcslen(szPath) * sizeof(wchar_t);
cds.lpData = (void*)szPath;
SendMessage(hMCHWND, WM_COPYDATA, (WPARAM)hMyHwnd, (LPARAM)(LPVOID)&cds);
```

CPF_GOTOPATH tells MultiCommander that the data provided is a path and this it should go to this path

CPF_SOURCE tells MultiCommander that it is in the source panel the new path should be shown

CPF_NEWTAB tells MultiCommander that the new path should be opened in a new tab.

The following CopyData Flags can be combined.

Some flags can't be combined. Do not combine multiple action flags and multiple target panel/side flags

```
#define CPF_GOTOPATH      0x00000002L // Go to path
#define CPF_GETCURPATH     0x00000004L // Go current path - Result returned in
WM_COPYDATA message from MC
#define CPF_GETCURITEM      0x00000008L // Get name (File/Folder) of Current item in
focus
#define CPF_GETCURITEMFULL 0x00000010L // Get full path for (File/Folder) of Current
item in focus
```

```

#define CPF_LEFT          0x00000100L // Go to the new path in on the left side
#define CPF_RIGHT         0x00000200L // Go to the new path in on the right side
#define CPF_SOURCE        0x00000400L // Go to the new path in the source panel
side
#define CPF_TARGET        0x00000800L // Go to the new path in the right panel side
#define CPF_ANSI          0x00010000L // If not specified All string are assumed to
be Unicode
#define CPF_NEWTAB        0x00020000L // Open a new side on the specified side.
#define CPF_FOCUSITEM     0x00040000L // Change the focus to the file item provided
in the path

```

Set CPF_ANSI if the path that are sent are ASCII/ANSI if this is not set Unicode (wchar_t) is assumed.

C++ Helper

Included in the Sample project there is a MCCopyData.h file. This file contains a helper class that will do most of the work for you

```

// Get Current path of the source panel
MultiCommander::MCCopyDataHelper MCCopyData;

MCCopyData.Subclass(hDlg);
std::wstring path = MCCopyData.GetCurrentPathW(hDlg, CPF_SOURCE);
MCCopyData.ReleaseSubClass(hDlg);

```

hDlg is the HWND of the calling program. And this HWND is used when MultiCommander is sending back the answer of the request.

MCCopyData.Subclass(hDlg);

This will hook the helper class into the message loop of the calling program and it will handle all of the WM_COPYDATA response that are sent from MultiCommander.

if your program is not a pure Win32 program or you can not do this. You need to handle the answer of the WM_COPYDATA your self.

You find example of how to do that in the helper class.

In this example we tell MultiCommander to go to D:\MyFolder\SubFolder and that it should be opened in a new tab

