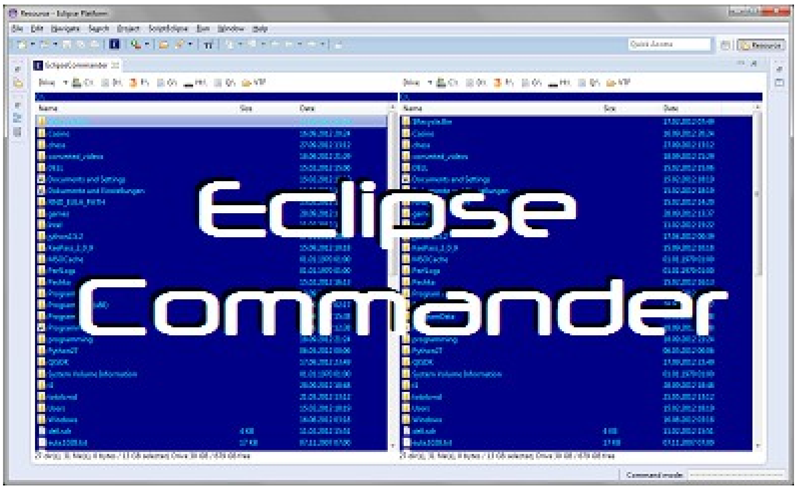


User Manual v1.3.0

Table of contents

Table of contents

Description

EclipseCommander is the worlds first two-pane file manager built with Eclipse. If you still remember the best file commander in the DOS area you will know all the shortcuts by heart. As sugar on top there is a new self-invented concept called „Virtual Tag Folders“: the easiest way to organize your files and folders without even touching your original files!

Read more in the Virtual Tag Folder chapter.

Installation

Installation for Windows

Download either the windows 32bit or 64bit package from the homepage and extract it. Start EclipseCommander.exe

Installation for Linux

Download either the linux 32bit or 64bit package from the homepage and extract it. Make the EclipseCommander file executable by „chmod +x EclipseCommander“ and start it with „./EclipseCommander“

Installation for MacOSX

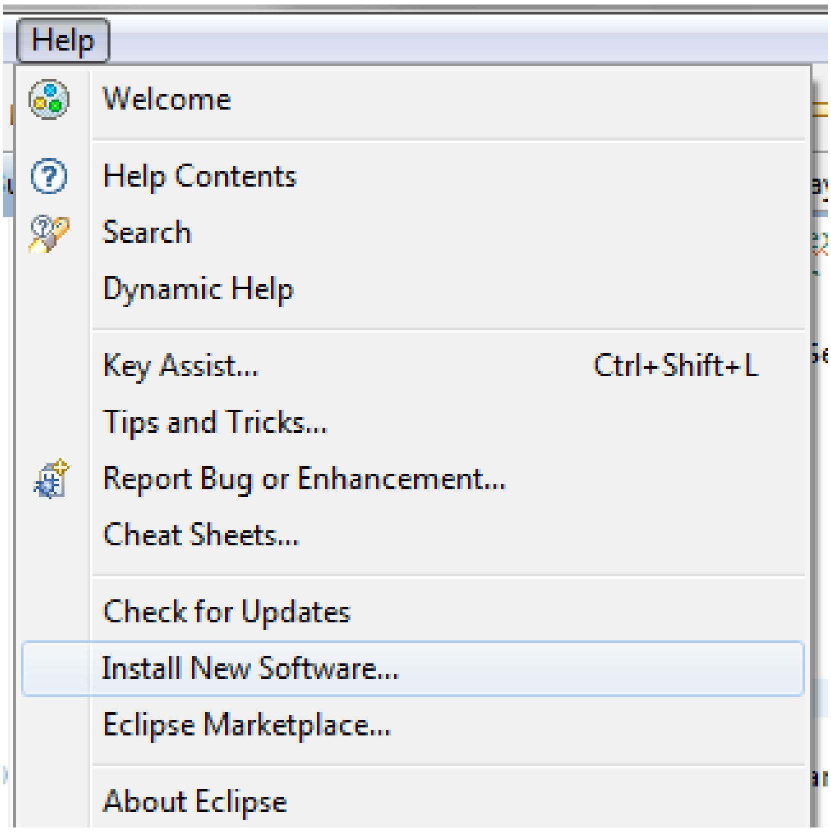
Download either the MacOSX 32bit or 64bit package from the homepage, extract and run it.

Installation as Eclipse Plugin

There are two ways to install EclipseCommander as Eclipse Plugin.

Installation via update site

Click Help → Install New Software...



Enter http://viplugin.com/eclipsecommander into the textfield and press Add...

Select and install the EclipseCommander plugin.

Installation via zip file

Download the latest version as zip file from the homepage http://viplugin.com/eclipsecommander (download box is on the left side). Extract the zip file to the dropins folder of your Eclipse installation. Restart Eclipse (if running)

Usage

Start EclipseCommander.

Note for EclipseCommander as Eclipse plugin users:

Open the EclipseCommander view by either clicking on the „E“ icon in the toolbar or by clicking Window → Show View → Other → Eclipse Commander.