```

MultiCommander::MCCopyDataHelper MCCopyData;
MCCopyData.GoToPath(L"D:\\MyFolder\\SubFolder\\", CPF_SOURCE | CPF_NEWTAB, hDlg);

```

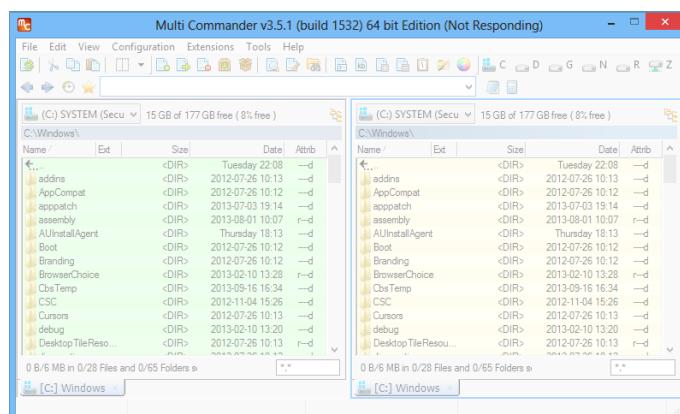
[Download sample project and test exe](#)

Multi Commands hangs and is not responding

Symptom

When Multi Commander hang it is often because it is waiting for some external operation to finished. But it can also be because of a problem inside Multi Commander.

A hang is when Multi Commander becomes not responsive. If the main user interface no longer accept any input and you see "Not Responding" in the title bar.



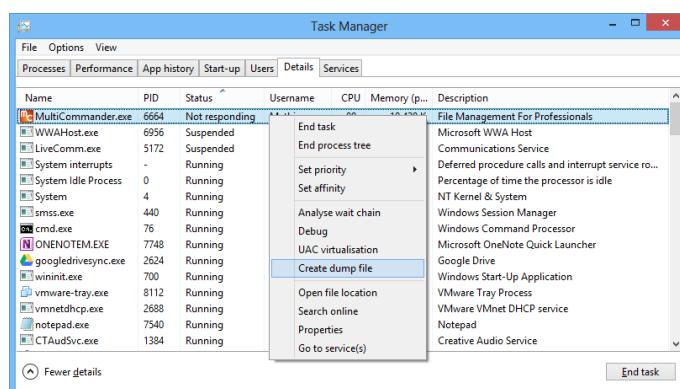
What to do.

If Multi Commander crashes a mini dump file is created and sent to the crash report system and we can den analyze why it crashed. But for a hang this is not done automatically. For us to find out why it hangs and to be able to fix it, We need to know what Multi Commander is doing when it hangs. For us to do that we need a dump of the hanged MultiCommander.exe process. But you will need to create that dump for us and send it to us.

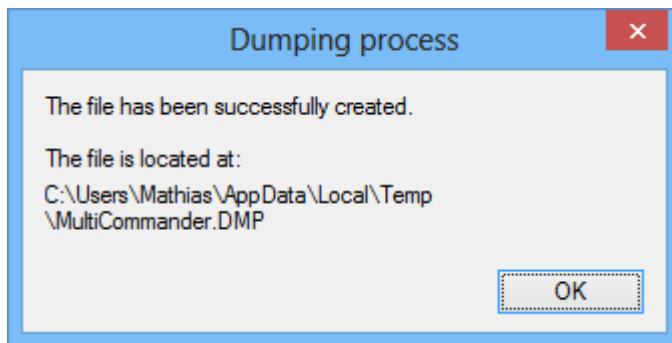
Create a dump file

To create a dump file you can use the Windows Task Manager (For Windows XP / Windows Server2003 you have to use a tool like [procdump](#)).

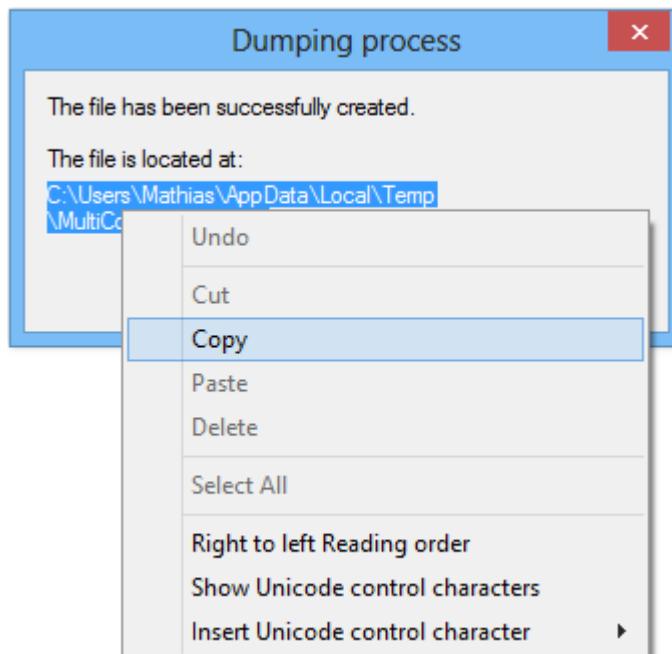
Start the task manager by pressing Ctrl+Shift+Esc, Select the detailed tab, Find the MultiCommander.exe process in the list that is hanged and right click on it and select "Create Dump File"



When the dump file is saved you will be presented with a dialog telling you where it wrote the dump file.



The path in the dialog can be selected so that you can copy and paste that path into Multi Commander or Windows Explorer to go to that location.



Then compress the dump file with Zip/7-zip/Rar. (Fastest way is to just right click on the file and select Send To > Compressed Folder).

Then upload that file to <http://multicommander.com/upload> and we can take a look and see what Multi Commander is waiting for and why it hangs.

If you want any feedback then also post a message in the forum or send us a mail.

Multi Commander Crashes

If Multi Commander Crash

A crash can happen because of many different reason. It can be because of a bug inside Multi Commander, But it can also be because of external factors. Like bad memory or bad hardware, or other external issues like conflicts with other software.

When Multi Commander crashes it will by default try to create a mini crash dump and then send it to our crash report system.

The crash report system will then do a automatic analyze of the crash and then send then the result for review. If the issue need more investigation it will be analyzed manually.

With the crash dump we can see where inside Multi Commander the crash happen and sometimes the reason why. But the crash dump file that is created by default has very limited information and sometimes the crash dump does not have enough information for us to find out what it is happening.

Shell Extensions

The most common reason for crashes are because of third party programs that get executed inside the Multi Commander process. This happens often when Multi Commander is using Windows Explorer Shell Extensions for building the Context Menu (Right Click menu).

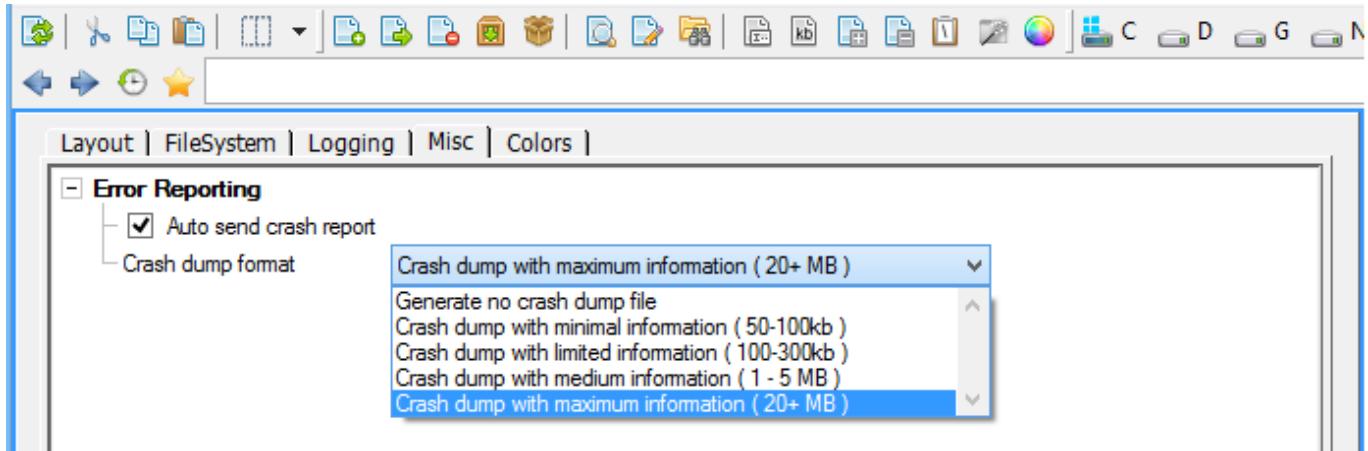
A lot of shell extensions are only tested on Windows Explorer, So when other programs like Multi Commander is calling them they can cause a crash of Multi Commander because a Shell Extension by a thrid party has an error, and there is nothing Multi Commander can do about it.

One thing you as a user can do is to disable Shell Extension one by one to find out what extension is not playing nice. A tool to disable/enable ShellExtension the free tool [ShellExView by NirSoft](#)

Provide Better Crash Information

Since the default crash report is very limited with information they do not always provide enough useful information. If you have regular crashes, then please consider changing the Crash Dump Format to a higher level. This will provide us with better information for finding why it crashes.

You find the settings under Core Settings



Also if you did something special to trigger the crash then please send a mail or leave a post in the forum letting us know what you did to trigger the crash. The more information we get the bigger the chance is that we can fix the problem.

The crash dumps with maximum information contains all the memory that the MultiCommander.exe process uses. So they can be large. But they also give us a good view on what is going on inside Multi Commander on how the internal state of different parts is.