Command	Shortcut
Switch to other pane	<TAB>
Show same directory in other pane	ALT+C
Switch panes	ALT+S
Choose drive left pane / right pane	ALT+F1 / ALT+F2
Rename	F2
Edit with text editor	F3
Edit with associated eclipse editor (only when used as Eclipse plugin, otherwise it's the bundled text editor)	F4
Copy	F5
Move	F6
Create directory	F7
Delete	F8 or <BACKSPACE>
Create File	F9
Search	F10
Virtual Tag Folder View	F11
Select file under cursor	<SPACE>
Invert current selection (e.g. select all if nothing is selected)	*

Select files by pattern	+
Deselect file by pattern	-

Virtual Tag Folder

Virtual Tag Folders are a completely new concept to organize your files and folders without even touching your original files! Just open the VTF view with F11, create a folder with F7 (which corresponds to a tag name) and copy files there. You can copy, move, execute, edit, search, ... them but they don't use any harddrive space and are copied in an instant.

Looking at an example might make this easier:

Say you have a collection of 100 movies and you want the same 100 movies in folders for genre and the same 100 movies in folder by director. In your normal filesystem that would mean copying around all the files, using up 3 times the space on your harddrive and copying 300GB would take a very long time.

Not so with Virtual Tag Folders!

Just create folders for the genre and folders for directors and copy the files there! You can watch them like the original files, you can remove them (without losing your original file) and you reorganize them instantely, because they are not really copied!

File Index Database

The file index database can be used to speed up searches in the filesystem dramatically. For this to work you have to do three things.

- go to the preferences and add all the paths you want to index
- click File->Create File Index... (this may take a very long time, depending on the folders you selected)
- click F10 to search and select the „Use file index“ combobox

Note that clicking the „use file index“ combobox while searching in a folder that is not in your file index path will return an empty result.

Scripting

Scripting has been added to EclipseCommander to be a true developers friend.

Currently only JavaScript is allowed as scripting language. Adding Python, Ruby or Groovy is easy and will be done in one of the next releases.

In the preferences you can select the path where your scripts are. The default path contains a few sample scripts to get you up to speed.

Here is the current API:

```
/**
 * select the given file in the current pane
 * @param file
 */
public static void moveCursorToFile(File file)

/**
 * select the given file index in the current pane
 * @param index
 */
public static void moveCursorToIndex(int index)

/**
 * select the file under the selection cursor
 */
public static void selectFileUnderCursor()

/**
 * set the filesystem path to show in the current pane
 * @param path the path to use as input for the current pane
 */
public static void setInput(String path)

/**
 * copy the currently selected files to the other pane
 * @throws ExecutionException
```

```
     */
    public static void copy() throws ExecutionException

    /**
     * move the currently selected files to the other pane
     * @throws ExecutionException
     */
    public static void move() throws ExecutionException

    /**
     * create folder with the given folderName in the current pane
     * @param folderName
     * @throws ExecutionException
     */
    public static void mkdir(String folderName) throws ExecutionException

    /**
     * open the virtual tag folder view in the current pane
     * @throws ExecutionException
     */
    public static void switchToVTF() throws ExecutionException

/**
 * delete the file under the selection cursor
 * @throws ExecutionException
 */
public static void deleteFileUnderCursor() throws ExecutionException

/**
 * Call an Eclipse Action
 *
 * @param action
 *            the id of the action you want to call
 * @return true if the call was successful
 */
```

```
public static boolean callHandlerAction(String action)

/**
 * show the ScriptEclipse console view
 */
public static void showConsole()

/**
 * send a keypress event (e.g. acknowledge dialogs)
 * @param key the key to send
 */
public static void keypress(String key)

/**
 *
 * @return the collection of files and folders in the current directory
 */
public static Collection<File> getAllFilesAndFoldersInCurrentDirectory()

/**
 *
 * @return the collection of files in the current directory
 */
public static File[] getAllFilesInCurrentDirectory()

/**
 *
 * @return the file under the selection cursor
 */
public static File getFileUnderCursor()

/**
 *
 * @return the currently selected files
 */
```

```
    public static File[] getSelelectedFiles()

    /**
     * refresh the current pane (needed if files are copied/moved/deleted by directly manipulating the filesystem)
     */
    public static void refresh()

    /**
     * rename the given file
     * @param file the file to rename
     * @param newFileName the new filename
     */
    public static void rename(File file, String newFileName)

    /**
     * update the file index database
     */
    public static void updateFileIndex()

    /**
     * search in the file index database
     * @param searchString
     * @return the files in the file index database matching the searchString
     */
    public static File[] searchInFileIndex(String searchString)

    /**
     * move the cursor up
     * @return true if cursor was moved up, false if we are already at the top
     */
    public static boolean cursorUp()

    /**
     * move the cursor down
```

```
 * @return true if cursor was moved down, false if we are already at the bottom
 */
public static boolean cursorDown()

/**
 * execute the given command (e.g. open an application to grep through files, ...)
 * @param command the command to execute
 */
public static void execute(String command)
```

Credits

This product includes the following software:
Apache Commons
The license can be found in the doc folder within the plugin.